Scientific Visualization: Interactions, Features, Metaphors

Edited by

Hans Hagen

DAGSTUHL
FOLLOW-UPS

Editors

Hans Hagen
AG Computergrafik und HCI
University of Kaiserslautern
67653 Kaiserslautern, Germany
`hagen@informatik.uni-kl.de`

ACM Classification 1998
I.3 Computer Graphics, I.4 Image Processing and Computer Vision, J.2 Physical Sciences and Engineering, J.3 Life and Medical Sciences

ISBN 978-3-939897-26-2

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at `http://www.dagstuhl.de/dagpub/978-3-939897-26-2`.

Publication date
October, 2011.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Digital Object Identifier: 10.4230/DFU.Vol2.SciViz.2011.i

ISBN 978-3-939897-26-2 **ISSN 1868-8977** **www.dagstuhl.de/dfu**

DFU – Dagstuhl Follow-Ups

The *DFU – Dagstuhl Follow-Ups* series offers a frame for the publication of peer-reviewed papers based on Dagstuhl Seminars. DFU volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

ISSN 1868-8977

www.dagstuhl.de/dfu

Contents

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: H. Hagen
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany

Preface

The Schloss Dagstuhl seminars on Scientific Visualization provide a dynamic setting for ongoing and future research in visualization. Numerous contributions in this active field originated at Schloss Dagstuhl, and were extended to large-scale collaborative research and high-impact works. This volume of the Dagstuhl Follow-Ups series contains the proceedings from the 2009 seminar.[1]

Resulting from a growth in data set size, complexity, and number of covered application areas, modern Scientific Visualization combines research from a wide variety of theoretical and practical fields such as mathematics, physics, biology and computer science. These research efforts yield a large number of different analysis, processing, and visualization techniques, allowing the efficient generation and presentation of visual results. This in turn directly contributes to the way domain experts are able to deduce knowledge from abstract data.

Emphasizing the heterogeneity of this research field, the Dagstuhl Seminar *Scientific Visualization* in 2009 focused on a wide range of visualization topics such as "Knowledge Assisted Visualization", "Visual Exploration Environment", "Biomedical Visualization", and "Visualization of Vector- and Tensorfields". The seminar aimed to provide an open and international environment for the discussion of recent trends, breakthroughs and future directions of research in the area of visualization, fostering scientific exchange and collaboration among researchers of the Sci-Vis community and identifying new research directions.

In the course of the seminar, leading international scientists presented state-of-the-art summaries as well as novel research results and ideas. Among the discussed key topics were:

- Interaction Techniques/Frameworks
  To efficiently perform visual data analysis, end users and domain experts need not just be presented with visualization results, but have to be offered intuitive and efficient real-time interaction techniques and frameworks. User-centered approaches demonstrate, how human factors can influence the way data is processed and presented. Presentations and results from this seminar illustrated and devised methods for interactive data exploration and analysis.

- Feature Definition and Extraction/Reconstruction
  New data types and application fields require new types of features, novel extraction techniques and visualization algorithms. Work from a broad context of feature extraction and reconstruction in areas such as scalar-, vector- and tensorfield visualization was presented in the course of this seminar.

- Visualization Metaphors
  As existing work from the field of visualization is adapted to new application areas or visualization problems, an increase in size, structure or complexity of the given data necessarily leads to the development of optimized algorithms. This seminar identified algorithms and data structures for performance and accuracy improvement in key areas of scientific visualization such as (vector) field analysis.

[1] See www.dagstuhl.de/09251.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: H. Hagen
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany

Besides these topics, participants gave valuable presentations about conceptual, philosophical and psychological questions in visualization regarding the impact and benefit of user-centered approaches, research classification and other topics. The productive setting at Dagstuhl made it possible, that a selection of ideas presented at this seminar as well as scientific results of this gathering are made available as Proceedings.
We would like to thank all the participants and many thanks go to Dr. Inga Scheler for her help editing this book.

Hans Hagen

List of Authors

Sean Ahern
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
ahern@ornl.gov

Yasmin I. Al-Zokari
Department of Computer Science
University of Kaiserslautern
Gottlieb-Daimler-Strasse
67663 Kaiserslautern, Germany
alzokari@informatik.uni-kl.de

Andriy Anishkin
Department of Biology, University of Maryland
College Park, MD 20742
anishkin@icqmail.com

Cornelia Auer
Zuse Institut Berlin
Takustrasse 7
14195 Berlin, Germany
auer@zib.de

Jean-Paul Balabanian
University of Bergen
Bergen, Norway
jean-paul.balabanian@ii.uib.no

Aaron Barsky
Department of Computer Science
University of British Columbia
201-2366 Main Mall, Vancouver BC V6T 1Z4, Canada
barskya@cs.ubc.ca

Chongke Bi
Graduate School of Frontier Sciences
The University of Tokyo
5-1-5 Kashiwanoha
Kashiwa, Chiba 277-8561, Japan
bichongke@visual.k.u-tokyo.ac.jp

Sergey Borovikov
University of Alabama
Huntsville, USA
snb0003@uah.edu

Blazej Bulka
Clark & Parsia, LLC
926 N St NW Rear, Studio #1
Washington DC, 2001, USA
blazej@clarkparsia.com

Hamish Carr
School of Computing and Informatics,
University College Dublin
Belfield, Dublin 4, Ireland
and School of Computing, Univ. of Leeds
Woodhouse Lane, Leeds, LS2 9JT, U.K.
h.carr@leeds.ac.uk

Silvia M. Castro
Laboratorio de Investigacion y Desarrollo en Visualizacion y Computacion Grafica (VyGLab)
Departamento de Ciencias e Ingenieria de la Computacion (DCIC)
Universidad Nacional del Sur (UNS)
Bahia Blanca, Buenos Aires, Argentina
smc@cs.uns.edu.ar

Hank Childs
Lawrence Berkeley National Laboratory
Berkeley, California, USA hchilds@lbl.gov

Joao L. D. Comba
Instituto de Informatica, UFRGS
Porto Alegre, RS, Brazil, CP 15064
comba@inf.ufrgs.br

Eric S. Cowgill
Department of Geology,
UC Davis, CA, USA
cowgill@geology.ucdavis.edu

Eduard Deines
University of California
Davis, USA
edeines@ucdavis.edu

David S. Ebert
Purdue University
Visual Analytics Center
West Lafayette, IN, USA
ebertd@purdue.edu

DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, Germany

Thomas Ertl
VISUS, Universität Stuttgart
Allmandring 19
70569, Stuttgart, Germany
ertl@visus.uni-stuttgart.de

Sebastian Escarza
Laboratorio de Investigacion y Desarrollo en Visualizacion y Computacion Grafica (VyGLab)
Departamento de Ciencias e Ingenieria de la Computacion (DCIC)
Universidad Nacional del Sur (UNS)
Bahia Blanca, Buenos Aires, Argentina
se@cs.uns.edu.ar

Leila De Floriani
Dipartimento di Informatica e Scienze dell Informazione
Universita di Genova
Genova, Italy
deflo@disi.unige.it

Issei Fujishiro
Department of Information and Computer Science
Keio University
3-14-1 Hiyoshi, Kohoku-ku
Yokohama 223-8522, Japan
fuji@ics.keio.ac.jp

Kelly P. Gaither
University of Texas
Austin, TX, USA
kelly@tacc.utexas.edu

Christoph Garth
University of California
Davis, USA
cgarth@ucdavis.edu

Ruth Gill
Washington University School of Medicine
St Louis, MO, U.S.A.

Eduard Gröller
Vienna University of Technology
Vienna, Austria
groeller@cg.tuwien.ac.at

Hans Hagen
Department of Computer Science
University of Kaiserslautern
Gottlieb-Daimler-Strasse
67663 Kaiserslautern, Germany
hagen@informatik.uni-kl.de

Bernd Hamann
Institute for Data Analysis and Visualization (IDAV), Department of Computer Science,
UC Davis, CA, USA.
hamann@cs.ucdavis.edu

Martin Hering-Bertram
Fraunhofer ITWM
Kaiserslautern, Germany
Rhine-Waal University of Applied Sciences
47533 Kleve, Deutschland
martin.hering-bertram@hochschule-rhein-waal.de

Ingrid Hotz
Takustrasse 7
14195 Berlin, Germany
hotz@zib.de

Cheuk Yiu Ip
Department of Computer Science and
University of Maryland Institute for
Advanced Computer Studies
College Park, MD 20742
jpcy@cs.umd.edu

Heike Jänicke
Swansea University
Swansea,Wales, UK
h.jaenicke@swansea.ac.uk

Yun Jang
ETH Zurich
Zurich, Switzerland
jangy@inf.ethz.ch

Marie desJardins
University of Maryland, Baltimore County
1000 Hilltop Circle
Baltimore MD, 21250, USA
mariedj@cs.umbc.edu

Kenneth I. Joy
Institute for Data Analysis and Visualization
Computer Science Department
University of California
Davis, California, USA
joy@cs.ucdavis.edu

Patric Keller
Department of Computer Science,
University of Kaiserslautern, Germany
pkeller@cs.uni-kl.de

Louise H. Kellogg
Department of Geology,
UC Davis, CA, USA
kellogg@geology.ucdavis.edu

Youngmin Kim
Department of Computer Science and
University of Maryland Institute for
Advanced Computer Studies
College Park, MD 20742
ymkim@cs.umd.edu

Taimur Khan
Department of Computer Science
University of Kaiserslautern
Gottlieb-Daimler-Strasse
67663 Kaiserslautern, Germany
tkhan@informatik.uni-kl.de

Christopher Koehler
Wright State University
3640 Col. Glenn Hwy, Dayton OH 45431
koehler.11@wright.edu

Andrea Kratz
Zuse Institute Berlin (ZIB)
Department Visualization and Data Analysis
Takustrasse 7, 14195 Berlin, Germany
kratz@zib.de

Oliver Kreylos
Department of Geology,
UC Davis, CA, USA
kreylos@cs.ucdavis.edu

Jörg Kuhnert
Fraunhofer ITWM Kaiserslautern
Kaiserslautern, Germany

Martin L. Larrea
Laboratorio de Investigacion y Desarrollo en
Visualizacion y Computacion Grafica
(VyGLab)
Departamento de Ciencias e Ingenieria de la
Computacion (DCIC)
Universidad Nacional del Sur (UNS)
Bahia Blanca, Buenos Aires, Argentina
mll@cs.uns.edu.ar

Dianne P. O Leary
Department of Computer Science and
University of Maryland Institute for
Advanced Computer Studies
College Park, MD 20742
oleary@cs.umd.edu

Peter Liggesmeyer
University of Kaiserslautern, Germany
liggesmeyer@informatik.uni-kl.de

Lars Linsen
Visualization and Computer Graphics
Laboratory
Jacobs University Bremen
Campus Ring 1
Bremen, Germany

Ross Maciejewski
Purdue University
Visual Analytics Center
West Lafayette, IN, USA
rmacieje@purdue.edu

Sergio R. Martig
Laboratorio de Investigacion y Desarrollo en
Visualizacion y Computacion Grafica
(VyGLab)
Departamento de Ciencias e Ingenieria de la
Computacion (DCIC)
Universidad Nacional del Sur (UNS)
Bahia Blanca, Buenos Aires, Argentina
srm@cs.uns.edu.ar

Daniel F. Martin
Lawrence Berkeley National Laboratory
Berkeley, USA
dfmartin@lbl.gov

Jeremy Meredith
Oak Ridge National Laboratory
Oak Ridge, Tennessee, USA
ahern@ornl.gov

Björn Meyer
Zuse Institute Berlin (ZIB)
Department Visualization and Data Analysis
Takustrasse 7, 14195 Berlin, Germany
bjoern.meyer@zib.de

Mark Miller
Lawrence Livermore National Laboratory
Livermore, California, USA
miller86@llnl.gov

Vladimir Molchanov
Visualization and Computer Graphics
Laboratory
Jacobs University Bremen
Campus Ring 1
Bremen, Germany

Tamara Munzner
Department of Computer Science
University of British Columbia
201-2366 Main Mall, Vancouver BC V6T
1Z4, Canada
tmm@cs.ubc.ca

Eoin Murphy
School of Computing and Informatics,
University College Dublin
Belfield, Dublin 4, Ireland
eoinomurchu@gmail.com

Harald Obermaier
University of Kaiserslautern
Kaiserslautern, Germany

Christian Pagot
Instituto de Informatica, UFRGS
Porto Alegre, RS, Brazil, CP 15064
capagot@inf.ufrgs.br

Robert Patro
Department of Computer Science and
University of Maryland Institute for
Advanced Computer Studies
College Park, MD 20742
rob@cs.umd.edu

Stefan K. Plontke
Tübingen Hearing Research Center (THRC)
Tübingen, Germany
stefan.plontke@uni-tuebingen.de

Bernhard Preim
University of Magdeburg
Department of Simulation and Graphics
39106, Magdeburg, Universitätsplatz 2,
Germany bernhard.preim@ovgu.de

Penny Rheingans
University of Maryland, Baltimore County
1000 Hilltop Circle
Baltimore MD, 21250, USA
rheingan@cs.umbc.edu

Paul Rosenthal
Visualization and Computer Graphics
Laboratory
Jacobs University Bremen
Campus Ring 1
Bremen, Germany

Filip Sadlo
VISUS, Universität Stuttgart
Allmandring 19
70569, Stuttgart, Germany
sadlo@visus.uni-stuttgart.de

Alec N. Salt
Washington University School of Medicine
St Louis, MO, U.S.A.

Daniel Schneider
Department of Computer Science
University of Kaiserslautern
Gottlieb-Daimler-Strasse
67663 Kaiserslautern, Germany
dschnei@informatik.uni-kl.de

Thomas Schultz
Computer Science Department and
Computation Institute
University of Chicago, USA
t.schultz@uchicago.edu

Norbert Siedow
Fraunhofer ITWM
Kaiserslautern, Germany
norbert.siedow@itwm.fraunhofer. de

Jaya Sreevalsan-Nair
IIIT- Bangalore
Electronics City, Hosur Road
Bangalore, 560100, India
jnair@iiitb.ac.in

Brian Van Straalen
Lawrence Berkeley National Laboratory
Berkeley, USA
bvstraalen@lbl.gov

Sergei Sukharev
Department of Biology, University of Maryland
College Park, MD 20742
sukharev@umd.edu

Shigeo Takahashi
Graduate School of Frontier Sciences
The University of Tokyo
5-1-5 Kashiwanoha
Kashiwa, Chiba 277-8561, Japan
takahashis@acm.org

Yuriko Takeshima
Institute of Fluid Science, Tohoku University
2-1-1 Katahira, Aoba-ku
Sendai 980-8577, Japan
takesima@vis.ifs.tohoku.ac.jp

Oliver Tse
Technische Universität Kaiserslautern
Fachbereich Mathematik
Erwin Schrödinger Strasse, 67663 Kaiserslautern
tse@mathematik.uni-kl.de

Dana K. Urribarri
Laboratorio de Investigacion y Desarrollo en Visualizacion y Computacion Grafica (VyGLab)
Departamento de Ciencias e Ingenieria de la Computacion (DCIC)
Universidad Nacional del Sur (UNS)
Bahia Blanca, Buenos Aires, Argentina
dku@cs.uns.edu.ar

Amitabh Varshney
Department of Computer Science and University of Maryland Institute for Advanced Computer Studies
College Park, MD 20742
varshney@cs.umd.edu

Joachim Vollrath
VISUS, Universität Stuttgart
Allmandring 19
70569, Stuttgart, Germany
vollrath@visus.uni-stuttgart.de

Gunther H. Weber
Lawrence Berkeley National Laboratory
Berkeley, USA
ghweber@lbl.gov

Daniel Weiskopf
VISUS, Universität Stuttgart
Allmandring 19
70569, Stuttgart, Germany
weiskopf@visus.uni-stuttgart.de

Kenneth Weiss
Department of Computer Science University of Maryland, College Park, MD
kweiss@cs.umd.edu

Matt Williams
Department of Computer Science
University of British Columbia
201-2366 Main Mall, Vancouver BC V6T 1Z4, Canada
mwill@cs.ubc.ca

Thomas Wischgoll
Wright State University
3640 Col. Glenn Hwy, Dayton OH 45431
thomas.wischgoll@wright.edu

Yi Yang
University of Kaiserslautern, Germany
yang@informatik.uni-kl.de

Dirk Zeckzer
Department of Computer Science
University of Kaiserslautern
Gottlieb-Daimler-Strasse
67663 Kaiserslautern, Germany
zeckzer@informatik.uni-kl.de

Valentin Zobel
Takustrasse 7
14195 Berlin, Germany
zobel@zib.de

CakES: Cake Metaphor for Analyzing Safety Issues of Embedded Systems *

Yasmin I. Al-Zokari, Taimur Khan, Daniel Schneider, Dirk Zeckzer, Hans Hagen[1]

[1] **Department of Computer Science**
University of Kaiserslautern
Gottlieb-Daimler-Strasse
67663 Kaiserslautern, Germany
{alzokari, tkhan, d_schnei, zeckzer, hagen}@informatik.uni-kl.de

Abstract

Embedded systems are used everywhere. They are complex systems whose failure may cause death or injury to people or may damage the environment are required to be safety safe. Therefore, these systems need to be analyzed. Fault tree analysis is a common way for performing safety analysis. It generates a large amount of interconnected data that itself needs to be analyzed to help different domain experts (e.g., engineers and safety analysts) in their decisions for improving the system's safety. Additional difficulties occur for the experts in communication and in linking the data (e.g., information of basic events or minimal cut sets) to the actual parts of the system (model). Therefore, a large amount of time and effort is being spent on discussions, searching, and navigating through the data. To overcome this, we present a new metaphor called "CakES" consisting of multiple views visualizing the data generated by fault tree analysis and linking them to the actual parts of the model by intuitive interaction. Using the interaction techniques of CakES the user can directly explore the safety related data without navigating through the fault tree while retaining an overview of all critical aspects in the model.

1998 ACM Subject Classification I.3 [Computer Graphics]: Reliability, Testing, and Fault-Tolerance

Keywords and phrases Fault Tree Analysis, minimal cut sets, basic events, information visualization, scientific visualization, engineering, tiled-wall, multiple monitor system, color vision deficiency, embedded systems, safety critical systems

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.1

1 Introduction

The size and complexity of embedded system's increase steadily forcing users/experts of different domains (e.g., safety analysts and engineers) to analyze a large amount of data. However, not only the embedded systems grow but the data generated by safety analysis grows too. Thus, obtaining total knowledge about the data in the different domains becomes more and more important. In [29], the authors show that increasing human involvement in tasks (understanding, exploring, navigating) through large and/or complex data sets leads to slower and error prone results. Examples are: forgetting to explore a sub-tree, missing a BE, or incorrectly combining the events of a MCS. In addition, the users/experts need to find the interesting/useful data and link them to support in their decision making process (e.g., which

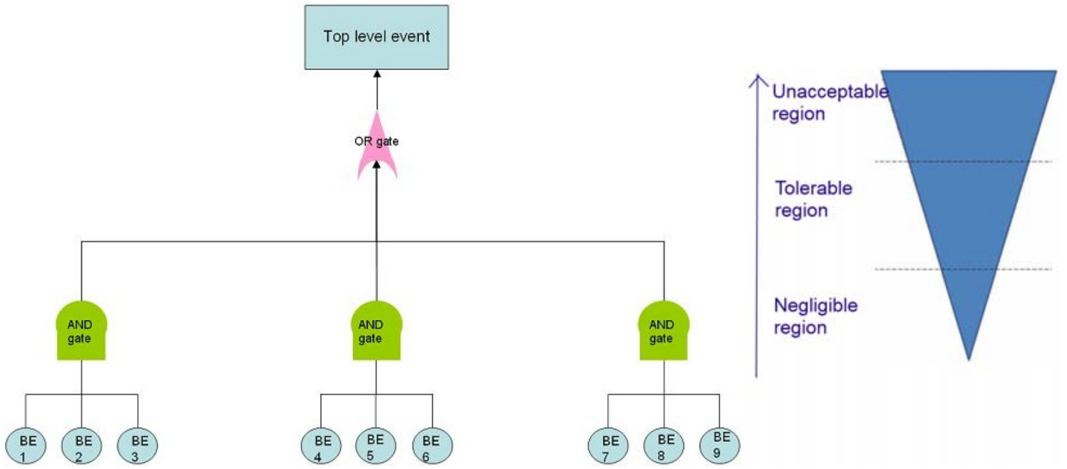
* This work was partially supported by DAAD, IRTG, ViERforES, BMBF.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 1–16
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

parts of a model should be improved or replaced given limited time and/or cost). Because each user has a different domain background communication difficulties appear. Therefore, there is a need for methods/tools to support users/experts during their analysis, exploration, and comprehension of the data from both domains (generated from the safety analysis and given by the engineers). To be able to deal with enormous data, we propose reducing the complexity of the data being analyzed by first extracting the most important information of the fault tree analysis and by then providing dynamic filtering to the user. Filtering is used to ease and speedup understanding and exploration. We present a new metaphor, "CakES". CakES consists of multiple views visualizing the physical model (engineering data), minimal cut sets, and basic events' information generated from the fault tree analysis (safety data). The views are linked in an interactive way to support exchanging and combining the knowledge of the different domains. Providing different correlating environments for the users [12] enables the analysts to explore data in one glance by four views: three visualizing the data (minimal cut set, basic event, and model view) and one interaction view that also shows the exact values (menu view). We applied an informal expert evaluation that provided us with useful and productive feedback and criticism. The paper is structured as follows. Section 2 provides the definitions of terms used in this work. Section 3 describes the related work. Our approach, the visual metaphors and the interaction devices, are presented in Section 4. Section 5 illustrates the use of our approach with a real world data set and Section 6 describes our design decisions. The setup and outcome of an informal evaluation performed on this approach are described in Section 7. The future work is presented in Section 8. We close this paper with the conclusions in Section 9.

2 Definitions

For an introduction into the topic of safety analysis and visualization, the relevant concepts used in this paper are defined. In [14] pp. 74, risks are divided into three levels: Unacceptable, Tolerable, and Broadly Acceptable. Figure 1b shows these levels [31], [19]. For both tolerable and acceptable levels, risks have to be reduced until the cost is more than the risk "As Low As Reasonably Practicable"(ALARP). Risk acceptance depends on many factors such as applicable laws or public opinion [2]. A safety-critical system could cause different levels of harm (hazard)(i.e., delay, injury, death, etc.) to living beings or/and to its environment [21]. A risk is a combination of the severity of the consequences of a hazard and the probability of its occurrence [7]. *Safety analysis* is used to prove that the system's risk is negligible [20] and involves detecting hazards and their causes to reduce the likelihood of probable accidents and their consequence. *Fault Trees Analysis (FTA)* is a traditional methodology in safety analysis (hazard analysis in more detail) to asses systems' safety [13], [18], [7] pp. 8. Fault Trees (FTs) are usually constructed using conventional logic gate symbols. Figure 1a shows an example of a FT. *Basic Events (BEs)* are the lowest-level influence factors (system faults) in the FT that may cause (with a certain probability [11]) an undesired state in the system (top level event or called hazard) [7]. The top level event is the root of the FT. In Figure 1a the BEs are BE_1 through BE_9. *Minimal Cut Sets (MCSs)* are a unique combination of the BEs–so it contains BEs–, e.g., a system malfunction [4], [6]. When any BE of a MCS is removed, the remaining BEs no longer form a cut set [10]. "A fault tree can be represented as the collection of its minimal cut sets" [7]. To improve the safety of a system the MCSs should be specified and isolated or removed [7]. In Figure 1a, the MCSs are the inputs of the AND gates: $MCS_1 = \{BE_1, BE_2, BE_3\}$, $MCS_2 = \{BE_4, BE_5, BE_6\}$, and $MCS_3 = \{BE_7, BE_8, BE_9\}$. In safety analysis two factors are important [7]: First, the severity of the



(a) An example of a Fault Tree.

(b) The ALARP carrot diagram.

Figure 1 Foundations.

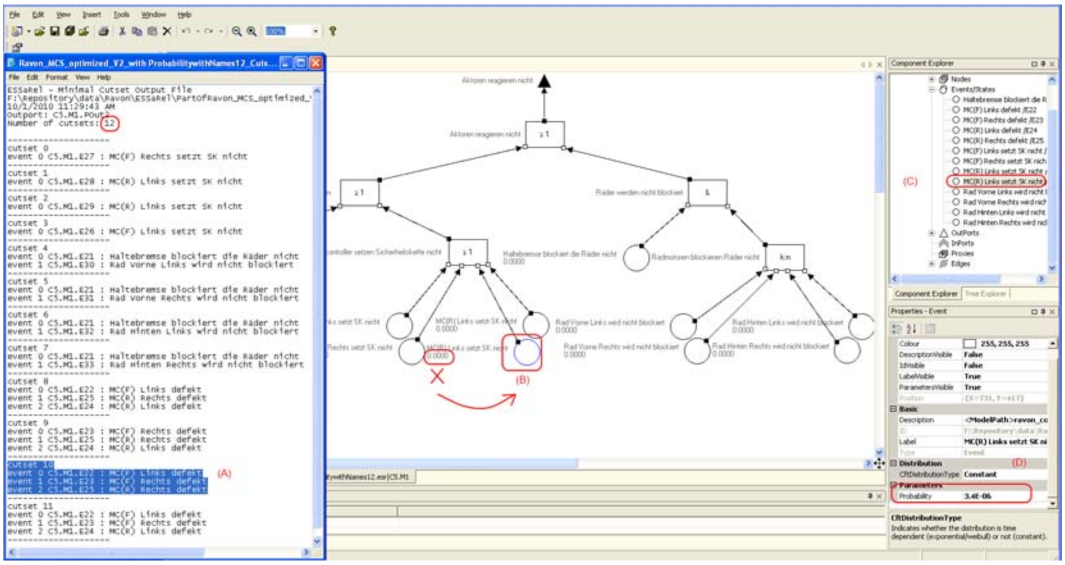
hazard's possible consequences. Second, the probability of the hazard occurrence, which is our concern in this paper. This is done usually by adding redundancy and/or improving the elements of the system.

3 Related Work

Visualization of safety and security data in embedded systems is an emerging research topic. With the growth of embedded systems and the growth of the data generated by fault tree analysis, visualization is becoming an important method to ease and speedup the developer's analysis. In this section a brief overview about the state of the art is given.

3.1 Fault Tree Analysis

We mention only some of safety analysis methods/tools, because others mainly follow the same principles. With the *ESSaRel tool* [10], users –usually safety analysts– can easily create FTs and generate MCSs analysis results (model generation and analysis). Here, component fault trees, gates, and BEs are visualized in 2D representations. However, the data generated by the MCS analysis –which is considered important for analysis– are not visualized and are represented in an external text file. This file includes some information of the MCSs such as their BEs, the BEs names, and their IDs (Figure 2). It is clear the user has to search for the FP of the BE from view (B) by clicking it to get view (D). Additionally, the names of the BEs in view (A) are not equal to the BEs names in views (C) and (D), which makes the exploration difficult. Further, there is no information about the MCSs' FPs the analysts can begin their search from. Therefore, the analyst has to find the other BEs of the examined MCS to compute its FP. As a result, the time and effort for exploring this small data set increases. In summary, the user has to navigate through the views to get simple informations such as which MCS is the most risky one, and still, no information about how the BEs look like or where they locate in the system is provided (especially if the naming of the BEs are inconsistent, which is usually the case when having different developers or engineers). Other tools/methods in safety analysis represent MCSs analysis data either in textual files



Ravon_MCS_optimized_V2_with ProbabilitywithNames12_Cuts...

```
ESSaRel - Minimal Cutset Output File
F:\Repository\data\Ravon\ESSaRel\PartofRavon_MCS_optimized_
10/1/2010 11:29:43 AM
Outport: C5.M1.POut2
Number of cutsets: 12

--------------------
cutset 0
event 0 C5.M1.E27 : MC(F) Rechts setzt SK nicht
--------------------
cutset 1
event 0 C5.M1.E28 : MC(R) Links setzt SK nicht
--------------------
cutset 2
event 0 C5.M1.E29 : MC(R) Links setzt SK nicht
--------------------
cutset 3
event 0 C5.M1.E26 : MC(F) Links setzt SK nicht
--------------------
cutset 4
event 0 C5.M1.E21 : Haltebremse blockiert die Räder nicht
event 1 C5.M1.E30 : Rad Vorne Links wird nicht blockiert
--------------------
cutset 5
event 0 C5.M1.E21 : Haltebremse blockiert die Räder nicht
event 1 C5.M1.E31 : Rad vorne Rechts wird nicht blockiert
--------------------
cutset 6
event 0 C5.M1.E21 : Haltebremse blockiert die Räder nicht
event 1 C5.M1.E32 : Rad Hinten Links wird nicht blockiert
--------------------
cutset 7
event 0 C5.M1.E21 : Haltebremse blockiert die Räder nicht
event 1 C5.M1.E33 : Rad Hinten Rechts wird nicht blockiert
--------------------
cutset 8
event 0 C5.M1.E22 : MC(F) Links defekt
event 1 C5.M1.E25 : MC(R) Rechts defekt
event 2 C5.M1.E24 : MC(R) Links defekt
--------------------
cutset 9
event 0 C5.M1.E23 : MC(F) Rechts defekt
event 1 C5.M1.E25 : MC(R) Rechts defekt
event 2 C5.M1.E24 : MC(R) Links defekt
--------------------
cutset 10
event 0 C5.M1.E22 : MC(F) Links defekt
event 1 C5.M1.E23 : MC(F) Rechts defekt
event 2 C5.M1.E25 : MC(R) Rechts defekt
--------------------
cutset 11
event 0 C5.M1.E22 : MC(F) Links defekt
event 1 C5.M1.E23 : MC(F) Rechts defekt
event 2 C5.M1.E24 : MC(R) Links defekt
--------------------
```

Component Explorer

- Nodes
- Events/States
  - Haltebremse blockiert die R
  - MC(F) Links defekt /E22
  - MC(F) Rechts defekt /E23
  - MC(R) Links defekt /E24
  - MC(R) Rechts defekt /E25
  - MC(F) Links setzt SK nicht /
  - MC(F) Rechts setzt SK nich
  - MC(R) Links setzt SK nicht
  - MC(R) Links setzt SK nicht
  - Rad Vorne Links wird nicht
  - Rad Vorne Rechts wird nich
  - Rad Hinten Links wird nicht
  - Rad Hinten Rechts wird nid
- OutPorts
- InPorts
- Proxies
- Edges

Properties - Event

Colour	255, 255, 255
DescriptionVisible	False
IdVisible	False
LabelVisible	True
ParametersVisible	True
Position	{X=731, Y=417}
Basic	
Description	<ModelPath>ravon_co
ID	F:\Repository\data\Ra
Label	MC(R) Links setzt SK ni
Type	Event
Distribution	
CftDistributionType	Constant
Parameters	
Probability	3.4E-06

CftDistributionType
Indicates whether the distribution is time dependent (exponential/weibull) or not (constant).

File
Edit
View
Insert
Tools
Window
Help
100%
Aktoren reagieren nicht
≥1
&
k/n
Räder werden nicht blockiert
Haltebremse blockiert die Räder nicht
Radmotoren blockieren Räder nicht
MC(R) Links setzt SK nicht
0.0000
Rad Vorne Links wird nicht blockiert
Rad Vorne Rechts wird nicht blockiert
Rad Hinten Links wird nicht blockiert
Rad Hinten Rechts wird nicht blockiert
(A)
(B)
(C)
(D)
Component Explorer
Tree Explorer

Figure 2 ESSaRel representations of a small sub-system containing 12 "minimal cut sets" with text (left). Fault tree and basic events (right).

or tables with or without their FPs depending on the methods, such as *FinPSA* [24], *RAM Commander* [1], *FSAP/NuSMV-SA Safety Analysis Platform* [6]. *Relex Fault Tree Analysis software* [17] has a feature for highlighting minimal cut sets. However, some other do not represent MCSs such as *ISAAC* [5] which is the new version of "Enhanced Safety Assessment for Complex Systems" (ESACS) [8]. In [15] the MCSs are represented as a subtree and in [26] as textual notation.

3.2 Visualization

A *3D Carousel View design* is presented in [23]. It takes the basic carousel model and elaborates it to hold an arbitrary number of 2D items. Here, the interactions used are spinning, zooming, and selection. Animation is included. Many video games use carousel views for displaying menus. They are usually used to select an item from a small set of items. The rotation begins slowly and the speed increases when the cursor gets close to the edge of the screen. The placement of the items is given by logarithmic equations. To avoid the problems caused by the clipping area, which causes difficulties for the users, they made the far items transparent. This method is applied when performance efficiency is not crucial. The *Hierarchy based 3D Visualization of Large Software Structures* ([3, 25]) presents a 3D visualization technique for the static structure of object-oriented software using distributions of 3D objects on a 2D plane. They adjust the transparency of object surfaces to the distance of the viewpoint to reduce the visual complexity. This method is used for hierarchical data (software packages). These data are represented as nested hemispheres. The *TrustNeighborhoods: Visualizing Trust in Distributed File Sharing Systems* in [9] is a tool for visualizing document trust relationships in large scale distributed file sharing systems. It uses a metaphor of a multilayered city to represent trust as geographic relations. It uses a radial space-filling layout, with a 3D mode for exploration. The interactions applied are dragging and dropping documents, overview and navigation, and smooth zooming (also called "fly to").

3.3 Summary

In summary: almost all safety analysis methods/tools are powerful in creating FTs and in generating the safety results (e.g., MCSs). However, analyzing the generated data needs: -Navigation through the textual files (upwards and downwards) for comparisons, and exploration. -Navigations through other views to compute or link the data together whenever the information is not complete in the textual reports (e.g., ESSaRel). Additionally, there is no information about the shapes and locations of the BEs. This implies that the user will have difficulties finding the real world relation between the physical hardware components and the fault trees and the MCS text representation. Finally, filtering these data according to Failure Probabilities is cumbersome if not impossible. Therefore, we propose to visualize the generated safety data to support the analysis.

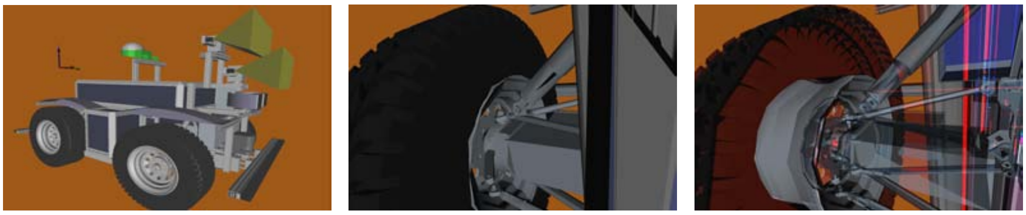
4 Our Approach

In our approach both the benefits of the "Carousel View" [23] and the "Hierarchy 3D view" [25] (Section 3) and a "simplified circle packing algorithm" [34] are combined and extended to support our goal. In this work, the items are not only placed at the border of the table as in [23] but also in the inner area of the circle for space efficiency. The transparency of the containers (MCSs, Figure 5c in Section 4.3) depends on the user's selection, not on his distance as in [25]. Here, even though all the transformations are available, the users do not need to navigate forward and backwards to explore the contents of the containers as in [25]. Additionally, the containers are placed in a symmetric fashion in the inner area of each level, because the author in [28] pp. 192 shows that humans perceive symmetry as being important and thus it provides a powerful organizing principle. In our case, a vertical view (2D view) is sufficient to illustrate the MCSs that are the best candidates to be chosen first for analysis. Additionally, a horizontal view (3D view) is used when the user selects a MCS to display the physical parts related to the BEs in this MCS. Thus, the user is directly involved in the data analysis process without the need to navigate through the system's fault trees and text files. The fault tree information is hierarchically structured, whereas the model information is flat. Thus, we modified this data to get a hierarchical structure for the model information, because naturally the model parts are in a hierarchical form. Then, we mapped the relationships of both data domains, to get completely related data to work with. CakES is a multi-window, or multi-screen system supporting different levels of focus and context at the same time.

4.1 Overview of CakES

In this section, the views and the interactions of CakES are be presented. There are four views, one for interaction and switching the window focus and three for visualizing the data:

1. Model View: displays the system's model being analyzed and its parts.
2. MCS View: displays the collection of MCSs, their probabilities, and the physical parts related to their BEs, so this view contains the safety related data. Therefore, this view contains the information of both the "text file" and the "Properties-Event" view of ESSaRel as shown in Figure 2. Additionally, the shapes of the physical parts related to the BEs and the overall FP of each MCS are shown. This information comes from the model and is computed from the FT information. Finally, it also provides the overall system's safety.
3. Menu View: displays a menu and the safety data in values.
4. BE View: displays the physical parts related to the BEs that are contained in the selected MCS, this is an auxiliary view for future usage.



(a) RAVON robot in the model view. **(b)** Outside the model. **(c)** Inside the model.

Figure 3 Part of RAVON robot displayed in the Model view.

Table 1 Keyboard interaction.

Keyboard's buttons	
Key	Function
y, Y	Rotate in +y, -y
x, X	Rotate in +x, -x
z, Z	Rotate in +z, -z
→	Move right
←	Move left
↑	Move up
↓	Move down
PgUp	Zoom out
PgDn	Zoom in
b, B	Translucent, Opaque
a, A	Animate, Stop Animation
o and O	Return to default position
s, S	Spotlight on, off
PrntScrn	Snapshot of the model view
ctrl + 1	Switching between standard and stereo view

4.2 Model View

In this view the model of the system being analyzed is displayed. For example, we have a robot model called RAVON (Robust Autonomous Vehicle for Off-road Navigation) shown in Fig 3a. This model is given in OpenInventor format [30]. OpenInventor is a C++ Object Oriented 3D graphics API providing higher layer programming of OpenGL that helps in efficient and convenient programming [22], [32]. The user can interact with the model using keyboard, mouse, and space-mouse. The interactions provided are rotation and translation such as zooming(z axis) and panning (x, y axis). These interactions are shown in the Tables 1 and 2. Further, the user has the ability to explore the model not only from the outside but also from the inside by placing the near clipping plane in a suitable position. Figure 3 shows an example.

4.3 MCS View

The *Cake* is our metaphor to visualize the failure probability (FP) of MCSs and the shape of the physical parts related to their BEs. Figure 5c shows the MCS view. All examples in

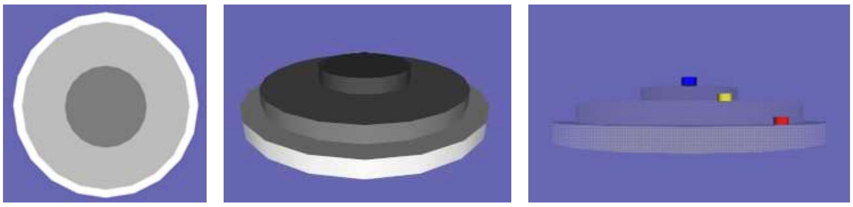
Table 2 Mouse and space mouse interaction.

(a) Mouse.

Mouse	
Button	Function
Left	rotate
Right	moving left & right
Scroll	zoom in/out
Middle	Return to default position

(b) Space mouse.

Space Mouse	
Action	Function
Roll, Pitch, Yaw	Rotate in (x, y, z)
Move	in x, y, z directions
Left button	Return to default position



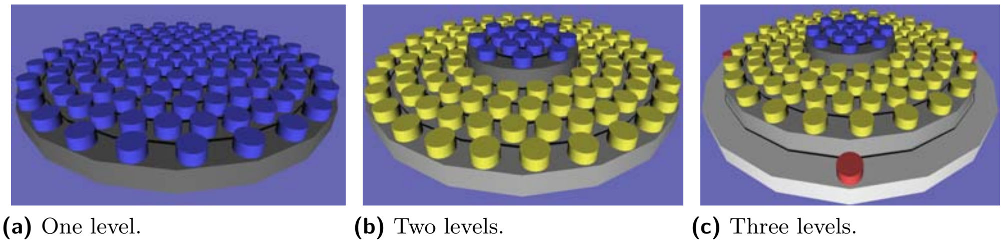
(a) A 2D view (top view).

(b) A 3D view.

(c) Placing MCSs depending on their failure probability.

Figure 4 Levels background and MCSs placement in the MCS view.

this section (only for the MCS view) are generated from artificial data for demonstration purposes. Two examples with real data will be presented in Section 5. The cake consists of a maximum of three levels that reflects FPs ranges shown in Section 2. There are four FP values that influence the number of levels displayed and the placement of MCSs: minimum, border between lowest and middle range (lower border), border between middle and highest range (upper border), and maximum. The first level, the one in the center, corresponds to the lowest range of FPs, those between the minimum and the lower border. This level includes MCSs with small FPs (could be considered as low risk MCSs). The second level corresponds to the middle range between lower and upper border. This level includes MCSs with higher FPs (could be considered as risky MCSs). The third level, the outer most one, corresponds to the highest range of FPs, those between the upper border and the maximum. This level includes MCSs with the highest FPs (could be considered as very risky MCSs). The first level has the darkest gray, the second level has a lighter gray, and the outer level has the lightest gray background. Figures 4a and 4b show the background coloring of the levels. The reason for giving the outer level the lighter gray color is that the user would then be able to directly concentrate on the most important level, which has the highest FPs [28] p. 143. Each level contains a number of MCSs (represented as cylinders) that have a FP within its range (Figure 4c). If there are no MCSs within the boundaries, then the level is empty and will not be displayed. Examples having only one or two levels are shown in Figure 5. Each MCS contains a certain number of BEs. The FP of a MCS is computed as the product of the FPs of its BEs. The user can manipulate the levels and the MCSs placements by changing the FP values using the menu sliders (Figure 9) described in Section 4.4. We computed the placements of MCSs using the following equations: The radius of the MCSs plus an empty distance around each MCS R_s is computed as: $R_s = 0.5 + \frac{1}{8}$. The radius of the ring holding the MCS R_c is $R_c = R_s \cdot RingNo$. The level radius R_h is $R_h = R_s + R_c$. The number of the MCSs in a ring is given by $m_r = \max(NoMCS) = \frac{\pi}{\arcsin\frac{R_s}{R_c}}$. The number of MCSs in the



(a) One level. **(b)** Two levels. **(c)** Three levels.

Figure 5 3D MCS view while interacting with the menu sliders.

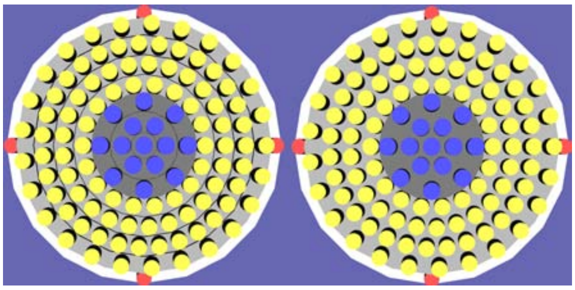


Figure 6 CakES: the MCS view. MCS view in 2D: With rings (left), With no rings (right).

level is $m_h = \sum_{r \in ring} m_r$. The number of MCSs in the cake is $m = \sum_{h \in levels} m_h$ Around each group of MCSs a black ring is placed to improve the distinguishability and to reduce any errors that could be caused by color contrast [28] p. 143. Figure 6 shows the view with and without rings, respectively. Table 3 shows the coloring of the MCSs in each level. The color vision deficiency entries work for Protanopia, Deuteranopia, and Tritanopia. Since the cyan has a level of green and blue, the blue-yellow problem of the Tritanopia type is avoided. Further, the red-green problem of the Protanopia and Deuteranopia type is solved, because red and green are not neighbors in the visualization (Figure 7). We added two different saturation levels to the MCSs colors, to provide more visualized details about the FPs of the MCSs to the users. These saturation levels are computed using the logarithmic mean of the range. High saturation is assigned to the upper part and low saturation is assigned to the lower part of this division. Figure 8 (right) shows MCSs of the second level of the cake (yellow) having different saturation values, where the MCSs having less saturation are the ones with less importance than the others in the same level. The MCSs of the first level are fully saturated blue and the ones of the third level (that contains red MCSs) are less saturated. The *BEs* of a MCS relate to both hardware and software parts. The physical parts related to the BEs of the fault tree are visualized inside each MCS to give the user the relationship between the cake's view and the BE view. Figure 8 shows two parts of the control box of RAVON in the selected MCS. The positioning of the BEs in the MCSs depends on the number of the BEs in a MCS. From our safety experts we know that the most important MCSs are the ones having 1-3 BEs. For example, for a single point of failure there will be a MCS with only one BE. This case is the most risky one, because the possibility for this MCS to occur is higher than the MCS that has two or more BEs, etc. However, we visualized a maximum of six BEs of MCSs that have six BEs or above to provide tolerability. Let x, y, z be the coordinates, h be the height of the MCS's cylinder in y direction, and n be the number of the BEs in a MCS. Then, the following equations are used for the placement of BEs depending of their number of BEs. If $n = 1$, then it will be placed in the center of the

Table 3 Coloring.

Risk level	Normal vision	Color vision deficiency
Low	Blue	Cyan
Medium	Yellow	Yellow
High	Red	Orange

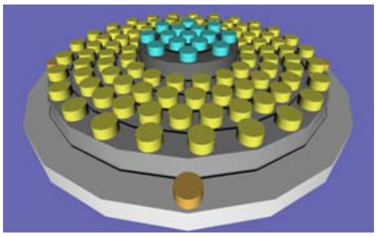


Figure 7 Cake using colors adapted to Protanopia, Deuteranopia, and Tritanopia.

MCS (x, y, z). If $n = 2$, they will be placed in a straight line, on points $(x, y \pm \frac{1}{4} \cdot h, z)$. If $n = 3$, they will be placed in a triangle, on points $(x - \frac{1}{2} \cdot r, y \pm \frac{1}{4} \cdot h, z)$ and $(x + \frac{1}{2} \cdot r, y, z)$. If $n = 4$, they will be placed on a polygon, on points $(x \mp \frac{1}{2} \cdot r, y \pm \frac{1}{4} \cdot h, z)$. If $n = 5$, they will be placed in the form of a cross, on points $(x \mp \frac{1}{2} \cdot r, y \pm \frac{1}{3} \cdot h, z)$ and (x, y, z). If $n = 6$, they will be placed on the six corner points of a polyhedron $(x \mp \frac{1}{2} \cdot r, y \pm \frac{1}{3} \cdot h, z)$ and $(x, y, z \pm \frac{1}{2} \cdot r)$. The most common approach for an analysis is to start with an overview and then to zoom into details as needed. The cake can be rotated using the same interactions as the model interaction. Two types of zooming are available in this view: spatial and semantic zooming. This approach is supported by providing the data at different levels of granularity. Using spatial zooming, the cake can be enlarged to pick the required MCS. Picking is done by (ctrl + left mouse button) and invokes a semantic zooming. When this is done, the MCS becomes transparent, the inner contents of the MCS will appear (BEs), the BEs are also displayed in the BE view, and the related parts in the model are highlighted.

4.4 Menu View

The menu view is primarily used for interaction. There are three interaction areas in the menu. The first area ((A) in Figure 9) consists of four buttons. Three of them can be used to select the views in focus and one terminates the application. The second area ((B) in Figure 9) consists of four range sliders with logarithmic scale. They determine the minimum, lower border, upper border, and maximum probabilities of the levels (Section 4.3). The sliders act as filters allowing to select FP ranges dynamically depending of the standard of the system being analyzed. Changing the values removes those MCSs from the view that are below the minimum or above the maximum FP. Further, MCSs are assigned to different levels when the borders are changed. Finally, two radio buttons in the third area determine whether to use normal coloring or colors adapted to a certain color vision deficiency type. We reduced the size of the menu in the figure for demonstration purposes. The second function of the menu is to display additional information. Statistical quantitative information of the safety file is provided, such as the number of MCSs and the number of BEs in the system. Further, if the user selects a MCS, its size (number of BEs it contains), its FP, and information about its BEs are displayed. The information about the BEs includes their ID, name, and FP.

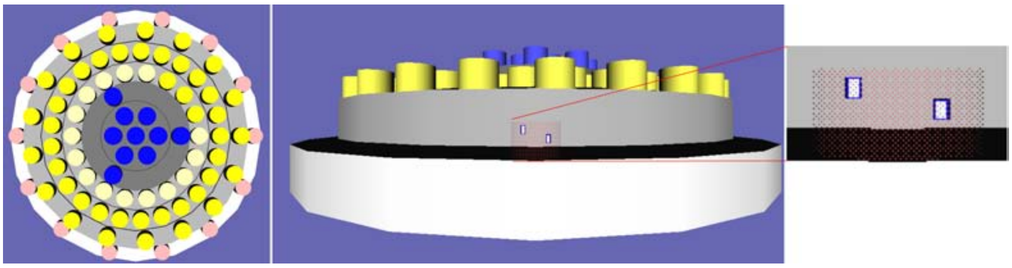


Figure 8 MCS view. (right) Saturation difference in the second level. (middle and left) Two parts of the controller box related to the BEs shown in the selected MCS.

4.5 BE View

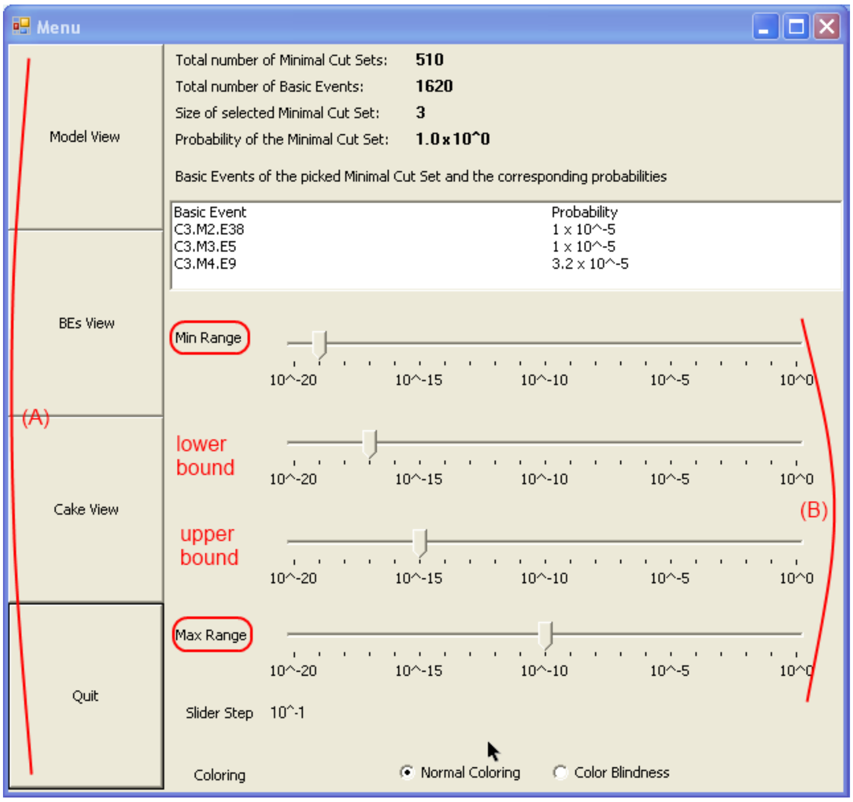
The BE view has a direct relation to the cake and to the model view. In this view, the hardware parts related to the BEs of the selected MCS in the MCS view are displayed. This view shows these parts in more detail. As in the MCS view, a maximum of six BEs are displayed. This view is an auxiliary view for future requirements. The interaction is the same as in the MCS and the model view.

5 Example using Real World Data

We applied our method to a real model in the context of the ViERforES project [27] funded by the German Federal Ministry of Education and Research (BMBF). It aims at optimizing the analysis of safety, security, and reliability aspects of embedded systems. The Robotics Research Lab of the University of Kaiserslautern, Germany [33] provides the Robot RAVON (Robust Autonomous Vehicle for Off-road Navigation) [16]. This complex embedded system is analyzed and tested using fault tree analysis (FTA) by safety analysts in the software engineering department. The results were fault trees containing basic events and additionally, after applying a MCS analysis, a list of minimal cut sets (540 MCSs) and their BEs. We adapted the system to two different environments. The first consists of a standard monitor and a stereo monitor (Section 5.1). The second is a tiled wall display (Section 5.2).

5.1 Standard and Stereo Monitor

This configuration consists of a standard monitor with 1920×1200 pixels and a Zalman Trimon passive stereo monitor with a resolution of 1600×1050. The computer has an AMD Phenom™ Quad-Core Processor with 2.60GHz, 3.25 GB of RAM, and an NVIDA GeForce GTX 280 graphics card. A 3Dconnexion Space Navigator is used for interaction. The multiple window system is used in an environment with two monitors (two standard or one standard and one stereo). The model view is displayed on the second monitor that is either a standard or a stereo monitor, and the three other views are displayed side-by-side on the standard monitor as shown in Figure 10. Here, a yellow MCS is selected in the MCS view as shown in Figure 10a (right). The hardware components associated with these BEs are displayed in the BE view (middle). Information about the BEs in the selected MCS is displayed in the menu view (left). Additionally, the consequence of this selection is shown in the model view automatically (Figure 10b). Here, the Ravon model is displayed on the second monitor, either in normal or in stereoscopic view. The BEs from the MCS that was selected in the MCS view (Figure 10a) are opaque, while the rest of the model is transparent.



Total number of Minimal Cut Sets: 510
Total number of Basic Events: 1620
Size of selected Minimal Cut Set: 3
Probability of the Minimal Cut Set: 1.0x10^0

Basic Events of the picked Minimal Cut Set and the corresponding probabilities

Basic Event	Probability
C3.M2.E38	1 x 10^-5
C3.M3.E5	1 x 10^-5
C3.M4.E9	3.2 x 10^-5

Slider Step 10^-1

Coloring ● Normal Coloring ○ Color Blindness

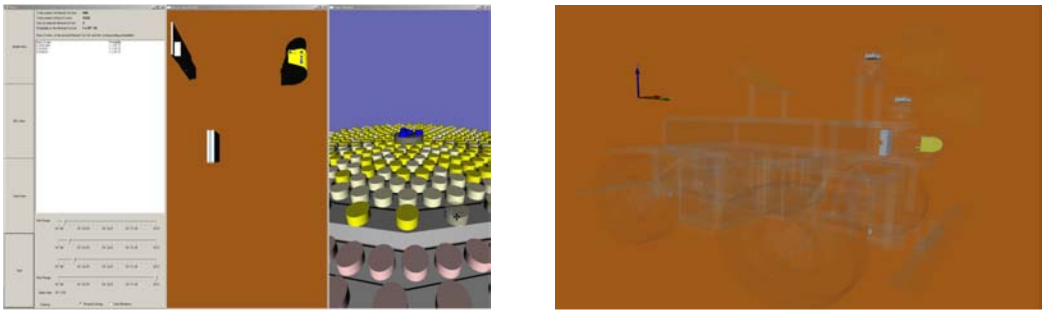
Figure 9 Menu view.

5.2 Tiled Wall

The 3 × 3 tiled wall consists of 9 displays that are connected to 5 computers. It is a multi-screen environment shown in Figure 11. The top 6 screens display the model view and the bottom 3 screens display the menu view, the BE view, and the MCS view (from left to right) as shown in Figure 11. As a result, this visualization can work in any environment having different views, windows, or screens.

6 Reasons for Some Choices

Our choices for the visualization metaphors are based on the following considerations. Having MCSs in different sizes, i.e., if a MCS has more BEs then it is larger, is better to be avoided. Large MCSs will have mostly a lower importance, since the probability of a MCS is calculated as the product of its BEs probabilities. Therefore, even though they have larger sizes, they are less critical. Assigning a small size to a MCS having a large number of BEs and vice versa causes a counter-intuitive visualization, that we wanted to avoid. Having only six BEs visualized in a MCS is sufficient, for the same reason described above. Whenever a MCS has more than six BEs, it is most probably not critical. In addition, when visualizing more than six BEs, the sizes of these elements will decrease too much and will no longer be distinguishable. Finally, MCSs having more than six BEs are crowded. Having MCSs visualized as cylinders avoids lines crossing on the surfaces of polygonal shapes that will partially occlude the shape of the BEs inside the selected MCS. Additionally, spherical shapes limit scalability and waste the surface area that could be used in the case of cylinders with the



(a) The menu (left), the BE (middle), and the MCS view (right).

(b) The Ravon model on a standard monitor.

Figure 10 CakES on standard monitors.

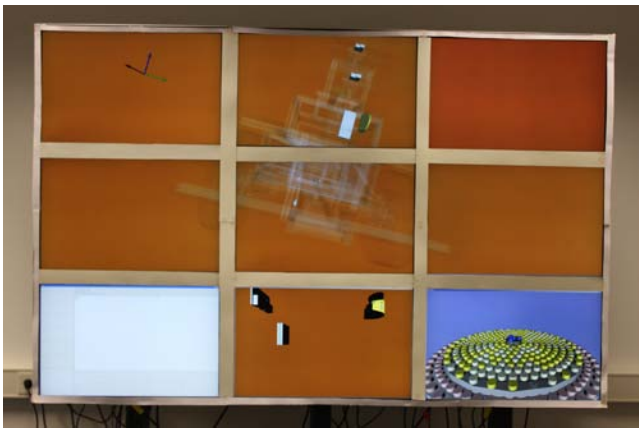


Figure 11 The CakES system on a tiled wall.

same radius. Having spaces between the cylinders enables the user to move freely inside the cake, reduces over crowdedness of MCSs, and eases the user's ability to distinguish between the BEs in the MCSs, when a MCS is selected. According to the risk acceptance diagram (Section 2, Figure 1b, [19]) there are three levels: Negligible, Tolerable, and Unacceptable. Therefore, we prefer having three levels of risks in the cake. Using different saturations for the same level increases the granularity inside each safety level. This helps the user to find the most critical MCSs. Neither 2D nor 3D text is used to avoid the drawbacks of text in 3D environments. For example, having text on top of the MCSs is only clearly visible to the user, when the scene is vertical, however, when the scene is not vertical, the text is less readable. Additionally, if we repeat the text many times or if we hang the text close to the MCSs, the scene will get crowded, even if the number of MCSs that are visualized is small. Finally, when zooming out, the text and textures become unreadable and the exploration will become difficult. As we will analyze systems having a large number of MCSs (normally, more than a hundred), having a unique color for each MCS is not feasible. According to [28], the amount of colors being easily distinguishable lies between five and ten different colors. For the same reason, textures are also not used.

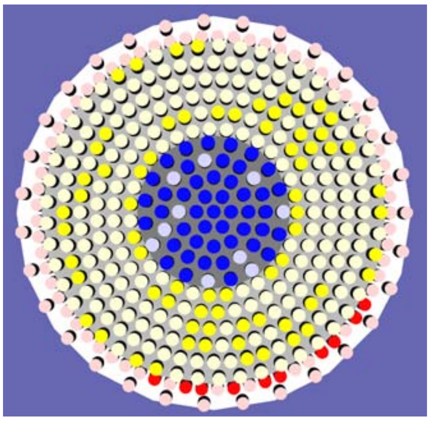


Figure 12 MCS view needs ordering.

7 Informal Expert Evaluation

An informal expert evaluation was performed on this approach by a group consisting of specialists from the domains software engineering, interaction, engineering (robotics department), and safety analysis, as well as information visualization. The aim of this evaluation was to get as much feedback as possible to improve our approach for the need and the questions of the experts of the different fields. The final feedback from this group was used to enhance the visualization in various aspects. They observed and suggested the following:

- The cylinders are not smooth.
- There are occlusions in the MCS view, because of the distortion.
- The most important MCSs should be in the first level, so the placement of the important and less important MCSs in the MCS view should be exchanged.
- It is better to have the MCSs in order from inside to outside, so the user can get a feeling about the importance degrees between them.
- The MCSs should be sorted depending on their probabilities.
- The color of the least important MCS should be green instead of blue in agreement with standard dependability requirements.
- Show clearly the connection between BEs and physical parts.
- Have a bigger, clearer text font in the menu.
- The interaction with the sliders are difficult.

As a conclusion, we should enhance our method based on this feedback.

8 Future Work

We plan to improve CaKES by taking the recommendations of the informal expert evaluation into account. Especially, ordering the MCSs by their probabilities in each level level would be beneficial. Figures 11 and 12 show the problem of unordered data. This would even increase more, if the data is large. Further, we would like to apply our approach to other display configurations, e.g., CAVE environments. Additionally, we will perform a user study and examine the usefulness of CakES. improve the sliders usability. Because the transparency caused delay in the interaction we need to change it to another function or by using a transparency node. Therefore, we changed to the *SORTED_OBJECT_BLEND* transparency type. Finally, we want to test our approach using other examples from ViERforES and related projects.

9 Conclusion

From the related work, we found that it is necessary to come up with new approaches and/or visualizations to explore the results of safety analysis, because these results are complex and/or large in nature. Additionally, the information about the geometrical location of the events that cause failures to the system and that are most significant to the embedded system is important. The CakES metaphor provides the user with the ability to explore data with more than five different levels of focus and context: the model, a combination of the MCSs' Failure Probability, the components of each MCS, the BEs' Failure Probability, and the shape and the locations of the BEs in the model. Linking the views shows the relationship of the data in a realistic interactive approach. This work draws immediate attention to the most important MCSs of the embedded system that cause its failure. The user (a safety analyst and/or an engineer) can apply this solution in different environments such as standard monitor configurations, tiled wall, and the combination of standard and stereo monitors [12]. To our knowledge, we introduced the first 3D representation for safety analysis features, linking the existing knowledge of this domain with the engineering domain's knowledge reducing the time of searching for relevant information, and leading to a reduction of human errors, effort, and cost. Additionally, CakES provides the ability of collaboration between specialists in different domains (e.g., safety analyst and robotics engineer). Finally, after gaining experience in either domain the specialists could efficiently work alone exploring and judging how to choose the parts to maintain and the parts to replace depending on their importance.

10 Acknowledgments

This work is partially supported by the DAAD, the IRTG sponsored by the DFG, and by the BMBF project ViERforES. We thank all our colleagues for helpful discussions and suggestions, especially Martin Proetzsch, Yi Yang, and Guo Zhensheng.

References

1 ALD Reliability Software Advanced Logistics Development and Services Worldwide. Fault tree analysis (fta). `http://www.aldservice.com/en/reliability/fault-tree-analysis.html`; Online; accessed 6-August-2010.

2 Colin Atkinson, Christian Bunse, and Hans Gross. *Component-Based Software Development for Embedded Systems: An Overview of Current Research Trends*. Springer Berlin Heidelberg, 2005.

3 Michael Balzer and Oliver Deussen. Hierarchy based 3d visualization of large software structures. In *VIS '04: Proceedings of the conference on Visualization '04*, page 598.4, Washington, DC, USA, 2004. IEEE Computer Society.

4 Tim Bedford and Pieter van Gelder. *Safety and Reliability: Proceedings of the ESREL 2003 Conference, Maastricht the Netherlands, 15-18 June 2003*. Taylor & Francis, 2003.

5 Piergiorgio Bertoli. *Model Checking and Artificial Intelligence*. ERTS, 2006. `http://www.springerlink.com/content/rq430x354035p654`.

6 Marco Bozzano and Adolfo Villafiorita. The FSAP/NuSMV-SA Safety Analysis Platform. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(1):5–24, 2007.

7 Marco Bozzano and Adolfo Villafiorita. *Design and Safety Assessment of Critical Systems*. CRC Press (Taylor and Francis), an Auerbach Book, 2010.

8 Marco Bozzano, Adolfo Villafiorita, Ove Äkerlund, Pierre Bieber, Eckard Boede, Matthias Bretschneider, Antonella Cavallo, Charles Castel, Massimo Cifaldi, Alessandro Cimatti, Alain Griffault, Christophe Kehren, Benita Lawrence, Andreas Lüdtke, Silvayn Metge, Chris Papadopoulos, R. Passarello, Thomas Peikenkamp, Per Persson, Christel Seguin, Luigi Trotta, and Laura Valaccaand G. Zacco. ESACS: an integrated methodology for design and safety analysis of complex systems. In *In Proceedings of ESREL 2003*, pages 237–245, 2003. Maastricht, The Netherlands.

9 N. Elmqvist and P. Tsigas. Trustneighborhoods: Visualizing trust in distributed file systems. In *Proceedings of the Eurographics/IEEE VGTC Symposium on Visualization 2007*, pages 107–114, 2007.

10 ESSaRel. Background information — essarel, 2002. `http://www.essarel.de/background/background.html`; Online; accessed 6-August-2010.

11 Bernhard Kaiser, Catharina Gramlich, and Marc Förster. State/event fault trees - a safety analysis model for software-controlled systems. *Reliability engineering & systems safety*, 92:1521–1537, 2007. `http://www.ingentaconnect.com/content/els/09518320/1998/00000061/00000001/art00061`.

12 Taimur Khan, Daniel Schneider, Yasmin Al-Zokari, Dirk Zeckzer, and Hans Hagen. Framework for comprehensive size and resolution utilization of arbitrary displays. In Hans Hagen, editor, *Scientific Visualization: Advanced Concepts*, Dagstuhl Follow-Ups, Wadern, Germany, 2010. Schloss Dagstuhl–Leibniz Center for Informatics. accepted for publication.

13 Nikolaos Limnios. *Fault Trees (Control Systems, Robotics & Manufacturing Series)*. Wiley, John & Sons, 2007.

14 UK Ministry of Defence. Safety Management Requirements For Defence Systems, Defence Standard 00-56, 2007.

15 Frank Ortmeier, Wolfgang Reif, and Gerhard Schellhorn. Formal safety analysis of a radiobased railroad crossing using deductive cause-consequence analysis (DCCA). In *In Proceedings of 5th European Dependable Computing Conference EDCC, volume 3463 of LNCS*. Springer, 2005.

16 RAVON. AG Robotersysteme: Ravon, 2009. `http://agrosy.informatik.uni-kl.de/en/robots/ravon/`; Online; accessed 6-August-2010.

17 RELEX. Fault tree event tree. `http://www.relexsoftware.co.uk/products/ftaeta.htm`; Online; accessed 6-August-2010.

18 ReliaSoft Corporation. Fault tree analysis — weibull, reliability engineering resources, 2009. `http://www.weibull.com/basics/fault-tree/index.htm`; Online; accessed 6-August-2010.

19 Jørn Vatn. A discussion of the acceptable risk problem. *Reliability Engineering and System Safety*, 61:11–19(9), July 1998. `http://www.ingentaconnect.com/content/els/09518320/1998/00000061/00000001/art00061`.

20 Savive Pty Ltd. Risk analysis and assessment. `http://www.savive.com/inform/riskassessment.html` ; Online; accessed 6-August-2010.

21 Savive Pty Ltd. Safety-Critical. `http://www.savive.com/inform/safetycritical.html` ; Online; accessed 6-August-2010.

22 sgi. Open Inventor™, 2011. `http://oss.sgi.com/projects/inventor/`; Online; accessed 6-July-2011.

23 Wang Shuo, Poturalski Marcin, and Vronay David. Designing a generalized 3d carousel view. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 2017–2020, New York, NY, USA, 2005. ACM.

24 STUK. Finpsa - tool for professional living psa. `http://www.stuk.fi/ydinturvallisuus/ydinvoimalaitokset/en_GB/finpsa/`; Online; accessed 6-August-2010.

25 Alfredo Teyseyre and Marcelo Campo. An overview of 3d software visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):87–105, 2009.

26 Andreas Thums and Gerhard Schellhorn. Formal safety analysis in transportation control. In *Proceedings of the Workshop on Software Specification of Safety Relevant Transportation Control Tasks, VDI Verlag GmbH*, 2002.

27 ViERforES. Vierfores, 2009. `http://www.vierfores.de/`; Online; accessed 6-August-2010.

28 Colin Ware. *Information Visualization.* Morgan Kaufmann, 2004.

29 Markus Weber. A survey of semantic annotations for knowledge management, 2008. `http://www.mendeley.com/profiles/markus-weber/`; Online; accessed 6-August-2010.

30 Josie Wernecke. *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor™, Release 2, Chapter 9. Applying Actions.* Addison-Wesley, 1994.

31 Wikipedia. Alarp. `http://en.wikipedia.org/wiki/ALARP` ; Online; accessed 6-August-2010.

32 Wikipedia. Open Inventor, year = 2011, note = `http://en.wikipedia.org/wiki/open_inventor`; online; accessed 6-july-2011.

33 Wikipedia. Ravon — wikipedia, the free encyclopedia, 2009. `http://en.wikipedia.org/w/index.php?title=Ravon&oldid=278928053`; Online; accessed 23-November-2009.

34 Dirk Zeckzer, Fang Chen, and Hans Hagen. Computing an Optimal Layout for Cone Trees. In Hans Hagen, editor, *Scientific Visualization: Advanced Concepts*, volume 1 of *Dagstuhl Follow-Ups*, pages 11–29. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2010.

2D Tensor Field Segmentation*

Cornelia Auer[1], Jaya Sreevalsan-Nair[2], Valentin Zobel[3], and
Ingrid Hotz[4]

1 Zuse Institut Berlin
Takustrasse 7, 14195 Berlin, Germany
auer@zib.de

2 IIIT – Bangalore, Electronics City, Hosur Road, Bangalore, 560100, India
jnair@iiitb.ac.in

3 Zuse Institut Berlin
Takustrasse 7, 14195 Berlin, Germany
zobel@zib.de

4 Zuse Institut Berlin
Takustrasse 7, 14195 Berlin, Germany
hotz@zib.de

Abstract

We present a topology-based segmentation as means for visualizing 2D symmetric tensor fields. The segmentation uses directional as well as eigenvalue characteristics of the underlying field to delineate cells of similar (or dissimilar) behavior in the tensor field. A special feature of the resulting cells is that their shape expresses the tensor behavior inside the cells and thus also can be considered as a kind of glyph representation. This allows a qualitative comprehension of important structures of the field. The resulting higher-level abstraction of the field provides valuable analysis. The extraction of the integral topological skeleton using both major and minor eigenvector fields serves as a structural pre-segmentation and renders all directional structures in the field. The resulting curvilinear cells are bounded by tensorlines and already delineate regions of equivalent eigenvector behavior. This pre-segmentation is further adaptively refined to achieve a segmentation reflecting regions of similar eigenvalue and eigenvector characteristics. Cell refinement involves both subdivision and merging of cells achieving a predetermined resolution, accuracy and uniformity of the segmentation. The buildingblocks of the approach can be intuitively customized to meet the demands or different applications. Application to tensor fields from numerical stress simulations demonstrates the effectiveness of our method.

1998 ACM Subject Classification I.3.3 [Computer Graphics]: Picture/Image Generation

Keywords and phrases Tensorfield visualization, surface topology

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.17

1 Introduction

Tensor fields occur in engineering and scientific simulations, either as intermediate product or as final result. Mostly, the analysis of the resulting data is based on scalar fields derived from these tensor fields. Since this approach often is insufficient to understand the entire physical process, there is an increasing interest in the analysis of tensor data itself. The wealth of information contained in tensor data, however, induces high data complexity making visualization, analysis, and finally understanding of the data a challenging problem. In

[*] This work was supportet by the DFG Emmy Noether Research Project.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 17–35
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

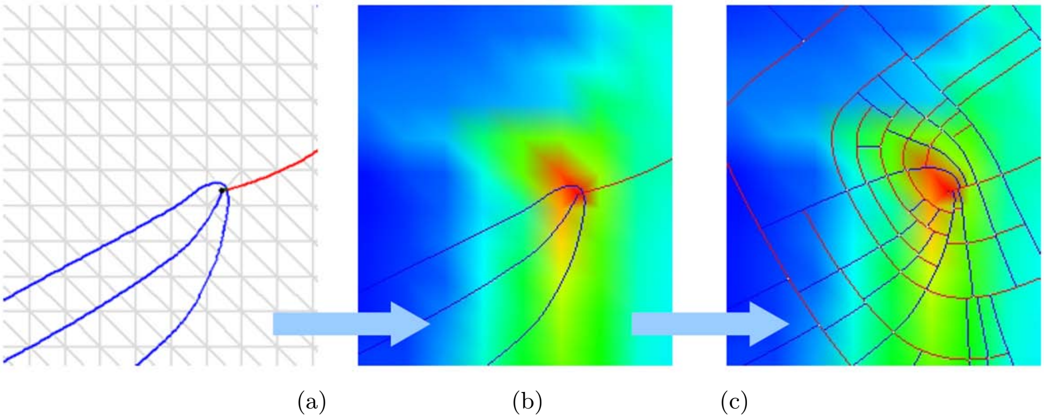


Figure 1 (a) Close up: Extraction of integral topological skeleton for pre-segmentation into cells of equivalent eigenvector behavior. (b) Color coding of scalar field reflecting the eigenvalue fields. (c) Adaptive segmentation into regions of similar eigenvector and eigenvalue behavior.

addition, researchers often do not exactly know what they are looking for, before getting used to the data. This lack of specific questions limits the use of feature extraction methods to reduce complexity. Thus, the goal is to provide an overview of the data without missing important details and without overwhelming the observer at the same time.

An essential step to reduce the amount of information is a segmentation that separates the field into regions of similar characteristic behavior. This higher level of abstraction allows a top-down exploration of the given dataset. Additionally the segmentation can be considered as a basis for visualization techniques using textures and glyphs.

This paper proposes a 2D tensor field segmentation guided by eigenvector and eigenvalue characteristics. Since the tensor is uniquely defined by these invariants, the segmentation gives the domain experts insight into the tensor field as a whole. The segmentation process consists of two steps:

- *Extraction of the integral topological graph considering both eigenvector fields* to provide a pre-segmentation, explained in Section 4. Thereby, the aim is not to show the correct topological structure but to use it as a basic frame, see Figure 1(a).
- *An adaptive segmentation workflow using the eigenvalue fields* to coarsen and subdivide the initial segmentation, explained in Section 5. The workflow consists of a set of building blocks, which can be flexibly combined to meet specific needs of the user or application, see Figure 1(c).

The adaptive refinement process of the segmentation is guided by the definition of a scalar invariant as similarity or dissimilarity measure, see Figure 1(b). Depending on the application, a variety of scalar invariants can be used. For example, anisotropy and maximum shear stress reflect the relation of eigenvalues and are of high importance in many applications. A generalization of the notion of anisotropy to non-positive definite fields allows us to extend our analysis to all tensor fields. In our approach, several dissimilarity measures can be flexibly applied either by themselves or as combinations.

The resulting segments themselves in the final segmentation can be considered as visualization glyphs in form of tiles. They are bounded by tensor lines which allows immediate interpretation of the eigenvector behavior within and color coding renders the eigenvalue characteristics in the segments. The geometry of the resulting segments is represented explicitly and offers statistical enquiry of properties inside each cell.

Section 5 explaines the basic concept by the implementation of a focus+context visualization. The segmentation is adapted to the demanded accuracy concerning eigenvector and -value similarity and to the given resolution of the displayed domain. Finally the extraction of *degenerate regions* in the field demonstrates the flexibility of the concept. Building blocks of the basic approach are adapted to a strategy capturing these often numerical unstable entities in tensor fields. The directional expression is weak, the only interest in these regions is in the isotropic behavior of the eigenvalues.

There has been a lot of recent research in extraction, simplification and visualization of tensor field topology, on which this work builds. Although, these methods extract valuable structural information from the eigenvector fields, they ignore the importance of the eigenvalues. The gap caused by the lack of interpretation of features in the eigenvalue field leads to decreased use of tensor topology in analysis. This work bridges this gap by providing a complete interpretation of the tensor field using the features present in both, the eigenvector as well as the eigenvalue fields. Its effectiveness is demonstrated using data sets from structural engineering, see Section 6.

2 Related Work

Still, the most common analysis methods for tensor fields are built on derived scalar fields. While this approach is often helpful it is not always sufficient. Due to the demand for tools representing the entire tensor information a variety of visualization methods have been developed. Since tensor fields have very application specific characteristics, these methods often are designed for concrete applications. Most efforts have been put into tensors from diffusion tensor imaging (DTI) and mechanical engineering applications, which is the focus of our work. In this area existing methods can roughly be classified into glyph-, texture- and topology-based methods.

Glyphs represent a direct visualization approach displaying tensor values in selected points. Related research issues are focused on the definition and placement of glyphs. Glyphs that are commonly used are ellipsoids, Haber glyphs [8], or superquadrics [12]. Different placement strategies are used to maximize the information displayed per image [7, 13]. A representation of tensor values on one-dimensional lines are hyperstreamlines. They are strongly related to streamline methods used for vector fields. They were introduced by Delmarcelle and Hesselink [6] and have been utilized in a geo-mechanical context by Jeremic et al. [11]. While glyphs are appropriate for characterizing single tensors, they are limited to low resolution and fail to give insight into the structure of the entire field. A more continuous view onto 2D fields can be obtained using tensor splats [3], or textures based on line integral convolution [9, 24].

For DTI , a lot effort has been put into tensor field segmentation, mostly with the goal of brain segmentation. Extending methods from image segmentation and clustering, the central research topic is the definition of an appropriate dissimilarity measure for tensors. Proposed methods range from active contours [21] and level sets [26] to graph-cut algorithms [23, 27]. Used metrics are the angular difference between principle eigenvector directions, or standard metrics considering the entire tensor, like the Euclidean or Frobenius distance. Recently, Wang et al. [21] introduced a distance measure from information theory designed for Gaussian distributions. Although, it is a good representation of the diffusion tensor characteristics, it is limited to positive definite tensors. A segmentation designed for meshes based on the curvature tensor was introduced by Lavoue et al. [15]. Vertices are clustered according to their principal curvature values using a k-means classification. The boundaries of resulting cells tend to be parallel to lines of minimum curvature but do not exactly represent the principal directions.

Methods concerned with the segmentation of general tensor fields are based on tensor field topology. They concentrate on the structure of the eigenvector fields neglecting the scalar entities. The idea of using topological methods to analyse the structures of 2D tensor fields goes back to Delmarcelle [5] and Lavin et al. [14] and builds the basis for the method proposed in this paper. They have introduced the topological skeleton consisting of degenerate points and connecting tensor lines as central features. Following this work, much effort has been put in simplifying and tracking of the resulting structures [19]. Alliez et al. [2] have proposed an application to curvature tensors for polygonal remeshing of surfaces. Zheng et al. [25] have initiated work in 3D tensor topology. Their analysis shows that in three dimensions degenerate features form one-dimensional structures. An eigenvector-based interpolation as basis for the topology extraction is proposed in [10]. An integral topological skeleton using both eigenvector fields has been used for a directional field segmentation in [17].

3 Basics and Notations

This section summarizes the basics for this paper. Definitions and notations are restricted to 2D tensor fields of second order, since they are the topic of this work. For a more complete and formal definition of tensors we refer readers to [22].

3.1 Tensors and Tensor Field

A tensor is a type of geometrical entity that generalizes the concept of scalars, vectors, and linear operators in a coordinate-independent fashion. With respect to a given basis of $I\!R^2$, a tensor $\mathbf{T}$ can be expressed by a 2×2-dimensional matrix of real numbers. $\mathbf{T}$ is called symmetric if for any coordinate basis, the corresponding matrix is symmetric. A tensor field over some domain $D \subset I\!R^2$ assigns to every point $P \in D$ a tensor $\mathbf{T}(P)$. In the rest of the paper, we will refer to symmetric 2D tensors of second order as tensor.

A tensor $\mathbf{T}$ is fully represented by its *eigenvalues* λ, μ and corresponding *eigenvectors* $\overset{\leftrightarrow}{v}$ and $\overset{\leftrightarrow}{w}$, implied by the eigenvalue equations $\mathbf{T}\cdot \overset{\leftrightarrow}{v}= \lambda\cdot \overset{\leftrightarrow}{v}$ and $\mathbf{T}\cdot \overset{\leftrightarrow}{w}= \mu\cdot \overset{\leftrightarrow}{w}$. The names λ and μ are assigned in a way, such that always $\lambda \geq \mu$. Since the multiplication of an eigenvector by any non-zero scalar yields an additional eigenvector, eigenvectors should be considered without norm and orientation, which distinguishes them from classical vectors. We use $\overset{\leftrightarrow}{v}$ and $\overset{\leftrightarrow}{w}$ when referring to eigenvectors to allude to the fact that they are bidirectional. We use $\mathbf{v}$ and $\mathbf{w}$ when referring to vectors representing normalized eigenvectors with an arbitrarily but fixed direction, e.g., using the unmodified results of the numerical computation used to generate them. The direction of $\mathbf{w}$ is defined in such a way that $\mathbf{v}$ and $\mathbf{w}$ form a right-handed system. Eigenvalues are computed by solving the characteristic equation, which is a quadratic equation in λ: $|\mathbf{T}-\lambda\mathbf{I}| = 0$, where $\mathbf{I}$ is identity matrix. For symmetric tensors, the eigenvalues are real, and the eigenvectors are mutually orthogonal. The eigenvector corresponding to the larger eigenvalue is called major eigenvector. Analogously, the eigenvector corresponding to the smaller eigenvalue is denoted as minor eigenvector. If both eigenvalues are positive, the tensor is called positive definite. Examples for positive definite tensor fields are diffusion tensor fields. Stress and strain tensor fields are in general not positive definite.

Integrating the eigenvector fields results in two orthogonal families of continuous curves. These curves are called *major* (red) and *minor* (blue) tensorlines according to the eigenvector field integrated. These tensorlines are used to bound the cells of the segmentation.

3.2 Tensor Field Topology

Similar to vector fields the structure of eigenvector fields is represented by their field topology. It defines a skeleton consisting of distinguished points (degenerate points), and connecting edges (separatrices). In this work, the topology of both eigenvector fields is considered as one integral tensor field topology. In the following, we shortly resume the basics of tensor field topology, concentrating on the aspects that we need later on. A more detailed discussion on this topic is given in [5, 17, 20].

Degenerate points – definition At most points in a tensor field, both eigenvectors are defined uniquely; each assigned to one eigenvalue. However, this is not the case for points with identical eigenvalues, that is $\lambda = \mu$. These points are called *degenerate* or *isotropic points*. This means the tensor is proportional to the identity matrix and all vectors are eigenvectors. They are the only location where tensorlines of the same color can intersect. Mostly, degenerate points appear as isolated points but also degenerate features of higher order are possible. These are degenerate lines and triangles. Degenerate points in tensor fields are equivalent to critical points in vector fields. However, due to orientation indeterminacy of tensorlines, these points exhibit structures that are different from the structures seen in vector and scalar field topologies, respectively.

Degenerate point – classification The field behavior in the vicinity of degenerated entities is characterized by a number of specific sectors. These sectors are separated by distinguished tensorlines which enter the degenerate point radially. The following behavior is possible, see Figure 2:

- A *hyperbolic sector* is bounded by one red and one blue radial line: Tensorlines in this sector approach, sweep past the degenerate point and leave the sector through one bounding radial line.
- A *parabolic or radial sector* is bounded by two radial lines of the same color: In this sector tensorlines of this color, start from the degenerate point and then diverge. Tensorlines of the opposite color enter and leave the sector through the bounding lines.
- An *elliptic sector* is bounded by one red and one blue radial line: Tensorlines in this sector start from the degenerate point, and leave the sector through one of the bounding lines.

A criterion to classify the sectors is the rotation angle of the eigenvectors $\Delta\alpha$, in comparison to the opening angle of a sector $\Delta\Theta$, for more details we refer to [17, 18].

$$\Delta\alpha \quad = \quad \begin{cases} \Delta\Theta & \text{radial, concentric} \\ \Delta\Theta - \pi/2 & \text{hyperbolic} \\ \Delta\Theta + \pi/2 & \text{elliptic} \end{cases} \tag{1}$$

Separatrices Radial tensorlines bounding hyperbolic sectors are called *separatrices*. They constitute the edges of the topological graph. The graph defined by the two eigenvector field builds the basis for the following segmentation algorithm.

3.3 Interpolation

Usually, tensor datasets represent a discretized tensor field, given on uniform or non-uniform grids which we store on a triangular mesh. For the extraction of topology and the integration of tensorlines, a linear eigenvector and eigenvalue interpolation is used [10]. The insertion of new vertices in degenerate points makes sure that a consistent eigenvector interpolation is

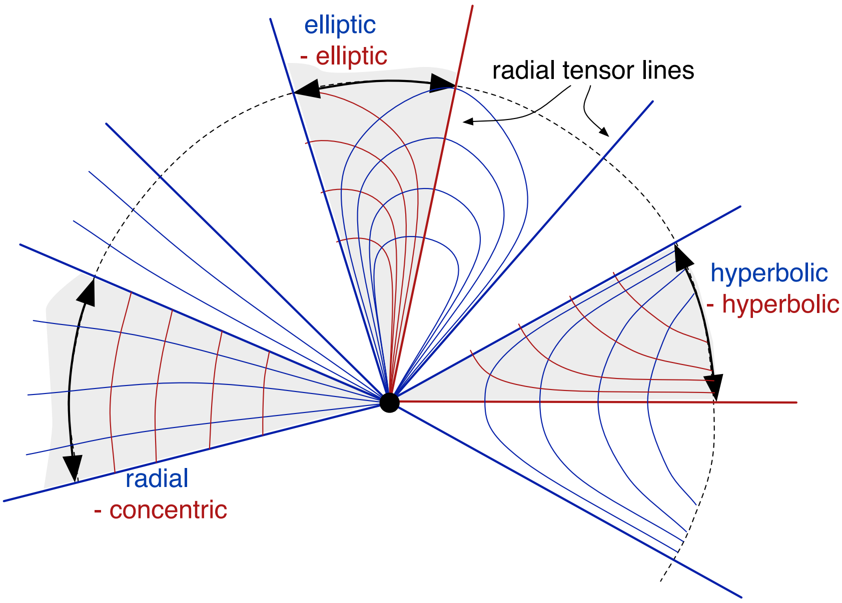


Figure 2 The neighborhood of a degenerate point is characterized by a number of sectors with specific behavior.

possible inside each triangle. The eigenvectors at these points are set to zero. The problem of eigenvector orientation is resolved by introducing edge labels, which encode the relative orientation of the calculated eigenvectors **v** and **w** in adjacent vertices. After computing these edge labels once, simple vector interpolation can be performed inside the triangles. The additional vertices together with re-triangulation lead to an increased number of degenerate entities of higher dimensionality when compared to linear interpolation of tensor components. This interpolation method has been chosen for performance reasons. It minimizes the number of eigenvector computations and makes an exact integration of the tensorlines possible. The interpolation can be easily replaced with any other consistent tensor interpolation, leading to slightly different results.

4 Initial Segmentation

In this section, we describe the steps leading to the initial segmentation of tensor fields using the topological skeleton. The degenerate points and intersections of the separatrices in both major and minor eigenvector fields define the vertices of the cells and the separatrices form the edges.

4.1 Topology Extraction

We extract the topology of the tensorfield using linear interpolation of eigenvector fields, given in Section 3.3. Alternatively, one can use other interpolation schema, e.g. [19]. Here, we summarize the steps taken for the topology extraction, which is described in detail in [17].

Location of degenerate points – The product of the edge labels, introduced for the interpolation, offers a simple criterion to detect triangles that contain degenerate points. The location of an interior degenerate point only depends on the eigenvector directions in the vertices

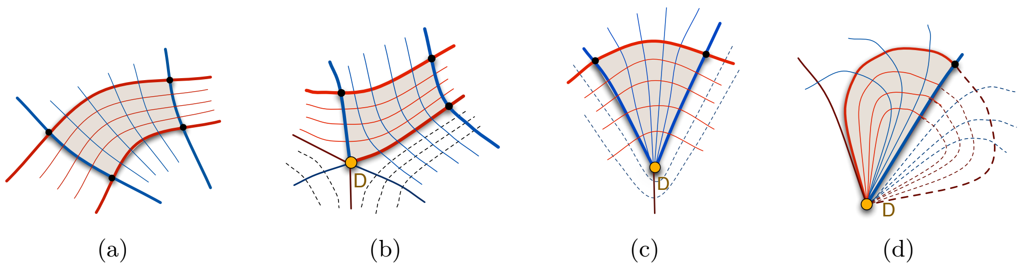


Figure 3 Cells defined by the topological skeleton: (a) regular cell without any degenerate points, (b) hyperbolic sector, (c) parabolic sector, and (d) elliptic sector.

$P_i, i = 1, 2, 3$ of the triangle. It is defined as the intersection of connections of the vertices and their so-called opposite points O_i. These are points on triangle edges e_i opposite to P_i, in which the eigenvector directions $\mathbf{v}_{oi}$ and $\mathbf{w}_{oi}$ are orthogonal to the eigenvector directions $\mathbf{v}_i$ and $\mathbf{w}_i$.

Determination and classification of radial directions – The neighborhood of a degenerate point is characterized by segments separated by radial tensorlines. For linear eigenvector interpolation, the radial tensorlines are straight lines defined by their intersection with the edges of the triangle. For the skeleton computation, only the radial lines which are boundaries of hyperbolic sectors are relevant. The classification is computed using Equation 1.

Non-isolated degenerate points – Degenerate entities like degenerate lines, polylines and triangle, can be treated similar to isolated degenerate points. The eigenvector field inside triangles adjacent to degenerate lines or triangles is constant. Thus, from a structural point of view, it is enough to consider the vertices of the polylines line as degenerate points. The sectors can be classified using the same angle criteria as for isolated degenerate points. While these configurations are rather rare in tensor fields for the component-based interpolation, they appear frequently for the eigenvector-based interpolation.

Separatrix computation and termination conditions – To complete the topological skeleton, relevant radial directions are integrated. The tensorline evaluation is done triangle-wise using 4th order Runge-Kutta. Eigenvector-based interpolation also allows an exact integration [16]. Following termination conditions are implemented to obtain a cleaner skeleton: (a) The separatrix leaves the domain. (b) The separatrix gets close to a degenerate point, line or triangle. (c) The separatrix describes a circle or spiral and passes itself closely in parallel integration direction. Separatrices are stored as polylines.

4.2 Cell Generation

After computing the topological skeletons for the major and minor eigenvector fields the intersections of the red and blue separatrices define the cells of the pre-segmentation. To increase the efficiency of these computations, every triangle keeps track of all separatrices passing through it. Thus, only the triangles that contain at least one red and one blue line are considered to compute the intersections. The vertices of the resulting curvilinear cells are either red-blue intersection points, degenerate points, or intersections of tensorlines with the boundary. They exhibit one of the following basic structures, see Figure 3:

1. Cells without a degenerate point are quadrangular with two red and two blue tensor lines

as boundary, in an alternating order. All red tensor lines passing through this segment enter at one blue boundary and leave the cell at the opposite boundary. All intersection angles are orthogonal.

2. Cells with one degenerate vertex lying in a hyperbolic sector are quadrangular. The angle at the degenerate point is in general not orthogonal.
3. Cells having a degenerate point in one vertex, lying in a parabolic segment, degenerate to a triangular shape.
4. In elliptic sectors, cells with either two or three vertices are possible.
5. Cells containing degenerate lines as edges can exhibit all kinds of complicated structures.

The edges of the cell are segments of the separatrices and hence are represented as polylines. The edges are ordered in counterclockwise orientation of the cell, and stored in a doubly-linked list, for efficiency in finding neighbors to the cell and adjacent edges in a cell. Each edge is represented using a half-edge data structure.

4.2.1 Half-edge Data Structure

A half-edge data structure [1, 4] is an edge-centered data-structure that maintains spatial information of vertices, edges and cells. Each edge is shared by two cells. An edge can also be considered as two opposite directed half-edges, called twins. Each half-edge stores its start point, the end point of a half-edge however is determined indirectly by referencing to the start point of the twin. The prime advantage of using this data structure is that a half-edge and its corresponding cell share a one-to-one relationship. Consequently, neighbor-searches and an iteration through the cells become very efficient.

Half-edge twins always belong to the same separatrix, except in the cases when the edges are part of either boundaries or degenerate lines. As separatrices are represented by polylines, the half-edge data structure is represented by polylines. Our implementation using CGAL [1] additionally has to support irregularities in the cell layout, namely T-junctions or hanging nodes, where twins have an n:m relation, such that they share common points of the separatrix polyline but do not share the same start and end points, see Figure 4(a). To resolve the issue of continuities in irregularities, our half-edge data-structure is modified as follows: (a) We store pointers to points representing the current edge; (b) Two sets of twins are supported for each edge - (i) *a geometric twin*: a single edge to identify the geometrical limits of the edge and (ii) *neighboring twins*: an array of twins to identify all neighboring cells in case of hanging nodes. The geometric twin of an edge is the flipped image of the edge with respect to its starting and end points, which would ideally be the twin but necessarily need not exist in the topological skeleton. The neighboring twins of an edge is the segmented set of the first twin, which are the edges that actually exist in the skeleton. In the absence of hanging nodes, the second set is a singleton set of the first twin.

4.2.2 Creating Cells from Topological Skeleton

The actual cell creation process involves physically creating the half-edges from the topological skeleton, and using them to build the curvilinear cells. Starting with a single cell as a seed cell, its neighborhood is grown to find the entire set of cells. Convex cells can be found by a strict rotation angle criteria at the vertices in a counterclockwise orientation. The remaining non-convex cells in the vicinity of degenerate lines or triangles are found by implementing a greedy walk of finding consecutive half-edges that are not associated with any cells. Consecutive half-edges are all that have the current half-edge's end point as start point.

5 Adaptive Segmentation Workflow

The segmentation resulting from the topology already decomposes the domain in regions where the eigenvector fields have a qualitatively similar behavior, but it does not yet fulfil all our criteria for a good segmentation. To represent the entire tensor information also the scalar invariants based on the eigenvalues have to be considered. This is achieved by adaptively modifying the cells, characterized by a specified degree of similarity with respect to eigenvalue behavior.

The segmentation strategy on the initial cell structure builds on two basic operations:

- **Coarsening**: Cells that do not exhibit enough structural information on their own get merged with adjacent cells.
- **Subdivision**: Cells which exceed the defined criteria of similarity are subdivided by new tensorlines.

Due to divergence and convergence of tensor lines, adaptive segmentation inevitably causes occurrences of hanging nodes in the edges. To keep these irregularities to a minimum, a growing strategy where we continue to merge or subdivide on consecutive cells as long as possible and necessary, see Figures 4(b,c). Algorithms for coarsening and subdivision operations are described in Section 5.3.

To guide the modification process by eigenvalue behavior one or more scalar fields are derived from the initial tensor field, which directly render the eigenvalue behavior, see Section 5.1. The degree of similarity and the need of modification is represented by weight functions defined on the edges of the cells, see Section 5.2. These edge-weights evaluate the derived scalar field but also reflect geometric properties of the cells. Combined with data dependent thresholds the edge-weights serve as decision basis whether cells have to be merged or subdivided. Further the edge-weights help to steer the modification by importance, as their values directly offer prioritisation to achieve a smooth segmentation. Using edge-weights for decision-making is an efficient choice as edges are one-dimensional structures, on which the weights are computed.

To extend the capabilities of the adaptive segmentation process, it is designed to have a high degree of flexibility to customize the workflow. For example using this segmentation as preprocessing step for glyph placement would have different demands than using it for texture mapping. We define the variables for customizing the workflow of the adaptive refinement as:

- **Operations:** The operations of the refinement, namely coarsening and subdivision, are the modules of the workflow. They can be repeated and the workflow customized by choosing the number and the order of implementation of the operations.
- **Control Parameters:** The control parameters of the refinement are the chosen edge-weights, the considered scalar fields and the thresholds defined by the demanded accuracy and resolution. The choice of these control parameters impact the priority queues used for the implementation of the operations.

The remainder of this section first presents the operations and control parameters of the approach. Then the basic workflow is demonstrated by calculating a segmentation of the tensor field. The segmentation follows accuracy towards tensor invariant similarity as wells as geometric criteria. Implemented as focus + context visualization, the refinement process is stopped for cells whose size falls below a value proportional to the resolution of the displayed domain. This provides an overview of the field, on demand the user can specify a focus

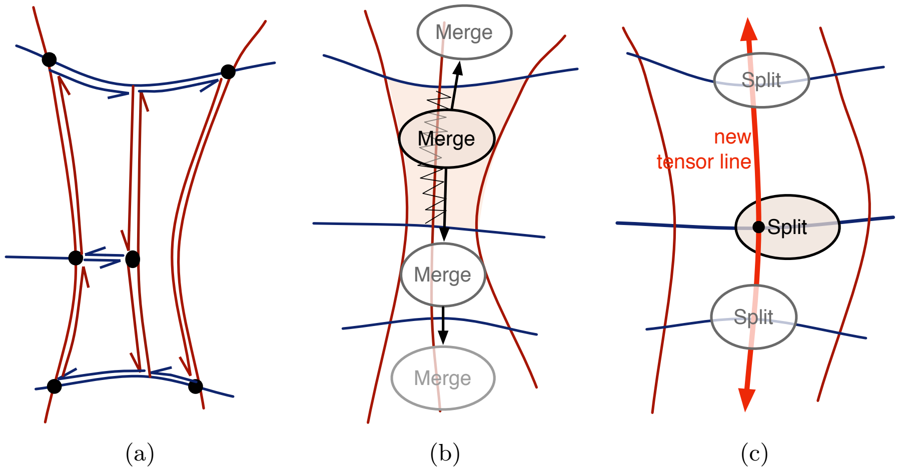


Figure 4 (a) T-junction or hanging node, where neighboring twins in half-edge structure have an n:m relation. Adaptive refinement operations: Recursive strategy for avoiding hanging nodes in (b) coarsening of cells, and (c) subdivision of cells started by insertion of new tensor line.

region to view further detail, see Section 5.5. The flexibility of the approach is finally shown by the extraction of degenerate regions, see Section 5.6.

5.1 Choice of Scalar Field

Several scalar fields can be considered as basis for the refinement depending on the specific application. We can either use both eigenvalue fields, an anisotropy value, the maximum shear stress (which is related to the anisotropy) or other tensor invariants.
Maximum shear stress S is defined as

$$S = |\lambda - \mu| \tag{2}$$

For the anisotropy we propose a generalized notion of fractional anisotropy FA^*, which can also be used for non positive definite tensors.

$$FA^* = \sqrt{\frac{(\lambda - \mu)^2}{\lambda^2 + \mu^2 + A^2}} \tag{3}$$

The positive constant A is added to the denominator, which eliminates the discontinuity close to zero for $\lambda, \mu \in IR$. This results in low anisotropy values for tensors with eigenvalues of different sign but small absolute value. The values of FA^* range from 0 to 1.

5.2 Edge-weight Definition and Thresholds

A set of pre-defined functions as weights assigned to the cell edges is provided. It consists of geometric measures representing the current cell size and shape as well as similarity measures for scalar fields derived from the tensor field. This set can be extended by user-defined functions. The dominant use of geometric edge-weights favours a more uniform segmentation, whereas the scalar field based weights lead to a higher adaptivity towards accuracy in eigenvalue similarity. In the following two weights of each class are proposed. These weights can be arbitrarily combined.

Let e be an edge of a cell consisting of k segments $(\mathbf{x}_i, \mathbf{x}_{i+1})$, by the virtue of being part of a polyline, where $\mathbf{x}_i$ is the position of the ith point on the edge. Further, let s be a scalar function defined along the edge and $s_i = s(\mathbf{x}_i)$.

- *Variance of scalar values $s(\mathbf{x}_i)$ on the edge $w_v(e)$:*
  $w_v(e) = \frac{\sum_{i=1}^{k}(s_{x_i}-\bar{s})^2\|\mathbf{x}_{i+1}-\mathbf{x}_i\|}{\sum_{i=1}^{k}\|\mathbf{x}_{i+1}-\mathbf{x}_i\|}$, where $\bar{s}$ is the mean of s along e.
- *Absolute difference of minimal and maximal scalar value along the edge $w_d(e) = abs(s_{min} - s_{max})$.*
- *Edge length $w_l(e) = \sum_{i=1}^{k}\|\mathbf{x}_{i+1} - \mathbf{x}_i\|$.*
- *Change of eigenvector direction along the edge $w_c(e)$:*
  $w_c(e) = \sum_{i=1}^{k}|\angle(\mathbf{v}_i, \mathbf{v}_{i+1})|$, v_i is the major eigenvector at position x_i.

We chose the proposed edge-weights to be as intuitive and universal as possible, independent of the various ranges that appear in different datasets. *Variance* is a commonly known statistical quantity and is a similarity measure which is robust to smaller perturbations of scalar values along an edge, such as noise. *Difference of minimal and maximal scalar value* in turn is strict towards any changes of scalar values along an edge and directly renders the absolute difference of scalar values appearing on an edge. *Edge length* can be used to adjust the size of the segmented cells to optimize perceivability by the user. The eigenvector directions are already well represented by the cell shape and tensor line boundaries, however if uniformity of the cells is required *change of eigenvector direction* represents the curvature of the cell boundaries and is therefore an appropriate measure.
In Section 5.5 we give a preset of thresholds for these weights, which are calculated as percentages of the given ranges in the field. These presets are universal and lead to stable results of good quality, which experiments with different datasets showed, see Section 6. However they can be intuitively strictened or loosened for different visualization purposes with immediate interpretation.

5.3 Refinement Operations

If not noted differently the refinement operations always respect the chosen thresholds towards the edge-weights. For example if a minimum edge length for the coarsening operation is specified - no edge subdivision is performed if one of the new edges would fall below the minimum edge length.

5.3.1 Coarsening

The main goal of the coarsening operation is to get rid of small cells that do not carry enough structural information on their own. Coarsening operation involves merges of cell pairs. Merging a pair of cells requires the merge of up to two pairs of edges and removal of the common edge of the cells. For this operation, we build an edge-weight based priority queue of pairs of cells that can be merged. We use queues to follow the FIFO (*first in, first out*) order, ascending or descending priority is fixed by minimum or maximum thresholds respectively.

Merge Prerequisites – Based on the geometry layout, two cells can only be merged if they share *a common edge* that can be deleted to join these cells. Technically, a common edge between two cells means that one of the cells has an edge whose geometric twin is an edge of the second cell. Edges containing hanging nodes cannot be common edges.

Priority Queue and Sorting – For the coarsening operation a priority queue of pairs of adjacent cells that can be merged is maintained. A multi-pass sort is performed based on the chosen edge-weights. Not only the smallest cells should be merged first but also for each cell two neighboring cells are candidates for merge (except for boundary cells). For the example of the segmentation workflow in Section 5.5 the priority queue is first sorted by minimum edge length of the edges involved and then by maximum edge length of the edges to be merged.

Algorithm

- Check adjacent cells for if they can be merged and sort these into the priority queue, based on the chosen edge-weight prioritization.
- While the priority queue is not empty, the pair with the highest priority is merged.
- Update the data structure by merging the appropriate edges of the pair cells, deleting the common edge and creating a new cell from the new edges.
- Update the priority queue with the new merged cell.
- If contiguous cell pairs in direction of the deleted common edge are to be merged move them on top of the queue to accomplish the recursive workflow. See Figure 4(b).

5.3.2 Subdivision

Single cells are subdivided by starting a new tensor line of opposite color on one of its edges that has to be subdivided. Similar to the coarsening operation an edge-weight based priority queue implemented as FIFO.

Start Point of Subdividing Tensor line – Two possibilities for the start point of the new tensor lines are provided. The first option favours the generation of equally sized cells, and starts the tensor line in the midpoint of the edge. The second option starts the tensor line between the extrema of the scalar values on the edge. This choice is more adapted to the data and guarantees to decrease the edge-weight when subdividing. There are no technical prerequisites to subdivide a cell.

Priority Functions and Sorting – Differently from the coarsening operation, a priority queue for edges is used rather than cells. Again the priority queue can be sorted according to multiple edge-weights. The growing strategy in subdividing consecutive cells is implemented by integrating a subdividing tensor line as long as possible and necessary, see termination conditions (a,b) in the algorithm below. No explicit prioritisation has to be done.

Algorithm

- Sort edges to be subdivided into priority queue, based on chosen edge-weights.
- While the priority queue is not empty, pop the top edge and start a subdividing tensor line of opposite color.
- Integrate tensor line until one of the following termination conditions is reached:
  (a) It intersects an edge, which is not in the priority queue and therefore should not be subdivided.
  (b) It intersects an edge and it's subdivision would generate edges violating fixed edge weight thresholds, for example minimum edge length.
  (c) It fulfils one of the termination conditions described in Section 4.1.
- Subdivide all cells corresponding to edges intersected by the new tensor line, as shown in Figure 4(c). Update data structure by subdividing intersected edges, generating new

edges along the tensor line, and finally generating new subdivided cells using the new edges.

- Update priority queue by deleting the original edges intersected by the new tensor line, and adding and sorting the newly generated edges if they are candidates for further subdivision.

5.4 Customized Workflow of Adaptive Refinement

The possibility to customize the workflow gives a high degree of flexibility in obtaining various analyses of the same dataset. Essentially the workflow consists of modules for operations, which are influenced by the control parameters and strategies adopted for implementation. Variations in the workflow are achieved by changing the number and order of the modules, by adjusting the thresholds used for each operation, and by deciding on the strategies to be used for the control flow of the modules. Strategies include the choice of appropriate edge-weights and scalar fields and choice of position of starting a new tensor line for subdivision of edges.

Table 1 Table to summarize the options when configuring for the segmentation process.

Basic operations	coarsening	subdivide
Edge-weight prioritisation	geometric	scalar field
Error measure for edge-weight definition	variance	max difference
Tensor line seeding	middle of edge	between max and min
Level of detail	resolution	accuracy

For a domain expert the flexibility of the approach ranges from using the presets with the scalar field of his choice over strictening or loosening thresholds to mixing and matching the implemented components to his needs. Developers can extend the basic set by implementing new elements, as e.g. edge-weights or tensorline seeding.

5.5 Workflow: Basic Segmentation

This workflow delivers a focus + context visualization, calculating an initial context segmentation which can be browsed in detail by selecting a focus region. The field is segmented in regions of similar tensor invariant behavior. We chose FA^* (see Section 5.1) as scalar field to render the eigenvalue characteristics. For all operations the same edge-weights and thresholds are used.

The default parameters are, if not differently noted:
(a) Geometric edge-weight steering resolution is the edge length w_l with minimum edge length threshold fixed to $\varphi_l = 1\%$ of the displayed domain range. By selecting a focus region the minimum edge length is automatically adjusted and the segmentation is refined displaying further details.
(b) Scalar field edge-weight steering accuracy is the absolute difference of minimum and maximum scalar value w_d, with its threshold set to 10 % of the scalar value range, which is for FA^* as scalar field φ_d= 0.1.
Resolution and accuracy are the basic level of detail parameters for focus + context visualizations, where the geometric edge-weight has higher priority than the accuracy edge-weight. This means for edge lengths smaller than the fixed minimum edge length a merge is performed even if the merged edge exceeds the given accuracy threshold.

The following operations composite the workflow

1. First coarsening: in merging as many similar cells as possible cleans up the pre-segmentation , especially very small cells are removed. The priority queue is first sorted by minimum edge length of the edges involved and then by maximum edge length of the edges to be merged. Merging of small cells with rather large cells favours the goal of a smooth segmentation.
2. Subdivision: the cells are refined to the pre-defined accuracy, unless this violates the resolution criterion. The tensor line seeding is between the extremal points. Experiments showed that the best strategy for a smooth segmentation is to do a 2-pass sort of the priority queue first based on maximum edge-length, the second pass based on maximum scalar edge weight.
3. Final coarsening: cells with highest similarity are merged. The priority queue is sorted first by minimum edge length of the edges involved and then in ascending order by a pre-calculated scalar edge weight w_d of the edges to be merged.

For a chosen focus step 2 and 3 are repeated with adjusted geometric edge weight, the accuracy edge weight remains. As the cells are given as explicit entities, cells exceeding the accuracy edge weight can be highlighted on demand.

This workflow and thresholds can be used in any tensor field segmentation as stable presets. Results for using *variance* as scalar field edge-weight are given in Section 6.

5.6 Workflow: Degenerate Regions

In the direct vicinity of degenerate entities tensors are almost isotropic and thus directional behavior is not strongly expressed. Such areas are collectively represented as *degenerate regions*, following the paradigm of the coarsening operation, that all cells are combined which do not exhibit enough structural information on their own. These degenerate regions are "grown" from the degenerated entities using two passes of the subdivision operation. FA^* is used as scalar field.

1. First subdivision: In the first pass only edges, called *degenerate edges*, emerging from a degenerate entity are considered for subdivision using a specific edge-weight, which is the maximum anisotropy occurring on the edge. The start point for the subdividing tensor line is the point on the edge with anisotropy below a fixed threshold $\varphi = 0.05$ and with maximum distance from the degenerate entity. prioritisation is done by the least anisotropy value of the start points.
2. Second subdivision: As it cannot be guaranteed that the subdividing tensor lines will intersect other degenerate edges at points with anisotropy below φ, a second subdivision pass is performed, where all edges participating in the intermediate degenerate regions from the first subdivision step are considered.

It should be noted that the generation of degenerate cells may induce subdivision of neighboring cells that may result in small not well shaped cells, which may not get merged in a later step, as seen in Figure 9. It is still of benefit to extract degenerate regions, as the weak expression of direction in such areas can result in numerical instabilities.

6 Results and Discussion

We tested our algorithm on three datasets from structural engineering, which are finite element simulations of forces acting upon solid blocks resulting in stress tensor data. These

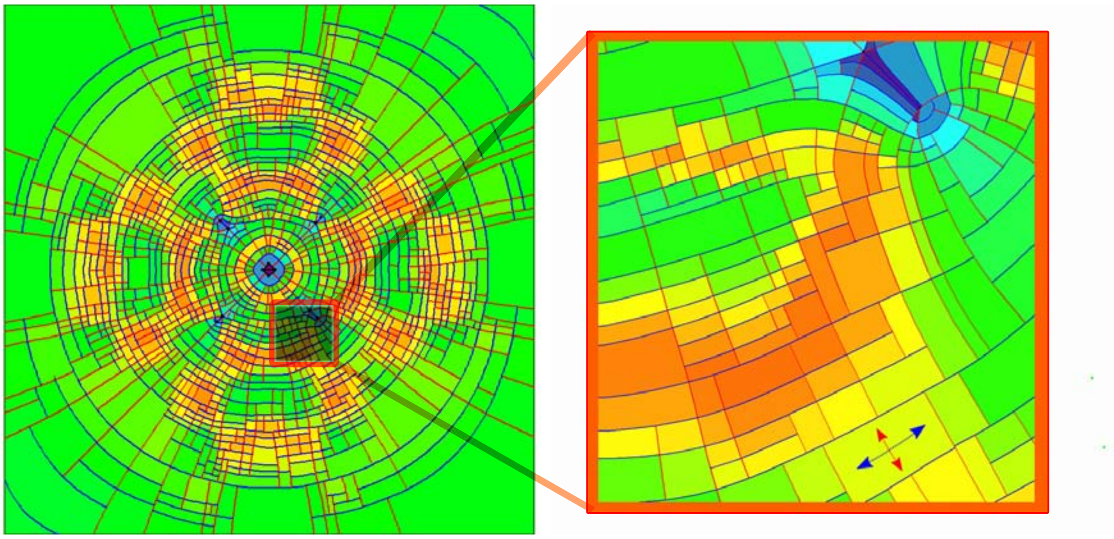


Figure 5 Focus + context visualization.

are simulations of one and two forces applied to the top of a solid block and of multiple forces applied to a notched block, the latter using hp-adaptive finite elements. In this section, we will refer to them as *one point load* (1PL), *two point load* (2PL), and *notched block* (NB), respectively.

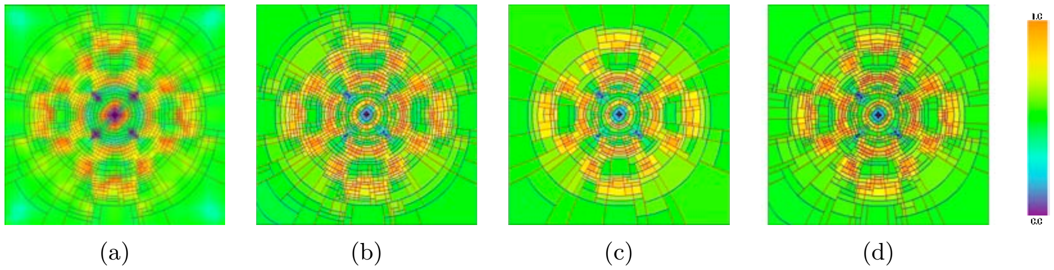
Figure 5 shows the focus + context implementation proposed in Section 5.5 on the 1PL dataset. The right image shows the selected focus further refined with the resolution threshold, minimum edge length, adapted to 1% of the displayed domain range, the accuracy threshold remains. In this visualization the cells delineate regions of similar eigenvalue behavior, the color coding of the cells renders the relation of the eigenvalues. The two arrows in the lower right corner of the focus image indicate exemplary how the eigenvector behavior can be interpreted from the cell boundaries.

We conducted further analysis on thresholds, specific control parameters and strategies. To evaluate the quality of the overview segmentation we used the following methods:

- *Image representation of local error*: The error is sampled in an image of resolution $u \times v$ where each pixel (i, j) is mapped to a point (x, y) in the tensor field, and is assigned a color from a red-shaded colormap mapping to error value $err(i, j) = \|s(x, y) - \bar{s}_{cell(x,y)}\|$, where $cell(x, y)$ is the cell containing (x, y) and $\bar{s}_{cell(x,y)}$ is the average scalar value on the edges of cell $cell(x, y)$. Results are shown in Figure 8(a) and 7.
- *Average error:* While computing the image representation of the local error, we calculate the average error for the field as $\oslash Err = \frac{\sum_{i,j} err(i,j)}{u*v}$.
- *Number of cells needed for pre-defined quality:* We aim to have as few cells as possible in an adaptively segmented field, which makes the number of cells needed to achieve a segmentation of predefined quality an important criterion.

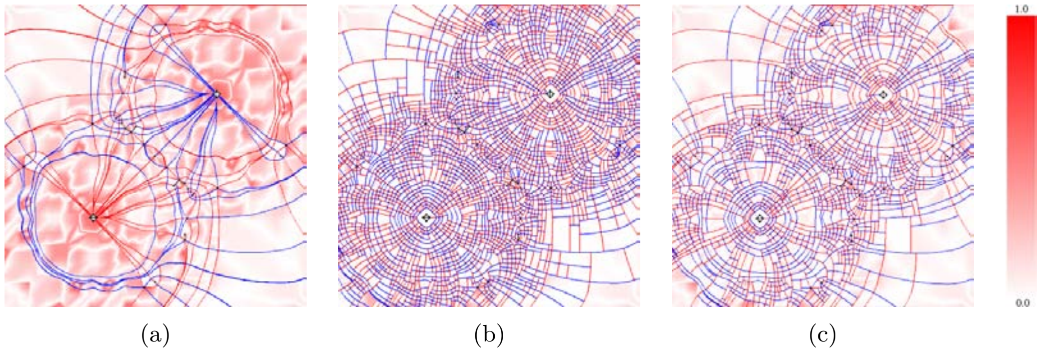
The first example examines level of detail according to accuracy, see Figure 6. In Figure 6(b) φ_d is twice as strict as for Figure 6(c). Using superimposition of the original scalar field on segmentation result of the one point load dataset, Figure 6(a) demonstrates that the segmentation and original scalar field match.

The second example focuses on resolution as level of detail. In Figure 7(b) the minimum



(a) (b) (c) (d)

Figure 6 Adaptive segmentation of a slice in the one point load dataset. (a) Superimposition of scalar field on adaptive segmentation of tensor field. Choice of threshold for edge-weight φ_d of value (b) 0.1 and (c) 0.2 for customizing the workflow for segmentation. (d) Segmentation with edge-weight variation and $\varphi_v = 0.015$



(a) (b) (c)

Figure 7 Image representation of error for a specific slice of two point load dataset. (a) Initial segmentation of tensor field. (b) Adaptive segmentation using w_d as edge-weight. (c) Adaptive segmentation using double threshold for minimum edge length.

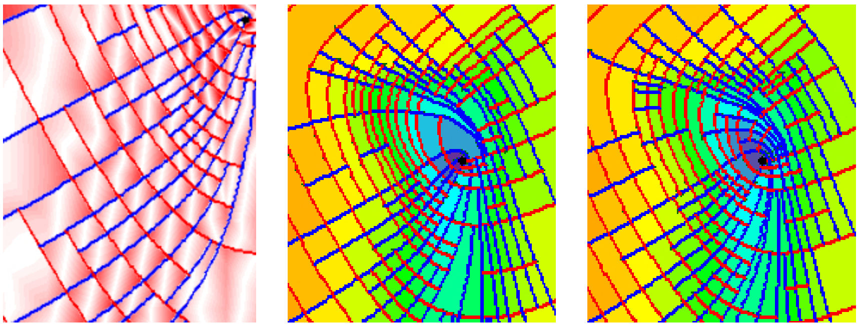
edge length φ_l is twice as strict as for Figure 7(c). All images are superimpositions of error images and the segmented cell boundaries. Figure 7(a) shows the pre-segmentation, and 7(b,c) the results for the different levels of detail.

As differences in images can be hard to perceive we display results of further evalutation in tables, associated figures are denoted. The first block in Table 2 is used as reference for the subsequent evaluation, it lists the values for the overview segmentation of Section 5.5.

Test 1: Comparison of edge-weights, variance w_v vs. absolute difference of extremal values of scalar field w_d:

As threshold for w_v 0.1% of the scalar value range is used, given by $\varphi_v = 0.001$. The middle block in table 2 and Figure 6(d) demonstrate that w_v leads to fewer cells at a higher mean error, compared to w_d used in the basic workflow. Variance is a criterion that regards mean values and ignores smaller variations along an edge. Thus fewer cells have to be subdivided, see Figure 6(d). The smoothing effect of the variance-based edge-weight can be extensively used to achieve desired results. w_d is an edge-weight that is rather strict and regards any change in the scalar field along the edge, which results in more cells, but higher accuracy.

Test 2: Comparison of strategies in choice of start point of tensor lines for subdivision operation: midpoint of an edge vs. point between extrema of the scalar field along an edge. Using the midpoint of an edge is a simple, straightforward technique, which leads to a slightly



(a) (b) (c)

Figure 8 Close up of adaptive segmentation of a slice in the notched block dataset: (a) Image representation of error shows how the edge-weights only reflect the behavior of the scalar field on the edges of the cell but not in its interior. Results of the choice of start point of tensor lines for the subdivision operation, at (b) the midpoint of edge and (c) the midpoint of extrema of scalar value along edge.

Table 2 Results for tests on control parameters and strategies 5.5.

Data	**2PL**	**1PL**	**NB**
Basic workflow:			
$\#Cells$	2830	1424	420
$\oslash Err$	4.67e-2	3.38e-2	4.02e-2
Ref. Figure	Fig. 7(b)	Fig. 6(b)	Fig. 8(a,c)
Accuracy edge weight *variance*:			
$\#Cells$	2497	1129	285
$\oslash Err$	5.039e-2	3.64e-2	4.93e-2
Ref. Figure		Fig. 6(d)	
Start point for subdividing tensor line in the middle:			
$\#Cells$	2870	1414	402
$\oslash Err$	4.75e-2	3.38e-2	4.02e-2
Ref. Figure			Fig. 8(b)

higher mean error, as shown in Table 2 last block. Starting the subdividing tensor lines between the minimum and maximum scalar values guarantees decrease in the edge-weights. This leads to qualitatively higher subdivisions as shown in Figure 8(b,c).

7 Conclusions

We have shown that the presented segmentation approach is able to generate segmentations aligned to the tensor field with low error measures. The weights and strategies can be chosen dependent on the demands of the specific application and allow a wide variety of data representation. The results shown in this paper are based on the anisotropy of the tensor field, and we can extend our work to other scalar fields used individually or as a combination of several fields. The main challenge for a segmentation based on tensor lines is the fact that feature lines resulting from the scalar field are in general not aligned with the eigenvector

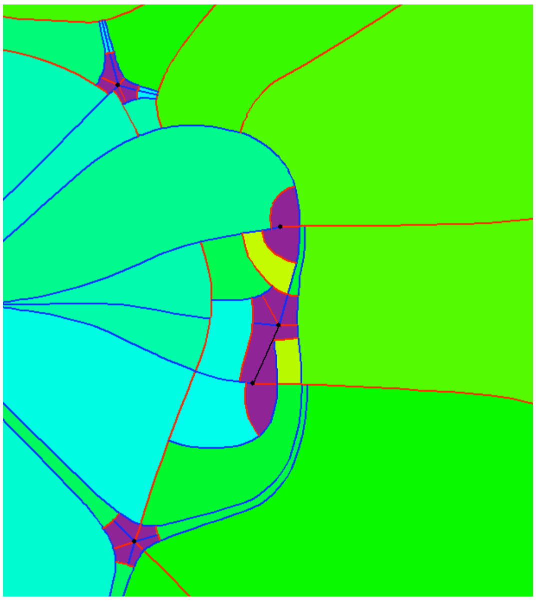


Figure 9 Close up: degenerate regions marked in violet.

field and thus they can only be approximated by a step function (see Figure 8(a)).

The strength of the application lies in its flexibility - restricting the modification process to geometric edge-weights delivers a uniform segmentation, whereas steering it by accuracy edge-weights generates a highly adaptive segmentation. There is a variety of options to customize the segmentation and use it for example for tensorline seeding or as preprocessing step for glyph placement or texture mapping.

During the design of the algorithm at many points decisions were made to balance its efficiency and accuracy. One example is the choice to define weights only on basis of the scalar field along cell edges, which is very efficient. But it cannot be guaranteed that the error contained in a cell is always well represented by the error along its one-dimensional edges. An integrated analysis of all cell edges, which leads to use of cell-weights as opposed to edge-weights is one of the interesting options for the future.

References

1 Cgal, Computational Geometry Algorithms Library. http://www.cgal.org.

2 Pierre Alliez, David Cohen-Steiner, Oliver Devillers, Bruno Levy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *SIGGRAPH 03*, 22(3):485–493, 2003.

3 A. Bhalerao and C.-F. Westin. Tensor splats: Visualising tensor fields by texture mapped volume rendering. In *6th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 294–901, 2003.

4 M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry, Algorithms and Applications*. Springer, 2nd edition, 1998.

5 Thierry Delmarcelle. *The Visualization of Second-order Tensor Fields*. PhD thesis, Stanford University, 1994.

6 Thierry Delmarcelle and Lambertus Hesselink. Visualization of second order tensor fields and matrix data. *IEEE Computer Graphics & Applications*, pages 25–33, 1993.

7 Louis Feng, Ingrid Hotz, Bernd Hamann, and Kenneth Joy. Anisotropic noise samples. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):342–354, 2008.

8 Robert B. Haber. Visualization techiques for engineering mechanics. *Comp. Systems in Engineering*, 1(1):37–50, 1990.

9 Ingrid Hotz, Louis Feng, Hans Hagen, Bernd Hamann, Boris Jeremic, and Kenneth I. Joy. Physically based methods for tensor field visualization. In *VIS '04: Proceedings of IEEE Visualization*, pages 123–130, 2004.

10 Ingrid Hotz, Jaya Sreevalsan-Nair, and Bernd Hamann. Tensor field reconstruction based on eigenvector and eigenvalue interpolation. In Hans Hagen, editor, *Scientific Visualization: Challenges for the Future*, 2008.

11 B. Jeremic, Gerik Scheuermann, Jan Frey, Zhaohui Yang, Bernd Hamann, Kenneth I. Joy, and Hans Hagen. Tensor visualization in computational geomechanics. *Int. Journal for Numerical and Analytical Methods in Geomechanics*, 26:925–944, 2002.

12 Gordon Kindlmann. Superquadric tensor glyphs. In *Proceeding of The Joint Eurographics - IEEE TCVG Symposium on Visualization*, pages 147–154, 2004.

13 Gordon Kindlmann and Carl-Fredrik Westin. Diffusion tensor visualization with glyph packing. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1329–1336, 2006.

14 Yingmei Lavin, Rajesh Batra, Lambertus Hesselink, and Yuval Levy. The topology of symmetric tensor fields. *AIAA 13th Computational Fluid Dynamics Conference,*, page 2084, 1997.

15 Guillaume Lavoue, Florent Dupont, and Atilla Baskurt. A new cad mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design*, 37(10):975–987, 2005.

16 Gregory M. Nielson and Il-Hong Jung. Tools for computing tangent curves for linearly varying vector fields over tetrahedral domains. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):360–372, 1999.

17 Jaya Sreevalsan-Nair, Cornelia Auer, Bernd Hamann, and Ingrid Hotz. Eigenvector-based interpolation and segmentation of 2d tensor fields. In *submitted to TopoInVis 09*.

18 Xavier Tricoche. *Vector and Tensor Field Topology Simplification, Tracking and Visualization.* PhD thesis, University of Kaiserslautern, 2002.

19 Xavier Tricoche, Gerik Scheuermann, Hans Hagen, and Stefan Clauss. Vector and tensor field topology simplification on irregular grids. In *VisSym '01: Proceedings of the symposium on Data Visualization 2001*, pages 107–116, 2001.

20 Xavier Tricoche, X. Zheng, and Alex Pang. Visualizing the topology of second order, time-varying two-dimensional tensor fields. In *Visualization and Image Processing of Tensor Fields*, pages 225–240, 2005.

21 Zhizhou Wang. *Diffusion Tensor Field Restoration and Segmentation.* PhD thesis, University of Florida, 2004.

22 Joachim Weickart and Hans Hagen, editors. *Visualization and Processing of Tensor Fields.* Mathematics and Visualization. Springer, 2006.

23 Y. Weldeselassie and G. Hamarneh. Dt-mri segmentation using graph cuts. In *Medical Imaging 2007: Image Processing. SPIE*, 2007.

24 Xiaoqiang Zheng and Alex Pang. Hyperlic. In *Proceedings of IEEE Visualization 2003*, pages 249–256, 2003.

25 Xiaoqiang Zheng and Alex Pang. Topological lines in 3d tensor fields. In *Proceedings of IEEE Visualization 2004*, 2004.

26 Leonid Zhukov, K. Museth, D. Breen, Ross Whitaker, and Alan Barr. Level set modeling and segmentation of dt-mri brain data. *Journal Electronic Imaging*, 12(1):125–133, 2003.

27 Ulas Ziyan, David Tuch, and Carl-Fredrik Westin. Segmentation of thalamic nuclei from dti using spectral clustering. In *Ninth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, pages 807–814, 2006.

𝒜

Jean-Paul Balabanian[1] and Eduard Gröller[2,1]

1 **University of Bergen**
Bergen, Norway
jean-paul.balabanian@ii.uib.no
2 **Vienna University of Technology**
Vienna, Austria
groeller@cg.tuwien.ac.at

Abstract

This paper describes the concept of 𝒜-space. 𝒜-space is the space where visualization algorithms reside. Every visualization algorithm is a unique point in 𝒜-space. Integrated visualizations can be interpreted as an interpolation between known algorithms. The void between algorithms can be considered as a visualization opportunity where a new point in 𝒜-space can be reconstructed and new integrated visualizations can be created.

1998 ACM Subject Classification I.3.7 Three-Dimensional Graphics and Realism

Keywords and phrases illustrative visualization, integrated visualization

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.36

1 Introduction

Illustrative visualization has been quite successful in recent years. The idea of illustrative visualization is to mimic the traditional illustrators' styles and procedures. Many techniques have been developed that span a wide range of traditional styles. These techniques include lighting models that resemble illustrative styles, exploded views, labeling, ghosting, and halos and have been successful at simulating the original illustrators' results. One strategy of illustrators is in principle to blend together very different styles. For example in one part of an illustration a photo realistic representation of the object is shown while in another part of the drawing the object is shown using ghosting effects, halos or outlines. Figure 1 demonstrates this heterogenous blending of different styles with several examples of a car.

This approach is similar to what illustrative visualization is doing and the idea of *blending different styles* is a concept that can be transferred, in a metaphorical way, to *blending of different algorithms*. Merging the results from one visualization with the results from another visualization, in a non-trivial way, can be considered as blending between the two algorithms. A simple example of this concept can be derived from slicing and volume rendering. These are two different visualization techniques for volume data. Blending between the two techniques could result in a visualization where the slice is integrated into the volume rendering. Figure 2 shows an example of what an integrated visualization combining direct volume rendering and slicing may look like.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 36–47
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

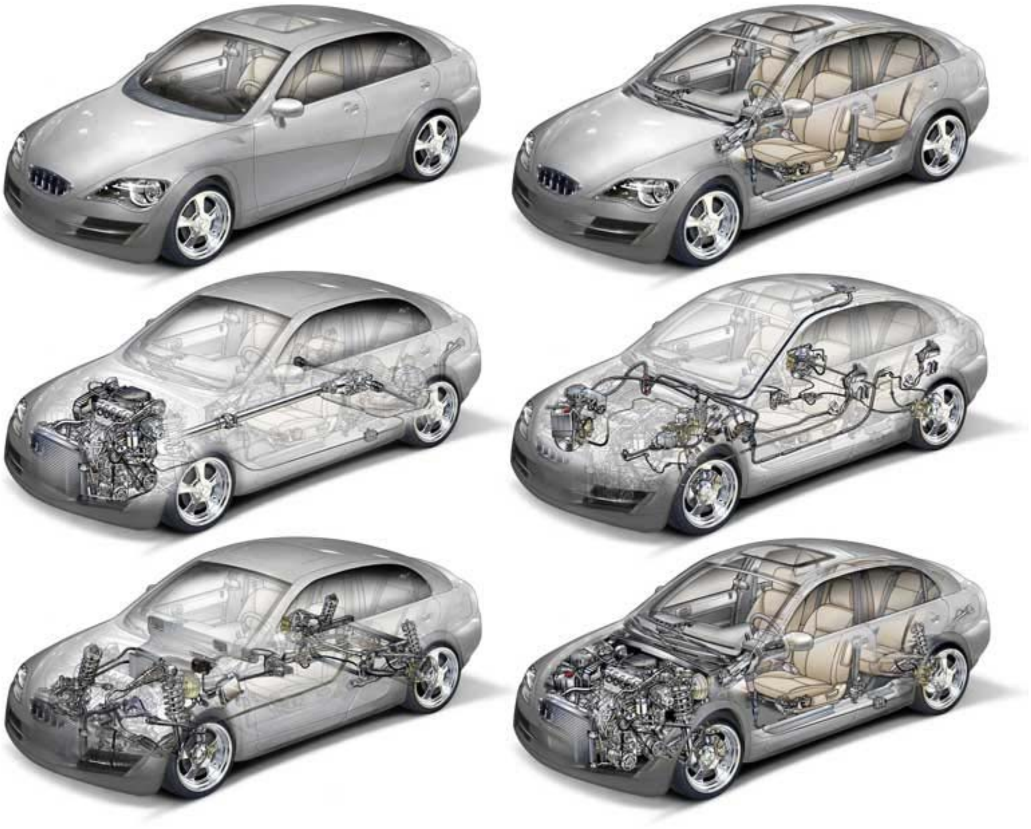


Figure 1 An illustration of a car with different parts ghosted. Image courtesy of Alan Daniels. Copyright beaudaniels.com

Integrated visualizations solve a limitation typical for linked views. In a linked-views setup the number of views increases with the complexity of the data. As the complexity increases the number of interesting aspects of the data also increases and more views are necessary to convey all of the important parts. Integrated visualizations alleviate this problem by providing a single frame of reference for all visualizations. They incorporate all of the important aspects of the data into the same view. Creating an integrated visualization is not straightforward, though, and so far only rather ad hoc approaches are known. A more systematic approach to create integrated views might be $\mathcal{A}$*-space*. $\mathcal{A}$-space is a space where all visualization algorithms reside. In $\mathcal{A}$-space every algorithm is represented by a unique point. $\mathcal{A}$-space is sparsely covered by the known visualization algorithms and there are many voids. Filling the voids between the points leads to reconstruction in $\mathcal{A}$-space and new integrated visualizations.

In the next section we will give examples of different visualizations that are reconstructions in $\mathcal{A}$-space and we will describe what type of integration each example is. In Section 3 we will describe $\mathcal{A}$-space in more detail and we will conclude in Section 4.

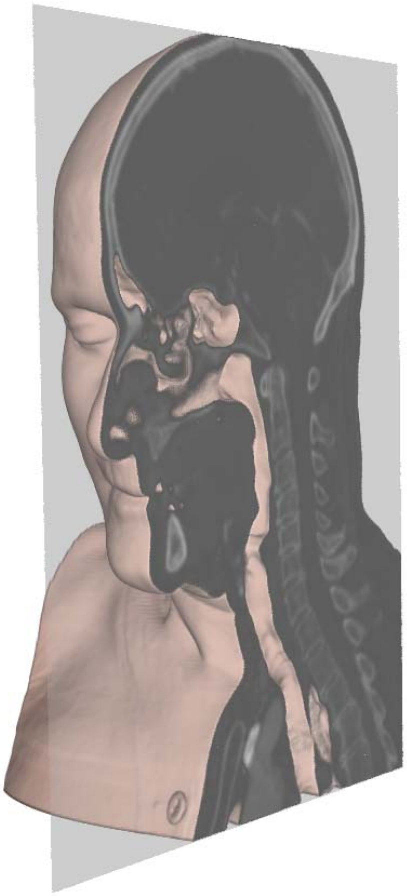
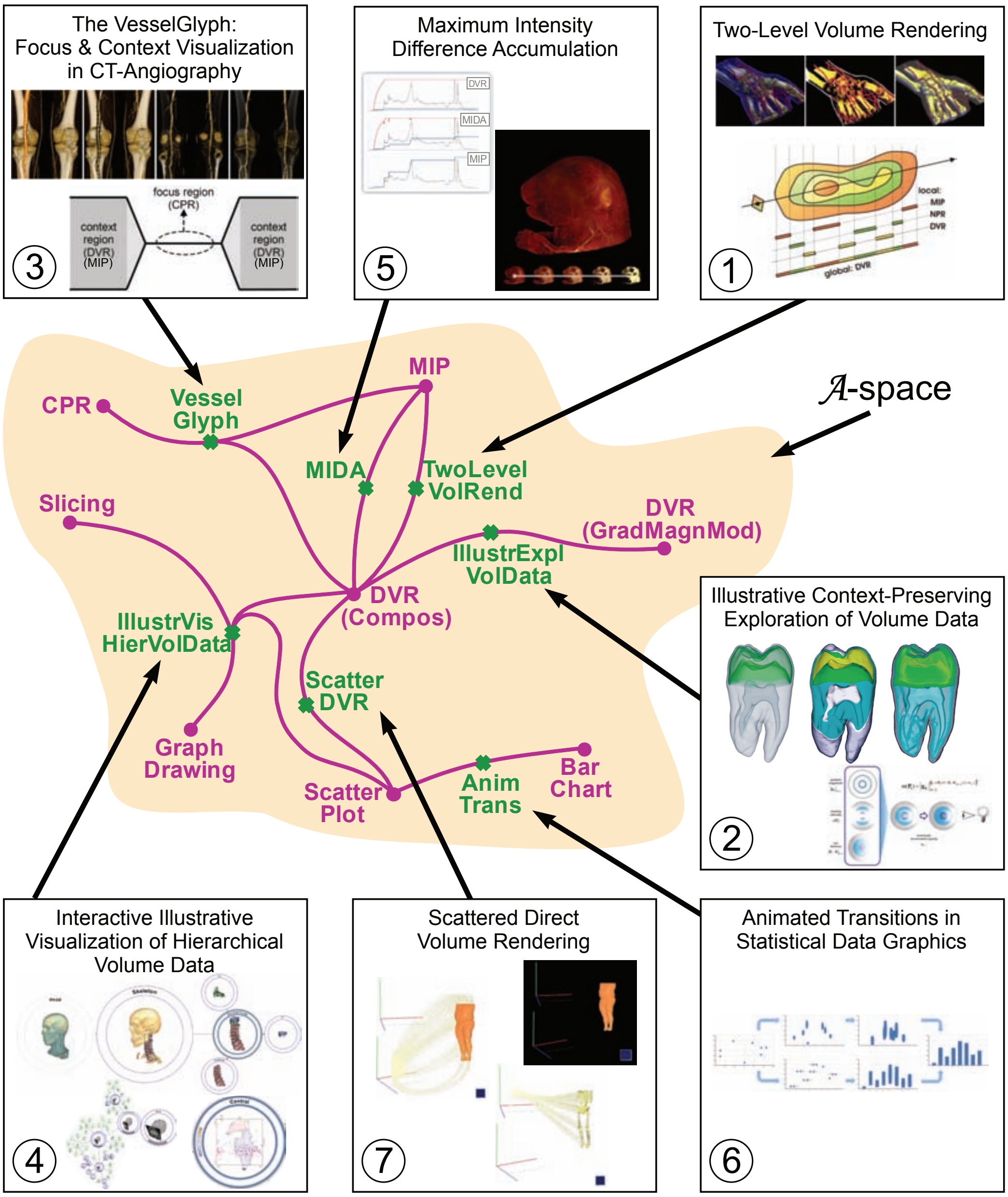


Figure 2 A volume-rendering with an integrated sagittal slice.

2 Examples of Reconstruction in 𝒜-Space

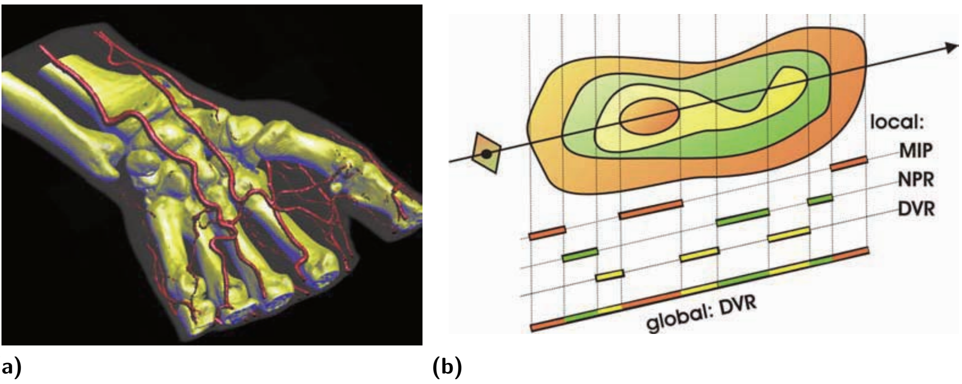
In this section we will showcase several examples of visualizations that are a *blending of algorithms*. The integrated visualizations presented can be considered as reconstructions in 𝒜-space with known starting points. We will show where in 𝒜-space these visualizations reside and the component algorithms required to perform the reconstructions. Figure 3 is a schematic of 𝒜-space with several algorithms indicated. The pink points represent well known algorithms that in principle are not integrated visualizations. Between these points paths have been drawn with green crosses indicating the reconstructed algorithms. An interesting observation is that between MIP and DVR there are two different paths. A path represents one way of blending algorithms. In 𝒜-space several paths may exist between algorithms and may result in fundamentally different visualizations. In the following sections we will describe all of the example visualizations that are present in Figure 3. We will discuss them in the order indicated by the number in the lower left of each frame. The chosen examples are just a subjective selection to illustrate a few nice places in 𝒜-space. A comprehensive overview on previous integrated views is beyond the scope of this paper.


The VesselGlyph: Focus & Context Visualization in CT-Angiography
focus region (CPR)
context region (DVR) (MIP)
context region (DVR) (MIP)
3
Maximum Intensity Difference Accumulation
DVR
MIDA
MIP
5
Two-Level Volume Rendering
local: MIP NPR DVR
global: DVR
1
𝒜-space
CPR
Vessel Glyph
MIP
MIDA
TwoLevel VolRend
Slicing
DVR (GradMagnMod)
IllustrExpl VolData
DVR (Compos)
IllustrVis HierVolData
Scatter DVR
Graph Drawing
Scatter Plot
Anim Trans
Bar Chart
Illustrative Context-Preserving Exploration of Volume Data
2
Interactive Illustrative Visualization of Hierarchical Volume Data
4
Scattered Direct Volume Rendering
7
Animated Transitions in Statistical Data Graphics
6

Legend

CPR: Curved Planar Reformation
DVR: Direct Volume Rendering
MIP: Maximum Intensity Projection

Figure 3 𝒜-space with example population.



■ **Figure 4** (a) A result from the Two-Level Volume Rendering [5, 4] visualization technique. One specific technique is used for the bone, another one for the skin and a third one for the vessels. (b) A schematic of the algorithm selection during rendering (NPR: Non-Photorealistic Rendering).

2.1 Two-Level Volume Rendering

The Two-Level Volume Rendering approach proposed by Hauser and Hadwiger [5, 4] is a merger of several visualization techniques. The idea is to use different rendering techniques depending on the underlying data. The techniques available for the rendering are Maximum Intensity Projection (MIP), Direct Volume Rendering (DVR) and others. During volume rendering the appropriate visualization techniques for the underlying segmented regions are chosen. The integration is a spatially coarse one since there is no smooth transition between the techniques and the resulting pixel is a composite of the visual representations produced by the different techniques. Figure 4a shows an example of this visualization approach where one technique is used for the bone, another technique for the vessels, and a third technique for the skin. Figure 4b indicates that for different spatial regions different algorithms are employed.

2.2 Illustrative Context-Preserving Exploration of Volume Data

The Illustrative Context-Preserving Exploration of Volume Data technique proposed by Bruckner et al. [2] is a visualization technique that enhances interior structures during volume rendering while still preserving the context. During volume rendering one or several structures may occlude the one of interest. Many techniques exist that can help in reducing the occlusion. Reducing the opacity of the occluding structures or applying clipping are two such techniques. The problem with these techniques is that they might remove the context of the interesting feature. The proposed approach combines DVR based on compositing with Gradient Magnitude Modulated DVR to reduce the opacity of less interesting areas in a selective manner. Two parameters are used to decide how to continuously *interpolate* between the two algorithms based on the input data. The results are illustrative volume renderings where contextual structures are outlined and the focused structures are kept in a prominent way. Figure 5a shows an example image produced by this technique. The center of the hand is semi transparent showing, among other details, the blood vessels quite prominently. The edges of the hand are not ghosted and thus retain the context. As indicated in Figure 5b the technique is a smooth and seamless integration of gradient-based opacity modulation, selective occlusion removal, fuzzy clipping planes and multiple transparent layers' handling.

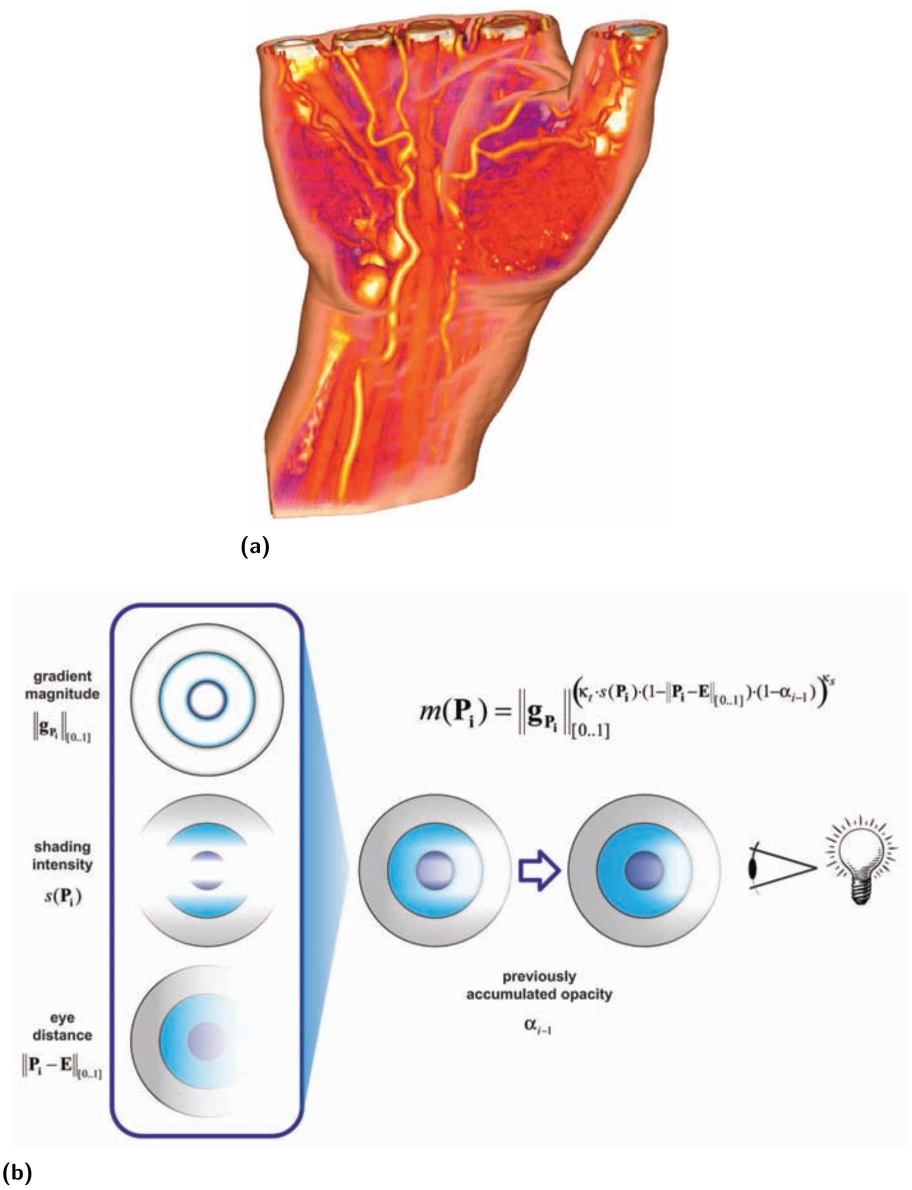

(a)
(b)
gradient magnitude
$\|\mathbf{g}_{\mathbf{P}_i}\|_{[0..1]}$
shading intensity
$s(\mathbf{P}_i)$
eye distance
$\|\mathbf{P}_i-\mathbf{E}\|_{[0..1]}$
$m(\mathbf{P}_i)=\|\mathbf{g}_{\mathbf{P}_i}\|_{[0..1]}^{\left(\kappa_t\cdot s(\mathbf{P}_i)\cdot(1-\|\mathbf{P}_i-\mathbf{E}\|_{[0..1]})\cdot(1-\alpha_{i-1})\right)^{\kappa_s}}$
previously accumulated opacity
α_{i-1}

Figure 5 (a) The context-preserving volume rendering [2] of a CT-scanned hand produces similarities to the ghosting effect used by illustrators. (b) Illustrates how different rendering techniques are seamlessly integrated while preserving the context [2].

2.3 The VesselGlyph: Focus & Context Visualization in CT-Angiography

The two previous examples have shown integration between techniques that all operate in 3D. The following example is an integration between a 2D technique and a 3D technique. The result is a visualization that exploits the complementary strengths and avoids the complementary weaknesses of both.

The VesselGlyph, proposed by Straka et al. [8], is a technique that combines Curved Planar Reformation (CPR) with DVR. CPR is a technique that takes a feature like a blood vessel and cuts it with a curved surface revealing the inside structures. The shape of the

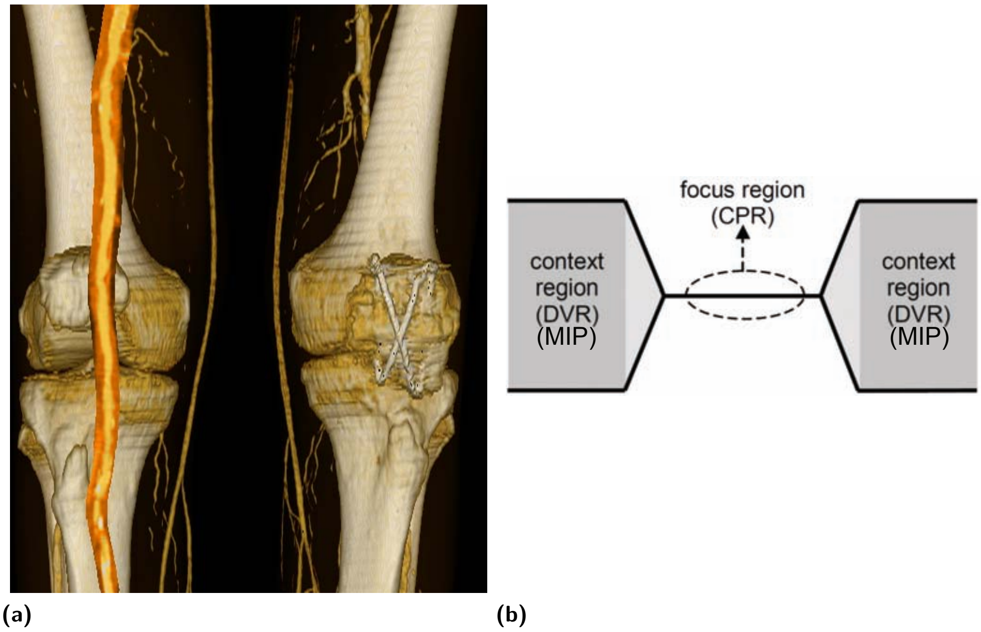


Figure 6 (a) The VesseGlyph [8] in action. The CPR, the vertical orange and red band on the left side, is projected onto the DVR of the same structure. (b) The concept of the VesselGlyph where CPR, the focus region, is integrated smoothly into the DVR, the context region.

surface is adapted so that it follows the curving and twisting of the structure. The resulting visualization is a 2D slice of the inside of the vessel. The VesselGlyph technique incorporates the CPR slice into the DVR visualization of the context structures. The resulting visualization shows interior details of blood vessels with CPR presented in the correct context rendered with DVR. The type of integration employed in this visualization is the merging of two spatially registered visualizations using for example image compositing techniques. Figure 6a shows an example of this type of visualization. With DVR alone the interior calcifications of blood vessels would not show up appropriately. With CPR alone the context region would be sliced arbitrarily which greatly reduces overview. Figure 6b depicts the concept of the VesselGlyph within an axial slice where the blood vessel would show up as a circular region in the slice center. CPR is considered for the focus region and is smoothly integrated into the DVR which is considered for the context region. It is also indicated that the context could alternatively be visualized using MIP.

2.4 Interactive Illustrative Visualization of Hierarchical Volume Data

The following example is more complex than the previous ones. The visualization performs integration between 3D and 2D techniques and also between scientific visualization and information-visualization techniques. The result is a visualization that in 𝒜-space blends more than two different algorithms.

Hierarchical Visualization of Volume Data by Balabanian et al. [1] is an integrated visualization that uses graph drawing to visualize the hierarchical nature of structures in a volumetric dataset. Graph drawing in 2D is used as a guiding space where other 2D or 3D visualizations are embedded. The nodes in the graph drawing are enlarged and serve as a canvas for the other visualizations. These visualizations include DVR, slicing, and scatter

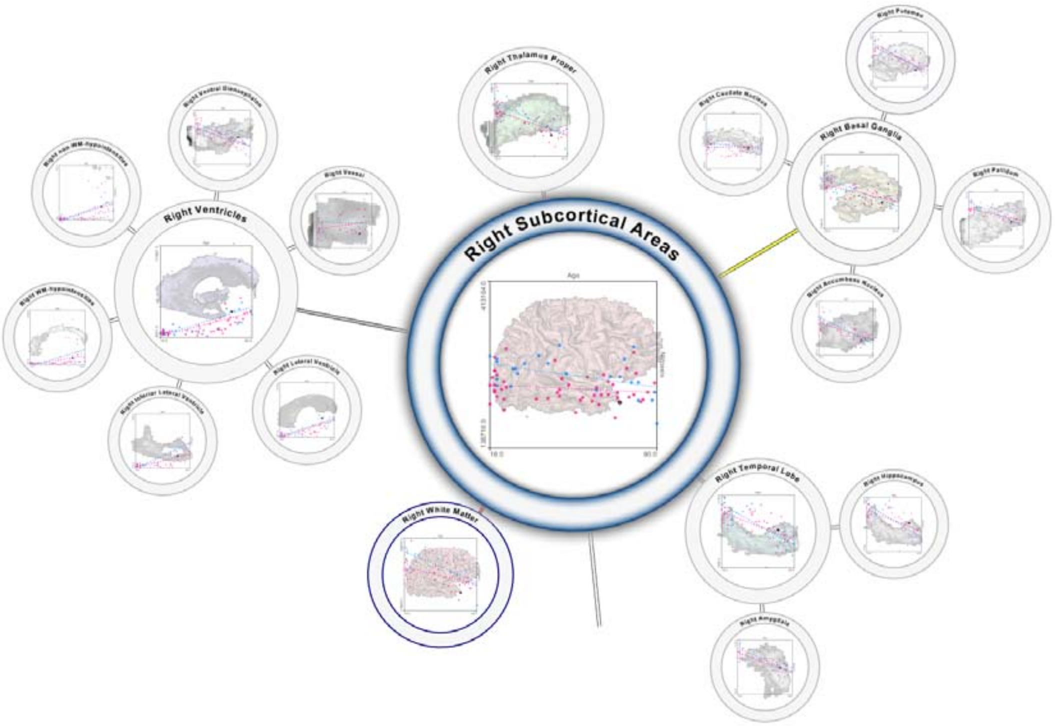


Figure 7 Hierarchical rendering of volume data [1]. A node-link diagram represents the hierarchical structuring with embedded volume renderings and scatter plots.

plots and are all integrated into one space. The type of integration employed in this visualization is at different levels. DVR and slicing are integrated in the same way as shown in Figure 2. The object that is actually rendered is defined by the hierarchical structure visualized by the graph drawing. The graph drawing is specified by the hierarchy information and every node is rendered as a circle. With statistical data available for the structures a scatter plot is added. Figure 7 shows an example of the subcortical areas of the brain (slicing not included here). The hierarchy of the substructures is visible with semi-transparent scatter plots on top of the embedded volume renderings.

In this example the integration is steered by the graph drawing. The abstract data is used to create a structure to present both the abstract and spatial data. It is also possible to envision an approach that uses the scientific-visualization space as the embedding space. In Figure 8 we have sketched the interpolation between the two spaces that are part of the visualization, i.e., the abstract and the spatial space. The red circle indicates where this work is located but using scientific visualization as the embedding space will result in a visualization located in the dashed square. Such an integrated view might be an exploded view in 3D space where the abstract hierarchical relationships are indicated through arrows.

2.5 Maximum Intensity Difference Accumulation

We now present another example of DVR-MIP integration in $\mathcal{A}$-space. This demonstrates that there are more than one possibilities to perform *interpolation* between points in $\mathcal{A}$-space. Maximum Intensity Difference Accumulation (MIDA) is a technique proposed by Bruckner and Gröller [3]. It is a volume rendering technique that integrates MIP and DVR. The integrated visualization preserves the complementary strengths of both techniques, i.e.,

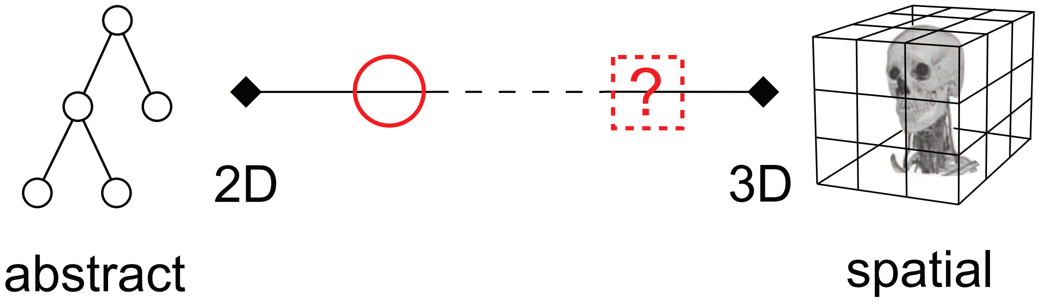


Figure 8 The red circle indicates the location where the interpolation between spaces takes place in the work by Balabanian et al. [1] while the dashed square indicates an alternative approach to this work.

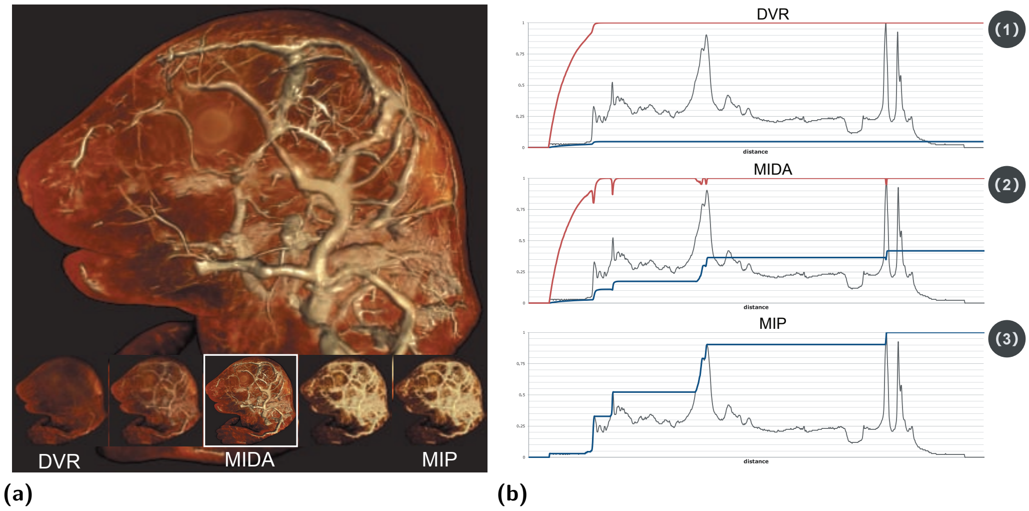


Figure 9 (a) Ultramicroscopy of a mouse embryo showing the MIDA [3] rendering enhanced with possible *interpolations* from DVR to MIP in the bottom. (b) Shows typical ray profiles for (1) DVR, (2) MIDA and (3) MIP [3].

efficient depth cuing from DVR and parameter less rendering from MIP. Since some datasets look better with DVR and others are best viewed with MIP, MIDA lets the user interpolate smoothly between DVR and MIP. Compared to the two-level volume rendering technique described in Section 2.1, MIDA is a spatially fine-grained integration approach and provides smooth transitions between the techniques. At each spatial position elements of both techniques are incorporated, whereas in two-level volume rendering algorithms are applied spatially disjoint. Figure 9a shows the result of using MIDA on an ultramicroscopy of a mouse embryo. Figure 9b shows the typical ray profiles generated with the different techniques.

2.6 Animated Transitions in Statistical Data Graphics

The Animated Transitions in Statistical Data Graphics proposed by Heer and Robertson [6] is a 2D to 2D integration performed in the information-visualization domain. The visualization techniques created provide smooth transitions between different visualizations of

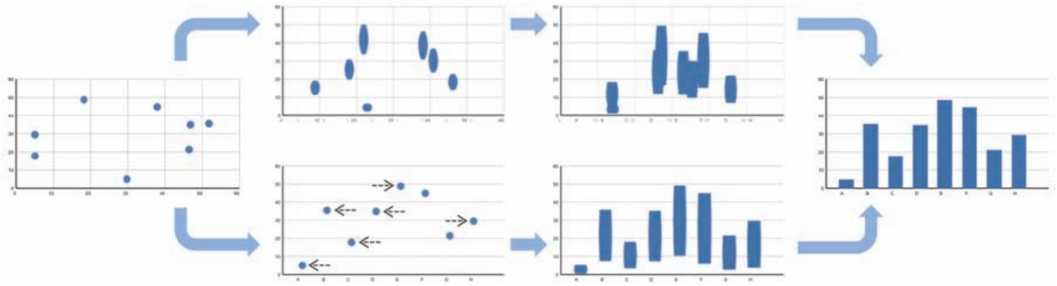


Figure 10 A schematic overview showing two possible transitions from a scatter plot to a bar chart [6].

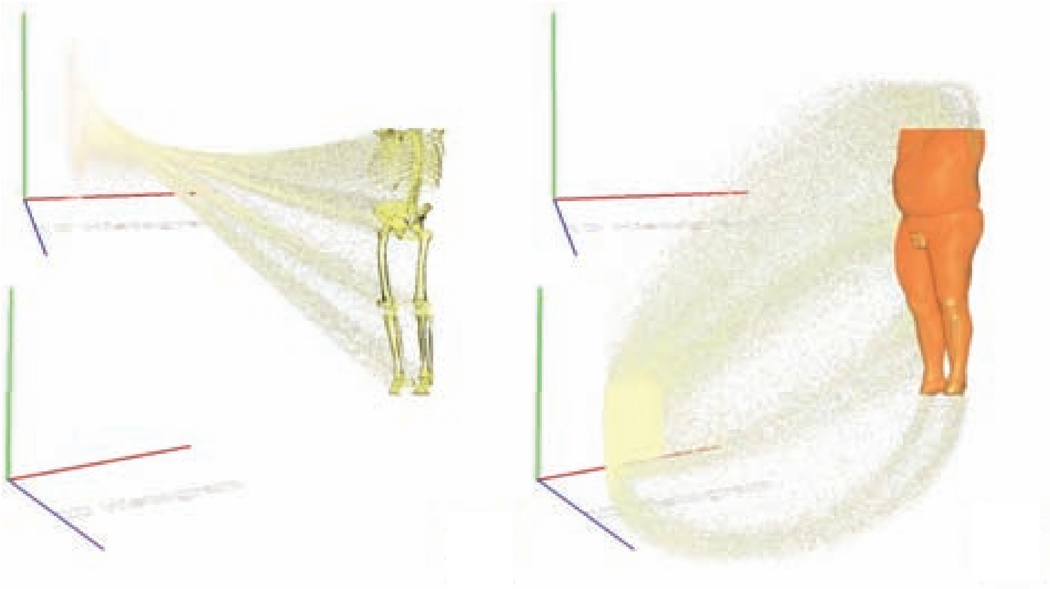


Figure 11 Two separate frames of the transition from DVR to scatter plots. [7]

statistical data. The example reconstructed in $\mathcal{A}$-space is an integration between scatter plots and bar charts. A benefit of this visualization is that the spatial relationship between sample points is visualized in the transition. The technique allows several different transitions between the statistical visualizations. Figure 10 shows in a schematic way two possible transitions from a scatter plot to a bar chart. This visualization technique is just one example of many approaches that exist for blending 2D to 2D algorithms.

2.7 Scattered Direct Volume Rendering

Our last somewhat speculative example from $\mathcal{A}$-space is the work by Rautek and Gröller [7] where a quite unusual integration is taking place. The integration is between 3D and 2D, specifically between DVR and scatter plots. On one side a volume rendering is shown and on the other side a scatter plot. In an animated transition the voxels are moving from the 3D volume-rendering space to their appropriate location in the 2D scatter plot and vice versa. Figure 11 shows two separate frames of the animated transition between DVR and scatter plots.

3 On the Nature of $\mathcal{A}$-Space

In the previous sections we have sketched the concept of $\mathcal{A}$-space. Via examples we have shown how visualization algorithms are points in $\mathcal{A}$-space and reconstruction is the process of blending between algorithms. However this concept entails more than this and in this section we will indicate further aspects of $\mathcal{A}$-space and open issues. $\mathcal{A}$-space is not a space in the strict mathematical sense. It shall act as a thought-provoking concept. Real-world phenomena are increasingly measured through several heterogeneous modalities with quite different characteristics. We believe that this increased data complexity can be tackled through integrated views. $\mathcal{A}$-space may help to more systematically explore the possibilities to blend together diverse visualization approaches with complimentary strengths. In the following we shortly discuss various open issues concerning $\mathcal{A}$-space.

Interpolation and reconstruction In the examples we have shown various types of blending algorithms together. Loosely we have called this blending interpolation and reconstruction. Can other types of interpolation be transferred to $\mathcal{A}$-space? Would it be possible to use barycentric coordinates to smoothly and simultaneously interpolate between three or more algorithms? Having several data sources for the same phenomenon makes fusion often a necessity. Fusion at the data and image level have been around for a long time. The visualization pipeline, however, consists of many more steps from the data to the final image. It is here where algorithm fusion and $\mathcal{A}$-space come into play. We cannot fight increased data complexity with increased visual complexity. Therefore a sensible step would be to move from linked to integrated or combined views.

Dimensionality and units What is the dimensionality of $\mathcal{A}$? What are the dimensions of $\mathcal{A}$? Would knowing the *coordinates* of an algorithm give some insight into the possibilities of reconstruction or the compatibility of algorithms. For example do algorithms in the same plane share some features? What kind of units does $\mathcal{A}$-space use? Is $\mathcal{A}$ a metric space? What is the distance between two algorithms and how does one measure this distance? Currently algorithms are often categorized according to their spatial and temporal asymptotical complexity. Could visualization algorithms be categorized according to other measures like visual complexity, number of algorithms integrated, algorithm length? Does $\mathcal{A}$-space have a set of basis algorithms where all other visualization techniques can be reconstructed from?

Transformations Which transformations make sense in $\mathcal{A}$-space? Let us assume we are starting with a linked view as the initial visualization where all the component algorithms are known points in $\mathcal{A}$-space. Is it possible to create a generic transformation that would convert such a visualization into an integrated visualization based on the linking between the views? Would that process reconstruct a new point in $\mathcal{A}$-space or create a mapping to an already known point?

Iso-algorithms Iso-surfaces are very important in the scientific-visualization domain. Analogously are there iso-algorithms in $\mathcal{A}$-space, with the visual complexity as iso-value for example?

Subspaces Into what subspaces can $\mathcal{A}$-space be subdivided? Would the subspaces correspond to natural categorizations such as 2D and 3D techniques or information visualization and scientific-visualization techniques?

Local neighborhood Given a point in $\mathcal{A}$-space how would the local neighborhood look like? Example measures might be gradient, divergence, curl. What would be the gradient, divergence or curl of DVR or MIP, i.e., ∇DVR, ∇MIP?

Interaction Given the typically dense overlapping in integrated views, interaction has hardly been explored in this context. An interaction event might simultaneously navigate in several spaces. How can the user be supported to efficiently interact with integrated views?

The above list of aspects and open issues of $\mathcal{A}$-space is for sure not complete. Creating integrated visualizations is currently done in an ad hoc way. Sparse data is reasonably simple to integrate, but the difficulty of integration increases with the density of the data. Dense data may have many important features collocated both spatially and temporally and currently there is no general way of solving this. Maybe some sort of exploded views in space and time could be an answer for this?

Categorizing the algorithms in $\mathcal{A}$-space may help in defining the boundaries of $\mathcal{A}$. The categorization may be to differentiate between fine and coarse visualization integration or to differentiate at what stage the integration is performed, i.e., data stage, algorithm stage, or image stage.

4 Conclusion

Integrated visualization will become more important in the future. Integrated views are a not yet fully explored area and they are one answer to cope with increased data complexity. We have shown some of the possibilities in $\mathcal{A}$-space and we think it may be a new direction on how to look at ways to perform visualization integration. $\mathcal{A}$-space might be a useful tool for classifying and indicating the possibilities of integrated visualizations. There already exist many integrated visualizations that may benefit to be localized in $\mathcal{A}$-space. Increasing the population of $\mathcal{A}$-space would also indicate untapped regions where reconstruction is a possibility and could lead to new integrated visualizations. Fill in the holes of $\mathcal{A}$!!!

References

1 Jean-Paul Balabanian, Ivan Viola, and Eduard Gröller. Interactive illustrative visualization of hierarchical volume data. In *Proceedings of Graphics Interface 2010*, 2010.

2 Stefan Bruckner, Sören Grimm, Armin Kanitsar, and Eduard Gröller. Illustrative context-preserving exploration of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1559–1569, 2006.

3 Stefan Bruckner and Eduard Gröller. Instant volume visualization using maximum intensity difference accumulation. *Computer Graphics Forum (Proceedings of EuroVis 2009)*, 28(3):775–782, 2009.

4 Markus Hadwiger, Christoph Berger, and Helwig Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *VIS'03: Proc. of the 14th IEEE Visualization 2003 (VIS'03)*, pages 301–308. IEEE Computer Society, 2003.

5 Helwig Hauser, Lukas Mroz, Gian Italo Bischi, and Eduard Gröller. Two-level volume rendering. *IEEE Trans. on Visualization and Computer Graphics*, 7(3):242–252, 2001.

6 J. Heer and G.G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, 2007.

7 Peter Rautek and Eduard Gröller. Scattered direct volume rendering. Personal communication, 2009.

8 Matúš Straka, Michal Červeňanský, Alexandra La Cruz, Arnold Köchl, Miloš Šrámek, Eduard Gröller, and Dominik Fleischmann. The VesselGlyph: Focus & context visualization in CT-angiography. In *VIS '04: Proceedings of the 15th IEEE Visualization 2004 (VIS'04)*, pages 385–392. IEEE Computer Society, 2004.

Interpolants Induced by Marching Cases*

Hamish Carr[1] and Eoin Murphy[2]

1 **School of Computing and Informatics, University College Dublin**
Belfield, Dublin 4, Ireland
and School of Computing, University of Leeds
Woodhouse Lane, Leeds, LS2 9JT, United Kingdom
`h.carr@leeds.ac.uk`

2 **School of Computing and Informatics, University College Dublin**
Belfield, Dublin 4, Ireland
`eoinomurchu@gmail.com`

Abstract

Visualization depends among other things on the interpolant used in generating images. One way to assess this is to construct case tables for Marching Cubes that represent the chosen interpolant accuracy. Instead, we show how to construct the interpolants induced by Marching Cases for comparison and assessment, how to extend this approach to Marching Squares, Cubes and Hypercubes, and how to construct an interpolant which is computationally equivalent to the digital rules conventionally used in image processing. Furthermore, we demonstrate that unlike tetrahedral meshes, geometric measurements over multi-linear mesh cells are inherently non-linear and cannot be summed as in the Contour Spectrum.

1998 ACM Subject Classification G.1.1, I.3.5

Keywords and phrases Interpolation, Marching Cubes, Isosurfaces

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.48

1 Introduction

In scientific and medical visualization, properties such as object volume and surface area are commonly computed for individual isosurfaces. Since the introduction of the Contour Spectrum [1], these properties have been computed for entire families of isosurfaces and used either as part of a visual interface [1], for automated analysis of significant features in the data [16], and as a measure of importance for topological simplification [6].

However, while geometric properties for simplicial meshes are generally piecewise-polynomial splines [1], computations on non-simplicial meshes have used discrete approximations instead [6]. Recent work [3, 15] has however shown that discrete approximations such as histograms or other image statistics are a rather poor approximation of the underlying geometric properties, a problem which is exacerbated by aliasing for small features.

Given this, it is natural to ask whether piecewise polynomial splines exist for geometric properties of non-simplicial meshes, and in particular of rectilinear meshes. This in turn requires examining both multilinear interpolants and the conventional Marching Cubes cases that are commonly used for extracting isosurfaces.

It has also been known for some time that the standard Marching Cubes cases do not correspond exactly to the trilinear interpolant [8, 10, 14, 13]. It is therefore also natural to ask whether there is an interpolant whose contours are identical to these cases.

* This work was partially supported by Science Foundation Ireland.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 48–58
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

This paper will demonstrate that piecewise-polynomial splines do not in general exist for multilinear interpolants or for Marching Cubes cases. In the process, we will also construct the interpolant corresponding to Marching Squares cases with a construction that extends to Marching Cubes cases, and demonstrate that these interpolants have undesirable qualities.

2 Previous Work

Isosurfaces are formally defined as the inverse image $f^{-1}(h)$ of a continuous function $f : \mathbb{R}^3 \to \mathbb{R}$ for a particular *isovalue* h, and are a core technique in scientific and medical visualization. In practice, isosurfaces are approximated using variations on Marching Cubes [9], which generates a triangulated separating surface between those samples known to be inside the isosurface and those samples known to be outside the isosurface.

More generally, marching cases can also be used to extract contours in 2D (isolines), 3D (isosurfaces) or higher (isohypersurfaces), using any type of mesh, whether simplicial (triangles in 2D, tetrahedra in 3D, pentatopes in 4D, &c.), rectilinear (squares or rectangles in 2D, cubes or hexahedra in 3D, hypercubes or hyperblocks in higher dimensions) or other cell types. Much of the literature on Marching Cubes has recently been surveyed by Newman and Yi [11].

While the intent of Marching Cubes was to generate surfaces that matched the trilinear interpolant on cubes, it was soon discovered that the mathematically correct bilinear and trilinear contours were hyperbolic in nature [14]. As a result, the standard Marching Cubes cases are only approximations, and subsequent work [13] has developed more complex tables that capture the topology of the isosurfaces in triangulated approximations.

Statistical properties such as histograms have long been used in user interfaces. Bajaj, Pascucci & Schikore introduced the Contour Spectrum [1], where geometric properties such as contour length (in 2D), area (in 3D), enclosed area (in 2D) and enclosed volume (in 3D) were computed for all isovalues and displayed in a separate panel to help a user identify significant isovalues. This work also identified that for simplicial meshes these properties are piecewise-polynomial splines that can be computed efficiently, as we will see in Section 3.

Subsequent work [6] showed that these properties can be computed for single contours during the construction of the contour tree, and used for topological simplification. The authors also observed in passing that these properties were not piecewise-polynomial for cubic meshes, instead using discrete approximations such as sample counts and isovalue summations.

Although this approach was necessary at the time, it has since been shown [3, 15] that discrete approximations (including histograms) effectively assume the nearest-neighbour interpolant, and as a result give poor approximations of these geometric properties, especially for small objects or at low sampling resolution.

We therefore return to the question of whether it is possible to find closed form solutions for non-simplicial meshes such as rectilinear meshes, either for multilinear interpolants or for marching cases. We will also show that an interpolant can be constructed that is equivalent to the marching cases, in the sense that its contours are identical to those extracted by the marching cases.

3 Spline Linearity

In the Contour Spectrum [1], the observation was made that for simplicial meshes, geometric properties took on the form of piecewise-polynomial splines. As this is the point of departure

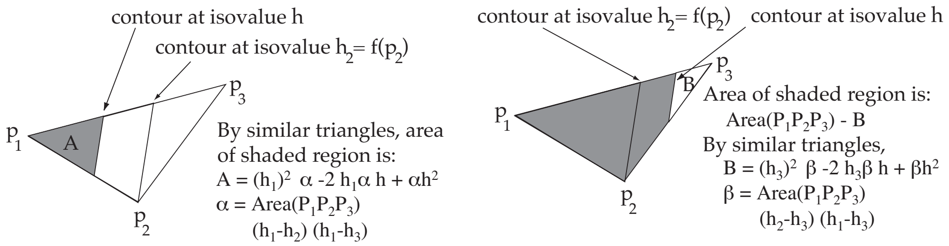

contour at isovalue h
contour at isovalue $h_2 = f(p_2)$
p_1
A
p_3
p_2
By similar triangles, area
of shaded region is:
$A = (h_1)^2 \ \alpha - 2 h_1 \alpha h + \alpha h^2$
$\alpha = \text{Area}(P_1P_2P_3)$
$(h_1 - h_2)(h_1 - h_3)$
contour at isovalue $h_2 = f(p_2)$
contour at isovalue h
p_3
B
p_1
p_2
Area of shaded region is:
$\text{Area}(P_1P_2P_3) - B$
By similar triangles,
$B = (h_3)^2 \ \beta - 2 h_3 \beta h + \beta h^2$
$\beta = \text{Area}(P_1P_2P_3)$
$(h_2 - h_3)(h_1 - h_3)$

Figure 1 Computing Area in a Linear Triangle.

for the balance of this paper, it is worthwhile reviewing the construction, which we will perform in 2D with linear contours in triangles for simplicity. As is conventional, we shall assume that the isovalues h_1, h_2, h_3 at the vertices p_1, p_2, p_3 of the triangle are distinct, i.e. that $h_1 < h_2 < h_3$.

As we can see in Figure 1, each contour of a linear interpolant on a simplicial mesh is simply a line. And, as each contour is defined by a single isovalue h, the length of the contour will be a function $L(h)$, whose input is an isovalue, and whose output is the length of the corresponding contour. Similarly, the area behind the contour will also be a function $A(h)$

Now, we know that the contours are the intersection of a horizontal plane with the graph of the function in $\mathbb{R}^3$, so it follows that all contours are parallel to each other, and in particular are parallel to the contour at isovalue h_2 that passes through p_2. Since this contour is fixed and has a known length and area associated, it follows that $L(h)$ and $A(h)$ can be computed using similar triangles, depending on which half of the triangle the contour is in. Because similar triangles have linear relationships, it then follows that properties such as length and area are polynomials in h, which allows easy tracking by summation of coefficients.

In a simplicial (tetrahedral) mesh in 3D, contours of the linear interpolant are planes, and the corresponding construction applies. In general, therefore, geometric properties in a simplicial mesh can be computed by laws of similar simplices, and generate piecewise-polynomial properties.

4 Bilinear Non-linearity

As we have just seen, geometric properties of isosurfaces on a simplicial mesh are polynomials, which are easy to track and combine. Unfortunately, the same is not in general true for higher-order interpolants. We will demonstrate this by examining geometric properties of the bilinear interpolant.

To do so, we start with the paradigm case of the area bounded by the contour at isovalue h. We know from Nielson & Hamann [14] that the contours of a bilinear cell are hyperbolae, and we show a single contour in a bilinear cell in Figure 2.

To compute the area of region A, we need to know the area B under the hyperbolic curve $y = \frac{k}{x - x_v} + y_h$, where x_v is the x coordinate of the vertical asymptote, y_h is the y coordinate of the horizontal asymptote, and k is a scaling constant. The area of B is then given by the integral:

$$\int_a^b \frac{k}{x - x_v} + y_h \, dx = k \ln |b - x_v| - k \ln |a - x_v| + (b - a) y_h \ . \tag{1}$$

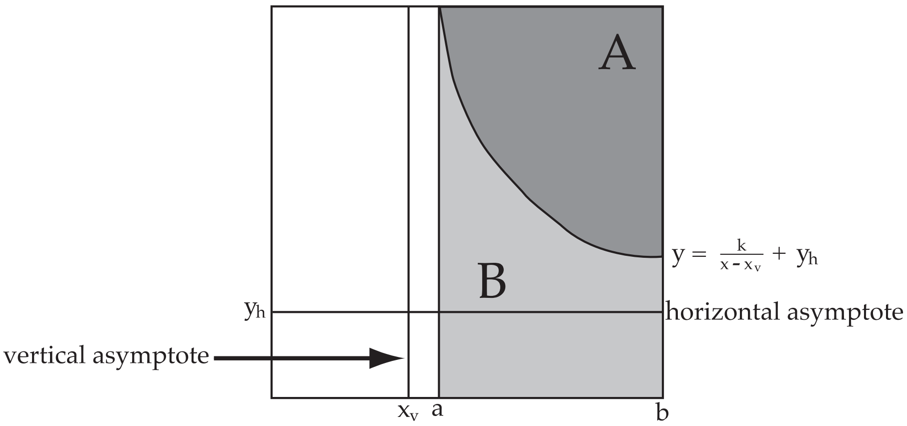


Figure 2 Computing Area in a Bilinear Cell. Since bilinear interpolants generate hyperbolic contours, computing local spatial measures involves logarithmic terms.

The appearance of a logarithmic term is problematic, because both the Contour Spectrum [1] and local geometric measures [6] rely on tracking the sum of polynomial terms with a single polynomial of $O(1)$ size.

It is possible on principle to sum logarithmic terms using the identity:

$$\log(a) + \log(b) = \log(ab) \ , \tag{2}$$

i.e. we can exponentiate each term and multiply them together, then take a logarithm any time we need it. For an extended sequence of log terms, however, we will have to worry about the numerical precision of performing a large number of multiplications.

The arclength of the contour is even worse, being defined by:

$$\begin{aligned} L(x) &= \int_a^b \sqrt{1 + (f'(x))^2} \\ &= \int_a^b \sqrt{1 + \left(\frac{-1}{x^2}\right)^2} \\ &= \int_a^b \sqrt{1 + \frac{1}{x^4}} \end{aligned}$$

for which no simple integration formula exists.

A further complication is that computing these formulae first requires finding the bounds of integration a, b. While this is feasible for the bilinear interpolant, for even the trilinear interpolant, it requires accurate analysis of the MC cases required to represent the trilinear interpolant correctly [12, 4].

Given the difficulties associated with non-polynomial function representations, analytically intractable integrals and hard-to-determine bounds of integration, it is natural to ask whether the task is any simpler if we simply use the known Marching Squares (or Cubes) cases.

5 Interpolant Induced by Marching Squares

Both for theoretical reasons and for practical reasons, therefore, it is desireable to identify what function, if any, corresponds to the contours generated by Marching Squares (Cubes) cases. Moreover, since the end points of the contour fragments are defined by linear interpolation,

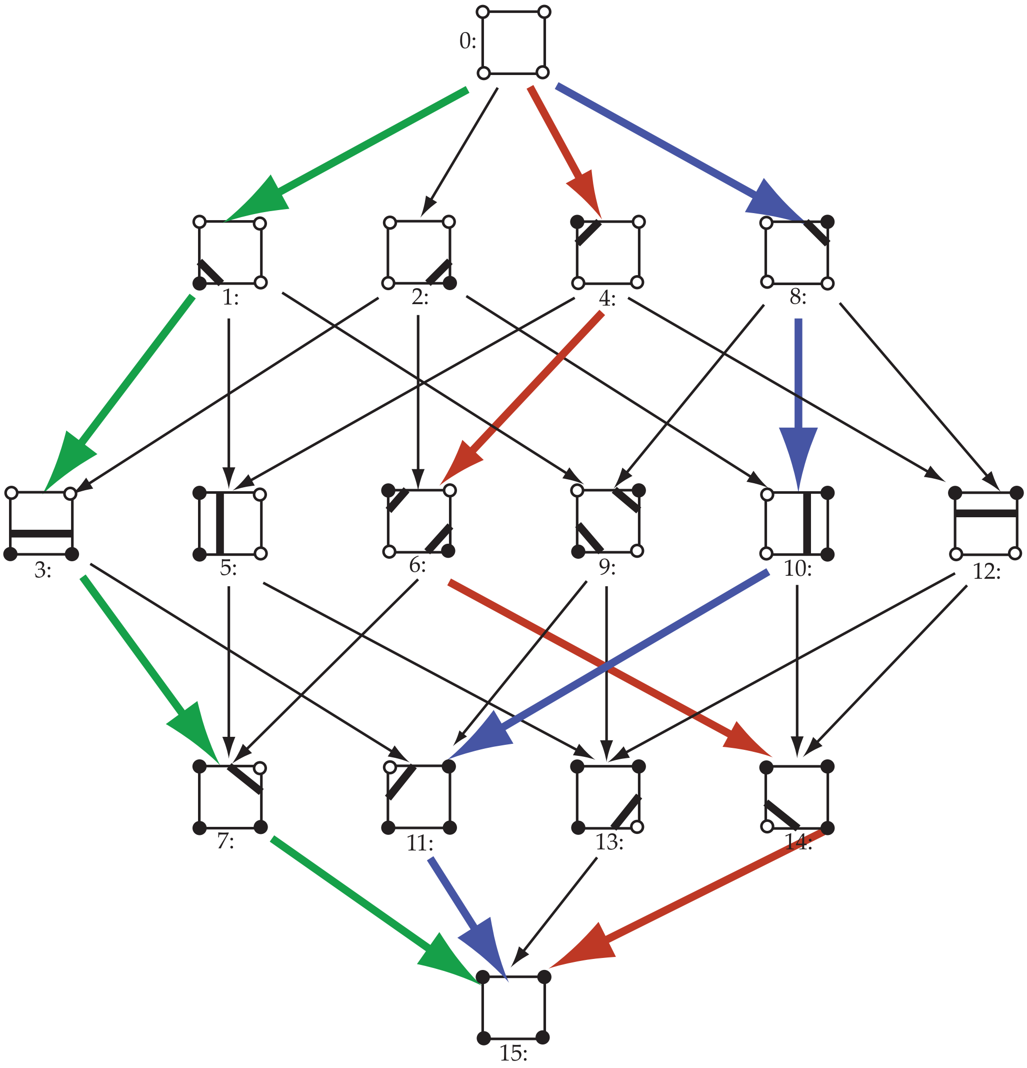


Figure 3 Subject to symmetry, only three distinct types of cell are possible, as shown by the coloured arrows.

we would like to know whether their geometric properties are polynomial splines as in the case of simplicial meshes.

We will see shortly that there is not in fact a single interpolant induced by Marching Squares. Instead, the interpolant depends on the set of Marching Squares cases swept through by the contour over all isovalues. Figure 3 illustrates the possibilities using a finite state machine to represent all possible sequences of cases [5].

In any given cell, if we sweep a contour from high isovalues to low, it will pass through a sequence of cases dictated only by the order in which the vertices are swept past (i.e. when they turn from white to black as the sweep passes their isovalue). Moreover, for the range of isovalues corresponding to the case, the points that define the contour fragments are linearly interpolated along the edges. Since the contours are linear combinations of these edge points, it follows that the contour sweeps out a continuous region of the cell, which can be represented as a surface, as in Figure 4. So, if each case sweeps out a surface over a range of isovalues, can the surfaces corresponding to different cases intersect? We will see in

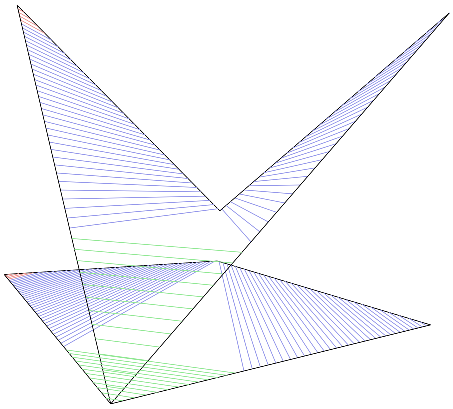


Figure 4 Illustration of Surface for Ambiguous Case with Projection to Ground Plane.

Section 7 that we can guarantee that they do not, but for now, we simply assume that they do not.

Consider an isovalued sweep through a particular cell. At high isovalues, all four vertices are white and the contour fails to intersect the cell (case 0). Some time later, one vertex becomes black, and the contour is in case $1, 2, 4$, or 8, which are rotationally symmetric to each other. If we follow the path marked in red in Figure 3, we sweep past vertex 2 to case 4, as illustrated. Thereafter, the next vertex to turn black can be diagonally opposite to the first (i.e. case 6), or adjacent to it (e.g. case 5 or 12). Again, following the red path, we sweep past vertex 1 to get case 14. Finally, the last vertex is swept to reach case 15.

Clearly, there are symmetries involved here, and several other paths equivalent to the red path. Up to these symmetries, there are three possible paths, marked in red, blue and green respectively in Figure 3. Having identify the set of possible paths, we now consider each type of path separately.

5.1 Ambiguous Face: The Red Path

Interestingly, the red path can only occur in a cell with an ambiguous face [14], as it requires diagonally opposing black and white vertices for case 6. While this might seem to be the hardest case to analyse, it is in fact the easiest, as can be seen in Figure 4.

This path uses cases $0, 4, 6, 14, 15$, of which 0 and 15 may be ignored as they have no contours. In case 4, the contour is defined by two edge points which move linearly, and thus sweep out a triangular surface until the isovalue of the opposite corner is reached. At this point, a separate contour starts at this new corner and similarly sweeps out a triangular surface, while the contour in the upper left corner continues to sweep out its triangular surface. Once the third vertex isovalue is reached however, these two contours disappear and a new one is swept towards the lower left corner. This is the exact opposite of case 2, and sweeps a final triangular surface.

Note however that this triangle only meets the previous ones at the edge points, and that a large region in the centre of the cell is never used by any contour. We can, however, set the isovalue of all points in this region to the vertex isovalue, forming a flat sheet that connects the three swept triangles. Thus, we have used the sweeping metaphor to construct a continuous function over the cell whose contours match those extracted using Marching

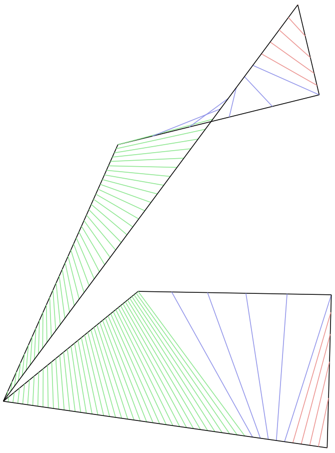


Figure 5 Illustration of Surface for Cyclic Case with Projection to Ground Plane.

Squares for this cell. Moreover, we note that this surface is composed of triangles, so that geometric properties will continue to be represented by tidy polynomials.

Unfortunately, however, this is not true for non-ambiguous faces, which we turn our attention to next. Moreover, the interpolant is data-dependent, and is not the same in every cell, which poses a problem for frequency domain analysis.

5.2 Cyclic Faces: The Blue Path

Our next possibility is the path shown in blue, where the vertices are swept in cyclic order around the face. Here, we start off with a single edge in case 8, and end up with a triangular surface in case 11. In between, however, we have one vertex moving along the top edge and one along the bottom edge. If we label these points $p(h)$ and $q(h)$, each linearly interpolated along the edge with respect to the isovalue h, we can derive the following formula for edge length:

$$\begin{aligned} L(h) &= \sqrt{\|p(h) - q(h)\|} \\ &= \sqrt{\left\| \left(v_{00} + \frac{h - h_{00}}{h_{01} - h_{00}} (v_{01} - v_{00}) \right) - \left(v_{10} + \frac{h - h_{10}}{h_{11} - h_{10}} (v_{11} - v_{10}) \right) \right\|} \end{aligned}$$

which is clearly non-polynomial.

5.3 Zigzag Faces: The Green Path

Finally, we deal with faces corresponding to the green path. Here, the vertices are swept in a zigzag fashion. As with the cyclic faces, we have triangles swept through when the edges defining the contour share a vertex, and an intermediate section with non-polynomial behaviour.

5.4 Asymptotic Decider

One additional possibility is to use the Asymptotic Decider [14] to adjudicate ambiguous faces. In this case, as shown in Figure 7, we end up with not three but four triangular sections induced by the actual contours, and a flat diamond in the middle connecting them.

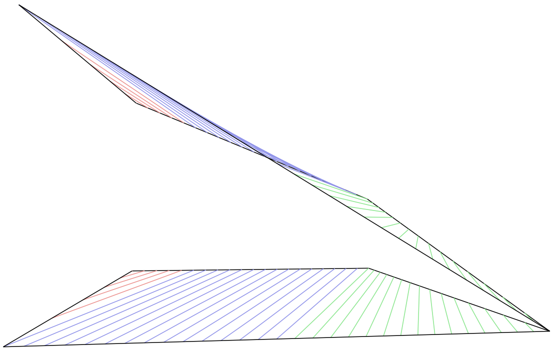


Figure 6 Illustration of Surface for Zigzag Case with Projection to Ground Plane.

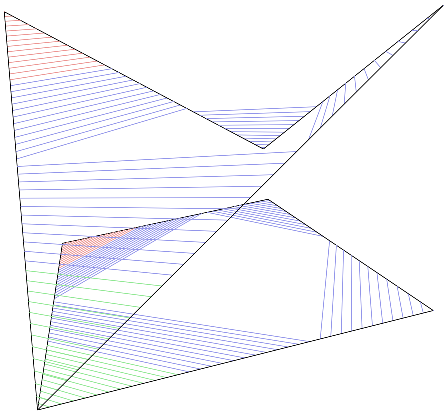


Figure 7 Illustration of Surface for Ambiguous Face with Asymptotic Decider.

5.5 Interpolant Properties

From the foregoing sections, we can see that the interpolant thus induced, while continuous, has several undesireable properties:

1. Piecewise: the interpolant is defined piecewise within each cell,
2. Non-Smooth: the interpolant is not smooth - i.e. it is only C^0 continuous
3. Non-Uniform: the details of the interpolant depend on the actual isovalues and their discrete relationships, rendering analysis based on signal processing difficult or impossible.

but, compared to the nearest neighbour (NN) interpolant, is at least piecewise continuous.

6 Interpolant Induced by Marching Cubes

Unsurprisingly, this approach can also be extended to Marching Cubes, following the same general principles:

1. There will be as many cases as there are distinct paths through the finite state machine. Note that while there will be reductions due to symmetry, it will not be true that the paths can simply be counted in a symmetry-reduced finite state machine. For example, consider our cyclic (blue) and zigzag (green) cases above. In both paths the sequence of symmetry reduced cases is: 0, 1, 2, $1C$, $0C$. But, as we can see from the corresponding surfaces, the details of the interpolants differ.
2. Each marching case in the path through the finite state machine will correspond to a patch of the interpolant, as will the isovalues at which the cases change.
3. For each marching case, the corresponding range of isovalues will generate a sequence of smoothly varying contours.
4. For each surface fragment, if the edges used to interpolate share a common vertex, the fragment will move linearly, and sweep out a barycentric interpolant over a simplex. Such fragments will have polynomial spline properties as in the Contour Spectrum [1].
5. For surface fragments whose vertices are not interpolated along edges with a common vertex, the corresponding patch will in general have non-linear behaviour, and will not generate polynomial spline properties.
6. The remaining regions of the cell will be defined by contours immediately above and below the isovalue of some vertex of the cell, and can be assigned that isovalue to generate a piecewise-continuous but not smooth interpolant.

Since the conclusion from Marching Squares is that non-linear behaviour is to be expected, neither the paths nor the details of the interpolants for Marching Cubes have been enumerated.

7 Extension to Marching Hypercubes

More generally, a similar approach can be applied to the Marching Hypercubes defined by Bhaniramka, Wenger & Crawfis [2]. For this, in any given case, the white vertices and the edge points are collected, and their convex hull is computed to serve as the exterior (lower level set) of the contour, the surface of the convex hull being used as the contour itself except at the boundaries.

In two dimensions, this rule generates the cases shown in Figure 3, and in three dimensions, it generates cases nearly identical to the standard set of crack-free Marching Cubes [10].

Because of the convex hull construction, the interpolated edge points expand continuously, and the convex hull surfaces can therefore be shown to be distinct from each other. Take two isovalues $h_1 > h_2$ in the same case and observe that an edge point e_2 for the isovalue h_2 must be in the interior of the edge from an edge point e_1 at isovalue h_1 to some vertex v with isovalue $h_2 > h$. Since all points on the contour at h_2 are convex combinations of edge points constructed in this way, it follows that they are strictly in the interior of the contour at h_1 and the result follows.

8 Digital Connectivity Rules

Carr & Snoeyink [5] showed that the connectivity of the standard crack-free Marching Cubes cases [10] and of Marching Hypercubes [2] are equivalent to connectivity rules from digital imaging, and it follows that the interpolants discussed in Section 5 and Section 6 are also equivalent to these connectivity rules.

9 Conclusions and Future Work

We have shown that the piecewise linear splines of the Contour Spectrum [1] do not easily generalize to non-simplicial meshes, that geometric properties of contours defined by Marching Squares or Marching Cubes cases are not piecewise linear, and and have shown how to construct continuous interpolants whose contours are exactly those of a given set of marching cases.

In practice, the implications of this are that for non-simplicial meshes, geometric properties for the Contour Spectrum [1] or for topological analysis [7] will continue to be computed either by discrete approximations such as voxel count or by explicit extraction of individual contours.

Although it may not be tractable, we would still like to find closed form equations for geometric properties of multilinear interpolants and of marching case contours. Subject to concerns about accuracy, we would also like to find approximate splines of geometric properties.

Acknowledgements

Thanks are due to Science Foundation Ireland for research grant funding and to University College Dublin for equipment grant funding. Further thanks are due to the many individuals who stimulated the ideas that led to this work.

References

1 Chandrajit L. Bajaj, Valerio Pascucci, and Daniel R. Schikore. The Contour Spectrum. In *Proceedings of Visualization 1997*, pages 167–173, 1997.

2 Praveen Bhaniramka, Rephael Wenger, and Roger A. Crawfis. Isosurface Construction in Any Dimension Using Convex Hulls. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):130–141, 2004.

3 Hamish Carr, Brian Duffy, and Barry Denby. On histograms and isosurface statistics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1259–1266, September/October 2006.

4 Hamish Carr and Nelson Max. Subdivision analysis of the trilinear interpolant. *IEEE Transactions on Visualization and Computer Graphics*, to appear 2009.

5 Hamish Carr and Jack Snoeyink. Representing interpolant topology for contour tree computation. In Hans-Christian Hege, Konrad Polthier, and Gerik Scheuermann, editors, *Proc. Topology-based Methods in Visualization 2007 Workshop*. Springer, 2007.

6 Hamish Carr, Jack Snoeyink, and Michiel van de Panne. Simplifying Flexible Isosurfaces with Local Geometric Measures. In *Proceedings of Visualization 2004*, pages 497–504, 2004.

7 Hamish Carr, Jack Snoeyink, and Michiel van de Panne. Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. *Computational Geometry: Theory and Applications*, 43:42–58, 2010.

8 M.J. Dürst. Letters: Additional Reference to "Marching Cubes". *Computer Graphics*, 22(4):65–74, 1988.

9 William E. Lorenson and Harvey E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21(4):163–169, 1987.

10 Claudio Montani, Riccardo Scateni, and Roberto Scopigno. A modified look-up table for implicit disambiguation of Marching Cubes. *Visual Computer*, 10:353–355, 1994.

11 Timothy S. Newman and Hong Yi. A survey of the marching cubes algorithm. *Computers And Graphics*, pages 854–879, 2006.

12 Gregory M. Nielson. On Marching Cubes. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):283–297, 2003.

13 Gregory M. Nielson. Dual Marching Cubes. In *Proceedings of Visualization 2004*, pages 489–496, 2004.

14 Gregory M. Nielson and Bernd Hamann. The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes. In *Proceedings of Visualization 1991*, pages 83–91. IEEE, 1991.

15 Carlos Eduardo Scheidegger, John M. Schreiner, Brian Duffy, Hamish Carr, and Cláudio T. Silva. Revisiting histograms and isosurface statistics. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1659–1666, 2008.

16 Shivaraj Tenginakai, Jinho Lee, and Raghu Machiraju. Salient Iso-Surface Detection with Model-Independent Statistical Signatures. In *Proceedings of Visualization 2001*, pages 231–238, 2001.

Comparative Visualization Using Cross-Mesh Field Evaluations and Derived Quantities

Hank Childs[1], Sean Ahern[2], Jeremy Meredith[2], Mark Miller[3], and Kenneth I. Joy[4]

1 **Lawrence Berkeley National Laboratory**
Berkeley, California, USA
hchilds@lbl.gov

2 **Oak Ridge National Laboratory**
Oak Ridge, Tennessee, USA
ahern@ornl.gov

3 **Lawrence Livermore National Laboratory**
Livermore, California, USA
miller86@llnl.gov

4 **Institute for Data Analysis and Visualization**
Computer Science Department
University of California
Davis, California, USA
joy@cs.ucdavis.edu

Abstract

We present a data-level comparative visualization system that utilizes two key pieces of technology: (1) cross-mesh field evaluation – algorithms to evaluate a field from one mesh onto another – and (2) a highly flexible system for creating new derived quantities. In contrast to previous comparative visualization efforts, which focused on "$A - B$" comparisons, our system is able to compare many related simulations in a single analysis. Types of possible novel comparisons include comparisons of ensembles of data generated through parameter studies, or comparisons of time-varying data. All portions of the system have been parallelized and our results are applicable to petascale data sets.

1998 ACM Subject Classification I.3.6 [Computer Graphics]: Methodology and Techniques

Keywords and phrases Comparative Visualization, Cross-Mesh Field Evaluation, Derived Quantity

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.59

1 Introduction

One of the most common activities of an analyst is to perform comparisons. These comparisons take different forms: comparing a simulation to experimental data, comparing a simulation to a legacy simulation result, comparing the results of a simulation before and after key algorithmic changes, comparing the results of two simulations with different initial geometries. These are generally "A-B" type comparisons, where two results are compared. Sometimes, however, the comparisons need to take place between a series of data sets, for example when doing a parameter study or looking at time-varying data.

In this paper, we present a powerful system for facilitating data-level comparisons. Our system incorporates multiple inputs and creates an output data set that reflects a comparison. The system combines two key technologies:

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 59–72
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

1. *cross-mesh field evaluation* (CMFE), where a field from a “donor” mesh is evaluated onto a “target” mesh, and
2. *derived quantity generation*, where new fields are created from existing ones

These two capabilities combine to form a powerful and flexible system where end users are given great latitude to create data sets tailored to application specific comparative metrics.

Given a set of donor meshes $M_1, M_2, ..., M_k$ each containing a corresponding field F_1, $F_2, ..., F_k$, and a target mesh M_C, We evaluate each field F_i onto M_C, creating a new field on the target mesh. These new fields are analyzed using the derived quantity subsystem, creating a new field over M_C that can be displayed using conventional visualization methods. Cross-mesh field evaluation is related to derived quantity generation because, from the perspective of the target mesh, it results in the creation of a new field. We exploit this relationship by integrating cross-mesh field evaluation with derived function generation in a data-parallel distributed memory implementation, making it applicable to a number of very large scale problems.

Research related to this effort is addressed in Section 2. Section 3 describes data-level comparison methods and illustrates the problems of cross-mesh field evaluation. Section 4 describes our system and the data-parallel implementation of cross-mesh field evaluation. Results of this effort are given in Section 5, and in the supplementary material.

2 Related Work

Three basic areas of comparative visualization – *image-based*, *data-level*, and *topological* comparisons – have been discussed in the literature (see Shen et al. [17]). Image-based methods generate multiple images, and provide methods by which a user can compare the image data; data-level methods compare the actual data between input data sets, and provide methods by which the differences can be compared; and topological-based methods compare results of features generated for each of the input data sets.

Most image-based comparison systems [9, 21, 22] perform image differencing algorithms on images from multiple inputs. These systems are limited to comparing visualizations where the data set can be represented by a single image. These techniques are extremely important in the context of comparison to experimental results, such as in the case with [9] and [22]. The VisTrails system of Bavoil et al. [3] coined the phrase *multiple-view comparative system* to describe image-based systems where plots are placed side-by-side or potentially overlaid. In multiple-view comparative systems, the burden is placed on the human viewer to visually correlate features and detect differences.

Topological methods [2, 10, 20] compare features of data sets. Typically, they survey the data set, create summaries of the data, and develop image-based or data-level methods to visualize these summaries.

The ALICE Differencing Engine of Freitag and Urness [11] is a data-level comparison tool. It is limited to comparing data sets that have identical underlying meshes and only allows data differencing comparisons. Shen et al. [17, 18] place data sets on an target mesh and utilize derived quantities to form comparisons. Sommer and Ertl [19] also base their comparisons on data-level methods. Their system employs only connectivity-based differencing, although they also consider the problem of comparisons across parameter studies.

Many visualization systems [1, 7, 12] provide subsystems for generating derived quantities. Moran and Henze give an excellent overview in [15] of their DDV system, using a demand driven calculation of derived quantities. McCormick et al. furthered this approach with Scout [14] by pushing derived quantity generation onto the GPU. Joy et al. [13]

have developed statistics-based derived quantities that greatly expand the uses of derived functions.

Our comparative visualization system has been deployed within VisIt [6], an open source visualization system designed to support extremely large scale data sets. As with many visualization tools [12, 1], it has a data flow network design that leverages pieces of VTK [16]. A key differentiating aspect is its strong contract basis [5], which enables many algorithms to be implemented in parallel. As with the system of Shen et al. [17, 18], the data sets to be compared are placed on a target mesh and derived quantities are generated. Our system contains the following extensions:

- comparison of more than two input data sets, which enables powerful applications for time-varying data and parameter studies,
- the flexibility of applying either position-based or connectivity-based comparisons, and
- a fully parallel, distributed memory solution.

3 Data-level Comparison Methods

Let $M_1, M_2, ..., M_k$ denote a set of donor meshes, each containing a corresponding field $F_1, F_2, ..., F_k$, and let M_C denote some target mesh. The cross-mesh evaluation step evaluates each field F_i, for $i = 1, ..., k$, onto M_C, creating new fields $F_{C,i}, i = 1, ..., k$ that represent each of the original fields on the target mesh. We then employ a function, $D(F_{C,1}, F_{C,2}, ..., F_{C,k}) \rightarrow \Re$, that takes elements of the k fields as input and produces a new derived quantity. The resulting field F_D, defined over M_C, is then visualized with conventional algorithms. (Most previous systems consider only M_1 and M_2 and visualize $F_D = F_{C,1}$ - $F_{C,2}$.)

M_i and F_i can come from simulation or experimental observation. Some common examples are: two or more related simulations at the same time slice, multiple time slices from one simulation, or a data set to be compared with an analytic function.

The target mesh M_C can be a new mesh or one of the donor meshes M_i. In practice, we frequently use the latter option. However, this system supports the generation of new arbitrary rectilinear grids with a user-specified region and resolution.

There are two choices for generating $F_{C,i}$: the evaluation can be made either in a position-based fashion or in a connectivity-based fashion. *Position-based evaluation* is the most common method. Here, the donor and target meshes are overlaid and overlapping elements (cells) from the meshes are identified. Interpolation methods are applied to evaluate F_i on M_C. This technique is difficult to implement, especially in a parallel, distributed memory setting, because M_C and M_i may be partitioned over processors differently – which requires a re-partitioning to align the data. This re-partitioning must be carefully constructed to ensure that no processor exceeds primary memory.

Connectivity-based evaluation requires that both the donor mesh, M_i, and the target mesh, M_C, are *homeomorphic*. That is, they have the same number of elements and nodes, and the element-to-node connectivities for all elements are the same. This is often the case if M_C is selected from one of the M_i, because for comparisons across time and/or parameter studies the remaining $M_{j:j \neq i}$ typically have the same underlying mesh. In this case, the value of an element or node in M_i is directly transferred to the corresponding element or node in M_C. Connectivity-based cross-mesh field evaluation is substantially faster; the difficult task of finding the overlap between elements is eliminated.

For Eulerian simulations (where node positions are constant, but materials are allowed to move through the mesh), connectivity-based evaluation yields the same results as position-

Table 1 Common derived quantities for comparisons. For simplicity, we assume only two donor meshes, M_1 and M_2. The target mesh (M_C) is M_2, and the fields from the evaluation phase are F_1 and F_2. Derivation 1 allows users to explore the differences between the two data sets. Derivation 2 is used in a similar fashion, but amplifies small changes on relatively small quantities. Derivations 3-5 (minima, maxima, and averages) are useful when going beyond simple "$A - B$" comparisons for a large number of donor meshes. When analyzing time series or parameter studies with many related inputs, these quantities allow for visualization of all data sets with a single, composite data set.

	Description	Definition
1	Difference	$F_2 - F_1$
2	Relative difference	$(F_2 - F_1)/(F_2 + F_1)$
3	Maximum or minimum	$(F_2 > F_1\ ?\ F_2 : F_1)$ $(F_2 < F_1\ ?\ F_2 : F_1)$
4	Determine the simulation containing the maximum or minimum	$(F_2 > F_1\ ?\ 2 : 1)$ $(F_2 < F_1\ ?\ 2 : 1)$
5	Average	$(F_2 + F_1)/2$

based evaluation but with higher performance. For Lagrangian simulations (where materials are fixed to elements, but the nodes are allowed to move spatially), position-based evaluations are used most often, because connectivity-based evaluations typically do not make sense in the context of moving nodes. However, connectivity-based evaluation allows for new types of comparisons, because comparisons can take place along material boundaries, even if they are at different spatial positions. Further, connectivity-based evaluations allow for simulation code developers to pose questions such as: how much compression has an element undergone? (This is the volume of an element at the initial time divided its volume at current time.)

There are limitless forms of derived quantities, F_D, that are necessary for different types of comparisons in different situations. We list a few of the most frequently used in Table 1 as examples.

4 System Description

Three critical pieces make up the comparative system: the derived quantity system, the cross-mesh field evaluation methods, and key portions that allow a distributed, parallelized implementation. We describe our comparative system within the constraints of VisIt [6].

4.1 Data-flow-based Visualization Systems

A number of visualization systems make heavy use of data flow networks, a common design for providing interoperability and extensibility. An instantiation of a data flow network, referred to as a pipeline, consists of a source, filters, and a sink. An example source is a file reader, an example sink is a rendering module, and an example filter is a visualization algorithm, such as slicing or contouring. In this model, data flows from the source, through the filters, and ultimately to the sink.

When operating in parallel, each of VisIt's processors instantiate an identical data flow network, which only differs in the data it operates on. So, when each filter executes, there

Table 2 Our expression language allows for all of these functions to be combined in arbitrary ways.

Math	+, -, *, /, power, log, log10, absval, ...
Vector	cross, dot, magnitude
Tensor	determinant, eigenvector, effective (e.g. strain), ...
Mesh	coordinates, polar, volume, area, element_id, ...
Field Operator	gradient, divergence, curl, Laplacian
Relational	if-then-else, and, or, not, $<, \leq, >, \geq, =, \neq$
Mesh Quality	shear, skew, jacobian, oddy, largest angle, ...
Trigonometric	sine, cosine,..., arctangent, degree2radian, ...
Image Filters	mean, median, conservative smoothing
Miscellaneous	recenter, surf. normal, material vol. fraction, ...

are multiple instances of this filter, one on each processor, and the inputs to these filters form a partition of the input data set.

4.2 Derived Quantities

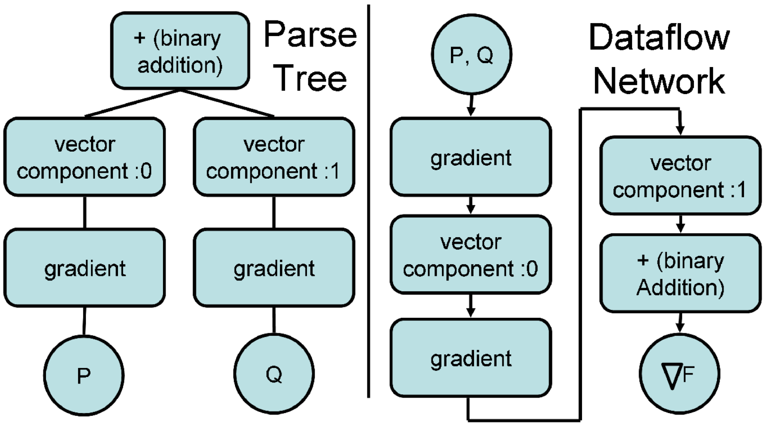
There are two key areas to VisIt's derived quantity system. One is the expression language that allows end users to create new, arbitrary derived quantities. The other is the mechanism that transforms an instance of an expression into a form suitable for data flow networks.

4.2.1 Expression Language

We have developed a functional, string-based system to allow users to create new derived quantities. This results in an expression language syntax that enables users to compose derived quantities in arbitrary ways. A major goal of the design of our expression language was to provide an intuitive interface where creation of new derived quantities required little to no learning curve for common operations. For example, the average of two fields, A and B, is as simple as "$(A+B)/2$".

Of course, users will want to create derived quantities that are more than simple mathematical constructs. Support exists in the language for composing scalar quantities into vectors or tensors, and for extracting scalar components back out. Notation for strings, lists, ranges, and strides allows selection of materials, parts, and other subsets of elements. For accessing other files, either within the same sequence or in different sequences, the language supports references by cycle, absolute and relative time index, and filename. Other named operations are referenced as functions, and a small selection of the over one hundred available are listed in Table 2.

The strength of our expression language lies in the richness of functionality and the interoperability between these expressions. Consider, for example, computing divergence ($\bigtriangledown$). If a two-dimensional vector F is defined as $P\hat{x}+Q\hat{y}$, then $\bigtriangledown F = \frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y}$. A user can calculate divergence directly using the built-in function, divergence(). But, for illustrative purposes, it is also straightforward to calculate divergence using other functions as building blocks: "divF = gradient(P)[0] + gradient(Q)[1]". VisIt has a custom scanner and parser that constructs a parse tree based on expressions like this one. Figure 1 contains the parse tree for this divergence expression.



■ **Figure 1** Creation of a derived quantity. On the left, we see an expression's parse tree. On the right, we see the linearized data flow network for that parse tree.

When VisIt assembles a data flow network, it relies on an Expression Evaluator Filter (EEF) to construct derived quantities. The EEF is typically inserted as the first filter in the pipeline, immediately following the file reader module. In a preparatory phase, the other filters place the names of their required variables into a list. When executing, the EEF cross-references this list with known expression names to determine what derived quantities need to be calculated.

The EEF dynamically creates a sub-network to create needed derived quantities. To do this, it first consults the parse trees of the expressions. For each node in each parse tree, a filter that can perform the corresponding operation is placed into the sub-network. Ultimately, this sub-network reflects a linearized form of the parse trees. The linearization process requires the EEF to do dependency checking between all of the parse trees for all of the expressions involved to ensure that every filter has the inputs it needs. VisIt's implementation supports the linearization of *any* parse tree, including this dependency checking.

Our system supports the accumulation of partial results onto the target mesh M_C so that individual $F_{C,i}$'s can be quickly discarded. While the sub-network is executing, the EEF is able to determine when intermediate variables are no longer needed and remove them. Through this mechanism, the EEF is able to successfully handle many related data sets that would otherwise exceed the available memory.

To perform a cross-mesh field evaluation, the user defines an expression involving built-in functions for the evaluation algorithms – position-based or connectivity-based. Like all other expressions, the comparison expressions have corresponding filters that can be placed in the sub-network to perform the cross-mesh evaluation. By combining these algorithms with expressions, users can direct the creation of new, derived quantities from a multitude of sources on the same target mesh. Furthermore, they can manipulate these quantities using all of the previously mentioned expressions to create interesting comparisons.

Finally, although derived quantity generation typically takes place immediately after reading the data, it is also possible to defer their evaluation until later in the pipeline. This ability allows for the target mesh to be transformed before the cross-mesh field evaluation takes place. This is important when registration is needed, for example for comparison with experimental data.

4.3 Cross-Mesh Field Evaluation

The implementations of the various filters for cross-mesh field evaluation (CMFE) are similar. They all have one input for the target mesh and they all are capable of dynamically instantiating an additional data flow network to obtain M_i and F_i. The differentiating point between the CMFE filters is how they evaluate the fields.

The connectivity-based CMFE algorithm is straightforward. For each element or node, it places F_i onto M_C to create $F_{C,i}$. The only subtlety is guaranteeing that the partitioning of the input data (in a parallel setting) is done so that each processor operates on the same chunks of data.

The position-based CMFE algorithm is complex. There are three major challenges:

1. the overlay step – identifying which elements of the donor mesh M_i overlap with an element in the target mesh M_C.
2. the interpolation step – fitting an interpolant for the field on the target mesh M_C such that it matches, as closely as possible, at key points on the donor mesh M_i.
3. Managing the distribution of data to maximize parallel computational efficiency in a distributed-memory environment.

4.3.1 The Overlay Step

We use interval trees [8] to efficiently identify elements from meshes M_i and M_C that overlap spatially. We start by placing all elements from M_i into the interval tree. Then, for each element of M_C, we use its bounding box to index the interval tree and find the list of elements from M_i with overlapping bounding boxes. We examine this list to find the elements that truly overlap (as opposed to only having overlapping bounding boxes). If M_C contains N_C elements and M_i contains N_i elements, then the time to generate the tree is O($N_i$log(N_i)) and the time to locate the elements of M_i that overlap with an element of M_C is O(log(N_i) + α), where α is the number of elements from M_i returned by the search. This gives a total time of O((N_C + N_i)*log(N_i)). Note that α is amortized out for all but degenerate mesh configurations.

4.3.2 Field Interpolation

For each position x on the target mesh M_C, we evaluate the field on the donor mesh M_i at x, and assign the value at that location to M_C. This method was chosen because it favors performance over accuracy. A good improvement to our implementation, however, would be to add the use of weighted averaging with weights based on volume overlaps.

4.3.3 Parallel Implementation

The final piece of the problem is to perform cross-mesh field evaluations in a parallel, distributed-memory environment. The key issue deals with spatial overlap. When a processor is evaluating a field from mesh M_i onto the target mesh M_C, it must have access to the portion of M_i that overlaps spatially with the portion of M_C it is operating on. Our strategy for this issue is to create a spatial partition to guide re-distribution of both meshes for the evaluation phase. Unfortunately, the spatial partition must be created with great care. If the partition divides space into regions that cover appreciably different numbers of elements, it will lead to load imbalances and potentially exhaust memory. Therefore, we focus on creating a *balanced spatial partitioning*, where “balanced" implies that every region contains approximately the same number of elements, E_t (see Figure 2). The E_t elements

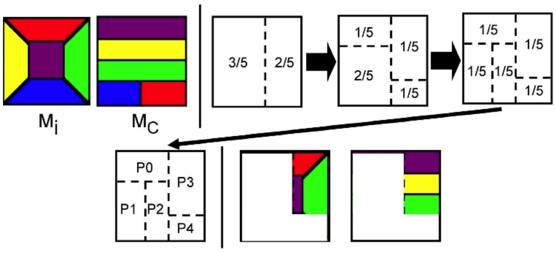


Figure 2 In the upper left, two meshes, M_i and M_C, are shown. Assume the red portions are on processor 1, blue on 2, and so on. We employ an iterative strategy that creates a balanced spatial partition. We start by dividing in X, then in Y, and continue until every region contains approximately $1/N^{th}$ of the data, where N is the total number of processors. Each processor is then assigned one region from the partition and we communicate the data so that every processor contains all data for its region. The data for processor 3 is shown in the last set of figures.

from each region may contain different proportions of elements from M_C and M_i; in general, it is not possible to have this proportion be fixed and E_t be equal on all processors.

The algorithm to efficiently determine a balanced spatial partitioning is recursive. We start by creating a region that spans the entire data set. On each iteration and for each region that represents more than $1/N^{th}$ of the data (measured in number of elements covered), we try to select “pivots", possible locations to split a region along a given axis. This axis changes on each iteration. All elements are then traversed, and their positions with respect to the pivots are categorized. If a pivot exists that allows for a good split, then the region is split into two sub-regions and recursive processing continues. Otherwise we choose a new set of pivots, whose choice incorporates the closest matching pivots from the previous iteration as extrema. If a good pivot is not found after some number of iterations, we use the best encountered pivot and accept the potential for load imbalance.

The implementation of this algorithm is complicated by doing many parallel pivot locations at one time. The above procedure, if performed on a single region at a time, would have a running time proportional to the number of processors involved, which is unacceptable. To overcome this, we concurrently operate on many regions at one time. When iterating over a list of elements, we avoid the poor strategy of interacting with regions that do not even contain the element. Instead, we employ a separate interval tree that stores the bounding boxes of the regions. Then, for each element, we can quickly locate exactly the regions that element spans. This variation in the algorithm gives a running time proportional to the logarithm of the number of processors, which is more palatable.

Balanced spatial partitioning only guarantees that the total number of elements from both M_C and M_i are approximately equal. Our interval tree-based approach gives the best results if the number of elements from M_i are balanced as well.

After the best partition is computed, we create a one-to-one correspondence between the regions of that partition and the processors. We then re-distribute M_i, F_i, and M_C with a large, parallel, all-to-all communication phase. If elements belong to multiple regions, they are sent to all corresponding processors. After the communication takes place, evaluation takes place using the interval-tree based identification method described previously. Finally, all of the evaluations are sent back to the originating processor and placed on M_C.

We used the data set from Section 5 for a rough illustration of performance. The evaluation is of a 1.5 million element unstructured grid onto a 1K x 1K x 676 rectilinear grid, in parallel, using eighty processors. The most expensive phase is evaluation. In this phase, each processor is doing nearly ten million lookups on its interval tree. Table 3 summarizes

Table 3 The time spent in the different phases of the parallelized cross-mesh field evaluation algorithm. The communication column represents communicating data to create the balanced spatial partitioning, and also the time to return the final evaluations.

Phase	Create Spatial Partition	Communication of Data	Build Interval Tree	Eval
Time	0.7s	2.9s	5.2s	27.4s

the times spent in different phases of the algorithm. The inclusion of this information clearly does not serve as a performance study, which will be studied further in the future. However, it does inform as to the general running time for large problems.

5 Results

We have provided an interface that allows the user to manage the entire comparison process, including what data sets are compared, how, and onto what target mesh. This system has been implemented in VisIt. We illustrate the systems use through the following examples.

5.1 Rayleigh-Taylor Instabilities

Rayleigh-Taylor instability simulations model the mixing of heavy and light fluids. For this study, we looked at two types of related data sets. First, we investigated a single simulation and its evolution in time. Then we looked at a parameter study, where turbulence parameters were varied to study how differences in these parameters affected the results.

5.1.1 Time-Varying Data

We started our analysis by looking at a single Rayleigh-Taylor instability calculation that simulated ten microseconds, outputting eighty-five time slices. Rather than focus on the differences between two time slices, we created visualizations that would summarize the whole data set. In particular, we were interested in summaries derived from a given binary condition, B_C(P, T), where B_C(P, T) is true if and only if condition C is true at point P and time T. For a given point P, the derived quantity was:

$$\mathit{time}(P) : B_C(P, \mathit{time}) \text{ AND } (\neg \exists\, t' : t' < \mathit{time} \text{ AND } B_C(P, t'))$$

This derived quantity is a scalar field that, for each point P, represents the first time that $B_C(P, T)$ is true. For our study, since we were observing the mixing of two fluids, we chose $B_C(P, T)$ to be whether or not mixing between the fluids occurs at point P at time T. From Figure 3, we can see that the mixing rate increased as the simulation went on (because there is more red than blue in the picture). We comment that the technique demonstrated here, showing a plot of the first time a binary condition is true in the context of time varying data, is very general. Further, we believe this is the first time that it has been presented in the context of creating these plots of this form (by using of data-level comparative techniques).

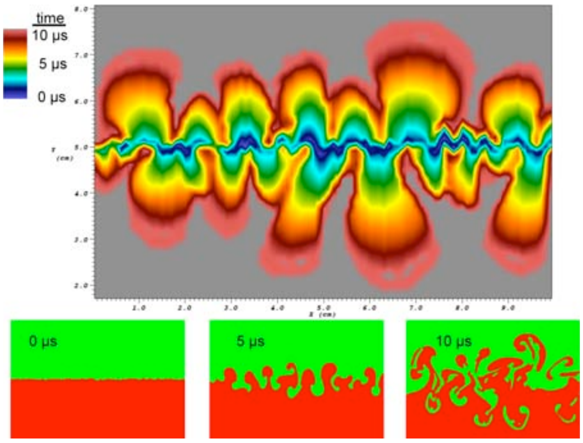


Figure 3 Along the top, we see a visualization comprising all time slices. Blue areas mixed early in the simulation, while red areas mixed later. Gray areas did not mix during the simulation. This plot allows us to observe mixing rates as well. Along the bottom, we include three time slices for reference. Heavy fluids are colored green, light fluids are colored red.

5.1.2 Parameter Studies

A simulation of a Rayleigh-Taylor instability is dependent on certain coefficients, which are adjusted in different situations. An important question is to understand how variation in these coefficients affects the outcome of a simulation. These effects can be monitored during parameter studies, where these coefficients are varied and the results are compared. For this parameter study, two coefficients were varied independently: the coefficient for turbulent viscosity and the coefficient of buoyancy. For each coefficient, five values were chosen. Twenty-five calculations were then performed, one for each pair of coefficients.

We focused on differences in magnitude of velocity, i.e. speed. This quantity had the most variation throughout the simulations and we wanted to characterize the relation between speed and the coefficients. We examined three different derived quantities defined over the whole mesh. The first quantity was the simulation index that resulted in the maximum speed at the given point. The second and third, respectively, were the coefficients of turbulent viscosity and buoyancy corresponding to that simulation index.

If i is a simulation identifier/index, $C_{tv}(S)$ and $C_b(S)$ are the turbulent viscosity and buoyancy coefficients for i, then the derived quantities, for each point P are:

1. $maxsid(P) = \underset{i \in 1,\ldots,25}{arg\,max}\ speed_i(\mathrm{P})$
2. $C_{tv}\_of\_max\_speed(P) = C_{tv}(maxsid(P))$
3. $C_b\_of\_max\_speed(P) = C_b(maxsid(P))$

The results of these derived quantities are displayed in Figure 4. From the $maxsid(P)$ plot, we can see that no one simulation dominates the others in terms of maximum speed. From the $C_b\_of\_max\_speed(P)$ plot, we can draw modest conclusions, but it would be difficult to claim that this term is greatly affecting which simulations have the maximum speed. From the $C_{tv}\_of\_max\_speed(P)$ plot, we can see that most of the high speeds either come from very low or very high turbulent viscosity coefficients (colored blue and red, respectively). We quantified this observation (see Table 4), and found that the simulations with extreme turbulent viscosity coefficients had over three quarters of the total area, meaning that the relationship between high speeds and turbulent viscosity is large.

Table 4 Quantifying how much space each coefficient covered in terms of percentage of the total space.

Coefficient	very low	low	middle	high	very high
Buoyancy	4.2%	21.1%	20.4%	17.5%	36.8%
Turbulent Viscosity	47.2%	8.1%	4.8%	8.7%	31.0%

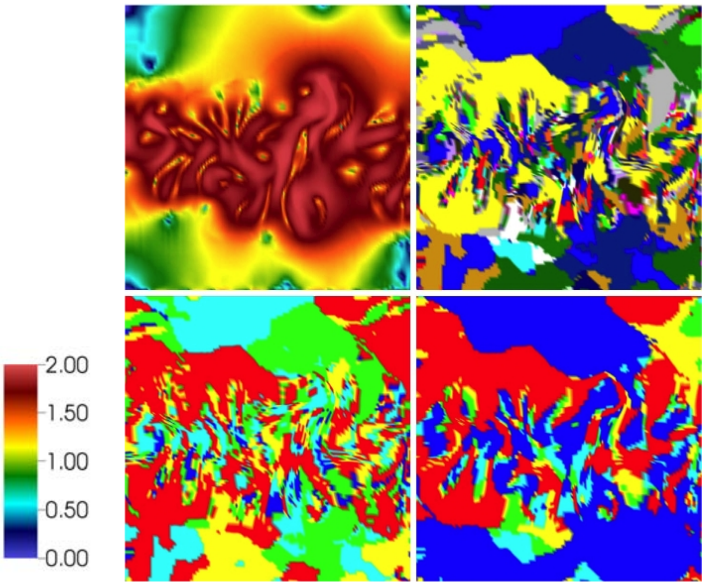


Figure 4 In the upper left, we see a normal rendering of speed for a single simulation. In the upper right, we color by $maxsid(P)$. In the lower left, we color by $C_b\_of\_max\_speed(P)$. In the lower right, we color by $C_{tv}\_of\_max\_speed(P)$.

Finally, we were also interested in quantifying the changes from simulation to simulation in our parameter study. We did this by calculating the following derived quantities:

1. $\frac{1}{25}\sum_{i=1}^{25} speed_i$
2. $\max\limits_{i=1,\ldots,25} speed_i$
3. $\min\limits_{i=1,\ldots,25} speed_i$
4. $\max\limits_{i=1,\ldots,25} speed_i - \min\limits_{i=1,\ldots,25} speed_i$

The results are shown in Figure 5. The fourth quantity informs an analyst as to the maximum differences possible for each point in space. Since we are performing these operations on a parameter study, we are effectively quantifying the uncertainty for this simulation. Of course, there are many alternative ways that an analyst may want to construct uncertainty information from ensembles of simulations. But we believe that through the examples we have presented in this section, we have motivated the capability of our system to do so and the importance of a flexible and powerful derived quantity system.

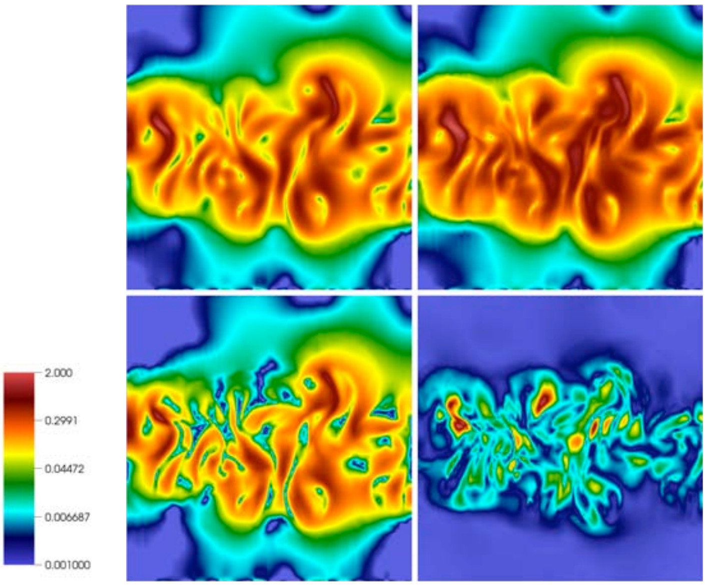


Figure 5 In the upper left, we are coloring by average speed, in the upper right by maximum speed, in the lower left by minimum speed, and in the lower right maximum variation. The first three plots vary in speed from zero to two and are colored using a logarithmic scale. The fourth plot (of differences) ranges from zero to one and is colored linearly. These plots effectively quantify the uncertainty for this ensemble of simulations.

6 Conclusions

We have demonstrated a powerful system for data-level comparative visualization and analysis. The system we have developed supports a wide range of comparative analyses. In addition, the system is very versatile not only in the modalities of cross-mesh field evaluation it offers, but also in the range of derived quantities that can be generated. Finally, it supports these operations in a highly scalable, distributed memory, parallel computational paradigm.

This methodology has greatly expanded the types of comparative visualization that can be addressed. By integrating this system into an open-source product, we have been able to get this technology in the hands of scientists and engineers. Future work will focus on the enhancements of these technologies to generate new comparative methods that impact their work.

Acknowledgments

This work was supported by the Director, Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-FC02-06ER25780 through the Scientific Discovery through Advanced Computing (SciDAC) program's Visualization and Analytics Center for Enabling Technologies (VACET). Portions of this work were performed in part under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

1 Greg Abram and Lloyd A. Treinish. An extended data-flow architecture for data analysis and visualization. Research report RC 20001 (88338), IBM T. J. Watson Research Center, Yorktown Heights, NY, February 1995.

2 Rajesh K. Batra and Lambertus Hesselink. Feature comparisons of 3-D vector fields using earth mover's distance. In *Proceedings of the IEEE Visualization Conference*, pages 105–114, 1999.

3 L. Bavoil, S.P. Callahan, P.J. Crossno, J. Freire, C.E. Scheidegger, C.T. Silva, and H.T. Vo. VisTrails: enabling interactive multiple-view visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):135–142, 2005.

4 Steven P. Callahan, Juliana Freire, Emanuele Santos, Carlos E. Scheidegger, Cláudio T. Silva, and Huy T. Vo. Vistrails: visualization meets data management. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 745–747, New York, NY, USA, 2006. ACM.

5 Hank Childs, Eric Brugger, Kathleen Bonnell, Jeremy Meredith, Mark Miller, Brad Whitlock, and Nelson Max. A contract based system for large data visualization. In *Proceedings of the IEEE Visualization Conference*, 2005.

6 Hank Childs and Mark Miller. Beyond meat grinders: An analysis framework addressing the scale and complexity of large data sets. In *SpringSim High Performance Computing Symposium (HPC 2006)*, pages 181–186, 2006.

7 Computational Engineering International, Inc. *EnSight User Manual*, May 2003.

8 T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms.* McGraw-Hill Book Company, 1990.

9 Willem C. de Leeuw, Hans-Georg Pagendarm, Frits H. Post, and Birgit Walter. Visual simulation of experimental oil-flow visualization by spot noise images from numerical flow simulation. In *Visualization in Scientific Computing*, pages 135–148, 1995.

10 Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. Local and global comparison of continuous functions. In *Proceedings of the IEEE Visualization Conference*, pages 275–280, October 2004.

11 L. Freitag and T. Urness. Analyzing industrial furnace efficiency using comparative visualization in a virtual reality environment. Technical Report ANL/MCS-P744-0299, Argonne National Laboratory, 1999.

12 C.R. Johnson, S. Parker, and D. Weinstein. Large-scale computational science applications using the SCIRun problem solving environment. In *Proceedings of the 2000 ACM/IEEE conference on Supercomputing*, 2000.

13 Kenneth I. Joy, Mark Miller, Hank Childs, E. Wes Bethel, John Clyne, George Ostrouchov, and Sean Ahern. Frameworks for visualization at the extreme scale. In *Proceedings of SciDAC 2007, Journal of Physics: Conference Series*, volume 78, page 10pp, 2007.

14 Patrick S. McCormick, Jeff Inman, James P. Ahrens, Charles Hansen, and Greg Roth. Scout: A hardware-accelerated system for quantitatively driven visualization and analysis. In *Proceedings of the IEEE Visualization Conference*, pages 171–178, Washington, DC, 2004. IEEE Computer Society.

15 Patrick J. Moran and Chris Henze. Large field visualization with demand-driven calculation. In *Proceedings of the IEEE Visualization Conference*, pages 27–33, Los Alamitos, CA, 1999. IEEE Computer Society Press.

16 William J. Schroeder, Kenneth M. Martin, and William E. Lorensen. The design and implementation of an object-oriented toolkit for 3d graphics and visualization. In *Proceedings of the IEEE Visualization Conference*, pages 93–ff. IEEE Computer Society Press, 1996.

17 Qin Shen, Alex Pang, and Sam Uselton. Data level comparison of wind tunnel and computational fluid dynamics data. In *Proceedings of the IEEE Visualization Conference*, pages 415–418, Los Alamitos, CA, 1998. IEEE Computer Society Press.

18 Qin Shen, Sam Uselton, and Alex Pang. Comparison of wind tunnel experiments and computational fluid dynamics simulations. *Journal of Visualization*, 6(1):31–39, 2003.

19 O. Sommer and T. Ertl. Comparative visualization of instabilities in crash-worthiness simulations. In *Data Visualization 2001, Procceedings of EG/IEEE TCVG Symposium on Visualization*, pages 319–328, 2001.

20 Vivek Verma and Alex Pang. Comparative flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):609–624, 2004.

21 Peter L. Williams and Samual P. Uselton. Foundations for measuring volume rendering quality. Technical Report NAS/96-021, NASA Numerical Aerospace Simulation, 1996.

22 Hualin Zhou, Min Chen, and Mike F. Webster. Comparative evaluation of visualization and experimental results using image comparison metrics. In *Proceedings of the IEEE Visualization Conference*, Washington, DC, 2002. IEEE Computer Society.

On the Computation of Integral Curves in Adaptive Mesh Refinement Vector Fields

Eduard Deines[1], Gunther H. Weber[2], Christoph Garth[1], Brian Van Straalen[2], Sergey Borovikov[3], Daniel F. Martin[2], and Kenneth I. Joy[1]

1 **University of California, Davis, USA**
`{edeines,cgarth,kijoy}@ucdavis.edu`

2 **Lawrence Berkeley National Laboratory, Berkeley, USA**
`{ghweber,bvstraalen,dfmartin}@lbl.gov`

3 **University of Alabama, Huntsville, USA**
`{snb0003}@uah.edu`

Abstract

Integral curves, such as streamlines, streaklines, pathlines, and timelines, are an essential tool in the analysis of vector field structures, offering straightforward and intuitive interpretation of visualization results. While such curves have a long-standing tradition in vector field visualization, their application to Adaptive Mesh Refinement (AMR) simulation results poses unique problems. AMR is a highly effective discretization method for a variety of physical simulation problems and has recently been applied to the study of vector fields in flow and magnetohydrodynamic applications. The cell-centered nature of AMR data and discontinuities in the vector field representation arising from AMR level boundaries complicate the application of numerical integration methods to compute integral curves. In this paper, we propose a novel approach to alleviate these problems and show its application to streamline visualization in an AMR model of the magnetic field of the solar system as well as to a simulation of two incompressible viscous vortex rings merging.

1998 ACM Subject Classification I.3 Computer Graphics

Keywords and phrases integration-based visualization, streamlines, interpolation, adaptive mesh refinement

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.73

1 Introduction

Simulation and visualization are key components of the modern scientific knowledge discovery pipeline across many different disciplines. In this context, the task of visualization is to provide insight into science problems from raw data generated by measurement or computer simulation. In certain application areas, such as astrophysics, the simulation must accommodate many different spatial scales. For example, in simulations of the solar system, the computational domain may span thousands of astronomical units (AU), whereas some physical structures that need to be resolved occupy significantly shorter length scales. As a consequence, it is not possible to use regular grids, which in many cases are a primary choice due to their conceptual simplicity, to discretize such simulation domains. The uniform resolution required to represent the smallest phenomena in a simulation would result in infeasible overall storage requirements. Unstructured grids, on the other hand, adapt well to differences in length scales, but they also introduce significant overhead by requiring an explicit representation

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 73–91
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

of grid connectivity. Adaptive mesh refinement (AMR) techniques are a hybrid solution that discretizes the simulation domain as set of overlapping, nested, axis-aligned rectilinear grids [4]. These grids are arranged in levels of increasing resolution, and information for a specific point of the domain is stored at the finest grid overlapping this point. Thus, AMR combines the adaptivity of unstructured grids with the implicit connectivity of regular grids, adding only little overhead in form of a layout description. This arrangement makes it possible to discretize adaptively simulations with comparatively little overhead.

In the visualization process, AMR data presents two challenges not found in other data representations. First, portions of the coarse grid that span the simulation domain are replaced with higher-accuracy information provided by grids at levels of higher resolution. Thus, it is necessary to detect this overlap and always consider data at the finest resolution available. Second, for methods that are built on the existence of a continuous interpolant over the entire data set, special care must be taken to reconcile coarse and fine information resolution at resolution boundaries to yield continuous interpolation. AMR simulations often specify cell-centered values, which further complicates interpolation.

Previous work on addressing these problems for the visualization of AMR data focused mainly on scalar field visualization techniques such as isosurface extraction and direct volume rendering (Section 2.2 provides a brief overview). In this paper, we focus on making it possible to use integration-based visualization techniques on AMR data sets. For this purpose, we investigate existing streamline construction algorithms and apply them to AMR vector fields. The goal of this effort is to identify problems that preclude efficient and accurate visualization of AMR vector fields and then develop a novel algorithm to address these problems. We perform a numerical evaluation of the accuracy of our method, compare it against existing approaches, and document the applicability of our scheme by using it to visualize two AMR data sets from astrophysics and fluid flow simulations. Finally, we discuss limitations of our approach and possible future extensions to overcome them.

2 Background and Related Work

2.1 Adaptive Mesh Refinement (AMR)

Adaptive Mesh Refinement (AMR), introduced by Berger and Oliger [3], combines the topological (structural) simplicity of regular, rectilinear grids with the adaptivity of unstructured grids to local changes in resolution: The computational domain is discretized using one or more coarse grids (also called boxes or patches) in a *root level*. During the simulation, an error estimate is computed for each cell of this root level, and all cells whose error estimate exceeds a given threshold are tagged for refinement. Subsequently, a set of rectilinear grids with an increased resolution is created that overlaps all tagged cells. These tagging and grid creation steps are repeated recursively until all regions are represented with adequate resolution. Simulation results from higher-refined levels are propagated back to the coarser levels, resulting in a consistent representation of the domain. In Berger and Oliger's original approach [3], newly created grids are structured, rectilinear grids that can be rotated with respect to the parent level. Berger and Collela published a modified version [4] of this algorithm where newly created grids are axis aligned with respect to the parent level, which is now widely used for simulations in applied science and engineering [35, 9, 12, 14, 31, 13, 1].

A Berger-Colella AMR hierarchy consists of axis aligned structured grids as building blocks (see Fig. 1 for an example). These grids are arranged in levels, starting at the coarsest root level up to the finest level of resolution. The hierarchy shown in Fig. 1, for example, consists of three levels, including the root level. Within a level, grids have identical resolutions.

Going from one level to the next finer level, a coarser grid cell is alway split into an integer number of finer grid cells, where the integer *refinement ratio* specifies the number of fine grid cells per coarse grid cell along each axis. The AMR hierarchy shown in Fig. 1 has a refinement ratio of two, meaning that each coarser grid cell is covered by two finer grid cells along each axis. Since the depicted data set is defined on a two-dimensional domain, each coarser grid cell is covered by a total of four finer cells. Furthermore, a fine grid always starts at the boundary of a coarse grid cell and a coarser grid cell is always either refined completely or not at all. An AMR level is always contained completely within the next coarser level. While a single grid (e.g., the grid comprising the third level in Fig. 1) at a given level may overlap more than one grid of the next coarser level, it is always enclosed in the union of all grids of that coarser level. Finally, transitions only occur between two consecutive levels. There is always a layer of grid cells of the next coarser level around all grids of a refining level.

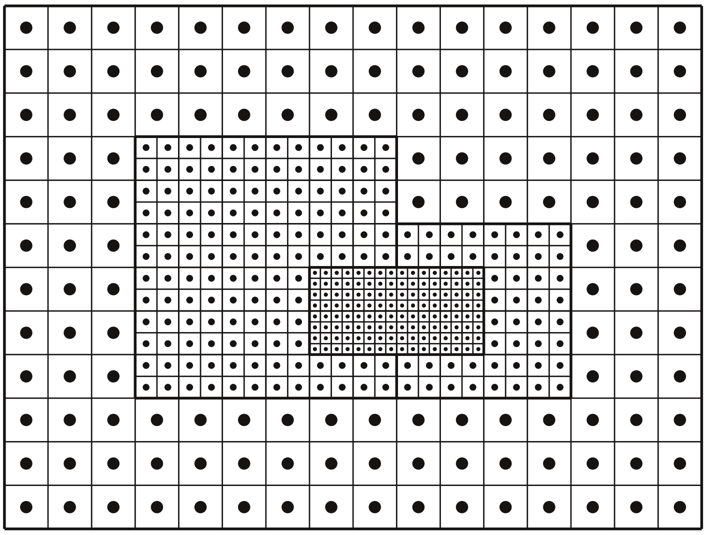


Figure 1 AMR mesh consisting of four grids arranged in three refinement levels.

An important aspect in the parallelization of simulations working on structured, rectilinear grids (AMR and other approaches that utilize multiple grids) is the concept of *ghost cells*, see Fig. 2. Most simulation codes require the computation of derivatives and approximate these derivatives based on the difference between values in cells adjacent to the current cell; however, cells at boundaries do not have neighbors in all directions. To make it possible to handle all cells in a grid in the same way, one introduces ghost cells to the simulation. These ghost cells can either be external to the simulation, i.e., lie outside the computational domain, or they can be located inside the domain. The latter case occurs if the domain is partitioned into multiple blocks. Where blocks are adjacent to each other, ghost cells replicate data from adjacent blocks, reducing the need for communication in parallel computation. In any case, ghost cells are not considered to be part of the actual simulation domain. Note that if one considers ghost cells, finer AMR levels do not necessarily start at coarser cell boundaries.

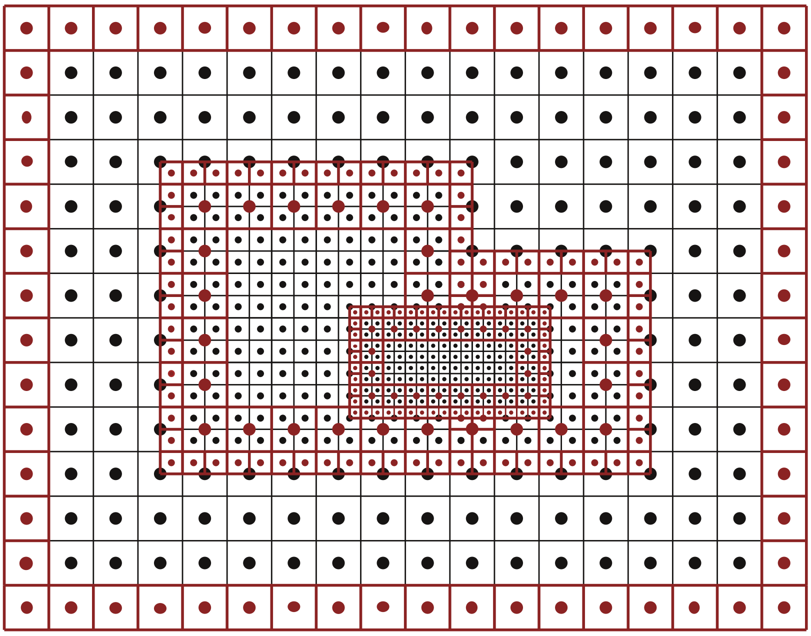


Figure 2 Ghost cells create overlap between resolution levels and allow all cells of a level to have a uniform neighborhood.

2.2 Visualization of AMR Data Sets

Much of the existing research on AMR visualization is focused on the visualization of scalar fields and particularly on the development of effective volume rendering and iso-surface extraction algorithms. Early approaches worked around problems with AMR data in visualization by converting it to a form compatible with existing software implementations, such as regular or unstructured meshes [28]. However, in this conversion the implicit mesh connectivity is lost together with the storage efficiency advantage of AMR over unstructured-grid-based approaches.

Specifically treating the case of iso-surface extraction, Weber et al. developed a method that addresses the discontinuities of interpolation across resolution level boundaries to extract crack-free iso-surfaces from cell-centered AMR data [38, 39]. Similar to our work described below, cell centers of each patch of an AMR hierarchy are interpreted as vertices of a new patch, forming the *dual grid* to the original patch. The use of dual grids leads to gaps between patches comprising different AMR hierarchy levels. Weber et al. used a procedural scheme to fill these gaps with so-called *stitch cells*. Fang et al. [10] presented an alternate iso-surface extraction approach for node-centered AMR data with the main goal of preserving the original patch structures.

Research concerned with volume rendering includes effective sorting schemes for cell-based volume rendering [26], accelerating AMR volume rendering using graphics hardware [37, 20, 16, 29, 18], dealing with interpolation issues [16, 40] and allowing for parallel approaches [23, 20, 41]. Furthermore, Kähler et al. [15] used a set of existing tools to render star formation simulation data produced by the framework developed by Bryan et al. [6]. Finally, Kähler et al. [17] also described a framework for remote visualization of time-varying AMR data.

2.3 Integration-Based Visualization

In the context of vector fields, *integration-based visualization* aims at using the trajectories of idealized, massless particles as basic primitives to construct insightful visualization methods. These particle trajectories, also called *integral curves*, are given as ordinary differential equations over the vector field of interest and are approximated using numerical integration schemes, such as e.g. the class of *Runge-Kutta* schemes. Here, we make use of a scheme by Dormand and Prince termed DoPri5 [32, 33], that is popular for the task of integral curve approximation [11, 34]. It offers 4th-order accuracy, adaptive stepsize control, i.e. the ability to automatically adjust the integration step size based on the complexity of the vector field in the traversed region, and dense output in the form of a polynomial curve describing every integration step.

Beyond direct depiction in the form of *streamlines* (in stationary vector fields) and *pathlines* (in time-varying vector fields), integration-based visualization techniques also include more general techniques. Integral surfaces depict the movement of a family of particles originating on a common curve or surface; this results in an illustrative depiction of vector field structures and avoids visual problems that are often incurred when visualizing many integral curves simultaneously [11]. Topological methods derive specific integral curves, so-called *separatrices*, from an analysis of a vector field's critical points, providing an abstract, skeleton-type overview of a vector field [36]. *Dense methods* focus on depicting all particles traversing a vector field simultaneously to e.g. mimic the effect of flow-induced oil smearing in physical experiment [21].

All these methods rely on the reliable and accurate computation of integral curves of the considered vector field. AMR vector fields present a number of challenges in this context. A number of publications demonstrate streamlines over adaptive meshes [2, 27, 22]; however, they are used as a visual benchmark for the described simulation or grid conversion methods using adaptive techniques, and a description of the used integration algorithm is not provided.

2.4 Problems with Integral Curves in AMR Data

Since the mesh consists of multiple refinement levels, each of which may contain several domains represented as rectilinear grids, the data in the finer levels replaces the coarser level information. Because individual domains across all levels overlap, it is necessary to obtain information from the finest level to represent the vector field data in full fidelity. Identifying the correct domain and level can be a difficult task; below, we present an approach to handle this efficiently. Moreover, due to the typical nature of AMR simulation codes, the vector field is provided on a per-cell basis. Thus, interpolation is required to achieve the requirement of continuity of the vector field for the application of numerical integration. While many interpolation schemes may accommodate this requirement, tri-linear interpolation is typically chosen for efficiency reasons. Here, we examine and compare the typically used cell-averaging interpolation and contrast it with a novel dual-mesh approach that promises better accuracy. Finally, due to the discontinuity of the vector field interpolant when traversing resolution level boundaries, significant errors can be introduced during numerical integration, leading to less accurate or incorrect integral curve approximations. Below, we demonstrate how these boundaries can be explicitly taken into account during integration to avoid this loss of accuracy and correctness.

3 Integral Curves in AMR Vector Fields

In this section we describe several possibilities how we can address the problems of integral curve computation in AMR data sets. Interpolating only in the root grid leads to large differences in resulting streamlines when compared to utilizing properly the data present in all hierarchy levels (see Figure 12). Thus, it is essential to consider the AMR hierarchy. We have examined two different possibilities to handle the AMR refinement grids. To calculate a single integration step, the Dopri54 algorithm needs to perform six vector field evaluations. In our first method, we identify the corresponding AMR grid in the finest possible level for each evaluation point and evaluate the vector field on this grid. The finest domain can be found efficiently using an interval tree structure. Here, we do not consider the boundaries between the different domains and levels. As a consequence, any possible discontinuity issues mentioned above are captured by the error estimation of the Dopri54 method, thereby reducing the integration step size and resulting in a larger number of integration steps (many failed steps) during integral curve computation. Additionally, the computational error in the numerical solution after passing the discontinuity is usually larger than the user prescribed tolerance (see [7]). The individual steps of this approach are summarized in Algorithm 1.

Algorithm 1 AMR_Ignore_Boundaries

```
start in domain in finest possible level
while not finished do
  advance integration step {
  for each vector field evaluation do
    find domain in finest possible level
    evaluate field in this domain
  end for
  }
  add step to curve
end while
```

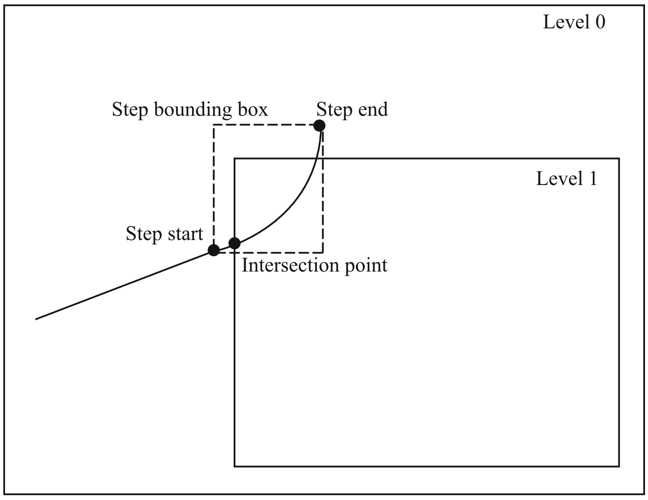


Figure 3 Streamline intersection with level boundary.

The error due to the discontinuity can be avoided by localizing the discontinuity position and stopping and re-starting integration at the discontinuity introducing minimal possible error (see [24, 7]). In our case, discontinuities are located at level boundaries. Thus, we have to calculate an intersection point of the integral curve with the boundaries of the nested refined grids. This is also the transition point where we have to descend into the finer level. We formulate the second approach as follows. We start the integration in the domain in the finest possible level. After each integration step we determine whether a refined region was crossed. For this purpose, we detect whether the bounding box surrounding the current integration step intersects the bounding box containing all the nested grids in the considered domain. If so, we then check for a possible intersection with the boundary of the child domains. Once we have determined an intersection point we can restart the integration in the corresponding nested domain with the intersection position as starting point. This process is illustrated in Figure 3. The DOPRI54 method provides a polynomial representation of the integral curve in each integration step. We use Newton's method to calculate the intersection point. The pseudo code of this approach is outlined in Algorithm 3.

Algorithm 2 Integrate_Domain

```
while inside domain do
  advance integration step
  if cross ghost cells then
    if inside children bounding box then
      if found child then
        intersect with child bounding box
        advance to intersection point
        set new domain to child domain
        outside domain
      else
        add step to curve
      end if
    else
      intersect with domain bounding box
      advance to intersection point
      outside domain
    end if
  else
    add step to curve
  end if
end while
```

Algorithm 3 AMR_Explicit_Boundaries

```
while not finished do
  find domain in finest possible level
  Integrate_Domain (see Algorithm 2)
end while
```

The advantage of the second approach, compared to the first, simpler method, is that we can process the curve integration in each AMR domain separately. In the first approach we additionally need at least the fields of the nested child domains and the neighbor domains in each integration step.

As mentioned in the previous section, the simulated data is mostly cell-centered whereas integration requires vertex-centered vector field data. Only using the vector value given in a cell independently of the position (nearest neighbor interpolation) of the evaluation point inside the cell is not sufficient. Thus, we have to calculate the vector values at grid vertices based on the given cell vector values. Computing the vector at the vertex position as average of the vectors inside the cells that surround a node yields a continuous trilinear interpolant. However, it also has limitations due to very strong smoothing and inaccurate calculation of field values at grid boundaries. Since an AMR hierarchy usually consists of a large number of grids, this method can result in a significant error during integral curve computation.

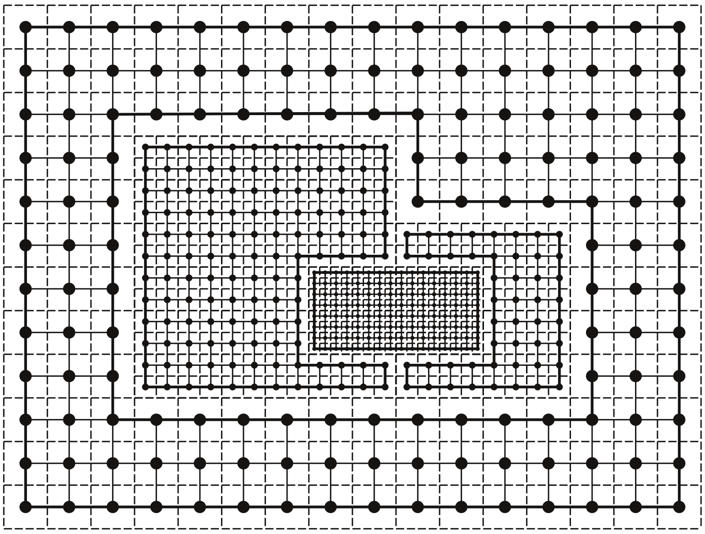


Figure 4 Dual-grid representation without additional ghost cells showing gaps between different AMR grids.

A proven approach in scalar visualization over AMR meshes with empirically better interpolation accuracy for a cell-centerd vector field is a dual-mesh interpolant. The set of all cell centers of the original grid form a grid, called the dual grid. In the case of block-structured AMR the dual grid is also rectilinear. AMR data contains several independent grids. Constructing the dual grid for each of them independently produces gaps between the neighboring domains as shown in Figure 4. A solution to this problem is the computation of an additional ghost layer surrounding the domain grid (see Figure 2). This implies extra data to be stored and processed, but the resulting dual grid of the entire AMR mesh is connected (see Figure 5). Thus, we can use the cell values for integral curve computation. This additional ghost cell layer is often generated during the simulation process.

4 Numerical Evaluation

To examine and evaluate the different methods for the integral curve computation described in the subsections above we perform the following test (see Algorithm 4). We used an 2D AMR mesh with 4 refinement levels and 172 domains. On this mesh we defined a cell-based vector field for which an analytical solution is easy to calculate by:

$$v = \begin{pmatrix} -rsin(\alpha) \\ rcos(\alpha) \end{pmatrix} \tag{1}$$

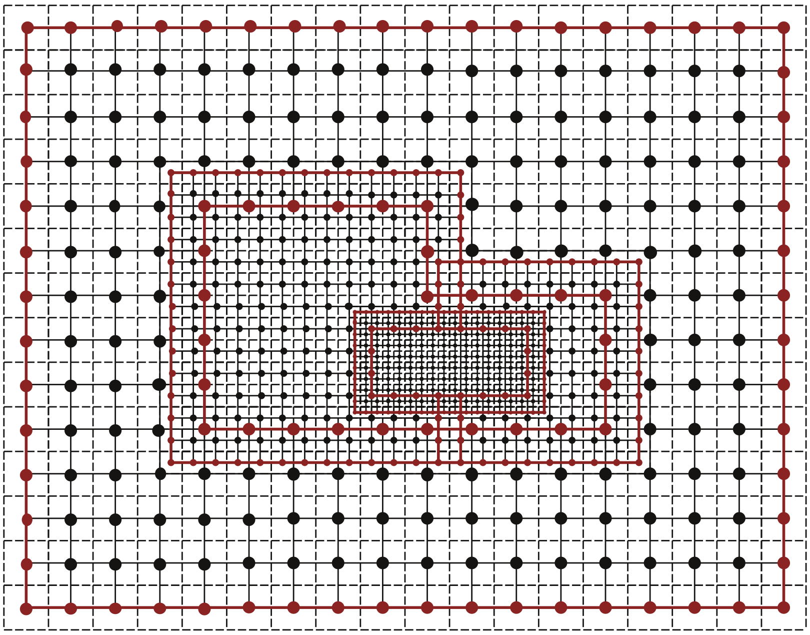


Figure 5 Dual-grid representation with additional ghost cells. The boundaries between different AMR domain grids (red lines) overlap showing no gaps.

with $r = \sqrt{x^2 + y^2}$, $\alpha = atan2(y, x)$, and (x,y) are the positions of the cell centers. We calculate integral curves over the interval $[0, 2\pi]$ in the field v and evaluated the relative error of the final point:

$$E_r = \frac{\|C(2\pi) - S(2\pi)\|}{\|S(2\pi)\|} \tag{2}$$

where C is the computed integral curve and S the exact solution.

In order to make the error calculation more significant and exclude artifacts, we use 100 different initial values and computed the average relative error of these runs. The results are plotted on a log-log scale. Figure 6 depicts the relative error of the test cases AI and DI. For comparison we calculated two additional error plots. The first one shows the error of the integration in the field without using any field interpolation. The integrator uses the

Algorithm 4 Test_Case

```
Case: AI
    cell data to point data using averaging
    execute AMR_Ignore_Boundaries
Case: AE
    cell data to point data using averaging
    execute AMR_Explicit_Boundaries
Case: DI
    build dual-grid
    execute AMR_Ignore_Boundaries
Case: DE
    build dual-grid
    execute AMR_Explicit_Boundaries
```

field value of the corresponding cell regardless of the evaluation position inside the cell. The second represents the error of the integral curve computation only in the coarsest level of the AMR mesh. In Figure 7 the error plots of all four test cases are compared. Finally, Figure 8 depicts the average number of vector field evaluations needed to calculate the curves for all four test cases.

As we can see from the resulting errors, interpolation is required in order to obtain reasonable results. By using the averaging method problems resulting from discontinuous behavior at level boundaries are clearly seen, introducing a large computational error and an increase in the amount of vector field evaluations. However, this error can be reduced by considering the domain boundaries. The dual grid interpolation shows better results compared to the averaging method. Even if we disregard the domain boundaries, the error is small. The cases DI and DE show equal behavior and the resulting error is smaller compared to the error resulting from the integration in the coarsest level of the AMR. One problem that may occur by disregarding the domain boundaries is that the integrator would not descend into refined levels and important features of the considered vector field can be missed (see for example sketch in Figure 3 and Figure 12). Furthermore, for the method in test case DI, we need to know the entire data set for each integration step, whereas in case DA the integration can be performed for each domain independently.

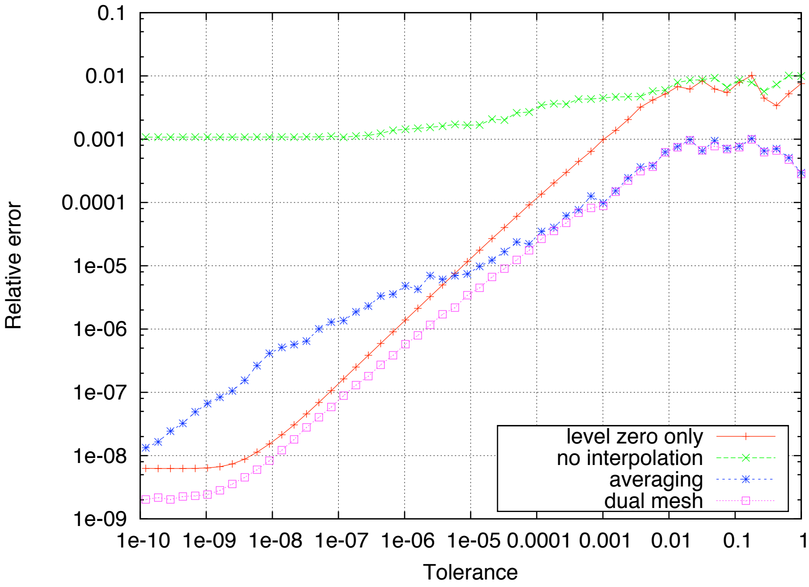


Figure 6 Comparison of relative error for test cases AI and DI.

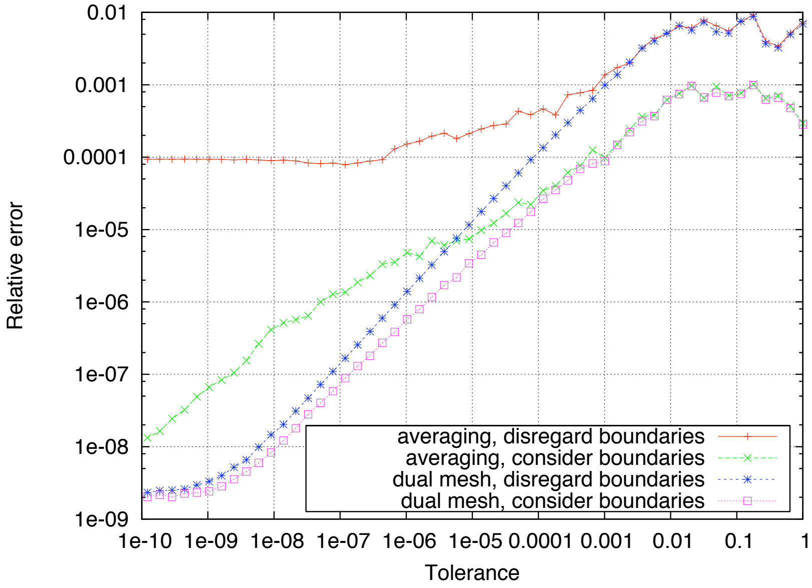


Figure 7 Comparison of relative error for the four test cases.

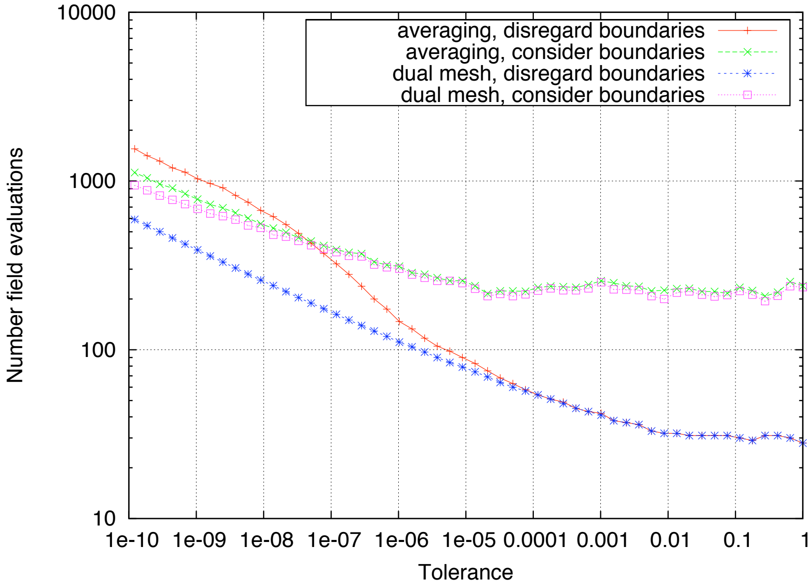


Figure 8 Number of evaluations for the four test cases.

5 Implementation

We have implemented our algorithm using VisIt [8], an open source visualization and analysis tool for large data sets. We chose VisIt as a basis for our work since it already supports a wide variety of AMR file formats. VisIt accommodates AMR as first class data type. It handles coarse cells in AMR data that are also available in a finer resolution level as a special case of "ghost data," i.e., data that is used to make computations more efficient, but which is not considered to be part of the simulation result. This marking is done by adding an array to each box/grid that designates the status of each cell (ghost or non-ghost).

VisIt's marking of refined AMR cells as ghost is facilitated by another data structure, the so called "domain nesting." This data structure specifies which grids/boxes are adjacent to each other and makes it possible to populate boundary ghost cells of one mesh with copied data from an adjacent box. Furthermore, this nesting information also specifies whether a box/grid contains any boxes of a finer level. For example, it is possible to query what boxes/grids of the next finer level intersect the current box/grid and what cells are covered. This information is given as a set of ranges.

VisIt also supports vector field visualization, for example with vector glyph plots and streamlines. However, the current implementation of streamlines in VisIt does not consider an AMR hierarchy. Instead, streamlines that are started in one grid/box of the domain only use information from that box/grid until they exit its *outer* boundaries. In particular, streamlines do not use ghost information to detect refined cells and appropriately descends into the AMR hierarchy.

Algorithm 5 AMR_Integrate_Curve

```
if no outer ghost zones then
    compute additional ghost layer
end if
while not finished do
    find domain in finest possible level
    build dual grid
    Integrate_Domain (see Algorithm 2)
end while
```

Based on the comparison results in section 4 we have implemented the method outlined in Algorithm 5 for the streamline visualization in our application cases. In order to descend into finer regions of the AMR grid, we utilize VisIt's data structures holding the nesting information, the ghost cell array, and the domain nesting described above. The ghost cell array specifies which cells of the coarse grid are refined. The domain nesting structure can be used to obtain the entire region of the finer domains (children domains) to construct the bounding boxes needed for the intersection calculations. If the simulation data do not provide additional ghost cells, we calculate them as follows: If an adjacent grid with equal refinement resolution exists, we use the values in the adjacent cells of the neighbor grid to generate the external ghost layer. If the domain is at the refinement level boundary we interpolate the ghost values using the coarser cell values. Note, that after the calculation of the dual grid we have to adapt the ghost cell array to the new grid, since the number of cells has changed.

6 Use Case – Application

In this section we will use our integration algorithm in order to visualize streamlines in two different application cases, the simulation of the solar wind interaction with the interstellar medium as well as a simulation of two incompressible viscous vortex cores merging.

6.1 The Solar Wind Interaction with the Local Interstellar Medium

Simulations of the solar system, particularly the interaction of solar wind with the local interstellar medium, are an important application area of AMR simulations. For example, scientists are particularly interested in plasma quantities in the solar system because they are currently measured by the Voyager 1 and Voyager 2 spacecraft. The global modeling of the heliosphere requires a computational region to be about 1000 AU, whereas length of plasma fluctuations can be 0.01AU, which are too fine to be modeled without AMR. The other application of the AMR is putting a very fine mesh at the spacecraft locations to compare the observations with the theoretical models.

The Voyager program was conducted to investigate distant planets of the solar system. The primary mission ended in 1989 and this was the beginning of the *Voyager Interstellar Mission*, which has the purpose to investigate outer solar system environment and search for the heliopause boundary. The magnetic field plays an important role in investigating our solar system. The solar flares produce strong magnetic field disturbances in the interplanetary magnetic field (IMF). The solar system is surrounded by local interstellar medium (LISM), which is weakly ionized plasma. The interstellar magnetic field (ISMF) has strong influence on heliosphere shape. The Voyager probes have a big set of instruments, however only magnetometers are still operational on both spacecraft. Therefore, modeling and visualization of the magnetic field are extremely important for this problem.

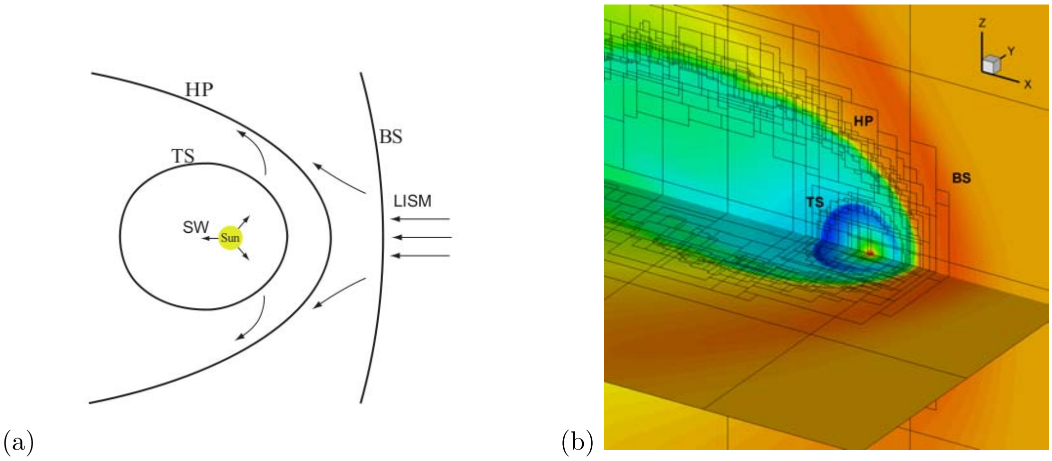


Figure 9 (a) The heliosphere. SW: Solar Wind, TS: Termination Shock, HP: Heliopause, BS: Bow Shock, LISM: Local Interstellar Medium. (b) The structure of the AMR mesh.

The heliosphere is the region of space filled by the expanding solar corona, a vast region indeed extending 150 – 180 astronomic units (AU) in the direction of the Sun's motion through the local interstellar medium (LISM) and several thousand AU in the opposite direction. The solar wind (SW) is a hydrogen plasma, expanding more-or-less radially from the Sun at speeds of 400 – 800 km/s, with a density that decays as r^{-2} (r denoting radial heliocentric distance from the Sun). The interstellar plasma, on the contrary, is rather weakly ionized. The Sun moves through the LISM with the velocity of about $25 - 26$ km/s. Collision of the SW and LISM flows creates a tangential discontinuity (the heliopause (HP), see Figure 9a) that separates these flows, a termination shock (TS), which decelerates supersonic SW at the inner side of the HP creating a so-called inner heliosheath, and possibly a LISM bow shock (BS).

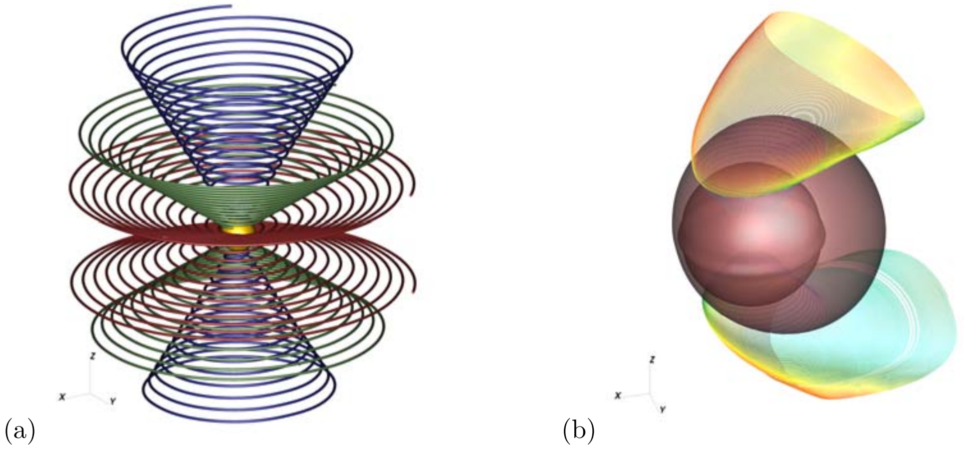


Figure 10 Magnetic field of the Sun. (a) Parker spiral shaped field lines. (b) Field lines passed the termination shock, colored by the field magnitude.

The AMR mesh used for the solar system simulation consists of five refinement levels and 20040 domains. The root grid on the coarsest level contains $100 \times 100 \times 80$ cells. In Figure 9b the structure of the AMR mesh around the heliopause is illustrated. Figure 10 shows the interplanetary magnetic field, which has the shape of the Parker spiral [30]. This distribution of the IMF is due to the expansion of the Sun's dipole magnetic field with SW velocity. As a result of the SW compression at the TS, the distance between two consecutive coils of the spiral becomes 3 times smaller in the inner heliosheath than in the supersonic SW. The streamlines behind the termination shock are shown in Figure 10b.

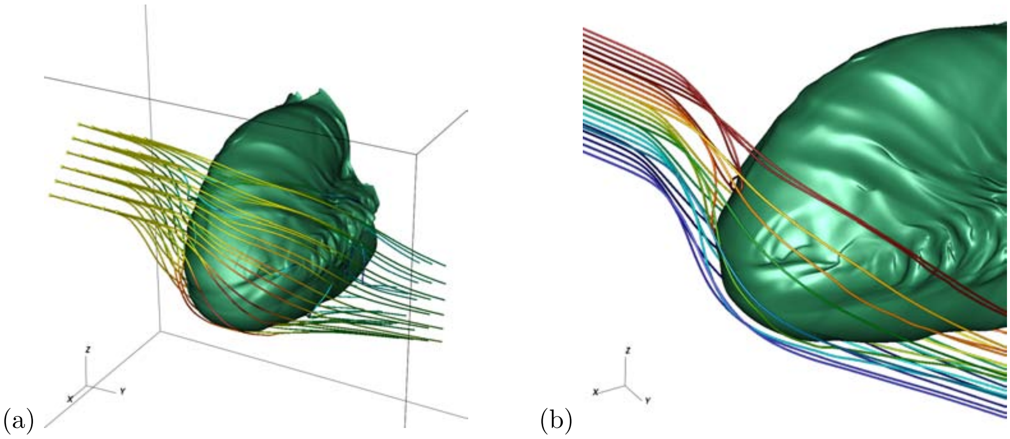


Figure 11 Interstellar magnetic field lines draping the heliopause. (a) Colored by the field magnitude. (b) Single color for each streamline).

Figure 11 shows interstellar magnetic field stream lines wrapping the heliopause, which is shown as the isosurface. The magnetic field squeezes the heliopause in the direction parallel to direction of the interstellar magnetic field at infinity. The wavy structure on the flanks of the heliopause is the combination of the Kevin-Helmholtz and Raleigh-Taylor instabilities (see [5]), which arises from the charge exchange between neutral hydrogen and plasma. The magnetic field acts as the stabilizing force, suppressing the instability, however, if magnetic field becomes perpendicular to the velocity of plasma, this stabilizing force becomes very small. This is the reason why the instability happens in the specific parts of the heliopause.

6.2 Simulation of Two Incompressible Viscous Vortex Cores

A second example comes from a 3D simulation of two incompressible viscous vortex rings merging, taken from [25]. Two co-rotating vortex rings approach each other at an angle, merging as they meet into a single vortex ring. The domain is periodic in the $z-$direction, with no-shear solid wall boundaries in the $x-$ and $y-$directions. This problem has been studied extensively due to its interesting vortex dynamics, both numerically and experimentally [19]. As the vortices merge, a complicated flow field is generated, providing a good testbed for the streamline algorithm. Streamlines provide a very useful visualization tool for interpreting the complex flow patterns produced as the flow evolves. Figure 12 clearly shows that the structure of the flow can not be captured without considering the AMR refined levels.

Figure 13 shows streamlines for the AMR data at the initial time (when there are two distinct vortex rings), after 60 timesteps (when the vortex rings are merging), and after 120 timesteps (after the two vortex rings have merged into a single flow structure). There are two levels of refinement in this simulation, each representing a factor of 4 refinement of the solution. The root grid on the coarsest level consists of 64^3 cells. In each figure, one can see tightly-wound streamlines around the vortex ring cores, along with streamlines farther from the vortex ring cores, which illustrate the far-field flowfield induced by the vortex rings.

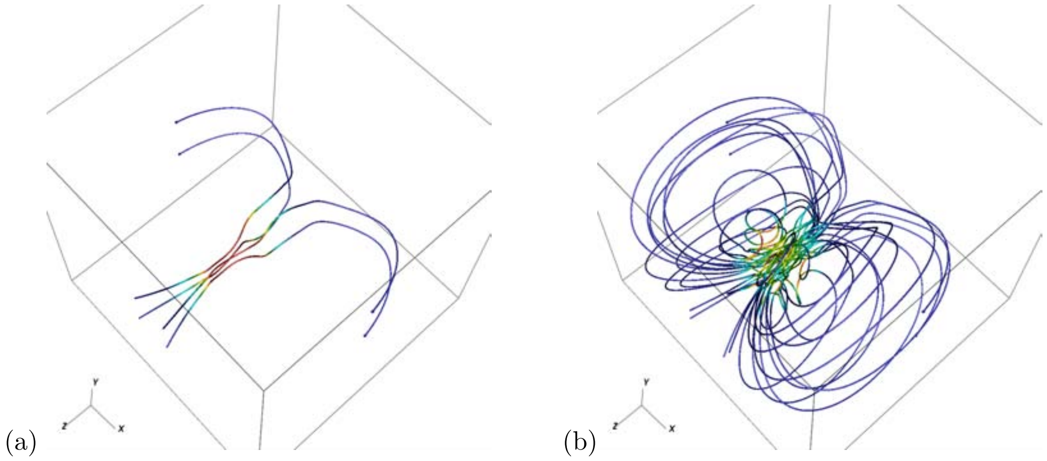


Figure 12 Streamlines in the vortex core data set (a) not considering the refinement levels, (b) using our algorithm. The color represents the velocity field magnitude.

7 Conclusions and Future Work

In our paper, we have examined the problem of the computation of integral curves in AMR simulations. We proposed a method for numerical integration which utilizes the data information provided at finest refinement level available and also handles interpolation issues

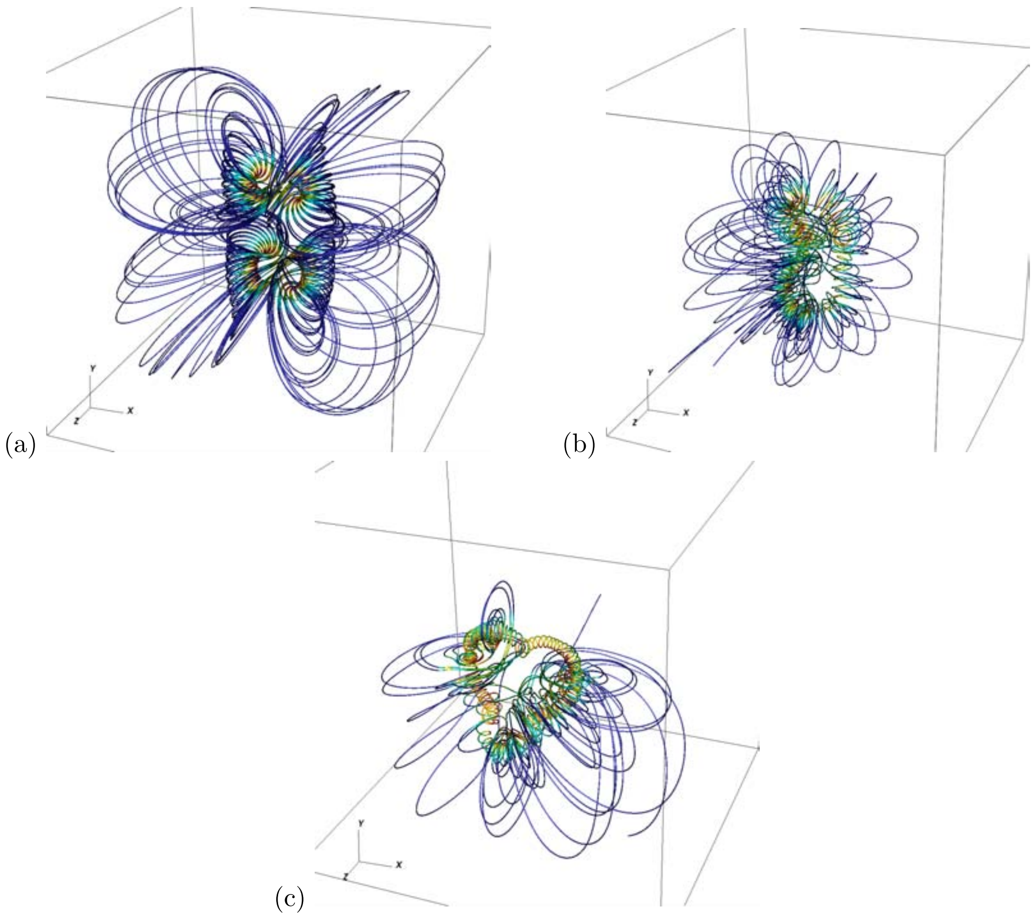


Figure 13 Streamlines visualization of the vortex core simulation for the time steps 0 (a), 60 (b), 120 (c), colored by field magnitude.

occurring at level boundaries. We applied our algorithm to visualization of streamlines in a model of the solar magnetic field as well as to a simulation of two vortex cores merging. Currently, our method is limited to block-structured rectilinear AMR grids. AMR methods have been expanded to simulate flow fields in complex geometries based on Cartesian grids with embedded boundaries or mapped grids. In future work we want to adapt our algorithm to handle these cases as well, which will make it possible to consider several additional interesting application areas, such as fusion. We further plan to implement a parallel version of our algorithm.

Acknowledgments

This work was supported in part by the Director, Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract Nos. DE-FC02-06ER25780 (University of California, Davis) and DE-AC02-05CH11231 (Lawrence Berkeley National Laboratory) through the Scientific Discovery through Advanced Computing (SciDAC) program's Visualization and Analytics Center for Enabling Technologies (VACET). We thank our colleagues from VACET, the members of the LBNL Applied Numerical Algorithms Group, and the VisIt development team.

References

1 A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. Welcome. A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *Journal of Computational Physics*, 142(1):1–46, May 1998.

2 A. Benkenida, J. Bohbot, and J. C. Jouhaud. Patched grid and adaptive mesh refinement strategies for the calculation of the transport of vortices. *International journal for numerical methods in fluids*, 40(7):855–873, 2002.

3 M. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53:484–512, March 1984.

4 Marsha Berger and Phillip Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64–84, May 1989. Lawrence Livermore National Laboratory, Technical Report No. UCRL-97196.

5 S.N. Borovikov, N.V. Pogorelov, G.P. Zank, and I.A. Kryukov. Consequences of the heliopause instability caused by charge exchange. *Astrophysics Journal*, 682:1404–1415, 2008.

6 Greg L. Bryan. Fluids in the universe: Adaptive mesh refinement in cosmology. *Computing in Science and Engineering*, 1(2), 1999.

7 M. Calvo, J.I. Montijano, and L. Randez. On the solution of discontinuous IVP's by adaptive Runge-Kutta codes. *Numerical Algorithms*, 1(33):163–182, 2003.

8 Hank Childs and Mark Miller. Beyond meat grinders: An analysis framework addressing the scale and complexity of large data sets. In *SpringSim High Performance Computing Symposium (HPC 2006)*, pages 181–186, 2006.

9 P. Colella and L. F. Henderson. The von Neumann paradox for the diffraction of weak shock waves. *Journal of Fluid Mechanics*, 213:71–94, 1990.

10 D. C. Fang, Weber, G. H., H.R. Childs, E.S. Brugger, B. Hamann, and K.I Joy. Extracting geometrically continuous isosurfaces from adaptive mesh refinement data. In *Proceedings of 2004 Hawaii International Conference on Computer Sciences (DVD-ROM conference proceedings)*, pages 216–224, 2004.

11 Christoph Garth, Han Krishnan, Xavier Tricoche, Tom Tricoche, and Kenneth I. Joy. Generation of accurate integral surfaces in time-dependent vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1404–1411, 2008.

12 L.F. Henderson, P. Colella, and E.G. Puckett. On the refraction of shock waves at a slow-fast gas interface. *Journal of Fluid Mechanics*, 224:1–28, 1991.

13 L. H. Howell, R. B. Pember, P. Colella, J. P. Jessee, and W. A. Fiveland. A conservative adaptive-mesh algorithm for unsteady, combined-mode heat transfer using the discrete ordinates method. *Numerical Heat Transfer, Part B: Fundamentals*, 35:407–430, 1999.

14 J. P. Jessee, W. A. Fiveland, L. H. Howell, P. Colella, and R. B. Pember. An adaptive mesh refinement algorithm for the radiative transport equation. *Journal of Computational Physics*, 139(2):380–398, January 1998.

15 Ralf Kähler, Donna Cox, Robert Patterson, Stuart Levy, Hans-Christian Hege, and Tom Abel. Rendering the first star in the universe – a case study. In *IEEE Visualization 2002*, pages 537–540. IEEE Computer Society, 2002.

16 Ralf Kähler and Hans-Christian Hege. Texture-based volume rendering of adaptive mesh refinement data. *The Visual Computer*, 18(8):481–492, 2002. Zuse Institut Technical Report ZR-01-30.

17 Ralf Kähler, Steffen Prohaska, Andrei Hutanu, and Hans-Christian Hege. Visualization of time-dependent remote adaptive mesh refinement data. In *IEEE Visualization 2005*, pages 175–182. IEEE Computer Society, 2005.

18 Ralf Kähler, John Wise, Tom Abel, and Hans-Christian Hege. GPU-assisted raycasting for cosmological adaptive mesh refinement simulations. In *Proceedings of Volume Graphics*. Eurographics Association, 2006.

19 Egon Krause. On vortex loops and filaments: three examples of numerical predictions of flows containing vortices. *Naturwissenschaften*, 1(90):4–26, 2003.

20 Oliver Kreylos, Gunther H. Weber, E. Wes Bethel, John M. Shalf, Bernd Hamann, and Kenneth I. Joy. Remote interactive direct volume rendering of AMR data. Technical Report LBNL 49954, Lawrence Berkeley National Laboratory, 2002.

21 R.S Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F.H. Post, and D. Weiskopf. The state of the art in flow visualization, part 1: Direct, texture-based, and geometric techniques. *Computer Graphics Forum*, 23(2):203–221, 2004.

22 Z. Li. An adaptive two-dimensional mesh refinement method for the problems in fluid engineering. *Lecture Notes in Computer Science*, 3314:118–123, 2004.

23 Kwan-Liu Ma. Parallel rendering of 3D AMR data on the SGI/Cray T3E. In *Proceedings of Frontiers '99 the Seventh Symposium on the Frontiers of Massively Parallel Computation*, pages 138–145. IEEE Computer Society, 1999.

24 R. Mannshardt. One-step methods of any order for ordinary differential equations with discontinuous right-hand sides. *Numerische Mathematik*, 1(31):131–152, 1978.

25 D. Martin, P. Colella, and D. T. Graves. A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions. *Journal of Computational Physics*, 227:1863–1886, 2008.

26 Nelson L. Max. Sorting for polyhedron compositing. In *Focus on Scientific Visualization*, pages 259–268. Springer-Verlag, 1993.

27 T. May, S. Schneider, M. Schmidt, and V. Luckas. Fast scalar and vector field visualization using a new progressive grid class. In *High Performance Computing Conference (HPC)*, pages 115–120, 2003.

28 Michael L. Norman, John M. Shalf, Stuart Levy, and Greg Daues. Diving deep: Data management and visualization strategies for adaptive mesh refinement simulations. *Computing in Science and Engineering*, 1(4):36–47, 1999.

29 Sanghun Park, Chandrajit Bajaj, and Vinay Siddavanahalli. Case study: Interactive rendering of adaptive mesh refinement data. In *Proceedings of the IEEE Visualization 2002*, pages 521–524. IEEE Computer Society, 2002.

30 E.N. Parker. The stellar wind regions. *Astrophysics Journal*, 134:20–27, 1961.

31 R. B. Pember, L. H. Howell, J. B. Bell, P. Colella, W. Y. Crutchfield, W. A. Fiveland, and J. P. Jessee. An adaptive projection method for unsteady, low mach number combustion. *Combustion Science and Technology*, 140:123–168, 1998.

32 P. J. Prince and J. R. Dormand. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980.

33 P. J. Prince and J. R. Dormand. High order embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 7(1), 1981.

34 D. Stalling. *Fast Texture-Based Algorithms for Vector Field Visualization.* PhD thesis, Freie Universität Berlin, 1998.

35 M. C. Thompson and J. H. Ferziger. An adaptive multigrid technique for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 82(1):94–121, May 1989.

36 X. Tricoche and C. Garth. Topological methods for visualizing vortical flows. In *Mathematical Foundations of Visualization, Computer Graphics, and Massive Data Exploration.* 2006 (to appear).

37 Gunther H. Weber, Hans Hagen, Bernd Hamann, Kenneth I. Joy, Terry J. Ligocki, Kwan-Liu Ma, and John M. Shalf. Visualization of adaptive mesh refinement data. In *Proceedings of the SPIE (Visual Data Exploration and Analysis VIII)*, volume 4302, pages 121–132, 2001.

38 Gunther H. Weber, Oliver Kreylos, Terry J. Ligocki, John M. Shalf, Hans Hagen, Bernd Hamann, and Kenneth I. Joy. Extraction of crack-free isosurfaces from adaptive mesh refinement data. In *Proceedings of the Joint EUROGRAPHICS and IEEE TCVG Symposium on Visualization, Ascona, Switzerland, May 28–31, 2001*, pages 25–34, 335, 2001.

39 Gunther H. Weber, Oliver Kreylos, Terry J. Ligocki, John M. Shalf, Hans Hagen, Bernd Hamann, and Kenneth I. Joy. Extraction of crack-free isosurfaces from adaptive mesh refinement data. In *Hierarchical and Geometrical Methods in Scientific Visualization*, pages 19–40. Springer Verlag, 2003.

40 Gunther H. Weber, Oliver Kreylos, Terry J. Ligocki, John M. Shalf, Hans Hagen, Bernd Hamann, Kenneth I. Joy, and Kwan-Liu Ma. High-quality volume rendering of adaptive mesh refinement data. In *Vision, Modeling, and Visualization 2001*, pages 121–128, 522. Akademische Verlagsgesellschaft Aka GmbH and IOS Press BV, 2001.

41 Gunther H. Weber, Martin Öhler, Oliver Kreylos, John M. Shalf, E. Wes Bethel, Bernd Hamann, and Gerik Scheuermann. Parallel cell projection rendering of adaptive mesh refinement data. In *Proceedings of the IEEE Symposium on Parallel and Large-Data Visualization and Graphics 2003*, pages 51–60. IEEE Computer Society, 2003.

Disclaimer

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Integrating Semantics into the Visualization Process*

Sebastián Escarza[1], Martín L. Larrea[1], Dana K. Urribarri[1], Silvia M. Castro[1], and Sergio R. Martig[1]

1 **Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica (VyGLab)**
Departamento de Ciencias e Ingeniería de la Computación (DCIC)
Universidad Nacional del Sur (UNS)
Bahía Blanca, Buenos Aires, Argentina
`{se,mll,dku,smc,srm}@cs.uns.edu.ar`

Abstract

Most of today's visualization systems give the user considerable control over the visualization process. Many parameters might be changed until the obtention of a satisfactory visualization. The visualization process is a very complex exploration activity and, even for skilled users, it can be difficult to arrive at an effective visualization. We propose the construction of a visualization prototype to assist users and designers throughout the stages of the visualization process, and the integration of such process with a reasoning procedure that allows the configuration of the visualization, based on the entailed conclusions. We are working on a formal representation of the Visualization field. We aim to establish a common visualization vocabulary, include the underlying semantics, and enable the definition of visualization specifications that can be executed by a visualization engine with ontological support. An ontological description of a visualization should be enough to specify the visualization and, thus, to generate a runtime environment that is able to bring that visualization to life. The visualization ontology defines the vocabulary. With the addition of inference rules to the system, we can derive conclusions about visualization properties that allow to enhance the visualization, and guide the user throughout the entire process toward an effective result.

1998 ACM Subject Classification I.3, I.3.6, I.2.4

Keywords and phrases semantic driven visualization, ontology, visualization, knowledge representation

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.92

1 Introduction

Visualization tools have strong dependency on the visualized data domain. This leads to a great heterogeneity in the field. If we compare, for example, a flow visualization with a graph visualization, we find differences between them in many respects. Data topologies in flow visualization are usually highly structured regular grids whereas in graph visualization are graphs. Data items in flow visualization have spatial locations that should be represented in the visual representation, but in graph visualization the layout of elements is a freedom

* This work was supported in part by the PGI 24/N020 and 24/ZN19, Secretaría General de Ciencia y Tecnología, Universidad Nacional del Sur, Bahía Blanca, Argentina.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 92–102
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

degree whereas the node-link structure is the key aspect to represent. As a result of such heterogeneity, different approaches need to be applied depending on the case.

However, there are many aspects in common across the visualization techniques. Data derivation, cleaning and filtering, and layout algorithms, are a few examples of tasks that are performed in similar ways in almost every visualization tool. A key issue to unify the Visualization area is the identification of such common aspects in order to perform developments that enable the reuse along the visualizations. Many efforts exist to gather these aspects, such as [25], [4], [26], [7], [3] and our Unified Visualization Model (UVM) [17].

The UVM is a reference model that gives users and designers a unique mental model in terms of which express their needs. It defines a theoretical framework for describing the intermediate states and transformations of the data, from its raw origin in the application domain to the final view construction. Additionally, the UVM enables the explicit definition of user interactions through the definition of tasks, basic interactions, and low level operators and operands.

The UVM and the other proposals are steps toward the foundation of a Visualization Base Theory and an extensive work has been done to unify the vocabulary in the field and overcome the heterogeneity. However, achieving an effective visualization is still more an art than a science. Although, with a framework for describing each aspect, each technique, and each possible choice in Visualization, how to configure the visualization process to obtain an effective visualization is a very difficult task, even for skilled users. Visualization systems can give the user considerable control over the visualization process. This freedom makes it difficult to choose a proper configuration that allows to obtain the desired results.

To overcome this difficulty, we propose to integrate semantics into the visualization in order to guide the user in the selection of the visualization parameters. We found necessary to include semantics for describing the data, the visual representation, the interactions, and the visualization process itself, plus a reasoning mechanism to derive the features of an effective visualization from such semantics. In consequence, our proposal relies upon three major components: a description about how each visualization technique represents each aspect of the data (i.e. a visualization ontology), a specification about the meaning of the raw data (i.e. a visualization-oriented data taxonomy), and a mechanism that performs the selection of the most suitable visual parameters to be applied to a concrete scenario (i.e. a rule system for semantic-based visualization).

The general objective of our proposal is the integration of the visualization process with a reasoning procedure that allows the configuration of the visualization, based on the entailed conclusions. In particular, we are working to obtain an operational visualization prototype with semantic support that allows to assist users and designers throughout all the stages of the visualization process. As part of our ongoing work, we are defining a visualization ontology based on the UVM, and formalizing a data taxonomy. Additionally, we are working on a preliminary set of rules that enable the entailment of conclusions that suggest to the user the selection of an appropriate configuration for an effective visualization.

This work is organized as follows. In Section 2, we present some related work on integrating semantics in visualization. Next, in Sections 3, 4, and 5, we describe the main components of our ongoing work. Finally, we give some closing remarks.

2 Related Work

The integration of semantics in visualization has evidenced an ever increasing interest in the community. Much work has been done to formalize the visualization topics and define visualization reference models. In [8], the authors argument the need for increasing the rigourousness in visualization descriptions to explicitly define a visualization ontology, and give some clues about how it can be achieved. In this sense, data and/or task oriented taxonomies such as [25], [4], [7], [3], and the UVM [17] are partial views of the semantic shared conceptualization for Visualization that must be explicitly modeled by every valid visualization ontology.

There are some good examples of how semantic information can be integrated into visualization tasks ([32], [15], [18], and [31]). However, in all these examples, the role of the semantics is to improve the integration, querying, and description of the data in the visualization; in none of these cases the semantics associated with the data is used to create the visualization.

Despite that, there are cases where semantics is used, although limited in some way, for aiding in the visualization creation. In [13], it is presented a semi-automatic visualization assistant that helps users to construct perceptually sound visualizations for large multidimensional datasets. In [10], it is proposed a framework that uses context information and a set of rules to automatically select a suitable visualization method. In [16], the authors present a method for aiding in the visualization pipeline design. A database of pipeline configurations is used to properly complete the configuration of the user's pipeline. In [9], the authors combine a domain ontology with a visual representation ontology to automatically select a proper visualization for web data.

Several efforts to define generalized visualization ontologies have also been made. A seminal paper on this matter is [2]. In that work, a top level ontology is outlined and future tendencies are given. Moreover, authors in [6] discuss the role of the information and the knowledge in visualization and identify possible trends. In [14] a graphics ontology representing the semantics for a 3D-scene graph is proposed to define the semantics of graphical primitives, using a very similar approach to ours.

Additionally, there are semantic specifications for particular visualization aspects such as the user domain data classification, the visual representation, and the visual mapping. Some examples are size-based data classifications [30, 29], a taxonomy for visualization algorithms that is based in assumptions over their inputs [28], the characterization of visual variables to represent visual representations at a higher level of abstraction [5], the use of fuzzy logic semantics to replace the traditional transfer function setup in illustrative volume rendering [21], and a specific semantic model created by a machine learning mechanism that uses representative dataset collections as training sets [24].

Finally, we can mention a rule-based related work. PRAVDA (Perceptual Rule-Based Architecture for Visualizing Data Accurately) [1] is a rule based architecture for assisting the user in making choices of visualization color parameters. This architecture provides appropriate choices for visualization, based on a set of underlying rules [23, 22], which are used to select a colormap.

3 Toward a Visualization Ontology

An ontology for visualization is the first necessary step to enable the building of visualization tools with underlying ontological support. An ontology is a formal, explicit specification of a shared conceptualization [12]. That shared conceptualization is given in visualization by the reference models discussed earlier, which provide the foundations and the necessary theoretical background. Our purpose is to formalize the theoretical framework given by the UVM and make it explicit and manipulable by a software platform automatically. To

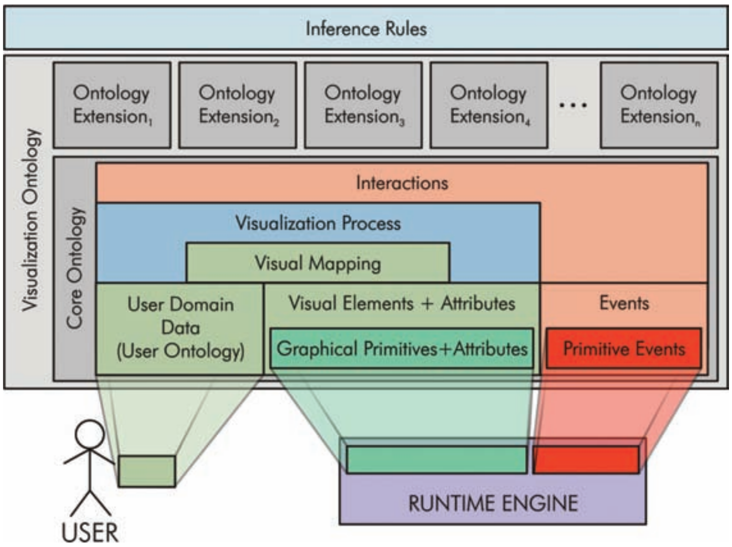


Figure 1 The ontology architecture. The figure shows the main components of the core ontology, the ontology extensions, and the inference rules. Users provide their own domain ontology and the selected runtime engine adds its primitives.

emphasize the shared nature of our ontology, we have adopted the Ontology Web Language (OWL) [20] as the language to define it. OWL is the W3C standard ontology language for the Semantic Web initiative and has been under active development for about six years. The utilization of a heavily supported standard facilitates the sharing and the interoperability of our proposal, and enables us to exploit the vast set of tools that have been implemented. In particular, we are interested in the definition of an OWL DL visualization ontology (i.e. the subset of OWL whose semantics is based in Description Logics), in order to keep the reasoning decidable.

The design of the ontology follows a modular approach (see Fig. 1). It has a core composed of a minimal set of basic concepts and relationships that can be extended by concepts and relationships of higher level of abstraction. The definitions in each ontology extension are based directly or transitively on the core definitions. Ontology extensions have a particular purpose and may be used depending on the user needs. In this way, only the extensions that model aspects of interest for the current application are used. This simplify the reasoning process, facilitates the decision making about the visualization configuration, and allows to focus on the outstanding topics of Visualization required for such application. This setting of core definitions plus their respective extensions leads to a layered and more extensible architecture.

From a static perspective, a visualization can be described in terms of the data in the user domain, the visual representation, and the visual mapping. From a dynamic perspective, the visualization can be perceived as a process of transformation that takes the user domain data as input, processes it to get the visual representation, and can be affected by the user interactions. The core of the ontology supports these both perspectives and it is presented in the following subsections (see Fig. 1).

3.1 User Domain Data Representation

Data in the user domain and its relevant properties are characterized by a domain ontology provided by the user. This ontology is imported and its concepts and relationships are made available to establish visual mappings from them.

Users are the experts in their domains. They know all the subtleties of the knowledge area that they want to visualize. Thus, it is reasonable that they provide their own representation for that area. Additionally, for a particular application domain, many domain ontologies may exist differing in how they represent such a domain, its purpose, level of detail, specificity, ontology commitment, etc. We prioritize the maximal flexibility in our design by allowing users the inclusion of their most adequate OWL domain ontology to represent their data and metadata. By requiring an OWL DL user ontology, we enable the use of all the OWL DL constructors, and ensure a successful integration between user and visualization ontologies.

3.2 The Visual Representation

The definition of the visual representation must describe the spatial substrate (how the elements are arranged in the visual representation space), the visual elements, and their attributes. Concepts for *space* and *geometric transformation* are used to describe the visualization layout.

In order to enable the automatic execution of the visualization represented by an ontological description, the visualization system must be able to perform the rendering of the elements represented by the involved concepts. But rendering engines may have different rendering capabilities. To overcome this issue without losing flexibility in our design, our system allows modular runtime engines for rendering and event handling. Each runtime engine provides concepts describing its supported graphical primitives (e.g., vertex based primitives, nurbs, etc) with its attributes (e.g., color, opacity, shader models, etc), the supported transformations (e.g., translations, rotations, etc), the user events that can be handled (e.g., mouse click, mouse drag, keystrokes, etc.), and every platform dependent aspect. These primitive concepts do not rely on any other concepts and they have a semantics given by how they are processed at runtime. Primitive concepts can also be combined to define derived concepts for describing more complex scenes (see Fig. 1).

3.3 The Visual Mapping and The Visualization Process

Concepts for the data in the user domain and the visual representation are combined to define the visual mapping. From a static viewpoint, the visual mapping consists mainly in ontological relationships that associates data items and attributes in the user domain with graphical elements and attributes in the visual representation. In many cases, the mapping follows a data structure or topology, or some data values are mapped specifically depending on their datatype.

However, the visual mapping can be the result of a complex data transformation. From a dynamic viewpoint, the visualization can be perceived as a process that takes the data in the user domain (i.e., the input data or raw data) and process it, giving back the view as a result. This processing is decomposed in a visualization network or pipeline where the intermediate data stages, the transformations, and the interconnections between them are described (see Fig. 2).

The visualization ontology defines concepts and relationships to describe the visualization stages and transformations. In each transformation, a formal description of the calculations performed there, and the conditions that must hold to carry out them, are provided.

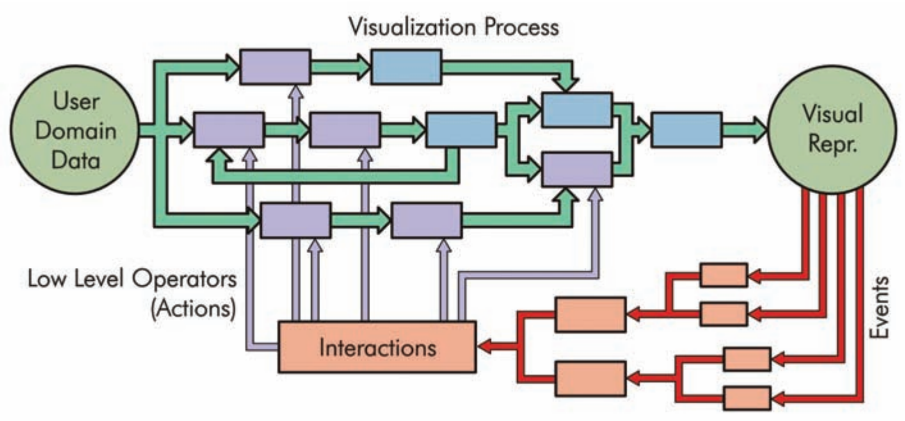


Figure 2 The visualization process and the interactions. The figure shows a visualization network that obtains a visual representation from the input data in the user domain, and the interaction feedback loop. A set of events are combined to characterize the basic interactions. These interactions affect the visualization network through low level operators provided in each stage.

These descriptions involve the specification of control structures such as the sequence, the conditionals, etc. Additionally, each stage provides low level operators that enables its reconfiguration. To represent these aspects the Process Ontology of OWL-S can be used. OWL-S, the Semantic Markup for Web Services [19], is an OWL ontology that describes semantic web services. The description of the behavior of such web services could be useful to describe the semantics of the stages and transformations in the visualization process.

The formal description of the visualization process enables its dynamic reconfiguration, the re-execution of the proper stages in response to interactions, and the ability to reason about how to connect the available stages and transformations to obtain the desired visual representation.

3.4 Interactions in Visualization

After the visual representation is generated, the user can interact with the visualization triggering background processes that recalculates and re-executes several parts of the visualization. Then, the user receives some feedback from the visualization and the interaction cycle is repeated again and again.

Each interaction in the visualization starts when the user produces an event on the visual representation. The events that the visualization can handle are associated to basic interactions. These interactions reconfigure the visualization process through low level operators defined in each stage of the visualization network (see Fig. 2).

Ontologically speaking, the basic interactions are mappings from events to low level operators. Events that can be handled by the runtime engine are provided as a set of concepts. Low level operators are part of the set of concepts used to define the visualization process. Hence, the basic interactions are represented in the ontology by relationships that map those two set of concepts. In this way, each time the user produces the set of events that triggers some interaction, the runtime engine will report them, and then, the semantic information about such interaction will be used to perform the appropriate reconfiguration of the visualization network represented by the ontology.

3.5 Extensions of the Visualization Ontology

In the previous subsections the main parts of the ontology core have been discussed. The core concepts and relationships offer the ability to model the key elements of the visualization, but they present a low abstraction level for the user data model and the underlying software platform. In visualization, many decisions take into account more abstract information. Concepts such as shape, transfer function, visualization technique, among others, are more appropriate to reason and decide the optimal way to represent some particular dataset.

Extensions to classify datasets, handle transfer functions, manipulate color spaces and transformations, and define information channels in visualization (such as shape, opacity, color, etc.) can be defined by combining core concepts and used independently only when they are required. For example, a user interested in transfer functions can use such extensions without importing others. In this way, there is no need to perform the reasoning over the full visualization ontology. Only the relevant concepts are considered to assist the user. Additionally, since extensions are defined in terms of core concepts, and such concepts have support in the runtime engine, the extensions automatically acquire runtime support contributing to the automatic execution of the visualization represented by the ontology.

4 Building Up a Visualization-Oriented Data Taxonomy

Data classification is the categorization of data for its most effective and efficient use. Data can be classified according to any criteria. For example, it can be divided according to its topical content, file type, operating platform, average file size, creation date, etc. We need a suitable data classification to integrate in the UVM and in our ontology. An explicit description of the data provides the semantics necessary to enable the reasoning and the selection of effective visualization techniques to visualize it.

A visualization-oriented data taxonomy should consider not only the size in bytes of the dataset but also issues like:

- the amount of items, to determine how important it is the visual scalability of the technique to be used,
- the existing relationships among items, to determine how important is to use a technique that displays explicitly such relationships (e.g. a graph-based technique),
- the amount of attributes, to determine how important is to use a multidimensional visualization technique,
- and finally, the amount of different objects to be represented and their complexity, to determine the necessary interactions to explore and analyze the data.

These aspects are key features to select the most adequate technique to visualize a dataset. A dataset taxonomy also helps in the selection of the graphical elements and their attributes. For example, the techniques to visualize high-dimensional textual data are very different from the techniques required to visualize a vector field, and the graphical elements to represent tuples are very different from those used to represent vectors. Figure 3 shows the UVM's transformations where the taxonomy-driven selections are performed.

In the context of our Visualization Ontology, we need to provide concepts and relationships to enable the formal definition of visualization-oriented data taxonomies. This implies the description of the metrics needed to characterize the datasets accordingly to the aspects previously mentioned. Based on such metric values, it is possible to define hierarchies for dataset classification. Then, for each category, a set of proper visualization techniques can be made available ensuring an adequate visual representation. All these concepts and

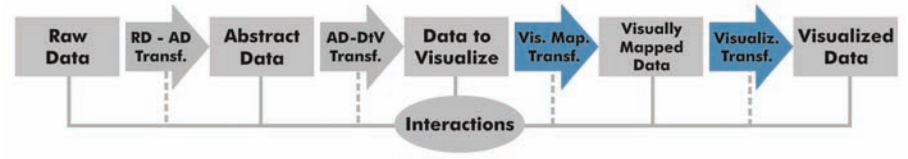


Figure 3 The role of a visualization-oriented data taxonomy in the context of the UVM's pipeline. The taxonomy is involved in the selection of how the elements and its attributes are represented (Visual Mapping Transformation), and it is essential to choose an appropriate visualization technique (Visualization Transformation).

relationships will be integrated into the visualization ontology through one or more extensions created for this particular purpose.

5 Fulfilling Semantics Through Rules: Semantic-Based Visualization

An ontology describes the meaning of the domain that it represents. But such ontology cannot be useful if that meaning is not exploited in all its extent. An OWL specification of concepts and relationships only describe the axioms of the knowledge base. It has information about what holds and what does not, and is able to be queried for ontology consistency, concept satisfiability, concept subsumption, and instance checking. These operations are valuable for validation purposes and provide richer "object oriented" axiomatic descriptions. However, to assist the user in his search for an effective visualization, it is necessary to include inference rules into the system. This addition will allow to derive conclusions about visualization properties that will be used to enhance the visualization and guide the user throughout the process.

Inference rules are able to represent the way in which the system can derive new information. They are combined with the axiomatic descriptions in the visualization ontology to entail new facts that can be queried and used to suggest how to visually represent some dataset. Rules also allow the use of variables in their specification giving the ability to infer through a pattern matching based mechanism. Such a mechanism, not present at ontological level, is essential to bind visualization parameters, derive new information, and increase the overall expressive power. Additionally, inference rules allow a richer information management by making reference to concepts of a higher abstraction level, i.e., concepts described by specific extensions of the ontology. For example, rules stating that some visualization technique is appropriate for some particular kind of dataset, are rules that make reference to the concepts for the visualization-oriented data taxonomy discussed before.

From combining the semantics given by the visualization ontology with a carefully chosen set of inference rules, the available family of queries become more according to our objective: assist the user to get an effective visualization. For example, one could ask the system if some technique is suitable for the input data, if some color scheme is appropriate to make evident the differences between some attributes, if the visual mapping shows appropriately the presence of some data correlation, etc.

In order to include rules into our proposal in an elegant and consistent manner, we are analyzing the use of a rule language that has a good integration with OWL. Unfortunately, the rule language for the Semantic Web is not standardized yet. The most encouraged proposal under consideration is the Semantic Web Rule Language (SWRL) [27]. This language adds rules written in RuleML (an XML-based language for the denotation of rules) to OWL DL. The result is a language based in horn clauses which has the expressive power of a First Order

Logic, but, in consequence, it is undecidable. Another proposal is the use of Description Logics Programs [11], a combination of OWL DL with the decidable portion of a horn-like rule language. This last proposal emphasizes decidability over expressiveness resulting in a more limited but decidable language.

Although the rule support for the Semantic Web is not mature yet, the tendency indicates that some kind of horn-like clauses will be used to enable full reasoning over the web. In this context, we expect that our proposal also make use of such a kind of rules.

6 Final Remarks

We have presented a proposal to integrate semantics into visualization to aid users to obtain an effective visualization. The semantic specification is based on the definitions given by a visualization reference model (the UVM), and contributes to the formalization of the visualization base theory. Such specification requires a visualization ontology, plus a set of inference rules to entail information to properly configure the visualization. The ontology was designed following a modular design that separates the core ontology describing basic concepts, from the higher abstraction level concepts defined in particular purpose extensions. Additionally, the ontology design distinguishes between primitive and derived concepts to provide runtime support for the visualization represented by the ontology.

Several aspects compose the semantic description for a visualization. Static aspects such as the data in the user application domain, the visual representation, and the visual mapping, are combined with concepts describing the stages and transformations of the visualization process and the interaction feedback provided by a visualization tool. All this information is gathered and used to establish valid conclusions about how to configure a visualization that represents adequately and accurately the input dataset.

In addition to the core visualization ontology, we have outlined an ontology extension to describe a dataset taxonomy, and have described the use of inference rules in the reasoning process. These two topics are essential to make possible, to our system, the selection of a suitable visualization technique as a function of the input dataset characteristics.

This work is the first of a long series of steps toward the construction of a visualization system that helps users represent their data in an effective manner. We are currently working in the design and implementation of a service-oriented visualization model that will extend the UVM through the inclusion of semantic information and reasoning. Currently, we are working on the formalization of the UVM and its ontological representation, the definition of the core ontology, the development of a service-oriented version of the UVM, the characterization of the input data and an inference model based on semantic reasoning. In the future, we expect to complete the ontology definition with specific extensions to handle data categorization, color management, and information channels among other topics. Also, we pursue the implementation of a modular prototype for a concrete evaluation of the exposed topics, and the validation of the proposed formalisms by the Visualization community.

References

1 L. D. Bergman, B. E. Rogowitz, and L. A. Treinish. A rule-based tool for assisting colormap selection. In *VIS '95: Proceedings of the 6th conference on Visualization '95*, page 118, Washington, DC, USA, 1995. IEEE Computer Society.

2 K. W. Brodlie, D. A. Duce, D. J. Duke, and al. Visualization Ontologies: Report of a Workshop held at the National e-Science Centre. Technical report, e-Science Institute, 2004.

3 Ken Brodlie and Nurul Mohd Noor. Visualization Notations, Models and Taxonomies. In Ik Soo Lim and David Duce, editors, *EG UK Theory and Practice of Computer Graphics (2007)*, pages 207–212, Bangor, United Kingdom, 2007. Eurographics Association.

4 S. K. Card and J. Mackinlay. The structure of the information visualization design space. In *INFOVIS '97: Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, page 92, Washington, DC, USA, 1997. IEEE Computer Society.

5 M. S. T. Carpendale. Considering Visual Variables as a Basis for Information Visualisation. Technical report, University of Calgary, Calgary, AB, 2003.

6 Min Chen, David Ebert, Hans Hagen, Robert S. Laramee, Robert van Liere, Kwan-Liu Ma, William Ribarsky, Gerik Scheuermann, and Deborah Silver. Data, information, and knowledge in visualization. *IEEE Comput. Graph. Appl.*, 29(1):12–19, 2009.

7 Ed H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'00)*, pages 69–75. IEEE Computer Society Press, 2000.

8 David J. Duke, Ken W. Brodlie, David A. Duce, and Ivan Herman. Do you see what i mean? *IEEE Computer Graphics and Applications*, 25(3):6–9, 2005.

9 O. Gilson, N. Silva, P.W. Grant, and M. Chen. From web data to visualization via ontology mapping. *Computer Graphics Forum*, 27:959–966(8), May 2008.

10 Maria Golemati, Constantin Halatsis, Costas Vassilakis, Akrivi Katifori, and University of Peloponnese. A context-based adaptive visualization environment. *Information Visualisation, International Conference on*, 0:62–67, 2006.

11 Benjamin N. Grosof, Raphael Volz, Ian Horrocks, and Stefan Decker. Description logic programs: Combining logic programs with description logic. In *WWW 03 Proceedings of the 12th international conference on World Wide Web*, pages 48–57. ACM, 2003.

12 Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, June 1993.

13 Christopher Healey, Robert St. Amant, and Jiae Chang. Assisted visualization of e-commerce auction agents. In *In Proceedings graphics interface 2001*, pages 201–208, 2001.

14 Evangelos Kalogerakis, Stavros Christodoulakis, and Nektarios Moumoutzis. Coupling ontologies with graphics content for knowledge driven visualization. In *VR '06: Proceedings of the IEEE conference on Virtual Reality*, pages 43–50, Washington, DC, USA, 2006. IEEE Computer Society.

15 V Kashyap, K-H Cheung, D Doherty, M Samwald, MS Marshall, J Luciano, S Stephens, I Herman, and R Hookway. An ontology-based approach for data integration - an application in biomedical research. In J Cardoso, M Hepp, and M Lytras, editors, *Real-world Applications of Semantic Web Technology and Ontologies*, Semantic Web and Beyond. Springer-Verlag, Heidelberg, 2008.

16 David Koop. Viscomplete: Automating suggestions for visualization pipelines. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1691–1698, 2008.

17 Sergio Martig, Silvia Castro, Pablo Fillottrani, and Elsa Estvez. Un modelo unificado de visualización. In *Proceedings of the IX Congreso Argentino de Ciencias de la Computación*, pages 881–892, 2003.

18 Trong Dung Nguyen, Tu Bao Ho, and DucDung Nguyen. Data and knowledge visualization in knowledge discovery process. In *VISUAL '02: Proceedings of the 5th International Conference on Recent Advances in Visual Information Systems*, pages 311–322, London, UK, 2002. Springer-Verlag.

19 OWL. OWL-S: Semantic Markup for Web Services, Last visited on September 2009. `http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/`.

20 OWL. OWL Web Ontology Language Overview, Last visited on September 2009. `http://www.w3.org/TR/owl-features/`.

21 Peter Rautek, Stefan Bruckner, and Eduard Groller. Semantic layers for illustrative volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1336–1343, 2007.

22 B. E. Rogowitz and L. A. Treinish. Using perceptual rules in interactive visualization. In B. E. Rogowitz and J. P. Allebach, editors, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2179 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 287–295, May 1994.

23 Bernice E. Rogowitz and Lloyd A. Treinish. Data structures and perceptual structures. In Jan P. Allebach and Bernice E. Rogowitz, editors, *Human Vision, Visual Processing, and Digital Display IV, Volume 1913, Number 1*, pages 600–612. SPIE, 1993.

24 Christof Rezk Salama, Maik Keller, and Peter Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1021–1028, 2006.

25 Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. *IEEE Symposium on Visual Languages*, 0:336, 1996.

26 Chris Stolte and Pat Hanrahan. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. In *INFOVIS '00: Proceedings of the IEEE Symposium on Information Vizualization 2000*, page 5, Washington, DC, USA, 2000. IEEE Computer Society.

27 SWRL. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, Last visited on September 2009. `http://www.w3.org/Submission/SWRL/`.

28 M. Tory and T. Moller. Rethinking visualization: A high-level taxonomy. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 151–158, 2004.

29 Antony Unwin, Martin Theus, and Heike Hofmann. *Graphics of Large Datasets: Visualizing a Million (Statistics and Computing).* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

30 E. Wegman. Huge data sets and the frontiers of computational feasibility. *Journal of Computational and Graphical Statistics*, 4(4):281–295, 1995.

31 Z. M. Weng and D. Bell. Integrating visual ontologies and wavelets for image content retrieval. In *DEXA '98: Proceedings of the 9th International Workshop on Database and Expert Systems Applications*, page 379, Washington, DC, USA, 1998. IEEE Computer Society.

32 Zhao Xu, Huajun Chen, and Zhaohui Wu. Applying semantic web technologies for geodata integration and visualization. In Jacky Akoka, Stephen W. Liddle, Il-Yeol Song, Michela Bertolotto, Isabelle Comyn-Wattiau, Samira Si-Said Cherfi, Willem-Jan van den Heuvel, Bernhard Thalheim, Manuel Kolp, Paolo Bresciani, Juan Trujillo, Christian Kop, and Heinrich C. Mayr, editors, *ER (Workshops)*, volume 3770 of *Lecture Notes in Computer Science*, pages 320–329. Springer, 2005.

Simulation and Visualization of Medical Application to the Inner Ear of the Guinea Pig to Reduce Animal Experiments

Martin Hering-Bertram[1,2], Norbert Siedow[1], Oliver Tse[1], Stefan K. Plontke[3], Ruth Gill[4], and Alec N. Salt[4]

1 **Fraunhofer ITWM, Kaiserslautern, Germany**
2 **Hochschule Bremen University of Applied Sciences, Germany**
3 **Tübingen Hearing Research Center (THRC), Germany**
4 **Washington University School of Medicine, St Louis, MO, U.S.A.**

Abstract

We present a novel approach to simulate drug application to the inner ear of the guinea pig with the goal to reduce animal experiments and to increase the accuracy of measurements. The framework is based on a tetrahedral grid representing the individual compartments of the cochlea, associated with a finite element model used to simulate medical diffusion and clearance. In a first simulation scenario, we were able to compute transfer coefficients between the inner compartments of the ear, validating experiments from the literature, and to prove the existence of clearance at the inner scala tympani. In a second scenario, the cochlea was unwound to obtain a one-dimensional model for efficient simulation-based transfer coefficient identification. These coefficients are useful to predict the impact of novel medication application systems.

1998 ACM Subject Classification J. Computer Applications, J.3 Life and medical sciences, I. Computing Methodologies, I.6 Simulation and Modeling

Keywords and phrases Cochlea, Perilymph, Spiral Ligament, Round Window Application, Modiolar Communication Routes, Numerical Simulation, Parameter Identification

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.103

1 Introduction

Hearing loss is still one of the most frequent chronic diseases and it may affect people of any age. Irreversible impairment of the sensory hair cells inside the human cochlea (inner ear), is often the reason for acute hearing loss. Due to the high sensibility of these hair cells, there is a wide range of possible causes for hearing loss, ranging from certain antibiotic treatments to simple aging processes. A drug therapy against hearing loss may only be successful when the right dosage is applied.

A promising alternative to systemic therapy, where medication is absorbed by the entire body, is the local delivery of drugs to the inner ear [7]. This is typically performed by injecting a small quantity of substance through the ear drum on top of the round window (purple region illustrated in Figure 3(a)), from where it will enter the scala tympani and will eventually reach the hair cells in the region where the hearing loss occurs, since hair cells at the front and at the end of the cochlea are associated with different frequency ranges. State of the art is round window application, i.e. placing a gel containing the drug on the round window. Alternatives to that may also help placing the medication closer to the target region, in the near future.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 103–117
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

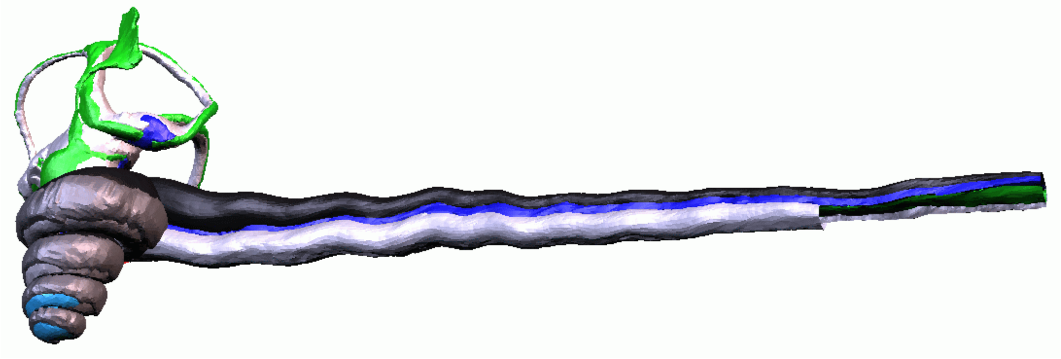


Figure 1 Unwinding the cochlea of a guinea pig, facilitating simulation and visualization of medical drug application to the inner ear.

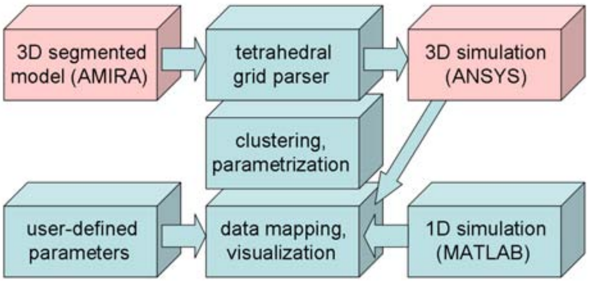


Figure 2 Simulation and modeling framework. The modules colored in blue belong to our own software, whereas red boxes denote commercial tools we used.

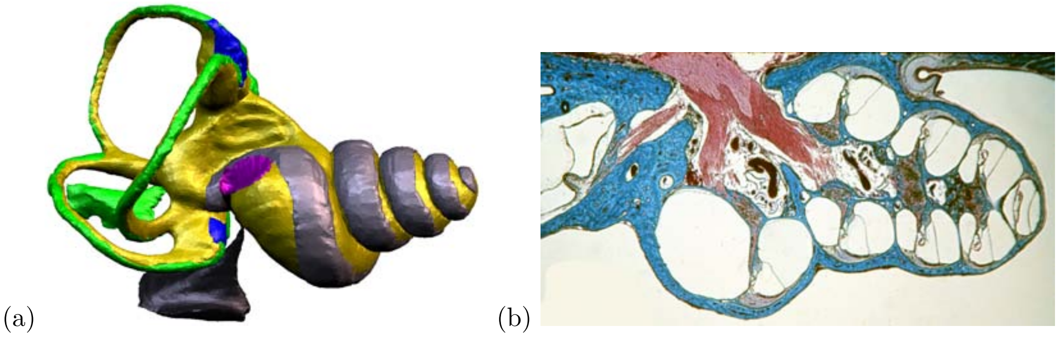


Figure 3 The cochlea (inner ear) of the guinea pig. (a) Segmented geometric model used for FEM-based simulation; (b) histological cross-section image of the cochlea, courtesy of Renate Lauf, Tübingen Hearing Research Center (THRC).

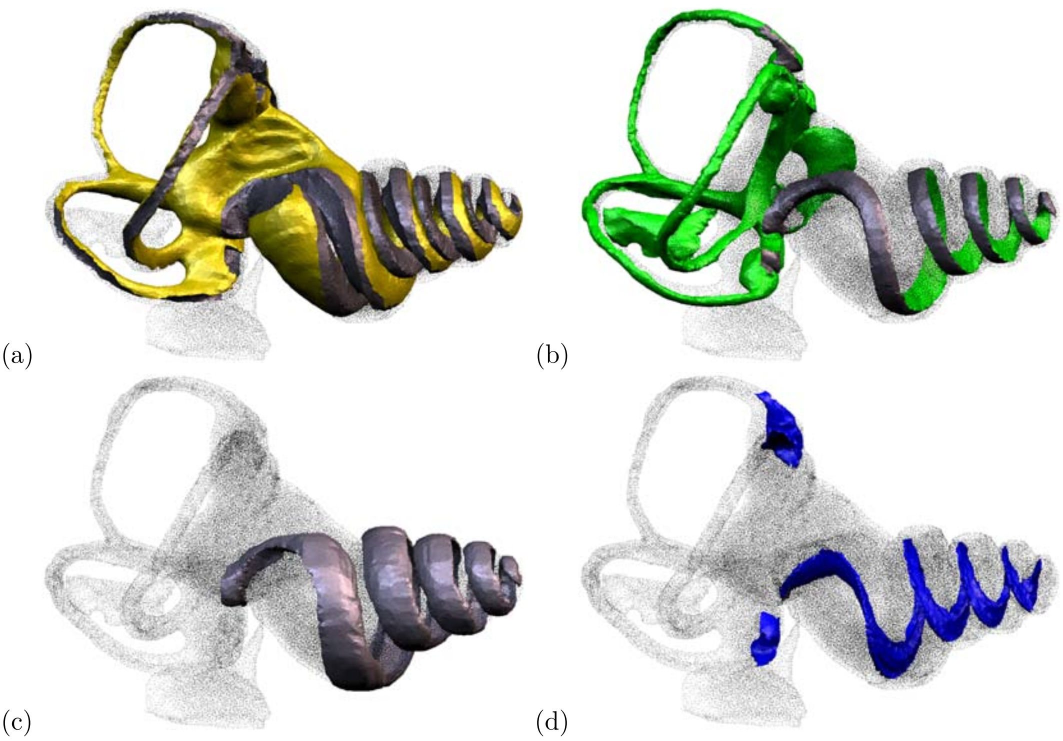


Figure 4 Segmented volume compartments: (a) scalae tympani and vestibuli; (b) endolymphic space; (c) spiral ligament; (d) organ corti.

Care must be taken not to cause further damage to the inner ear when locally applying medication. An over dosage of applied medication may increase hearing loss, as well as inner tissues may suffer from surgery. With under dosage, incorrect application frequency, or faulty medication formulation (e.g. too large molecules to penetrate round window tissue), no significant quantity will enter the cochlea. Due to these risks, at the current state approval of new application systems requires a large number of animal experiments.

The cochlea of the guinea pig (Figure 1) is shaped in a similar way as the human cochlea, except that it is much smaller and variations in size and shape are rather limited [3, 13]. Animal experiments required by law are used to assess the effect of drug application. The results are scaled up to the human ear without understanding the physical fundamentals of this process. Measurements of drug concentration are very difficult to perform due to the small size of a guinea pig's cochlea. Substances that cannot be measured with micro CT or magnetic resonance microscopy [11] may require destruction of the cochlea when measured. Unfortunately, the visible marker substances may have different transfer coefficients than the medication to be investigated.

In our present work, we contribute a simulation and modeling framework with the goal to reduce and replace animal experiments. Besides the ethical advantage of computer-simulated approaches, there are virtually no limitations in measuring results and in the accuracy for transforming the underlying physics into mathematical models. The only loss of precision is due to (small) numerical simulation errors and due to differences between the virtual model and reality, both in geometry and the physical phenomena developed here. Our algorithm simulates and visualizes the application process for the cochlea of a guinea pig with much greater accuracy than can be obtained with any state of the art animal experiment, provided

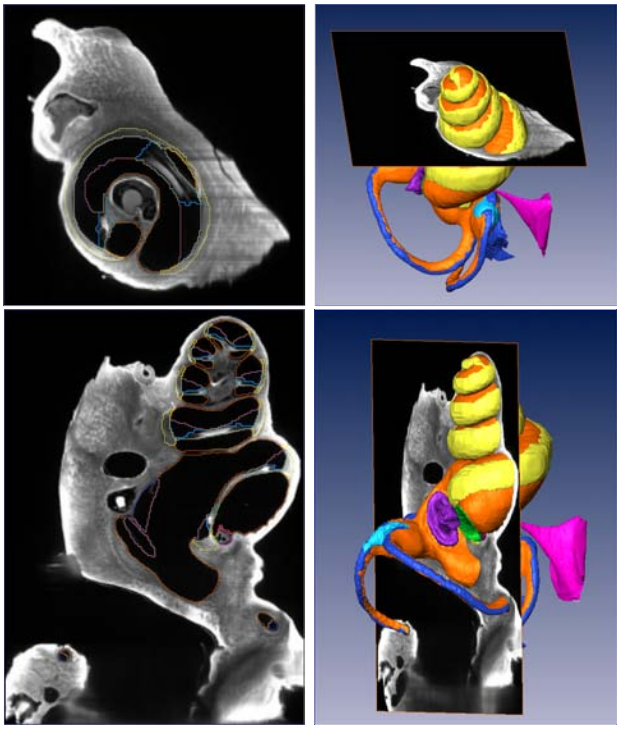


Figure 5 Volume modeling by cross-sectional segmentation with AMIRA.

that transfer and diffusion coefficients are known. We anticipate that similar methods will be available in the near future to simulate analogous processes in the human ear.

A schematic view of our framework is depicted in Figure 2. We used the AMIRA software to construct a segmented geometric model of the cochlea compartments. The resulting tetrahedral mesh is parsed by our system and exported for simulation with ANSYS. After the model and a number of ANSYS simulation runs used for validation have been established, our system forms a stand-alone application, as we obtain a parametrized 1D-representation by unwinding the cochlea and the simulation model. The 1D-version of the simulation is now implemented in MATLAB, whereas C++ with OpenGL was used for the remaining modules. Based on user-defined parameters, 3D ANSYS and 1D MATLAB simulations are mapped to the geometric model for visualization purposes.

The remainder of our paper is structured as follows: The next section summarizes previous and related work. Section 3 contains the parameter identification process determining transfer coefficients and validating them based on results from previous work. Section 4 is devoted to the model reduction process, unwrapping the cochlea geometry into a one-dimensional model. Segmentation and visualization of the geometry facilitates the construction of a parametrized 1D simulation approach that accurately matches the results of the full 3D finite element simulation. Section 5 presents a concluding discussion of the results obtained.

2 Previous and Related Work

An initial study of the cochlea geometry for the guinea pig has been described by Fernandez [3]. A more recent work by Shepherd and Colreavy [13] presents geometric examples for the micro structure of the main compartments of the human cochlea (inner ear), the scala tympani and the scala vestibuli (cavities in Figure 3(b)). Approaches directed at surgery planning propose segmentation of computer tomography data [2] and direct volume rendering [5]. Modeling of the human middle ear has been described for dynamic and acoustic simulation [4, 1].

The compartments of the inner ear are mostly filled with a liquid, such that the distribution of medical substances is mostly due to **diffusion** within homogeneous liquid and **transfer** across boundary tissues. Further effects are **clearance** (advection) due to blood flow and bio-chemical **degradation**, which also reduces the concentration on a longer time scale.

Recent work is concerned with localizing the most important transfer (communication) routes between individual scalae [7, 9]. Knowing these transfer routes, (and corresponding transfer coefficients) it is possible to locate and quantify the clearance due to blood flow. Compartments that are densely connected to blood flow, like the spiral ligament at the outer boundary (gray surface in Figure 3(a)) cause a reduction of medication substance. For controlling medication dosage, it is important to fully understand the interplay between diffusion, transfer, and clearance.

An early study based on animal experiments measuring the transfer between scalae tympani and vestibuli is described in [12]. In our present work, we used these results for validation and for parameter identification, i.e. specifying the transfer coefficients based on a 3D finite element model. A mathematical FEM-based diffusion approach has been described in [8], using an analytic snake-like shape as geometric model. For a more complete overview of the physical modeling, we refer to [6].

In the present work, an accurate 3D model of the cochlea is used for FEM-based simulation of medication concentration. Data sets for reconstructing the geometry of the guinea pig inner ear were obtained from computed tomography (CT) and orthogonal plane fluorescence optical sectioning (OPFOS; A. Voie, G. Saxon; Spencer Technologies). Fluid spaces of the inner ear were segmented using the AMIRA software system (MERCURY Computer Systems Inc.). The 3D construction of the geometric cochlea model was published in [10] and is illustrated in Figure 5.

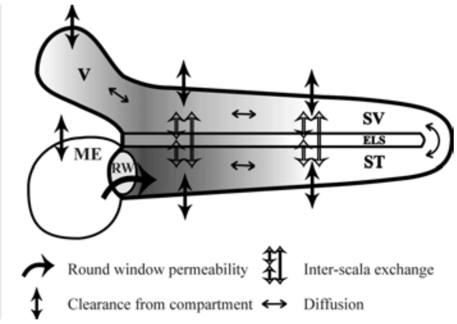


Figure 6 Schematic view of middle ear (ME), round window (RW), scala tympani (ST), scala vestibuli (SV), and vestibule (V). Both scalae are connected in the apex and also exchange material due to their proximity.

3 FEM-based Simulation and Parameter Identification

3.1 Simulation Model

The segmented geometric model constructed with AMIRA is composed of 117,773 vertices and 586,674 tetrahedra, each of which is marked with a label denoting its compartment.

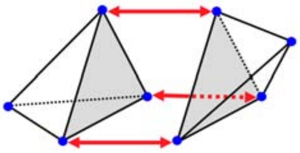


Figure 7 Duplicating inner compartment boundary vertices. Diffusion is based on linear ansatz functions associated with the ANSYS nodes (blue dots). Transfer across boundary surfaces is defined by ANSYS link elements (red arrows) between duplicated nodes.

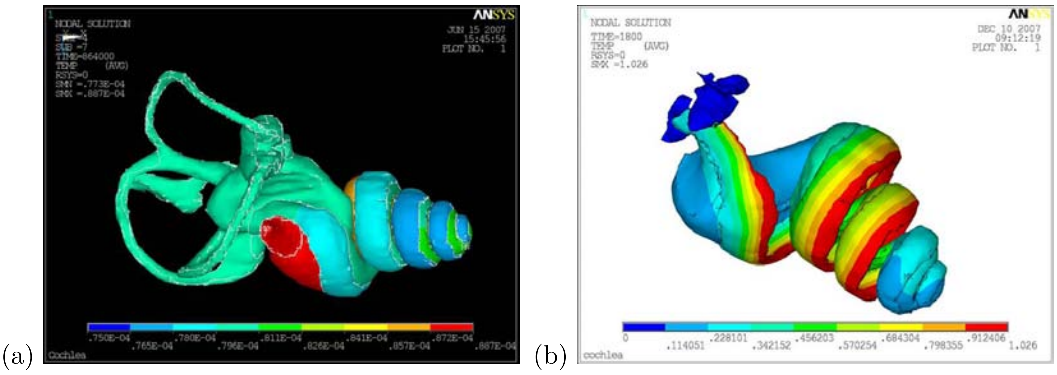


Figure 8 Simulation scenarios using ANSYS. (a) Round window application scenario; (b) validation of transfer coefficients for clearance estimation.

The four major compartments are illustrated in Figure 4. Most tetrahedra belong to the complex containing scala tympani, scala vestibuli, and the vestibule. The round window, where medication is usually applied, is located in front of the scala tympani (seen from the middle ear), which is twisted to a snake-like shape together with the other scalae. At the very end, the apex, it is connected to the scala vestibuli, as depicted in Figure 6. For the ANSYS simulation, this complex is split into two labels ST and SV (scala tympani and scala vestibuli).

The three other compartments illustrated in Figure 4 are the endolymphic space (ES, green) between the scalae, the spiral ligament (SL, gray) causing a clearance at the outside boundary due to blood flow, and the organ corti (OC, blue), which is part of the hearing nerve system. Our simulation model assumes that each compartment has a unique diffusion coefficient and that material can also be exchanged between the common boundaries, based on transfer coefficients. While the diffusion coefficients mostly depend on the medium, the transfer coefficients heavily depend on the molecule size of the drug and of the tissue. Modeling the tissue as a surface rather than a volumetric shape is already a simplification. The simulation model has been described earlier [8] using a simple spiral-like shape model. The equations are defined as follows:

Diffusion of medical concentration c_i ($i = ST, SV, ES, OC, SL$) in a compartment, due to its diffusion constant k_i:

$$\frac{\partial c_i}{\partial t} = k_i \Delta c_i, \tag{1}$$

where Δ denotes the Laplace operator.

Transfer between concentrations c_i and c_j in adjacent scalae linked by (directed) area n

with transfer coefficient k_{ij}:

$$k_i \frac{\partial c_i}{\partial n} = k_{ij}(c_i - c_j). \tag{2}$$

Inflow of medication concentration from the round window into the scala tympani, due to concentration D at round window:

$$k_{ST} \frac{\partial c_{ST}}{\partial n} = -\beta(c_{ST} - D). \tag{3}$$

Clearance at the outer boundary of spiral ligament:

$$k_{SL} \frac{\partial c_{SL}}{\partial n} = -\gamma c_{SL}. \tag{4}$$

The only well known quantities in the above model are the diffusion coefficients k_i. The transfer coefficients k_{ij} depend on the structure of the tissue between each pair of neighboring scalae and on the molecular size of the drug. We note that the transfer constraints in equation (2) are symmetric, seen from both sides. The coefficients β and γ are one-sided transfer coefficients for inflow and clearance, respectively. We assume that the drug concentration in the blood is zero, since the volume of the cochlea is much smaller than the volume of the entire body. An additional term for bio-chemical degradation reducing the drug concentration over time may be added when considering long time scales.

We assume that the drug or marker substance is either placed on the round window surface or (in secondary experiments) directly ejected into the scala tympani. The overall concentration drops down over time, mostly due to a clearance at tissues connected to major blood flow. In regions such as the spiral ligament, the drug is partially advected by blood flow into the entire body, where it does not play a significant role for ours simulation. We note that this effect is reversed when a systemic therapy is simulated, i.e. the drug is contained in the blood and gets into the cochlea via diffusion.

For the solution of above equations, we used ANSYS, a FEM-based simulation package mostly used for calculating heat transfer. Our diffusion model in fact differs from heat transfer only in the physical units used. The drug concentration inside the scalae is represented by a piecewise linear representation,

$$c(x, y, z) = \sum_i a_i \phi_i(x, y, z), \tag{5}$$

where a_i are coefficients and ϕ_i linear ansatz functions associated with ANSYS nodes located at the vertices of our tetrahedral mesh.

For the transfer between adjacent scalae, ANSYS link elements are employed. These couple the corresponding nodes and take into account a transfer coefficient and the directed surface i.e. an averaged normal vector times the orthogonal surface area. To properly specify these link elements, our preprocessor needs to duplicate all triangles (and the corresponding vertices) belonging to inner tissues, see Figure 7. The link element between two ANSYS nodes corresponding to a split vertex takes one third of the directed areas of all tissue triangles connected to this vertex. Directed areas are accumulated by summing up the corresponding vectors.

After the tetrahedral model has been preprocessed and exported for ANSYS, the simulation is started, based on an ANSYS script. Figure 8 depicts two simulation scenarios, (a) round window application where the concentration decays along the scala tympani and also spreads into the adjacent compartments and (b) a scenario to estimate transfer coefficients, described below.

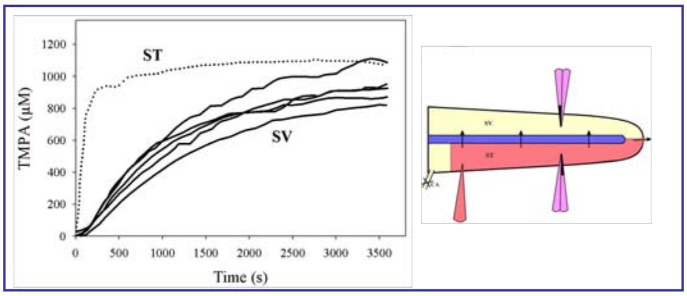


Figure 9 Results from first experiment [12] filling scala tympani with an ionic marker (TMPA) and measuring concentration in scala vestibuli.

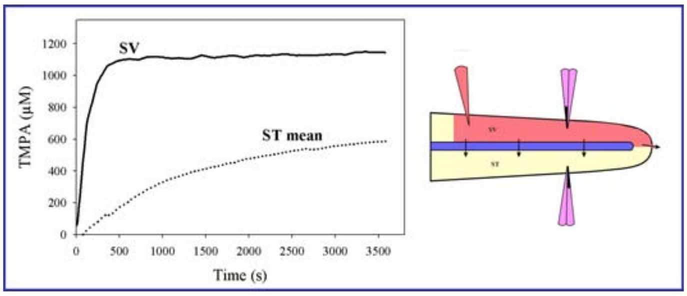


Figure 10 Results from second experiment [12] filling scala vestibuli and measuring concentration in scala tympani.

3.2 Clearance Estimation

The main difficulty in our simulation approach is the estimation of transfer and clearance coefficients. These are not known from the literature (like diffusion coefficients) and may depend on the molecular shape of a drug or marker substance. The only way of estimating these coefficients is a time-consuming parameter-identification process, where the FEM solver is iteratively being used with different parameter sets until the most proper solution has been found. This process requires some knowledge about the data, for example measurements.

In the following, we describe this parameter identification process validating measured results from the literature [12]. These measurements are taken from two experiments. First, the scala tympani of a guinea pig had been filled with a ionic marker (TMPA) substance and the concentration had been measured over time at two points inside the scalae tympani and vestibuli, see Figure 9. In the second experiment, the scala vestibuli was filled with the ionic marker and the concentration inside the scala tympani was measured. It is observed that this concentration does not reach the full level, since the clearance via the spiral ligament constantly reduces the concentration, while that one in the scala vestibuli is artificially kept at the same level.

Based on the data from the first experiment, we were able to estimate the clearance coefficient for the spiral ligament, see Figure 11. Without any clearance (left), the scala vestibuli is eventually filled up completely, i.e. with the same concentration as inserted. By adding clearance from the spiral ligament to the systemic blood flow the calculated final concentration at the measurement site (third winding of scala vestibuli) can be matched to the measured concentration (right of Figure 11). With this method, the transfer coefficient for the outer clearance is determined.

It is conjectured that also a clearance inside the cochlea due to blood vessels near the

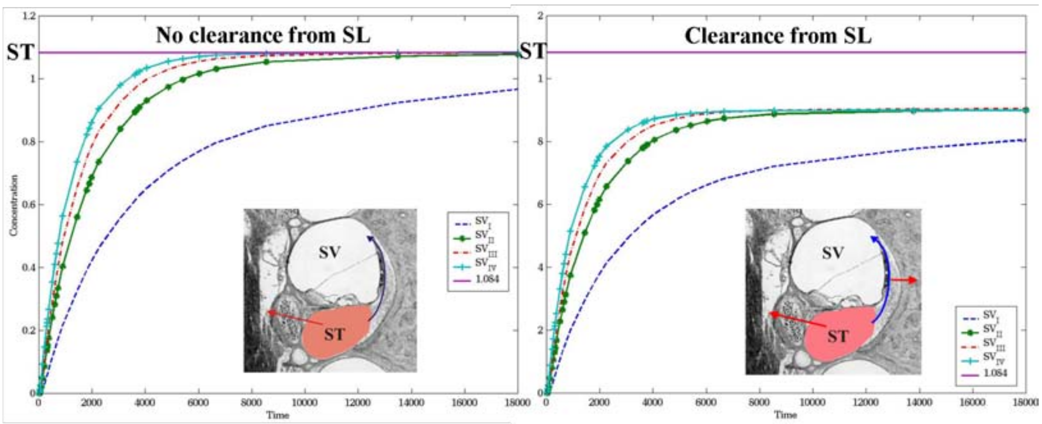


Figure 11 Validation of results of first experiment without clearance from spiral ligament (left) and with outer clearance identified by matching experimental results (right).

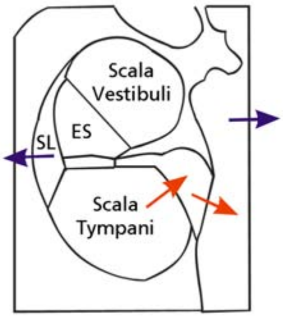


Figure 12 Schematic view of medication clearance due to blood flow. It is conjectured that in addition to the spiral ligament (SL), a second clearance must apply at the inner scala tympani.

scala tympani exists [9], see Figure 12. With our simulation we were able to prove this hypothesis. The corresponding transfer coefficient can be estimated, provided that the exact region of the inner clearance is known. Figure 13 shows the calculated concentration at pipette location in the third turn of the scala tympani while holding the level inside the scala vestibuli constant. When only adjusting the clearance coefficient for the spiral ligament, the actually measured concentration cannot be reached. An additional clearance of substance to the modiolus is needed to explain the data. The link nodes where we assumed that the clearance takes place are highlighted in Figure 14.

4 1D Model Reduction

In the above section we have described the parameter identification process. Since the full-resolution ANSYS simulation takes about 30 minutes (regardless of the simulation time, since time steps are adapted to the rate of change in “temperature”), this process is not very efficient. On the other hand, our model still contains a number of transfer coefficients that are not really accurate. For calculating these parameters, the FE-model needs to be reduced into a more efficient, yet accurate approach. Exploiting the one-dimensional shape of the scalae, we propose a 1D-parametrization of these, such that we obtain five univariate concentration functions $c_i(s)$, interconnected by corresponding transfer functions. This

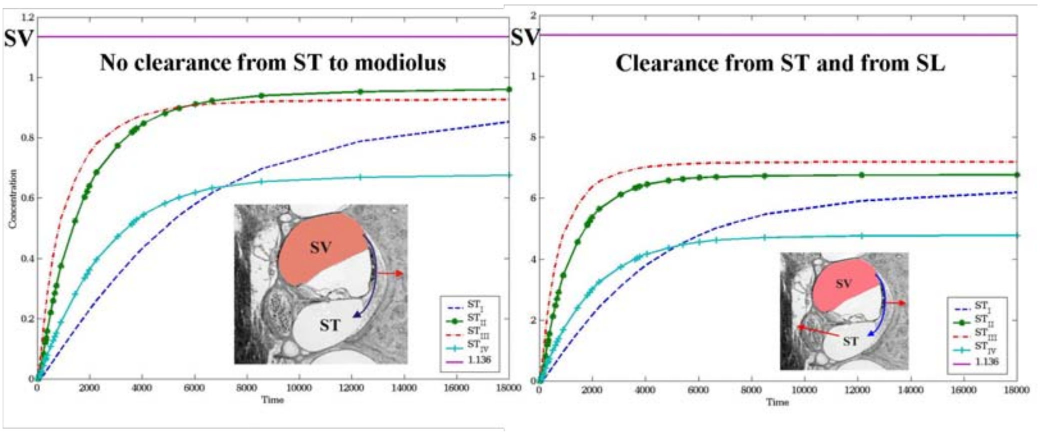


Figure 13 Validation of results of second experiment without clearance from inner scala tympani (left) and with inner clearance identified by matching experimental results.

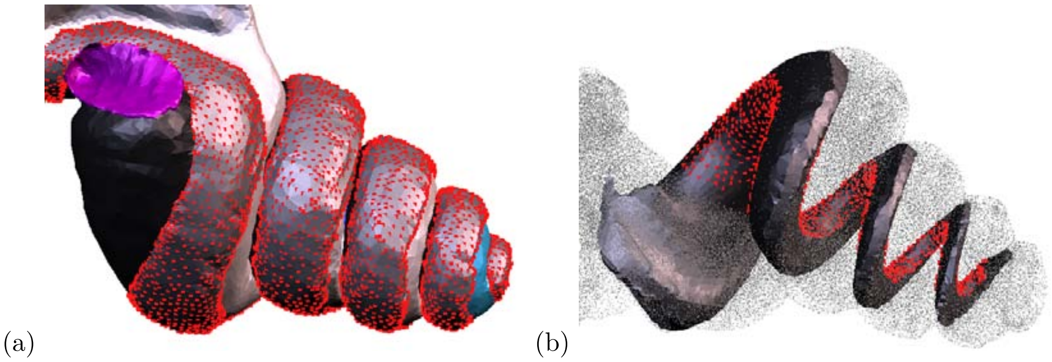


Figure 14 Link nodes used for clearance. (a) outside spiral ligament; (b) inside scala tympani.

approach corresponds to unwrapping and stretching the cochlea, as illustrated in Figure 1. With a one-dimensional parametrization of the individual scalae, also the simulation model is projected onto the topologically simpler 1D model space, according to Figure 15.

Unfortunately, to our knowledge no commercial tool for unwinding inner ear models exists. Hence, we extended our own modeling and visualization system (previously used to link AMIRA with ANSYS; all implemented in C++ based on OpenGL rendering) to also support these geometric operations (Figure 2).

In a first approach, we approximated the four main compartments (leaving scalae tympani and vestibuli connected) by polylines. These where obtained by a graph clustering approach in the following way:

- create a graph node for every tetrahedron
- connect the nodes of adjacent tetrahedra by edges
- repeatedly collapse shortest edges merging two nodes

For collapsing, we admit only edges that do not connect different compartments. Each node is associated with the center of a polyhedral complex and each edge carries the accumulated directed area of its corresponding surface. Based on this approach (with few more constraints eliminating small cycles) we obtained the clustering in Figure 16.

Though this cluster graph still contains enough information for simulation, its representation is too coarse to obtain accurate results. However, we could use the resulting polylines to

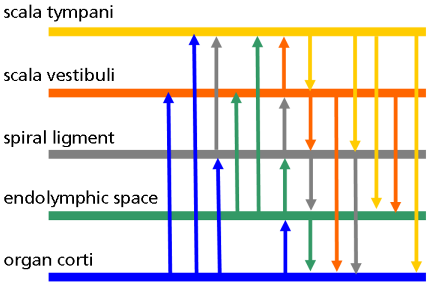


Figure 15 Schematic view of 1D model.

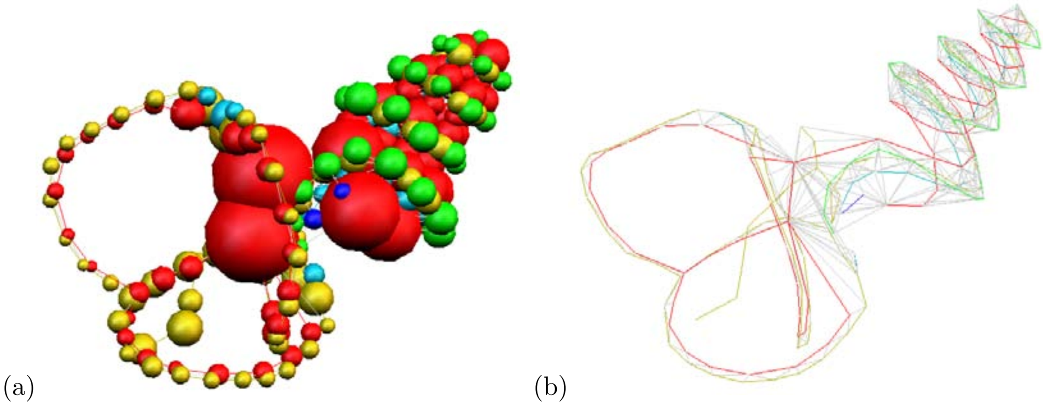


Figure 16 Clustering results. (a) visualization of associated polyhedral volumes; (b) connectivity graph for parametrization of scalae.

parametrize the different compartments. Thus, the clustering provided a (nearly) automatic way of separating the scalae tympani, vestibuli and the apex by simply marking tetrahedra associated with the nodes of corresponding polylines, see Figure 17.

In a second attempt, for each compartment a set of cross-sectional slices following a midpoint-curve (cubic interpolating spline for graph nodes) was constructed. Besides the matching problem for parametrizations of different scalae, this approach did not provide smooth transitions of the coordinate frame (Bishop Frame). One of the most significant reasons for this is that the normal plane of a guiding curve (i.e. a plane whose normal vector is tangential to the curve) is not the optimal choice for minimal cross sections. In some cases, these intersection planes produced connected 2D regions from multiple windings.

A clean approach to parametric cross sections was found by interpolating a linear axis inside the cochlea. With two fixed points on this axis and one parametrized point on a guiding curve (midpoint of spiral ligament), a smooth parametrization of cutting planes is obtained. The polygonal regions of the individual scalae can be combined to a tubular structure, as depicted in Figure 18.

Starting with a resolution of 500 slices, the region where scalae tympani and vestibuli are disconnected (408 slices) was identified. Interestingly, the scala vestibuli already ends at slice 324, where it enters the apex, see white tube in Figure 1 and compare to Figure 17. What on the first glance looks like a flaw in our visualization is caused by the fact, that this scala has one winding less than the other scalae.

The parametrization of cross-sectional slices of the five scalae allowed us to transform

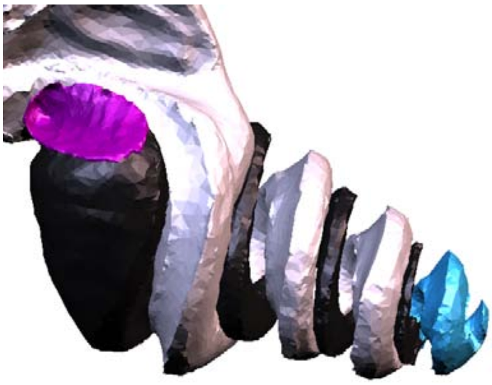


Figure 17 Automatic separation of scala tympani (black), scala vestibuli (white), and apex (cyan), based on clustering graph.

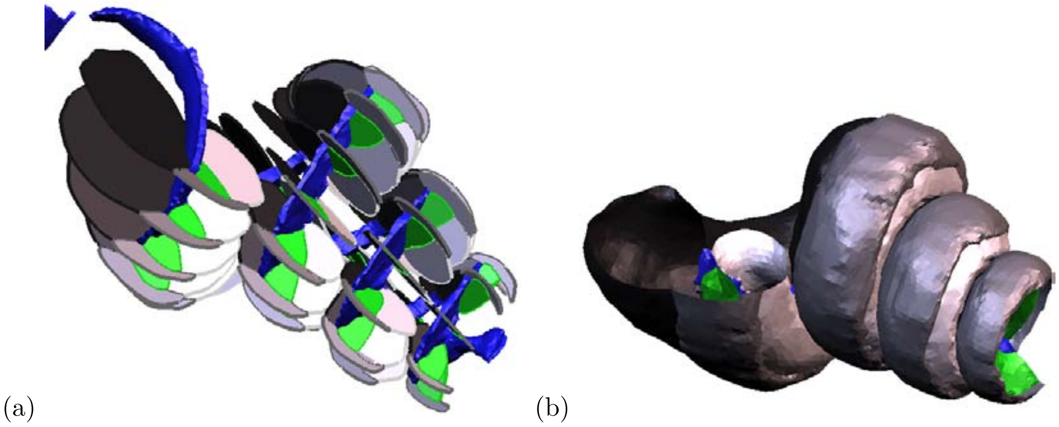


Figure 18 Construction of slices. (a) moving slices around organ corti (blue); (b) slices connected to tubes.

the equations (1–4) of the 3D model into a one-dimensional approach. This means that the system of equations is now a banded system with five equations for every cross-sectional slice. In order to transform the equations properly, we needed to calculate for every two consecutive slices the volume of the compartments in between and the common surface areas for every pair of compartments.

The simulation results for a round-window application scenario based on our reduced 1D model and the full 3D ANSYS simulation are compared in Figure 19. Therefore, we have re-loaded the ANSYS simulation into our own viewer and averaged it over the cross-sectional area for each scala. A 3D comparison of both results is depicted in 20. Despite of the much greater efficiency of the 1D approach, the solution is nearly as accurate as the 3D approach. The computation time for one simulation has dropped down from 30 minutes to a few seconds, which is efficient enough for sophisticated parameter identification.

5 Concluding Discussion

In the present work, we have combined previous constructions of accurate cochlea geometry [10] with a physical simulation method [8] for medical application to the inner ear. The model is accurate enough to validate measurement results from the literature [12] and to

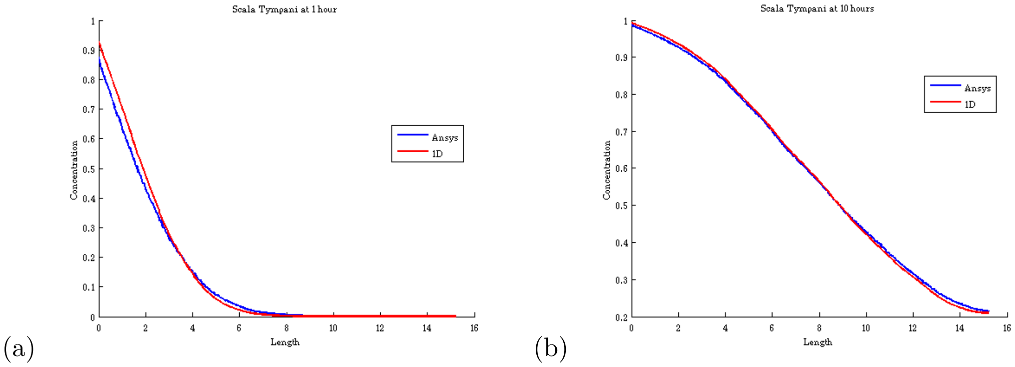


Figure 19 1D Comparison of 3D ANSYS simulation and 1D model reduction. (a) Simulated concentration inside scala tympani after one hour of application; (b) after 10 hours.

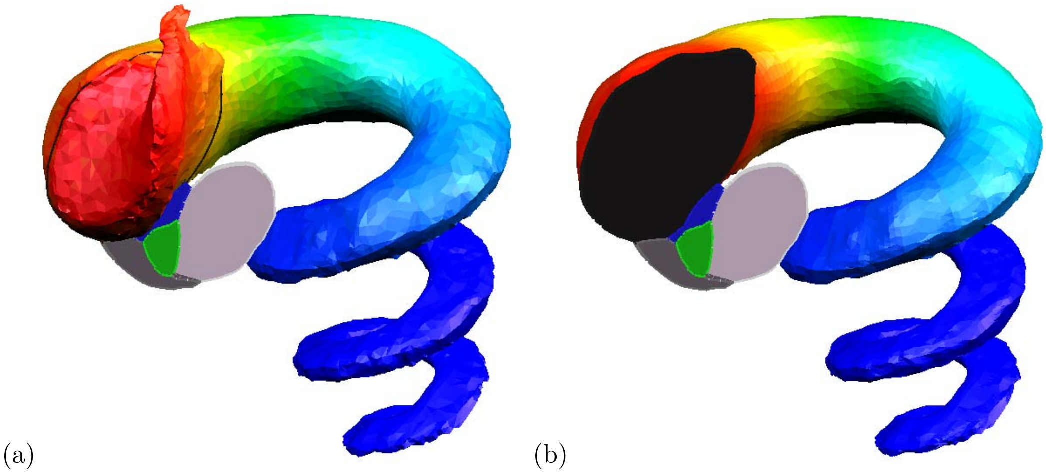


Figure 20 3D Comparison for scala tympani after one hour application time. (a) ANSYS-result; (b) reduced 1D simulation. The 1D simulation nearly coincides with the full 3D simulation, since concentration gradients are mostly oriented along the scalae in typical application scenarios. We used a similar color map as ANSYS, here without quantization. Alternative color maps are available (since rainbow colors work well for comparative visualization but do not convey gradient magnitude).

prove a hypothesis [9] that also a clearance at the inner boundary of the scala tympani exists.

The benefit of our modeling and visualization system to achieve this task is that simulation results obtained with ANSYS and with MATLAB can be integrated into the same system and compared on the same geometric model. For this case study, even simple visualization methods were extremely helpful to understand the entire process. For example, when unwinding the cochlea, we observe that the scala vestibuli (white in Figure 1) is shorter than the others, since it is connected earlier to the apex. Also the location and quantification of clearances is very convenient with the aid of a 3D visualization with user interaction.

A limitation of the simulation model is still that the exact values for a number of parameters are still unknown. These are mostly related to the material-dependent transfer coefficients that also may have local variations. A further increase of accuracy would be obtained by knowing the exact regions where clearance can occur. In order to determine these parameters, we have reduced our finite element model to a more efficient one-dimensional approach that matches the FEM-based simulation results for round window application scenarios. Simulation scenarios with steep concentration gradients across scalae boundaries may cause greater errors of the 1D approach. However, when using this efficient method for parameter identification (where the simulation is run hundreds of times with various parameters), the resulting parameter set can be validated by a single 3D simulation.

Our ultimate goal is the reduction of animal experiments required for approval of novel application systems designed for the human ear. This includes a more reliable method for predicting such processes compared to "up-scaling" results from animal experiments to the human ear geometry. The present work is a small step towards this goal, providing a methodology composed of geometric modeling, visualization, simulation, and validation. In the near future, a lot of work will be necessary for modeling and simulating human cochleas where the variation in geometric shape is somewhat greater than for the guinea pig. A further challenge is the validation of the results, based on non-invasive measurements.

Acknowledgements

This work was supported by grants BMBF 0313844 to NS and SKP and NIDCD DC01368 to ANS.

References

1 H.J. Beer, M. Bornitz, H.J. Hardtke, R. Schmidt, G. Hofmann, U. Vogel, T. Zahnert, and K.B. Hüttenbrink. Modelling of components of the human middle ear and simulation of their dynamic behaviour. *Audiology & Neuro-Otology*, 4(3-4):156–162, 1999.

2 J. Dornheim, S. Born, S. Zachow, M. Gessat, D. Wellein, G. Strauß, B. Preim, and D. Bartz. Bildanalyse, visualisierung und modellerstellung für die implantatplanung im mittelohr (in german). *Proc. Simulation and Visualization '08*, pages 139–154, 2008.

3 C. Fernandez. Dimensions of the cochlea (guinea pig). *J. Acoust. Soc. Am.*, 24:519–523, 1952.

4 R.Z. Gan, Q. Sun, R.K. Dyer, K.-H. Chang, and K.J. Dormer. Three-dimensional modeling of middle ear biomechanics and its applications. *Otology & Neurotology*, 23(3):271–280, 2002.

5 P. Hastreiter, C. Rezk-Salama, B. Tomandl, K. Eberhardt, and T. Ertl. Interactive direct volume rendering of the inner ear for the planning of neurosurgery. *Proc. Bildverarbeitung für die Medizin '99*, page 192–196, 1999.

6 R.K. Hobbie. *Intermediate physics for medicine and biology (3rd edition)*. Springer, 1997.

7 S. Plontke, A.W. Wood, and A.N. Salt. Analysis of gentamicin kinetics in fluids of the inner ear with round window administration. *Otology & Neurotology*, 23:967–974, 2002.

8 S.K. Plontke, N. Siedow, R. Wegener, H.P. Zenner, and A.N. Salt. Cochlear pharmacokinetics with local inner ear drug delivery using a 3d finite element computer model. *Aud & Neur*, 12:37–48, 2007.

9 H. Rask-Andersen, A. Schrott-Fischer, K. Pfaller, and R. Glueckert. Perilymph/modiolar communication routes in the human cochlea. *Ear Hear*, 27:457–465, 2006.

10 A.N. Salt, R. Gill, A. Voie, and G. Saxon. 3d representation of the guinea pig inner ear and the implications for modeling drug movements in the fluids. *Abstr. Assoc Res Otolaryngol*, 2006.

11 A.N. Salt, M.M. Henson, S.L. Gewalt, A.W. Keating, and O.W. Henson. Detection and quantification of endolymphic hydrops in the guinea pig cochlea by magnetic resonance microscopy. *Hearing Research*, 88:79–86, 1995.

12 A.N. Salt, K. Ohyama, and R. Thalmann. Radial communication between the perilymphatic scalae of the cochlea. i: Estimation by tracer perfusion. *Hearing Resiearch*, 56:19–36, 1991.

13 R.K. Shepherd and M.P. Colreavy. Surface microstructure of the perilymphatic space. *Arch Otorlarnygol Head Neck Surg*, 130:518–524, 2004.

Information-theoretic Analysis of Unsteady Data

Heike Jänicke

Swansea University
Swansea, Wales, UK
h.jaenicke@swansea.ac.uk

Abstract

The temporal evolution of scientific data is of high relevance in many fields of application. Understanding the dynamics over time is a crucial step in understanding the underlying system. The availability of large scale parallel computers has led to a finer and finer resolution of simulation data, which makes it difficult to detect all relevant changes of the system by watching a video or a set of snapshots. In recent years, algorithms for the automatic detection of coherent temporal structures have been developed that allow for an identification of interesting areas and time steps in unsteady data. With such techniques, the user can be guided to interesting subsets of the data or a video can be automatically created that does not occlude relevant aspects of the simulation. In this paper, we give an overview over the different techniques, show how their combination helps to gain deeper insight and look at different directions for further improvement. Two CFD simulations are used to illustrate the different techniques.

1998 ACM Subject Classification G.3 Time series analysis, E.4 Coding and information theory

Keywords and phrases Information theory, unsteady data

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.118

1 Introduction

One of the big challenges in visualization is the depiction of data that changes over time. Many interesting structures in fluid simulations owe their relevance to their temporal evolution and the effects they have on a wider region of the system. Wake vortices that occur behind aeroplanes, for example, originate from the aeroplanes wings and effect growing regions behind the plane with increasing time. One way to investigate such evolution over time is to watch a video of the data set or to take snapshots at relevant time steps. As many applications are inherently three-dimensional, it is difficult to find a good camera position to depict all relevant changes in an unsteady data set. A second problem is the fact that not all relevant properties are known a priori. Hence, the movie might be perfect to investigate large changing structures but smaller and more subtle phenomena might be overlooked or occluded.

One way to ensure that relevant structures are depicted in the video, is to automatically detect interesting phenomena beforehand using so called feature detection algorithms. Many such algorithms use a predefined mathematical description of the phenomenon and find corresponding structures in space and time. However, even for such basic structures as vortices there exists no unique detection criterion but rather a large variety of vortex measures. Structures so far unknown are likely to be missed as it is difficult to capture anomalies by mathematical descriptions. Such structures might easily evolve in simulations with conditions that did not exist before like abnormal weather patterns induced by climate change.

In order to detect all these different relevant patterns, several approaches have been recently proposed. Commonly, they identify coherent structures that are different from

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 118–128
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

what would be considered ordinary in the given data set. Most such techniques are based on concepts from information theory, a theory that is amongst other things concerned with measures that quantify how much information a subset of data contains about the entire system.

In the following we will give a brief overview over different techniques and use two different flow simulation data sets to show how they can be used in combination to explore time-dependent data. As the different techniques are related to a large variety of applications, we refer the interested reader to the related work sections of the papers cited for each technique.

2 Information-theoretic Data Analysis

Information theory is a vast field with a huge amount of measures to quantify information contents. However, only few of them are applicable to structured time-dependent spatial data. The concepts to come are based on the notion of causal states which can be thought of as stochastic spatio-temporal patterns capturing the dynamics of a local neighborhood. Hence, causal states are perfectly suited to investigate the dynamics in the applications under consideration. Afterwards, we will investigate measures to quantify which causal states feature unusual behavior, before we summarize the idea of ϵ-machines that can be used to investigate the evolution of entire systems over longer time intervals.

2.1 Causal States

A powerful concept from information theory is the notion of causal states as introduced by Crutchfield and Young [2]. A causal state is a stochastic spatio-temporal pattern that describes a position's or particle's past and future (compare figure 1). The past comprises all other positions that go directly or indirectly into the computation of the value at the central position. The future is the set of positions whose value is influenced by the value associated with the position or particle. A simulation's set of causal states is computed from the underlying data by dividing the occurring configurations into different classes according to their past and future. Each resulting causal state is a set of past configurations along with an associated distribution over possible futures that might occur after observing the past. For more details on the theory and computational aspects of causal states, we refer the interested reader to previous work [5]. In summary, the set of causal states summarizes the dynamics in the data set and gives an overview over the short-term temporal evolutions in it.

2.2 Eulerian vs. Lagrangian Frame of Reference

As shown in figure 1, the structural masks for past and future can be chosen differently. If we choose these structures such that they are analog to the finite difference model used in computational fluid dynamics (CFD), they have the shape of light cones as illustrated in figure 1 (top) labeled EulerianLSC. This structure supports the Eulerian description of motion, where the simulation domain is subdivided into cells and the evolution of values at different cell positions is recorded over time. An alternative frame of reference is the Lagrangian description of motion, where the flow is described using a number of particles that change their position according to the velocity field. In the Lagrangian frame of reference, the evolution of values associated with a certain particle is of interest. A simple example of the two frames of reference is the investigation of a river. We can either observe a fixed

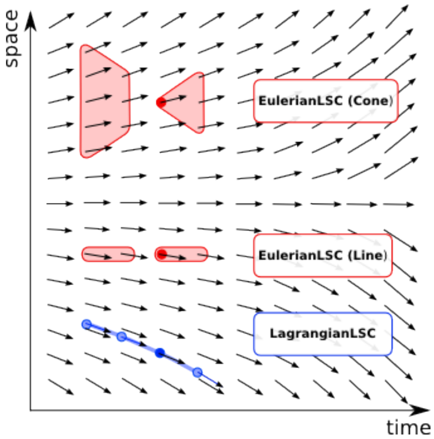


Figure 1 Past and future configurations for the different measures of complexity.

position and monitor the velocity, temperature or pressure at this particular position or we can put a small boat on the river and monitor the water properties beneath the boat that follows the current. Both frames of reference are of high importance in CFD research and can be supported by the causal states using different structures for past and future as illustrated in figure 1. The light cone structures at top and center support the Eulerian frame of reference and the one at the bottom the Lagrangian frame of reference. The cylinder shaped structure depicted in the middle is a computationally faster version of the EulerianLSC, which we found to give similar results as the full cone structure [4].

2.3 Local Statistical Complexity (LSC)

The goal of this work is to identify positions that form interesting structures in the data set. The causal states we have identified so far subdivide the data set into different classes of behavior. The next thing we need is a measure to quantify how interesting each class of coherent behavior is. Commonly, it is very difficult to say what a user might be interested in. If we assume that we are to show the user the unlikely events, the task becomes more feasible. Local statistical complexity [7, 3] is an information-theoretic measure that tells for a given causal state how much information from the past is required to predict the dynamics in the local future. If we only need little information and it is very easy to predict the dynamics, we found a pattern that is very common in the data set and the user probably knows that it is in there anyway. However, if we detect a pattern that is hard to predict and therefore unusual, it might be something that the user is interested in. Hence, by computing the local information contents, local statistical complexity assigns each causal state and thus each position in the (multivariate) data set a scalar value telling whether the local dynamics are common or not. The resulting time-dependent scalarfield can easily be used to guide the users attention or to compute a good position for the camera when generating a movie.

2.4 ϵ-Machines

In order to study the dynamics of the system as one, we need a depiction of the causal states that provides information on how they interact with each other. Such a visualization is provided by ϵ-machines [6]. ϵ-Machines can be thought of as directed graphs. The nodes are the causal states of the system and the edges depict the transitions between them.

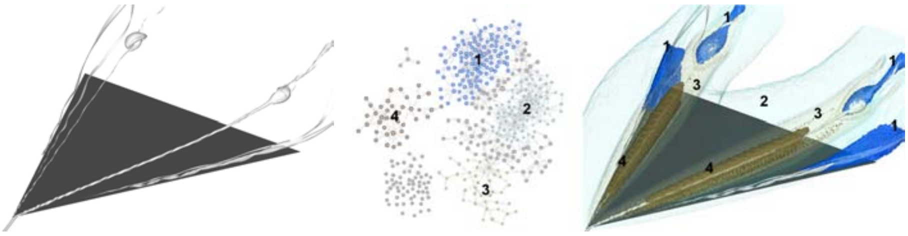


Figure 2 Relevant structures in the delta wing data set: (left) Streamsurfaces indicating the six vortices above the wing. (center) ϵ-machine of this data set with colored substructures. (right) Physical positions corresponding to the colored subregions of the ϵ-machine.

For example, if a position in the data set changes from causal state A in time step t to causal state B in time step $t+1$, we add an edge linking causal states A and B in the ϵ-machine and assign it weight 1. This procedure is repeated for all transitions in the entire data set. The resulting structure can be thought of as a finite state machine capable of simulating the dynamics in the given data set. Now that we have a model representing the entire data, we can analyze its properties. For example, we can investigate strongly linked components to detect coherent structures in space, find subsets with high LSC to identify unusual formations or track features over time.

2.5 Areas of Application of the Different Techniques

In the preceding sections, we have summarized the concepts of causal states with their application to Lagrangian and Eulerian flows, local statistical complexity and ϵ-machines. Before we continue with the data analysis let us briefly look at the type of information the different concepts can extract. Causal states form the building blocks of all the techniques. They subdivide a given system into a set of stochastic spatio-temporal patterns. Based on the structural mask of the causal state, we can decide if we want to investigate the data from an Eulerian or Lagrangian point of view. The Eulerian view gives site specific information and relates different positions to one another. The Lagrangian view supports the analysis of particle evolution. Local statistical complexity can be used in both scenarios to identify either positions or particles that feature extraordinary local dynamics and hence, indicate interesting regions. When the dynamics of the entire system are of interest, ϵ-machines are a useful tool. They depict all causal states occurring in a system and show how they interact. Different enhancements can be used to focus on specific information such as the evolution of coherent structures, stability of different phenomena or the interaction between different structures in the domain.

3 Data Analysis

In the following we will apply the different analysis techniques to two different flow simulations and show how they can be used in combination to gain deeper insight into the data.

3.1 Delta Wing

The Delta Wing data-set represents the airflow around a delta wing at low speeds with an increasing angle of attack. Multiple vortex structures form on top of the wing due to the rolling-up of the viscous shear layers that separate from the upper surface. These

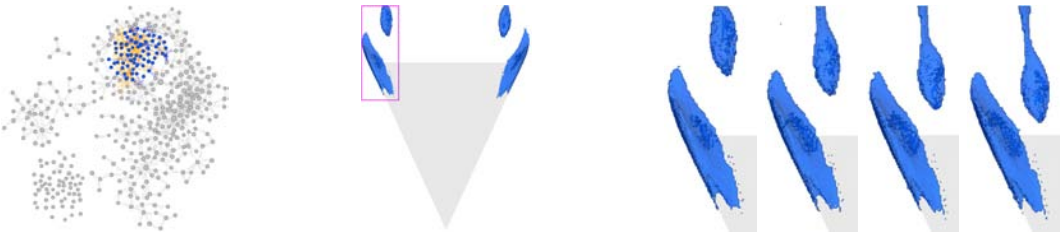


Figure 3 Evolution of the recirculating bubble in the delta wing data set: (left) ϵ-machine; Selected nodes are colored in blue. (center) Corresponding positions in time step 650. The pink frame indicates the area of the closeup. (right) Physical positions with corresponding causal states in time steps 650, 670, 690, and 710.

formations of three vortices can be observed on either side of the wing (Fig. 2(left)). With increasing angle of attack the intensity of the primary vortices (vortices nearest the symmetry axis) increases until in time-step 700 a vortex breakdown occurs (bubbles at the end of the vortices). The analysis of vortex breakdown is highly interesting, as it is one of the limiting factors of extreme flight maneuvers. The grid consists of approximately 3.1 million positions.

Figure 2(center) shows the ϵ-machine of the data set. Several highly interconnected structures were brushed and positions corresponding to these causal states are depicted in figure 2(right) using isosurfaces of the same color. We can distinguish four different structures: the causal states indicated by number one and colored in dark blue can be found at the recirculating bubbles and at the tips of the wing, number two is the surrounding domain, number three comprises positions forming an outer hull around the major vortices and number four are the positions in the main vortex. If we follow the causal states in the dark blue selection number one over time as depicted in figure 3, we see that the outline perfectly follows the changing structures of the recirculating bubble. Hence, we are able to detect and follow features without giving an a priori mathematical definition of the structure. Moreover, as such structures consist of a set of connected causal states the description is more flexible and we can easily adopt it to features whose values change over time, by adapting the selection to strongly linked nodes in the graph.

Now that we have identified the different structures in the data set, we can look for the unusual ones. Therefore, we can either color the nodes in the ϵ-machine according to local statistical complexity as illustrated in figure 4(left) or we can render an isosurface highlighting all positions with a LSC value above a given threshold. The colored ϵ-machines reveals that the recirculating zones are the most complex structures. Additionally some interesting formations occur in all other subparts but it is rather difficult to isolate additional coherent structures. Looking at the isosurface in the LSC field (fig. 5), we can clearly distinguish the minor vortices that are part of subset 2 in the ϵ-machine. Color coding the ϵ-machine according to the number of edges per node, as shown in figure 4(right), reveals that positions belonging to the major vortices and their hulls are very dynamic and often change their state in a very unpredictable manner as there are many possible successors. To investigate this behavior more closely, we will look at the Lagrangian representation of the data.

Figures 6a and 6b show a number of pathlines started in front of the delta wing where color indicates the norm of velocity or pressure at the given position. Comparing the two color distributions, the images look very different. However, if we color the particle traces using the LSC of the respective variables, the results become quite similar. This means that particles feature an unusual evolution of values for velocity as well as for pressure. The two

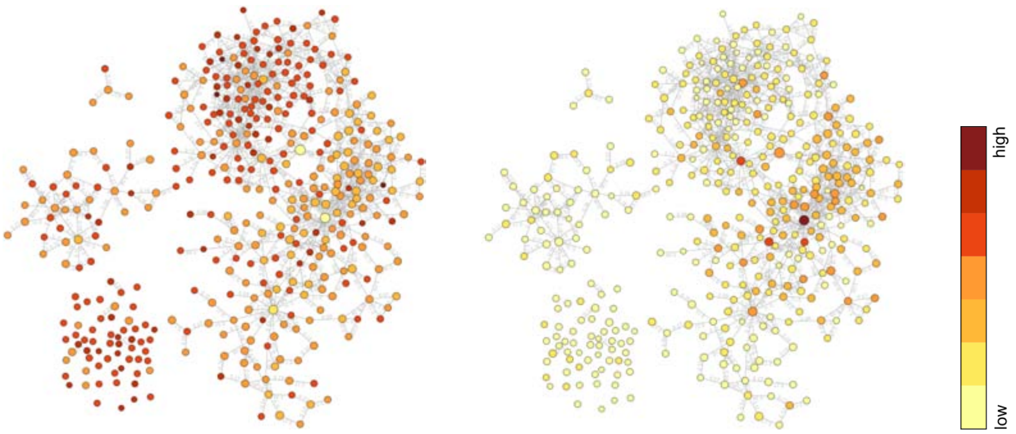
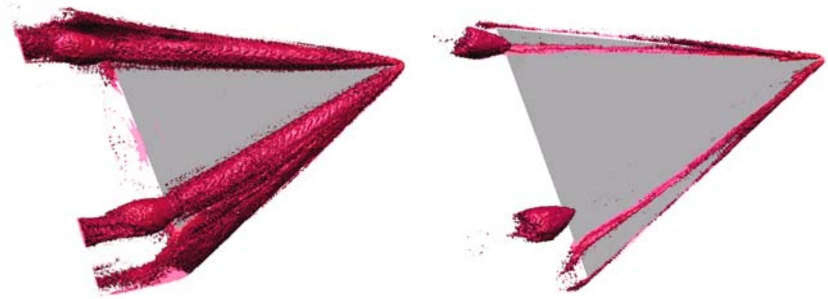


Figure 4 ϵ-Machine of the delta wing using different color codings: (left) local statistical complexity of causal state and (right) number of edges per node.

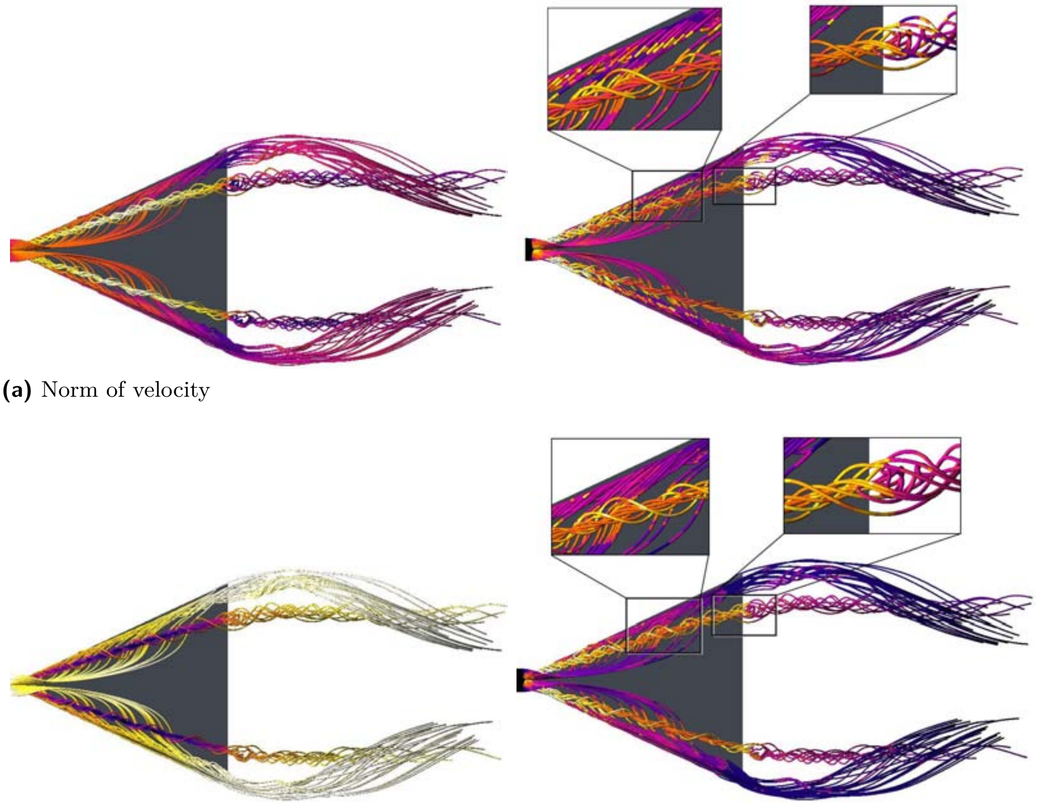


(a) LSC > 10 **(b)** LSC > 14

Figure 5 Isosurfaces in the local statistical complexity field of the norm of velocity of the delta wing data set.

close-ups focus on two different features. The left image depicts an area where particles move from the area of influence of the major vortex into the one of the minor vortices. We can see that the particle traces that first go underneath the spiralling major vortex feature yellow color, i.e., high LSC, when close to the first vortex, go back to more usual temporal dynamics when between both vortex structures and change to more unusual dynamics again as they enter the minor vortices. The close-up on the right hand-side depicts particles that change from the major vortex into the recirculating bubble. Here we see that the dynamics are most unusual at the transition point and more predictable inside the structure.

In summary, we saw that ϵ-machines are well suited to identify major coherent structures in a data set and their evolution over time. Moreover, they give an overview over the distribution of different quantities such as LSC over the different structures. For a more detailed analysis, we used the depictions of the Eulerian- and the LagrangianLSC. While the EulerianLSC is good at giving holistic impressions of the relevant structure, LagrangianLSC is better suited to investigate more subtle phenomena. Hence, all three techniques are a powerful combination and can answer a large variety of questions.



(a) Norm of velocity

(b) Pressure

Figure 6 Pathlines in the delta wing data set color-coded using (a) (left) norm of velocity, (right) LagrangianLSC of norm of velocity and (b) (left) pressure and (right) LagrangianLSC of pressure.

3.2 Swirling Jet

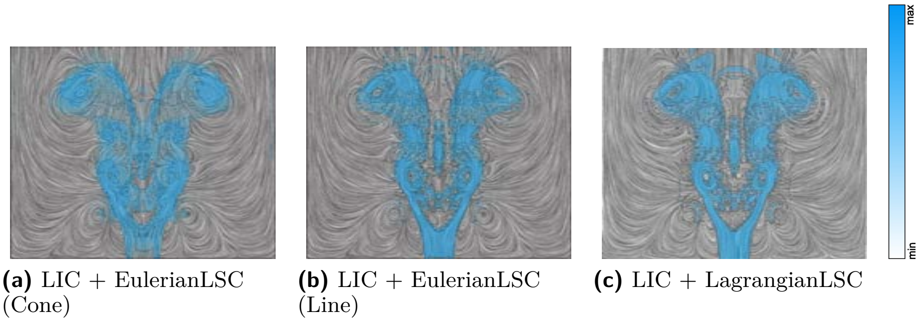
The development of a recirculation zone in a swirling flow is investigated by numerical simulation. This type of flow is relevant to several applications where residence time is important to enable mixing and chemical reactions.

The unsteady flow in a swirling jet is simulated with an accurate finite-difference method. The Navier-Stokes equations for an incompressible, Newtonian fluid are set up in cylindrical coordinates assuming axi-symmetry in terms of streamfunction and azimuthal vorticity. All equations are dimensionless containing the Reynolds number Re and the swirl number S as defined by Billant et al. [1]

$$Re \equiv \frac{v_z(0, z_0)D}{\nu} \qquad S \equiv \frac{2v_\theta(R/2, z_0)}{v_z(0, z_0)} \tag{1}$$

where $z_0 = 0.4D$, $D = 2R$ is the nozzle diameter and ν the kinematic viscosity, as dimensionless parameters.

The flow domain is the meridional plane $\mathcal{D} = \{(r, z) : 0 \leq r \leq R, 0 \leq z \leq L\}$ with $R = 5D$, $L = 8D$ and D denoting the nozzle diameter at the entrance boundary. The flow domain is mapped onto the unit rectangle which is discretized with constant spacing. The mapping is separable and allows to a limited extent crowding of grid points in regions of interest. The present simulation uses $n_r = 91$ and $n_z = 175$ grid points in radial and axial directions. The boundary conditions are of Dirichlet type at the entrance section and the



(a) LIC + EulerianLSC (Cone)

(b) LIC + EulerianLSC (Line)

(c) LIC + LagrangianLSC

Figure 7 Swirling jet: Relevant structures extracted using Eulerian- and LagrangianLSC.

outer boundary and at the exit convective conditions are imposed for the azimuthal vorticity. The initial conditions are stagnant flow and the entrance conditions are smoothly ramped up to their asymptotic values within four time units.

The depiction of time-dependent two-dimensional data is much easier than the visualization of 3D data as it is free of occlusion. All structures can directly be presented using texture-based techniques such as line integral convolution (LIC) and the movie approach. However, these approaches have two disadvantages. Texture-based approaches do not distinguish between high relevance features and those that are induced by noise. Hence, subtle irrelevant structures are overemphasized and it is difficult to distinguish between structures and noise. An additional color-coding based on the norm of velocity or another field might provide assistance in this direction, but the user has to know the relevant value ranges beforehand. Similar problems occur in the temporal domain, where no assistance on what is relevant and what is not is provided. In the following, we will show how the previously introduced methods can help detect interesting phenomena in space and time.

Figure 7 depicts the three different complexity measures applied to the swirling jet data. In general the structures look very similar. This is due to the fact that they change little over time and that the vortex structures remain at more or less fixed locations. Minor differences are present at the central part of the jet and the region at the top of the image where water leaves the domain. While the dynamics in the center are more unusual from an Eulerian point of view, those at the upper parts feature higher LagrangianLSC. EulerianLSC (Cone) marks large areas of the flow and the complexity slowly decreases at the boundaries. This happens as the cones consider a large area of influence that gradually moves out of the relevant structures at the boundaries. This more extended spatial pattern has its strengths when it comes to the analysis of structures that are in itself not very interesting such as vortex center, where the velocity is zero, but are relevant due to the surrounding flow. LSC is perfectly able to capture vortex corelines as it takes neighboring values into account. Hence, the unusual pattern is a stagnant center surrounded by stronger current. The two line-based measures, EulerianLSC (Line) and LagrangianLSC are not able to capture such structures, but nevertheless highlight the vortices as the surrounding flow is very unusual. The more crisp boundaries of the last two measures are, however, better suited to detect sharp boundaries than the conical shear region. Here the boundaries do not get washed out as with EulerianLSC (Cone) but precisely mark the outer and inner boundary. In summary we can say that all three techniques were able to detect the relevant structures and using a combined presentation of LIC and LSC gives a better impression of the dynamics in the data as relevant structures are immediately visible. Spatial patterns in the lower part of the images that are close to noise are not highlighted by the LSC and hence, attract less attention.

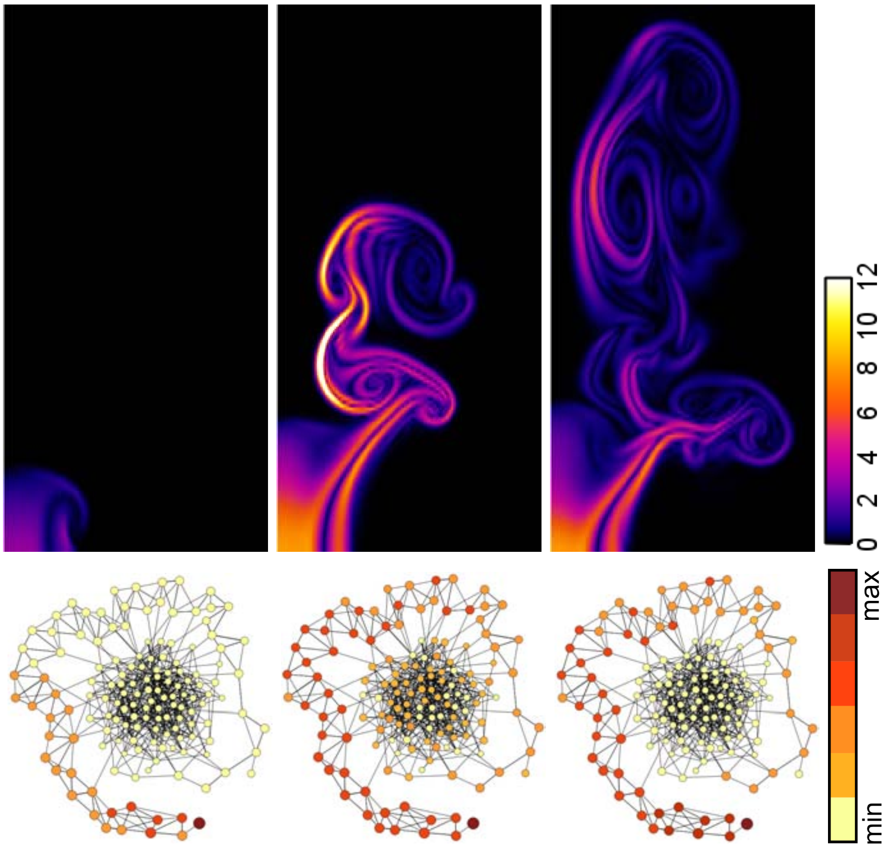


Figure 8 Evolution of the ϵ-machine of the swirling jet (norm of vorticity): Time steps (0), 2, (8), 56, 100.

Now that we have identified the unusual structures in space, we will focus on the temporal component. To investigate the longterm evolution, we look at the ϵ-machine of a related simulation as depicted in figure 8. The corresponding ϵ-machine depicted below looks like a curling up snail-shell. We see a well pronounced outer arc and a lump-like inner set of nodes that is highly interlinked. The color-coding of the nodes indicates the number of positions per causal state in the different time steps.

An easier overview over the temporal dynamics can be gained when representing these structures in a matrix view as depicted in figure 9. Nodes are aligned along the x-axis, the y-axis encodes time and color indicates the number of physical positions that are in a certain causal state at a given time. Two different features can be observed: First, some of the nodes are present throughout almost the entire run and have quite constant intensity, i.e., a stable number of corresponding positions. Second, around time steps 18 and 56 strong fluctuations occur and many of the otherwise unused causal states appear.

When we look at the rendering of the data set at these noticeable time steps, we can relate certain formations in the matrix to physical events. The flow starts with a resting fluid and all positions feature a single causal state, the dark red node at the spirals end in the machine and the corresponding dark red line in the matrix view. With increasing time, the velocity increases and larger parts of the domain leave the steady causal state. During this process (first eight time steps), the nodes in the outer arc of the machine become used more frequently. In the matrix view, this corresponds to the lower part of the matrix where

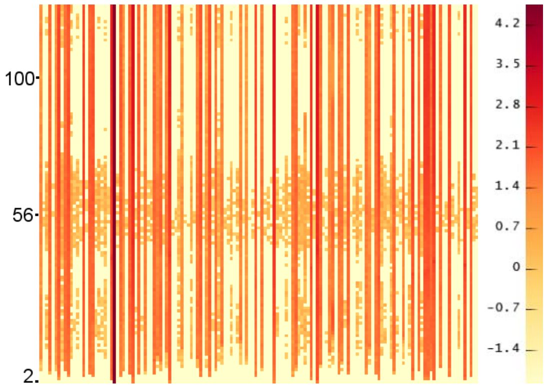


Figure 9 Transition probability in matrix view with logarithmic scaling.

the different vertical lines start, each representing a node in the outer arc of the ϵ-machine. After this initial period, the flow becomes more dynamic and nodes in the center of the machine are activated as well. Strongest dynamics can be observed around time step 56 (fig. 8(center)) when almost the entire inner part of the machine is active. Up to this time step, a strong conical shear region has formed and ring-like vortex structures evolved. Both the matrix view and the ϵ-machine show that the dynamics during this period are hard to predict. The matrix view contains many fluctuations and in the ϵ-machine the inner lump is active that is highly connected with transitions of very low probabilities. In time step 75 the flow becomes more predictable again. The inner lump is no longer active and only stable patterns with few transitions in the outer arc occur in the flow. This stable evolution lasts until the end of the simulation and time step 100 is given exemplarily in figure 8(right).

The LSC analysis in the first part revealed the different relevant structures in the data set, namely the conical shear region and the ring-like vortex structures. However, the measures provided little information about the temporal distribution of the structures and their evolution. This part can be better analyzed using the ϵ-machine and the representation of the causal states. Thus, we can easily go to time steps in the simulation that features an unusual formation in the matrix view of the causal state distribution, render the data using a standard texture-based visualization algorithm such as LIC and highlight relevant components using LSC. If the features are more dynamic, a similar analysis as used with the delta wing based on the LagrangianLSC is possible to investigate the local temporal evolution more closely. However, in this example we found the application of the ϵ-machine combined with any of the LSC measures sufficient to extract all relevant structures over time.

4 Conclusion

In this paper we gave a summarizing overview over recently introduced visualization techniques that aim at the extraction of coherent structures based on information theory. So far, causal states, local statistical complexity and ϵ-machines have always been introduced as independent techniques. While LSC is a measure to extract coherent structures in a time-dependent data set, ϵ-machines are applied to study the longterm behavior in unsteady dynamics. In this paper we showed how the different techniques can be used in combination to gain deeper insight into the data than can be provided by the individual approaches.

In the future we would like to extend the set of methods towards better investigation concepts for long-term correlations and effects. Moreover, the feature tracking ability of the information-theoretic concepts requires more attention to uncover its full ability.

Acknowledgements

The authors would like to thank Wolfgang Kollmann for the swirling jet data-set and Markus Rütten (DLR) for the delta wing data-set.

References

1 P. Billant, J.M. Chomaz, and P. Huerre. Experimental study of vortex breakdown in swirling jets. *JFM*, 376:183–219, 1999.

2 James P. Crutchfield and Karl Young. Inferring statistical complexity. *Phys. Rev. Lett.*, 63(2):105–108, July 1989.

3 Heike Jänicke, Michael Böttinger, Xavier Tricoche, and Gerik Scheuermann. Automatic Detection and Visualization of Distinctive Structures in 3D Unsteady Multi-Fields. *Computer Graphics Forum*, 27(3):767–774, 2008.

4 Heike Jänicke and Gerik Scheuermann. Measuring complexity in lagrangian and eulerian flows. *Under review*, 2009.

5 Heike Jänicke and Gerik Scheuermann. Steady visualization of the dynamics in fluids using ϵ-machines. *Computers and Graphics*, 2009.

6 Heike Jänicke and Gerik Scheuermann. Visual analysis of flow features using information theory. *Conditionally accepted for the special issue on knowledge-assisted visualization in Computer Graphics and Applications*, 2009.

7 Heike Jänicke, Alexander Wiebel, Gerik Scheuermann, and Wolfgang Kollmann. Multifield Visualization Using Local Statistical Complexity. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1384–1391, 2007.

Construction of Implicit Surfaces from Point Clouds Using a Feature-based Approach

Patric Keller[1], Oliver Kreylos[3], Eric S. Cowgill[3], Louise H. Kellogg[3], Martin Hering-Bertram[4], Bernd Hamann[2], and Hans Hagen[1]

1 **Department of Computer Science,**
University of Kaiserslautern, Germany.
2 **Institute for Data Analysis and Visualization (IDAV),**
Department of Computer Science, UC Davis, CA, USA.
3 **Department of Geology, UC Davis, CA, USA.**
4 **Fraunhofer ITWM in Kaiserslautern, Germany.**

Abstract

We present a novel feature-based approach to surface generation from point clouds in three-dimensional space obtained by terrestrial and airborne laser scanning. In a first step, we apply a multiscale clustering and classification of local point set neighborhoods by considering their geometric shape. Corresponding feature values quantify the similarity to curve-like, surface-like, and solid-like shapes. For selecting and extracting surface features, we build a hierarchical trivariate B-spline representation of this surface feature function. Surfaces are extracted with a variant of marching cubes (MC), providing an inner and outer shell that are merged into a single non-manifold surface component at the field's ridges. By adapting the iso-value of the feature function the user may control surface topology and thus adapt the extracted features to the noise level of the underlying point cloud. User control and adaptive approximation make our method robust for noisy and complex point data.

1998 ACM Subject Classification I.3 Computer graphics, I.3.5 Computational Geometry and Object Modeling

Keywords and phrases 3D Point Clouds, Surface Reconstruction, Implicit Surfaces;

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.129

1 Introduction

The exploration of point data sets, like environmental LiDaR (Light Detection and Ranging) data, is a challenging problem of current interest. Besides the vast amount of points obtained from terrestrial and airborne scanning, data complexity is increased by e.g., noise, occlusion, alternating sample density and overlapping samples. Despite of the high scanning resolution, undersampling occurs at small and fractured components like fences and leaves of trees.

In this research we introduce a new surface reconstruction approach that allows to recover surface meshes from relative complex point data sets. Since most reconstruction methods do not distinguish between surface and non-surface related point structures, they rather assume the input points always originate from surface structures, applying them to noisy and complex point data would not lead to reliable results. To perform a reliable surface reconstruction a feature-based reconstruction approach taking into account the structural composition of the acquired point cloud is proposed. It adapts the mesh reconstruction in a way that only surface-related point structures are extracted. This is accomplished by

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 129–143
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

first performing a structural analysis of the point cloud that filters the point data based on a classification of local point set neighborhoods. The feature values assigned to the individual points are used to define whether a point corresponds to a surface or not. The proposed approach constructs a *feature field* function approximating these feature values from which an initial surface representation is extracted. The locus of points associated with the local extrema of this implicit “feature function“ most probably represents the underlying surface structures. Thus, moving the initial surface toward these local extrema, incorporating operations adapting the topology of the surface meshes, yields the final surface approximation. Since the proposed approach is meant to complement the visualization of raw laser-scanned data, it provides a mesh-representation exhibiting a user specified topology. The topology can be controlled, similar to choosing a radius for alpha shapes in 3D space [11], in a way that the extracted surfaces adapt to the point cloud specific properties. A further advantage is that the extracted surface representation is continuous and thus can be used for further post-processing.

The remainder of this paper is structured as follows: Section 3 provides an overview about the steps of the proposed method. In Section 4, Section 5, Section 6 and Section 7 we discuss the individual steps of the reconstruction approach in detail. The results are presented in Section 8. The conclusions are provided in Section 9.

2 Related Work

Many methods exist for the construction of surfaces from scattered point data. The existing methods can be divided into explicit and volume-based reconstruction methods.

2.1 Volume-based (implicit) surface reconstruction

These reconstruction techniques transfer the input point cloud into volumetric representations implicitly defining the underlying surfaces. Using methods like the marching cubes algorithm (MC) [21] enables us to extract explicit surfaces representations in form of polygonal meshes. One of the first reconstruction methods within this field was introduced by Hoppe et al. [16]. The basic idea is to combine local signed distance functions (SDF) into a global representation from which the surface can be extracted by contouring. Following this idea the methods discussed in [33], [5] and [24] are based on defining SDFs whose zero level sets represent the desired surfaces. More recently, scattered data approximation with radial basis functions (RBF) is described in [30] [7] [6] to construct SDFs. Othake et al. [26] extended the original RBF construction to multi-scale RBFs increasing approximation quality. Shen et al. [29] improved the original method by incorporating additional normal information in the RBFs allowing a more stable reconstruction. Another possible technique for surface reconstruction concerns the usage of level set methods [31] [33]. This technique defines a function attracting the level set surface toward the data points. However, the application of these techniques is limited when working with noisy unfiltered point clouds like environmental LiDaR data.

2.2 Explicit surface reconstruction

Another possible way to obtain surfaces from unorganized point clouds is to fit parametric surface patches to the input points. The approach of Eck and Hoppe [10] fits a network of B-Spline patches in a way allowing them to reconstruct surfaces of arbitrary topology. Alexa et al. [2] introduced an approach based on moving least squares (MLS) also used in [1] [25] [12] [9] which fits polynomial surface patches to local point neighborhoods. In [22] the authors propose an MLS reconstruction scheme which takes into account local curvature approximations to obtain meshes of higher quality. One of the main problems associated with the MLS-based techniques is that they have to adapt in order to capture the correct underlying topology. Depending on the structure of the input data this can be hard to achieve. Another drawback is that the MLS-technique in general is not capable of constructing surfaces having sharp features. One approach for solving this problem was published by Fleishman et al. [12]. Alternative methods like the power crust (PC) algorithm introduced by Amenta et al. [3] the 3D Alpha Shapes method from [11], the Cocone algorithm [8] as well as some other approaches [23] [17] also produce good results concerning surface reconstruction from point data.

Most of the existing techniques make the assumption that the point cloud exclusively contains points corresponding to surface-like structures. In cases the point cloud includes objects having non-surface character, like trees or bushes as is the case in environmental point clouds featuring vegetation, most of the existing methods would fail. We propose an approach able to extract surfaces even from these types of point clouds. We do not make any assumptions on the structural composition of the point clouds in advance. This allows us to apply our method to nearly any type of point data. The provided results suggest that our method holds great promise in the field of surface reconstruction from unorganized environmental point cloud data.

3 Overview

The proposed method aims at generating an implicit representation of the underlying surface(s) in form of a 3D scalar field, named feature field. From this feature field the final surface representation is extracted using a MC-based approach. The proposed feature-based surface reconstruction consists of the following steps:

1. Pre-processing. This step includes the organization of the input points into an octree-based voxel structure as well as the determination and assignment of feature values to the individual points. These feature values in a certain sense express the likelihood that a point is part of a surface structure by measuring the ”planarity” of a point’s local neighborhood.
2. The approximation/interpolation of the computed feature values using 3D (trivariate) hierarchical B-splines provides the scalar-valued feature field function.
3. The extraction of an initial surface mesh, associated with an user-specified iso-value by using the MC-algorithm.
4. The adaption of the initial surface representation to approximate the meta-surface which represents the local maxima of the corresponding feature field. This includes additional refinement strategies increasing the quality of the final surface representation.

4 Pre-Processing

4.1 Hierarchical Point Cloud Decomposition

For purposes of efficient data processing the proposed reconstruction method applies the octree-based data organization scheme. Considering the applied surface reconstruction algorithm, this form of data management has several advantages: Besides the efficient handling and visualization of point clouds as well as the acceleration of certain point operations like k-nearest neighbor searches, it facilitates the construction of the feature field function since it makes possible applying the hierarchical B-spline approximation scheme. It also supports a straightforward MC-based iso-surface extraction by incorporating the edges of the multi-level voxel grid;

4.2 Determining Feature Values

The determination of the feature values expressing the correspondence of individual points to surfaces is based on a former approach described in [18]. The feature values are defined in a way that they measure the "planarity" of the local neighborhood $N(p)$ of a point $p \in P$, whereas $N(p)$ either can be defined by the set of k-nearest neighbor points of p, or the set of points lying within a specified radius r of p. In principle, we want the feature value of p to be high, if $N(p)$ corresponds to a smooth surface and low otherwise. This is achieved by analyzing the geometric shape of $N(p)$ by relating the eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_2$ of the covariance matrix $\mathbf{C}$ of $N(p)$ amongst each others. It holds that $N(p)$ has a spherical shape if $\lambda_0 \approx \lambda_1 \approx \lambda_2$, a planar shape if $\lambda_0 \ll \lambda_1 \wedge \lambda_1 \approx \lambda_2$ and a cylindrical shape if $\lambda_0 \approx \lambda_1 \wedge \lambda_1 \ll \lambda_2$. We combine this information and define the surface feature value f of a point p to be

$$f(p) = \frac{\lambda_1 - \lambda_0}{\lambda_2}. \tag{1}$$

The size of $N(p)$ has major impact on the obtained value f and is an important influence factor controlling the quality of the final surface representation. Choosing $N(p)$ to be small compared to the noise ratio or the point density results in strongly varying feature field interfering with the reconstruction process. Forcing $N(p)$ to be too large will degrade the quality of the extracted surface. This effect is discussed more detailed in Section 8.

5 Scattered Data Approximation

We consider the approximation of the introduced feature values $f_i \in \mathbb{R}$ assigned to the input points $p_i \in P \subset \mathbb{R}^3$. Obtaining a smooth global feature function F approximating the given feature values f_i becomes a scattered data problem. This problem can be formalized as follows: Given a finite set of scattered data points $\{(p_i, f_i)|p \in \mathbb{R}^3, f_i \in \mathbb{R}\}$, construct a function $F : \mathbb{R}^3 \to \mathbb{R}$ that minimizes the residuals $\|F(p_i) - f_i\|$. There are several possible ways for accomplishing this task like using approximation schemes based on, e.g., RBFs [13] [27] or spline fitting techniques [19] [15]. Good overviews of existing scattered data interpolation/approximation techniques are provided in [14] [20] [27] [28]. The introduced method follows the ideas presented in [4] [32] using hierarchical B-splines defined over the rectangular regions of a corresponding quadtree. This approach is able to provide a smooth and adaptive approximation of the feature field by a set of hierarchical trivariate B-spline functions. Extended to the 3D case, starting from the original bounding voxel enclosing the entire point cloud, the approximation scheme consists of the following steps:

1. The initial approximation of the determined feature values associated with the individual points by triquadratic Bernstein-Bézier polynomials.
2. Joining the individual Bernstein-Bézier polynomials into one continuous B-spline representation by applying the knot removal scheme proposed by Bertram et al. [4].
3. The refinement of the approximation by performing above steps to the voxel grid at the next finer octree level.

Recursively applying these steps provides a hierarchical approximation of the feature values.

5.1 Initial Bézier Fitting

Let $P_i = \{(p_j, f_j)\}$ be a set of n scattered data points associated with the voxel V_i. P_i is approximated by a trivariate Bernstein-Bézier polynomials of the form

$$F_i(p_j) = \sum_{i=0}^{l}\sum_{j=0}^{l}\sum_{k=0}^{l} c_{i,j,k}\phi^l_{i,j,k}(p_j), \tag{2}$$

where $\phi^l_{j,j,k}(p_j)$ are the corresponding trivariate basis functions of degree l. The solution for F_i is found by solving Equation (5.1) according the basis coefficients $c_{i,j,k}$. In the case that an overdetermined system is given, F_i is determined by minimizing the sum of the squared residuals φ with respect to the basis coefficients $c_{i,j,k}$

$$\varphi = \sum_{j=0}^{n} \|F_i(p_j) - f_j\|^2. \tag{3}$$

To deal with an under-determined system we add additional constraints forcing F_i to become zero in regions where no scattered data samples are given. The additional term has the form

$$\psi = \sum_{i=0}^{l}\sum_{j=0}^{l}\sum_{k=0}^{l}(c_{i,j,k})^2\ . \tag{4}$$

Minimizing

$$\varepsilon = \varphi + \psi \longrightarrow min \tag{5}$$

requires solving a relatively small linear system of equations yielding the desired function F_i. Since we are interested in obtaining a continuous representation we apply further processing.

5.2 Global B-spline Approximation

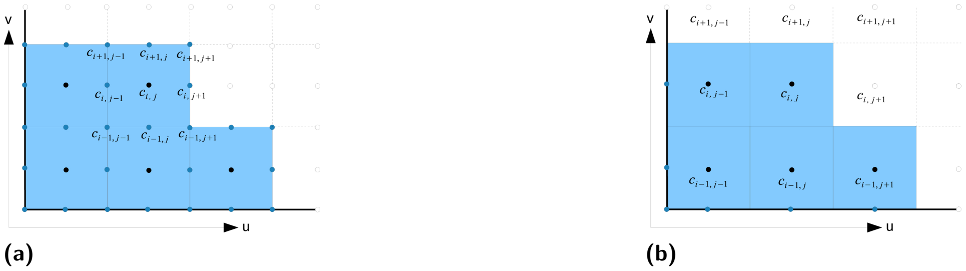


Figure 1 (a) Initial distribution of the basis coefficients defining corresponding biquadratic B-spline functions within an adaptive quad-grid consisting of empty and non-empty quadrilateral. Blue points represent zero-valued basis coefficients located at the domain boundaries of the data set having knot multiplicity three, black and white points are basis coefficients having knot multiplicity one. The white points are zero-valued basis coefficients of empty quadrilaterals. (b) quad-grid after knot removal.

To obtain a continuous approximation, we merge the individual Bernstein-Bézier polynomials F_i into a global B-spline representation F. The Bernstein-Bézier polynomials of degree l associated with the individual voxels can be represented as one global piecewise continuous B-spline function in which the control points at the voxel boundaries (faces, edges, vertices) have multiplicity l. The multiplicity of the knots associated with control points at the inside of a voxel is one. Figure 1a shows a 2D example that illustrates the distribution of the control points corresponding to a biquadratic B-spline over a quadrangular domain. The grid consists of empty (white) and non-empty (blue) quads. If a basis coefficient is located on the domain boundary of the point data set, its value is forced to be zero. The domain boundary of the data set in the 3D case is defined by the vertices, the edges and the faces of the smallest voxel enclosing the entire point set. In regions where voxels are classified as empty the control points are set to zero (depicted white). Removing knots at the voxel boundaries that are not located on the domain boundaries, yields the desired B-spline representation (see Figure 1b). In the triquadratic case the resulting B-spline is C^1 continuous. The knot removal may affect the quality of the approximation and thus demands re-fitting the remaining coefficients $c_{i,j,k}$. This is accomplished by sequentially readjusting each coefficient

$$c_{i,j,k} = \frac{\sum_{p \in V_i} \left(f_p - \sum_{x=0,x\neq i}^{l} \sum_{y=0,y\neq j}^{l} \sum_{z=0,z\neq k}^{l} c_{x,y,z} \phi_{x,y,z}(\vec{p}) \right)}{\sum_{p \in V_i} \phi_{i,j,k}(\vec{p})}, \tag{6}$$

where $\vec{p} \in \mathbb{R}^3$ denotes the coordinates of the points $p \in V_i$ in parameter space, and $f_p \in \mathbb{R}$ terms the feature value associated with p. Each adjustment of $c_{i,j,k}$ again has impact on the quality of the approximation in the neighbor voxel. Hence, the re-fitting becomes an iterative process. The process is terminated if the local approximation error $|\sum_{p \in V_i} F(p) - f_p|$ falls below a given error bound or a maximum number of iterations has been performed. The resulting function F represents a continuous approximation of the feature field function getting zero at the point clouds domain boundaries.

Problems arise in cases, where the least squares residual is small, but the slopes or curvatures of the trivariate B-spline functions are extremely high. This, especially, occurs in regions where the distribution of the points or the function values is one-sided. For example, in ”empty” voxel regions the function values may take arbitrarily large values which results in a poor representation. To encounter this problem, additional constraints are incorporated forcing the final representation to be smooth. The enhanced approximation scheme incorporates the minimization of the first principal derivatives of the B-spline representation F at the positions of the individual points.

5.3 Recursive Refinement

Let $L_j = \{(P_i^j, F_i^j), \cdots\}$ be a set where P_i^j are the scattered point sets associated with the individual voxels at octree level j and F_i^j be the corresponding approximating B-spline functions. Starting at octree level j we determine F_i^j for each $P_i \neq \emptyset$. The subsequent merging of adjacent functions F_i^j is performed by applying the above introduced knot removal. This leads to a globally C^1 continuous representation F^j at level j. In order to obtain L_{j+1} we recompute the feature values

$$\Delta f^{j+1}(p) = \Delta f^j(p) - F^j(p) \tag{7}$$

for each point, where $\Delta f_p^0 = f_p$ are the initial feature values assigned to p. This yields the basis for the subsequent approximation step performed at level $j+1$. Again, applying the least squares fitting procedure and merging the individual Bernstein-Bézier polynomials at level $j+1$ results in the continuous representations ΔF_i^{j+1} of the underlying residuals Δf_i^{j+1}. The final approximation is defined by

$$F^{j+1} = F^j + \Delta F^{j+1}. \tag{8}$$

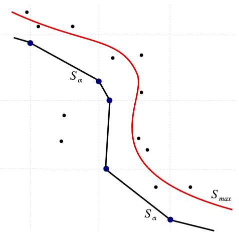


Figure 2 Two-dimensional example illustrating the extraction of the base mesh S_α via the MC algorithm regarding a specified iso-value α. The locus of the local maxima of F to be approximated by S_α is represented by the curve S_{max}.

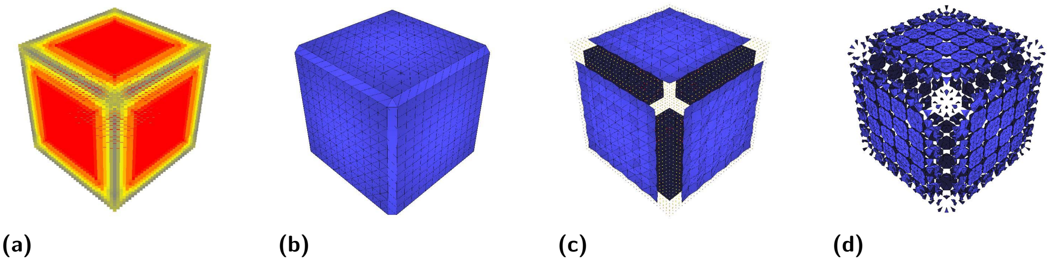


Figure 3 Example data set illustrating the effects of varying the iso-value at the initial extraction procedure. (a) Input point cloud of a simple test cube data set with points colored from blue to red according the correspondence of points to surface-related structures. (b)-(d) extracted base meshes obtained for iso-values (b) 0.4, (c) 0.6 and (d) 0.9.

The choice of the level at which the approximation is started and terminated has major impact on the performance and the quality of the final approximation.

For example, starting the computation at a low-resolution level e.g., on the root of the octree, and performing a large number of refinement steps would be computationally expensive. Otherwise, performing too few refinement steps may result in low-quality approximations. The B-spline refinement can be stopped, if a specific maximal refinement level is reached or the approximation error $|\sum_{p \in V_i} F^j(p) - f_p|$ approaches a predefined limit.

6 Initial Surface Extraction

Given a continuous function F approximating or interpolating the feature values assigned to the individual points, the first step aims at the extraction of a base mesh from F using the MC-algorithm [21]. The quality of the extracted mesh depends on the resolution of the voxel as well as on the quality of F. In general, the MC algorithm generates two-manifold meshes reflecting the topology of the iso-surface at the given extraction level (see Figure 2). Varying the iso-value results in base meshes having different topological characteristics. At this point the user has to control the surface topology by choosing the iso-value, such that the extracted mesh reflects the desired topology. This approach is inspired by the alpha-shape reconstruction method by Edelsbrunner and Mücke [11], where a Delaunay-complex is constructed for a point set, erasing all simplices whose circumscribed hypersphere exceeds a predefined radius alpha. Figure 3 shows some examples regarding a simple test data set. It illustrates the variation in topology, for extracting an initial surface from a precomputed feature function F, by varying the iso-value from low to high.

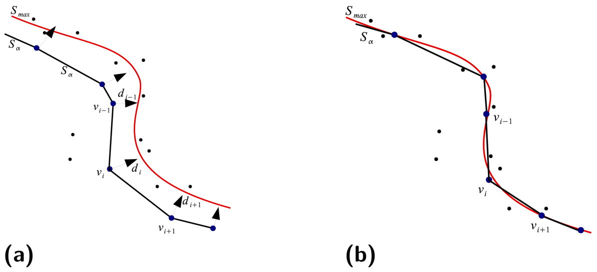


Figure 4 Examples illustrating the principles of approximating the meta-surface $S_{\max}$ representing the locus of local maxima of F. (a) Initial approximation S_α with vertices v_i and corresponding gradient vectors d_i, before the vertex displacement is applied. (b) Resulting mesh after the displacement of the vertices v_i along the vectors d_i toward S_{max}.

7 Approximation of Local Maxima

Next, the generated base is adapted to approximate the meta-surface representing the locus of local maxima of F. The set of points associated with this meta-surface is defined by $S_{\max}$. In a sense, $S_{\max}$ represents ”ridge-lines” within the feature field.

The locus of the local maxima of F is defined by

$$S_{max} = \{p \in \Omega | \nabla F(p) = 0 \wedge \nabla^2 F(p) < 0 \wedge F(p) > \alpha\}, \tag{9}$$

where $\Omega \subseteq \mathbb{R}^3$ defines the support domain of F defined by the bounding voxel enclosing the point cloud and α denotes the iso-value chosen for the the initial base mesh extraction.

Depending on the topological characteristics of F the subsets of S_{max} can be associated with surfaces, lines or isolated points. As mentioned above, the function F approximates feature values assigned to the points P within the interval $[0, 1]$ where the value 1 (0) is associated with points assumed to be represent surface (non-surface) related structures. Based on this, the surface related subsets of S_{max} most probably represent the point clouds inherent surfaces. The following procedure is intended to approximate these structures. Using the extracted base mesh, the meta-surface $S_{\max}$ is obtained by iteratively applying the following steps:

1. We move the mesh toward the local maxima of the feature field F by displacing the mesh vertices (cf. Figure 4).
2. In addition, an adaptive strategy for updating the mesh connectivity, to improve the quality of the final representation, is applied.

7.1 Vertex Displacement

Moving the mesh toward S_{max} is achieved by displacing the vertices along the vector field ∇F. Let $\vec{d}$ be the corresponding displacement vector of vertex v, then v is relocated by iteratively moving v along $\vec{d}$ toward S_{max}. Since F approximates the local trend of the feature values assigned to points distributed in 3D space, the maximum values of F most probably occur in regions the corresponding point cloud has high feature values. Hence, projecting the vertices along $\vec{d}$ into these regions is obvious. The new position $\vec{v}_{new}$ of v is defined by

$$\vec{v}_{new} = \vec{v} - \vec{d} \cdot (\vec{d} \cdot (\vec{v} - \vec{c})), \tag{10}$$

where $\vec{v} \in \mathbb{R}^3$ denotes the position of vertex v, and $\vec{c}$ represents the weighted centroid of the proximate point neighborhood $N(v)$ of v determined by

$$\vec{c} = \sum_{p \in N(v)} \frac{f_p}{\sum_{q \in N(v)} f_q} \vec{p}, \tag{11}$$

where f_p denotes the feature value assigned to the point p. The size of the neighborhood $N(v)$ is determined by the radius of the one-star neighborhood of vertex v. The one-star neighborhood of v is given by all vertices that are directly connected to v. Let r be the range to the most distant vertex corresponding to the one-star neighborhood of v, then $N(v)$ is defined to be $N(v) = \{p \in \mathbb{R}^3 | \|v - p\| < r\}$.

7.2 Adapting Mesh Connectivity

This step becomes necessary when triangles corresponding to different mesh regions converge or intersect each other. To obtain the final surface representation we collapse vertices of affected triangles. The collapsing criterion depends on the size of the involved triangles. Collapsing two vertices v and w, not connected, is allowed, if the following condition holds:

$$\|\vec{v} - \vec{w}\| < r_{\min}, \tag{12}$$

where $r = min\{\cdots, r_i, \cdots\}$ with r_i representing the radius of the smallest sphere enclosing the triangle $t_i \in T_v \cup T_w$. The sets T_v and T_w consist of the triangles connected to v and w. If condition (12) is satisfied, the vertices v and w are collapsed and substituted by a new vertex v_{new}. The position of v_{new} is computed by $\vec{v}_{new} = \frac{1}{2}(\vec{v} + \vec{w})$. The collapsing operation takes into account additional rules preventing the mesh to break or to collapse. Additional operations are incorporated maintaining the mesh consistence e.g., by removing duplicated triangles. As a consequence of the vertex collapse, the resulting mesh is no longer guaranteed to be two-manifold.

After repositioning the vertices the resulting mesh might not approximate S_{max} in an optimal way. One possible reason is that the resolution of the mesh in some regions is too coarse. Subdividing selected triangles introduces new vertices at the centers of the triangles. Again, forcing these vertices toward $S_{\max}$ further increases the quality of the surface mesh. The decision whether a triangle is subdivided or not depends on the error measure

$$\varepsilon = \left(\frac{\overline{f} - f_c}{\|\nabla F_c\|} \right)^2, \tag{13}$$

where $\overline{f}$ denotes the average feature value of the vertices of the considered triangle, f_c and ∇F_c represent the feature value and the gradient at the triangles center. We allow subdividing a triangle if $\varepsilon > r^2$, with r being the radius of the smallest sphere enclosing the corresponding triangle. Subdividing triangles introduces new vertices w at the center of the triangles. Those vertices, in a subsequent step, are moved toward S_{max} following the rules of vertex displacement (cf. Section 7.1). On the other hand, triangles are collapsed if they have a bad edge-to-edge aspect ratio.

Since the subsets of S_{max} in some cases represent points or curves, it may happen that the vertex displacement forces the mesh to collapse into point- or a curve-like structures. In this case the corresponding mesh structures are deleted.

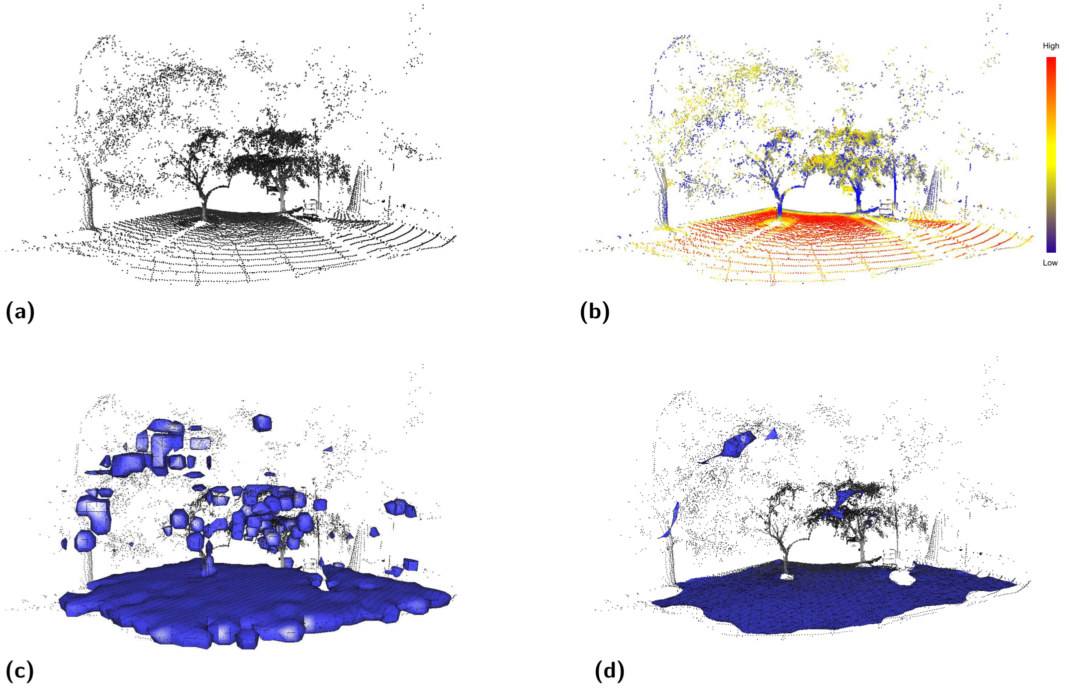
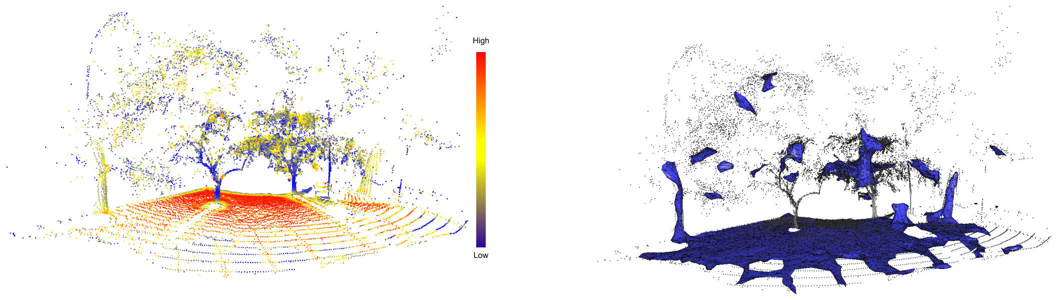


Figure 5 (a) original point cloud showing an environmental scene. (b) feature values, computed by the applied point classification scheme, which are mapped in form of color values to the corresponding points (red indicates surface-related, blue non-surface related points). (c) reconstructed initial mesh, (d) final surface representation.

8 Results

This section provides some results obtained by applying the proposed method to different point data sets. The used data sets are well-suited to demonstrate the ability of the method to reconstruct surfaces from complex and noisy point data sets.

Figure 5a presents the raw input point cloud showing an environmental scene (having about 50,000 points) consisting of quite complex structures. Applying the proposed point classification scheme delivers feature values for each point. The classification was performed using the local point neighborhood $N(p)$ determined by all points lying within a given range r of the considered point p. In the following, r is specified in terms of percentage of the diagonal of the smallest sphere enclosing the entire point cloud. Figure 5b visualizes these values by mapping colors ranging from blue to red to the points. Red colors indicate points in near-planar regions, whereas blue colors indicate points that are not assumed to be part of surfaces. In this example r was chosen to be 0.02. As the image shows, mainly the points at the ground are identified as surface-related structures, but also some isolated point structures at the crowns. This is due to the fact that the local neighborhoods of the isolated points are rather planar and thus the points misleadingly are assumed to lie on a surface. Figure 5c shows the base mesh extracted for the iso-value 0.6. Besides capturing the ground structure the initial mesh also captures the identified structures at the crowns. Figure 5c shows the final mesh after the application of the vertex displacement and the adaption of the mesh connectivity. The ground surface is almost completely recovered, except for a small part located at the tree trunk in the middle. Since the point cloud does not exhibit surface-like



(a) **(b)**

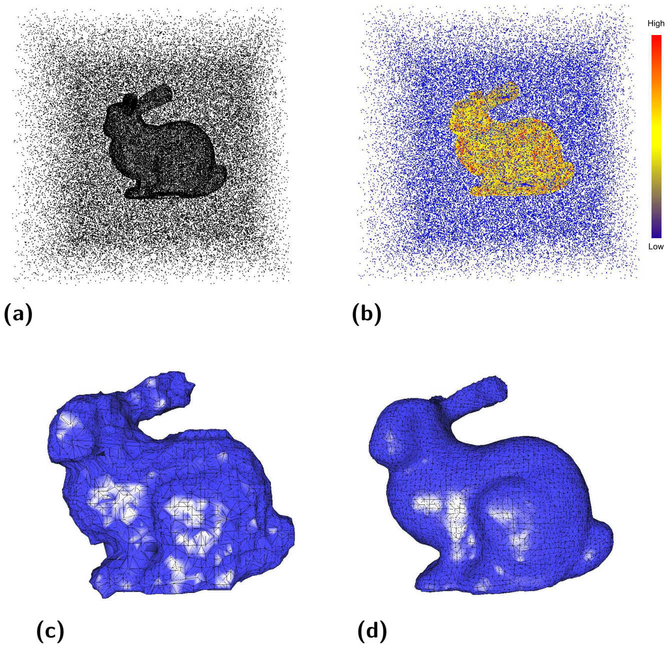
Figure 6 (a) Visualization of the feature values for a chosen neighborhood size $r = 0.016$. (b) the final surface representation.

character at the area at which the trunk merges with the ground, the local point environment rather has cylindrical or spherical character, the affected region is not captured by the final surface.

The reconstruction process strongly depends on the choice of the neighborhood size considered for performing the point classification. Figure 6a demonstrates the effect of choosing the neighborhood size to be to small with respect to the density of the points located at the ground. In this example the radius determining the neighborhood $N(p)$ was chosen to be $r = 0.016$. The generated surface mesh, depicted by Figure 6b, does not capture the entire ground. The points in the front at which the feature values are low (blue colors) are not approximated by the surface. Due to the small neighborhood size chosen the points are not assumed to be part of the ground surface and thus not part of the final surface mesh.

In both examples the chosen parameters are equal, except for r. The B-spline based approximation of the feature values was started at octree level five and terminated at level ten. The Cartesian grid used for the MC-based extraction was determined by partitioning the original bounding voxel into $40 \times 40 \times 40$ cubes.

The next example demonstrates the stability of the reconstruction approach by applying it to a modified point set. Figure 7a shows the point cloud of the Stanford bunny data set where noise is added in form of randomly distributed point samples. In this case, the boundary voxel, considered for adding the additional samples, is centered at the centroid of the point cloud and has the diagonal length of two times the length of the most distant point to the centroid. The number of additional noise samples in this example equals $80,000$ points, the original number of points of the bunny data set is approximately $35,000$. Figure 7b shows the color-coded point cloud indicating the computed feature values. Again, blue represents non-surface like, red represents surface like point structures. The chosen neighborhood size equals $r = 0.01$. It is observable that the classification process identifies most points corresponding to the original point cloud. Applying the approximation scheme followed by the MC mesh extraction delivers the initial base mesh depicted in Figure 7c. The approximation was started at level six and was terminated at octree level ten. Here, the cube grid used for the MC-based mesh extraction was determined by partitioning the original bounding voxel into $80 \times 80 \times 80$ cubes. The final mesh after performing several repositioning and smoothing steps is depicted in Figure 7d. This example shows that the method provides reliable results in cases the point cloud becomes extremely noisy. Several experiments have been performed using an increased number of additional noise samples. The reconstruction provides acceptable results up to $100,000$ noise samples. Theoretically, the reconstruction is



(a) **(b)** **(c)** **(d)**

Figure 7 (a)-(d) examples illustrating the surface reconstruction from a point cloud enriched with $80,000$ noise samples. (a) original point cloud containing the Stanford bunny data set consisting of approx. $35,000$ points, (b) point cloud featuring color values indicating the estimated strength of the correspondence of points to surface-related structures (blue indicates low, red indicates high correspondences). (c) the extracted base mesh, (d) the final surface mesh.

possible as long as the density of the points, corresponding to underlying surfaces, locally is higher than the density of the proximate noise samples. However, if the noise samples form surface-related point structures close to the original point samples the reconstruction might fail at these regions.

The average computational complexity of the proposed algorithm is estimated to be $O(k^2 \cdot n \cdot log(n))$, where k is the average number of points within $N(p)$ and n is the number of input data points. In case that $k << n$ the complexity $O(n \cdot log(n))$ rather holds. The complexity of the mesh extraction is estimated to be $O(m \cdot log(n))$ and primarily depends on the number of extracted mesh vertices m and the number of input points. Computationally most expensive are the approximation of the feature values by F and the vertex displacement. Hence, it was decided to perform the determination of feature values and their approximation in a pre-processing step. This allows the user to perform the actual surface reconstruction more efficiently. Table 1 presents the computation times for the results depicted by Figure 5 and Figure 7. The processing was performed on a desktop computer equipped with an Intel Core2 Duo Processor and 4GB main memory.

9 Conclusions

We have presented a novel approach to feature classification and selective surface reconstruction from large scale point clouds. Our method is used to complement visualization of raw data sets obtained from terrestrial and airborne laser scanning. The main problem when visualizing such vast amounts of points is that the user may not recognize structures of interest

Table 1 Computation times for each step of the algorithm concerning the two example data sets. The approximation was performed using four iterations at each processed octree level.

Computation step	Time [sec.] environmental data set (50,000 points)	Time [sec.] modified bunny data set (115,000) points
Computation of feature values	46.95	72.34
Performing approximation	145.11	1107.84
Extracting base mesh	13.2	141.39
Vertex displacement (1st iteration)	7.65 (3,900 vertices)	415.12 (26,000 vertices)

without additional visual cues. Therefore, we classify point neighborhoods into different feature groups by assigning corresponding feature values. These values assigned to surface features are approximated by a hierarchical trivariate B-Spline function which is the basis of our surface reconstruction approach. Topological and geometric detail can be controlled by a user-defined parameter. Future work will be directed at choosing these parameters locally and accelerating the approach further by more sophisticated data structures for streaming and view-dependent querying.

Acknowledgements

This work was supported by the German Research Foundation (DFG) through the International Research Training Group (IRTG) 1131, the University of Kaiserslautern and the Center for Mathematical and Computational Modeling $(\mathrm{CM})^2$. It was also supported in part by the W.M. Keck Foundation that provided support for the UC Davis Center for Active Visualization in the Earth Sciences (KeckCAVES), Department of Geology. We thank the members of the Visualization and Computer Graphics Research Group at the Institute for Data Analysis and Visualization (IDAV) at UC Davis.

References

1 M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3—15, 2003.

2 Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Point set surfaces. In *Conference on Visualization*, pages 21–28. IEEE Computer Society, 2001.

3 Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust. In *6th ACM symposium on Solid Modeling and Applications*, pages 249–266. ACM Press, 2001.

4 M. Bertram, X. Tricoche, and H. Hagen. Adaptive smooth scattered-data approximation for large-scale terrain visualization. In *Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization*, pages 177–184, 2003.

5 J. Bloomenthal, C. Bajaj, J. Blinn, M.-P Cani-Gascuel, A. Rockwood, B. Wyvill, and G. Wyvill. *Introduction to implicit surfaces.* Morgan KAufman, Inc., San Francisco, 1997.

6 J.C. Carr, R.K. Beatson, J.B. Cherry, T.J. Mitchel, W.R. Fright, B.C. McCallum, and T.R. Evans. Smooth surface reconstruction from noisy range data. In *Proc. SIGGRAPH Computer Graphics*, 2001.

7 J.C. Carr, R.K. Beatson, B.C. McCallum, W.R. Fright, T.J. McLennan, and T.J. Mitchel. Smooth surface reconstruction from noisy range data. In *Proc. ACM GRAPHITE*, pages 119–126, 2003.

8 T. K. Dey and S. Goswami. Tight cocone: A water tight surface reconstructor. In *Proc. 8th ACM Sympos. Solid Modeling Appl.*, pages 127–134, 2003.

9 Tamal K. Dey and Jian Sun. An adaptive mls surface for reconstruction with guarantees. In *3rd Eurographics Symposium on Geometry Processing*, pages 43–52, 2005.

10 M. Eck and H. Hoppe. Automatic reconstruction of b-spline surfaces of arbitrary topological type. In *ACM Siggraph*, pages 325–334, 1996.

11 Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

12 S. Fleishman, D. Cohen-Or, and C.T. Silva. Robust moving least-squares fitting with sharp features. In *ACM Siggraph*, pages 544–552, 2005.

13 T.A. Foley. Scattered data interpolation and approximation with error bounds. In *In Computer Aided Geometric Design*, pages 163–177, 1986.

14 R. Franke. Scattered data interpolation: Test of some methods. *In Mathematics of Computation*, 38(157):181–200, 1982.

15 B.F. Gregorsky, B. Hamann, and K.I. Joy. Reconstruction of b-spline surfaces from scattered data points. In *Computer Graphics International 2000*, pages 163–170, 2000.

16 H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *ACM SIGGRAPH '92 Conference Proceedings*, pages 71–78, 1992.

17 P. Keller, M. Bertram, and H. Hagen. Reverse engineering with subdivision surfaces. In *Computing 2007*, pages 127–134, 2007.

18 Patric Keller, Oliver Kreylos, Marek Vanco, Martin Hering Bertram, Eric S. Cowgill, Louise H. Kellogg, Bernd Hamann, and Hans Hagen. *Extracting and Visualizing Structural Features in Environmental Point Cloud LiDaR Data Sets (to be published).*

19 S. Lee, G. Wolberg, and S.Y. Shin. Scattered data interpolation with multilevel b-splines. *IEEE - Transactions on Visualization and Computer Graphics*, 3(3):228–244, 1997.

20 S.K. Lodha and R. Franke. Scattered data techniques for surfaces. In *Proceedings Dagstuhl Conf. Scientific Visualization*, pages 182–222, 1999.

21 William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

22 B. Mederos, L. Velho, and L. H. de Figueiredo. Moving least squares multiresolution surface approximation. In *Sibgraphi*, 2003.

23 L. V. Boris Mederos, N. Amenta, L. Velho, and L.H. de Figueiredo. Surface reconstruction from noisy point clouds. In *Eurographics Symposium on Geometry Processing*, 2005.

24 S. Muraki. Volumetric shape description of range data using "blobby model". In *ACM SIGGRAPH Proceedings in Computer Graphics*, pages 227–235. ACM Press, 1991.

25 A. Nealen. An as-short-as-possible introduction to the least squares, weighted least squares and moving least squares methods for scattered data approximation and interpolation. In *Technical Report, TU Darmstadt*, 2004.

26 Y. Othake, A. Belyaev, and H.-P. Seidel. A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. In *Shape Modeling Interantional 2003*, pages 153–161, 2003.

27 M.J.D. Powell. Radial basis functions for multivariate interpolation. In *Algorithms for Approximation of Functions and Data*, pages 143–168, 1987.

28 L.L. Schumaker. Fitting surfaces to scattered data. In *Approximation Theory II*, pages 203–268, 1976.

29 C. Shen, J.F. O'Brian, and J.R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *In Proc. of ACM SIGGRAPH 2004*, pages 896–904, 2004.

30 Greg Turk and James F. O'Brien. Shape transformation using variational implicit functions. In *The Proceedings of ACM SIGGRAPH 99*, pages 335–342, 1999.

31 R. Whitaker. A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, 1998.

32 W. Zhang, Z. Tang, and J. Li. Adaptive hierarchical b-spline surface approximation of large-scale scattered data. In *In Proc. Pacific Graphics '98*, pages 8–16, 1998.

33 Hong-Kai Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. In *IEEE Proceedings in Variational and Level Set Methods in Computer Vision*, pages 194–201. IEEE Computer Society, 2001.

Framework for Comprehensive Size and Resolution Utilization of Arbitrary Displays

Taimur Khan, Daniel Schneider, Yasmin I. Al-Zokari, Dirk Zeckzer, and Hans Hagen[1]

1 **University of Kaiserslautern**
67653 Kaiserslautern, Germany
`{tkhan,d_schnei,alzokari,zeckzer,hagen}@informatik.uni-kl.de`

Abstract

Scalable large high-resolution displays such as tiled displays are imperative for the visualization of large and complex datasets. In recent times, the relatively low costs for setting up large display systems have led to an highly increased usage of such devices. However, it is equally vital to optimally utilize their size and resolution to effectively explore such data through a combination of diverse visualizations, views, and interaction mechanisms. In this paper, we present a lightweight dispatcher framework which facilitates input management, focus management, and the execution of several interrelated yet independent visualizations. The approach is deliberately kept flexible to not only tackle different hardware configurations but also the amount of visualization applications to be implemented. This is demonstrated through a scenario that executes four interrelated visualizations equally well on both a 5 PC tiled-wall and a single desktop. The key contribution of this work is the ability to extend the tiled-wall to work with multiple applications for enhanced size and resolution utilization of such displays.

1998 ACM Subject Classification I.3 Computer graphics, I.3.3 Picture and Image Generation

Keywords and phrases Large and High res Displays, Coordinated and Multiple Views, Human Computer Interaction

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.144

1 Introduction

Innovations in large wall-sized displays have been yielding benefits to visualizations in industry and academia: to cater to a larger audience, for more efficient collaborative work, to further immerse the client in virtual reality applications, and to facilitate the visualization of large and complex datasets by maintaining both overview and detail views simultaneously [5]. It is these improvements that have led to the growth of large display implementations despite the limitations in size of a single such display and the costs associated with them. The single most influential factor in this progression has been the advent of tiled display systems - a large display consisting of tiled smaller ones driven by clusters of off-the-shelf PC systems. This leads to complete physical scalability over the eventual size of the display while keeping the costs relatively low. These facts have led to a number of research projects that have incorporated tiled displays and made it amongst the top ten visualization research topics in recent times [15].

Effective exploration of truly large complex datasets is a quandary that researchers are constantly trying to tackle in a number of domains. In the recent past various solutions have been employed including multiple views, data aggregation, and filter techniques. There is no single solution to this problem, as it varies depending on the nature of the domain,

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 144–159
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

the dataset, and the experts involved. However, the work presented in this paper proposes the use of coordinated multiple applications / views on tiled-wall displays as a means to efficiently explore such datasets irrespective of the domain. The core idea is to divide the tiled-wall into a large work area that utilizes a distributed rendering framework and several other subsections that run coordinated applications. The emphasis of this work has been on the management of different concurrent applications running on a tiled-wall through a dispatcher framework that handles coordination and input management.

There are a number of tiled-display rendering frameworks available that vary in their sophistication and as a consequence in their complexity and functionality as well [16, 22]. However, these choices were significantly reduced while imposing the following requisites: to hide the display setup from the developer, to have the flexibility to execute on several display configurations, and to keep the implementation relatively simple. These rationales assisted us in choosing TileRenderer [5] as an appropriate rendering framework, as the other frameworks examined (CAVELib [14], VRjuggler [3], Syzygy [19], Jinx [21], Chromium [9], NAVER [11], and CGLX [6]) were quite heavy-weight with extensive APIs. The common theme amongst the frameworks mentioned is that they provide extensive libraries for interactions, interfaces, and virtual contents for virtual reality environments. It is due to the various libraries required and the sheer quantity of modules involved that makes their usage highly complex and heavy-weight. In comparison, Deller et al. formed TileRenderer with a small and easy to use library that allows developers to shape or alter existing OpenGL applications through the use of a few callback functions.

TileRenderer much like most existing such frameworks distributes the geometry and material data to render a single applications graphics database on multiple rendering clients while we aim to tackle multiple applications. This left three vital tasks for our dispatcher framework to perform: inter-application communication, interaction, and focus management. The only other viable solution would be to have a single scene with various viewports; however, we choose to refrain from this approach to eliminate the possibility of artifacts from one model appearing in the background of another and to a larger extent due to the interaction limitations within these viewports.

The organization of this paper is as follows: examine related work (Section 2), discuss the proposed dispatcher framework (Section 3), demonstrate it using a visualization case study (Section 4), and lastly look at possible future work (Section 5). In the case study, a visualization scenario is presented on two distinct hardware configurations that utilizes our framework in order to execute four visualization applications simultaneously, handle interactions between them, and manage input and focus remotely.

2 Related Work

In order to obtain a working prototype of the dispatcher framework, the following issue needed to be addressed: distribute and synchronize input devices and applications over the network. In this section, software solutions aimed at distributing input devices and applications are examined for comparison.

2.1 Distributed Device Data

In the related context of collaborative environments, co-located and synchronous interactions lead to a query of how a specific form of input supports an application realized on a shared display [8]. APIs such as CAVELib [14], Syzygy [19], and VR Juggler [3] have embedded support for collaborative interaction between remote applications. In the case of the latter, this is in the form of a module called Gadgeteer, which distributes device data across machines and clusters.

Instead of integrating such a heavy-weight architecture, producing our own streamlined mechanism was preferred: switching the *virtual focus* and passing encoded/decoded Simple DirectMedia Layer (SDL) [20] messages - see Section 3.4.

2.2 Distributed Applications

There are a few standards or technologies that are designed towards application integration. Common Object Request Broker Architecture (CORBA) was a revelation in allowing developers to undertake distributed object-oriented programming to integrate diverse applications into heterogeneous distributed systems [24]. Similarly, the High Level Architecture (HLA) [4] has been instrumental in linking disparate simulations. Such frameworks provide rich functionality at the expense of API complexity, so we crafted our own method: each application registering for a message type with the dispatcher and processing it accordingly.

3 Framework Overview

The dispatcher framework presented in this paper works in conjunction with TileRenderer, however, it can be quite easily ported to any distributed rendering software such as the ones mentioned in Section 1.

First, TileRenderer was modified to work with Coin3D [12], an open source Open Inventor API, instead of pure OpenGL. With the incorporation of scene graphs the environment immediately improved significantly, providing two distinct advantages:

1. Improved rendering performance and optimal use of available hardware resources. Scene graphs by nature maintain a "retained" model of the virtual world allowing additional optimizations such as processing culling and drawing in parallel, and state sorting [18].
2. Introducing a high-level programming interface to eliminate low-level OpenGL commands such as primitive drawing commands, state settings, and matrix manipulations [18].

The next vital step towards our goal of enhanced tiled wall utilization required the ability to handle multiple applications. This led to the development of the dispatcher framework that is explored further in the next sections. Its main functionality is to act as a centralized message center where all messages are received and forwarded as needed. The main components of this framework are depicted in the figure below and described thereafter.

- *NetMessages* (see Section 3.1) are created by the applications and connected to the Dispatcher via the *MessageHandler* (see Section 3.3).
- Individual applications register for a message type at the *Dispatcher* (see Section 3.2).
- Appropriate messages are forwarded by the *Dispatcher* only to the subscribers (see Section 3.2).
- Input is fed directly to the application running TileRenderer, for all others we incorporate the concept of *Virtual Input* - a special type of message that is an encoded SDL event (see Section 3.4).

Figure 1 depicts the logical system structure of the dispatcher framework. The developer may add several disjoint applications to the framework, however these applications need to interface with the common *MessageHandler* for both interaction and communication purposes. It is through the *MessageHandler* that the different applications are able to communicate with the *Dispatcher*. Here, a particular communiqué is worth mentioning:

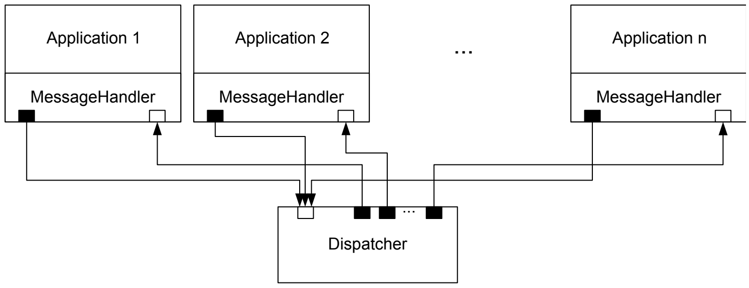


Figure 1 Logical System Structure.

encoded SDL events that are used to pass the *virtual focus*, hence providing the ability to interact with applications that may be running on a remote PC.

A centralized message center that is referred to as the *Dispatcher* is responsible for receiving and forwarding messages if necessary. The communication itself is realized by using sockets that are represented as black and white boxes in Figure 1. The distinction here is in the communication direction: a black box stands for an output socket while a white box stands for an input socket.

3.1 NetMessage

To be able to distinguish between different *NetMessages*, enumerated message types are used. Alternatively, one might have encoded the message type in the text message itself leading to a more complex process. Depending on the type of message, the processing of the message text varies:

- *INVALID:* An invalid message can not be processed. Non-blocking socket reading may return NULL, if there is no value to read. To avoid NULL references, this type is introduced.
- *DISPATCHER_REGISTRATION:* To take part in the communication process, the applications have to register at the *Dispatcher* by using this type of message.
- *DISPATCHER_MESSAGE_SUBSCRIPTION:* To be informed when a certain type of message is received at the *Dispatcher*, a client has to subscribe for it. This is done to avoid passing all messages to all the clients - flooding.
- *DISPATCHER_VIRTUALFOCUSCHANGE:* To force a change in which application has the *virtual focus*, this type of message has to be sent.
- *VIRTUALFOCUS_RECEIVED:* If an application receives this type of message, it has received the virtual focus. In case of *TileRenderer* this implies processing the *SDL_Events* captured by the input devices locally.
- *VIRTUALFOCUS_LOST:* If an application receives this type of message, it has lost the virtual focus. In case of *TileRenderer* this means to forward the *SDL_Events* captured by the input devices to the *Dispatcher*.
- *SDLEVENT:* This type of message signals that an *SDL_Event* is encoded in the *NetMessage*. This is described further in Section 3.4.
- *QUIT:* An application has to finish its current task and shut down.

There are a number of other application specific messages that can be added to exchange specific information as required by the developer, examples of these are presented later in the case study.

3.2 Dispatcher

In this section, we examine the logical structure of the *Dispatcher* (see Figure 2a) and the flow of its main loop (see Figure 2b). The *DispacherThread* is at the heart of the logical structure: it periodically invokes the main loop, polls the *InputSocket*, and maintains the *VirtualFocus* and the *Registry*.

The *DispatcherThread* is an asynchronous thread that implements a non-blocking read by continuously polling the *InputSocket* for an encoded character stream. This encoded stream is used to fill a *NetMessage* data structure, where a key field is the enumerated message type that determines how to process such a message. If the received message is of type *INVALID* this means there is no data available and the next *NetMessage* is polled. This action continues until there is valid data available on the socket - any message that is not of type *INVALID*.

As soon as a valid message is received, it is checked at the *Registry* whether there are subscribers for this type of message. If this is the case, the *NetMessage* is forwarded via the *OutputSocket* of the corresponding registry entry. Depending on the message type, further processing may be required:

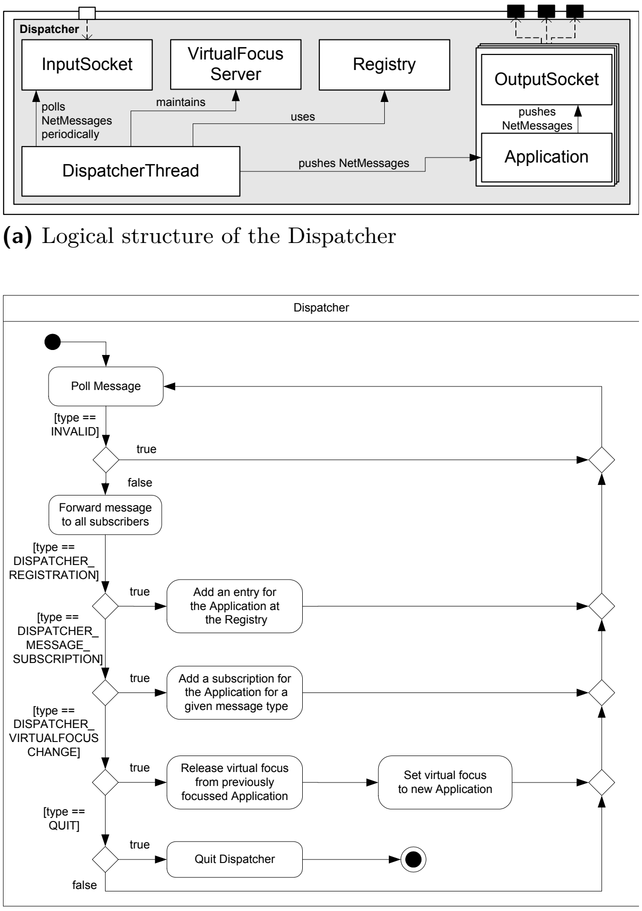
- *DISPATCHER_REGISTRATION:* If the received message is of this type, a new application has to be registered. This is accomplished through a new application entry created at the *Registry* that contains the *OutputSocket* connection to the remote application.
- *DISPATCHER_MESSAGE_SUBSCRIPTION:* If such a message is received, it implies that an application has subscribed for a new type of message at the *Dispatcher*. Whenever a message with the corresponding type is received by the *Dispatcher*, it will be forwarded to all its subscribers.
- *DISPATCHER_VIRTUALFOCUSCHANGE:* This type of message forces the *virtual focus* to switch to another application. The switching process is faciliated by the *VirtualFocusServer*, responsible for the release of the *virtual focus* from the previous application and for setting it to a new one. In other words, input device data captured by SDL is forwarded to the appropriate subscribing application.
- *QUIT:* If a message of type *QUIT* is received, it is distributed to all registered applications in order to initiate their respective shutdown.

If some other type of message is received by the *Dispatcher*, it is simply ignored as it holds no significance for the *Dispatcher* itself.

3.3 MessageHandler

It has been shown earlier in Figure 1 that all applications or views use the *MessageHandler* component to communicate with the *Dispatcher*. Here it is important to note that the current configuration utilizes TCP for both input and output sockets, thus ensuring that events are received in the same order as they occur. In this section we shall further examine the logical structure of the *MessageHandler* as shown in Figure 3a and the typical behavior of a view as in Figure 3b.

When a new application or view starts up, it initializes the *MessageHandler*. It is the *MessageHandler* that communicates backwards to the registered applications or views, through the use of callbacks. Thus, the first step is to register all callbacks at the *MessageHandler*, which in turn registers itself automatically for the corresponding message types at the *Dispatcher*. Once these callbacks are registered, both the *InputThread* as well as the *OutputThread* are started.


Dispatcher
InputSocket
VirtualFocus Server
Registry
OutputSocket
polls NetMessages periodically
maintains
uses
pushes NetMessages
DispatcherThread
pushes NetMessages
Application
Dispatcher
Poll Message
[type == INVALID]
true
false
Forward message to all subscribers
[type == DISPATCHER_ REGISTRATION]
true
Add an entry for the Application at the Registry
[type == DISPATCHER_ MESSAGE_ SUBSCRIPTION]
true
Add a subscription for the Application for a given message type
[type == DISPATCHER_ VIRTUALFOCUS CHANGE]
true
Release virtual focus from previously focussed Application
Set virtual focus to new Application
[type == QUIT]
true
Quit Dispatcher
false

(a) Logical structure of the Dispatcher

(b) Main loop of the Dispatcher

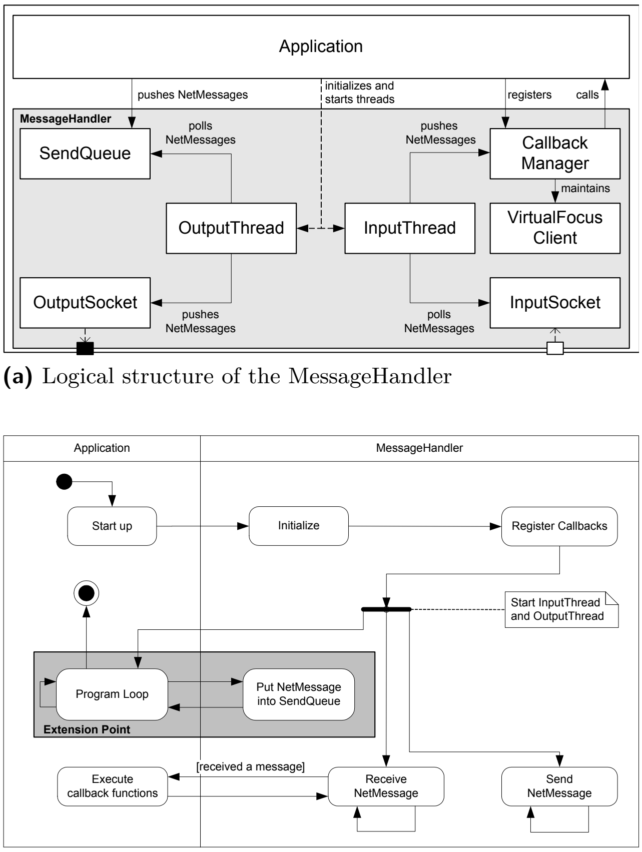
Figure 2 Structure and flow of the Dispatcher.

The original thread continues with the execution of the applications main program loop. While this program loop is in operation, some *NetMessages* may be put into the *SendQueue*. Similarly, the *InputThread* continuously receives messages in a non-blocking way. Much like the dispatcher functionality described in Section 3.2, as long as no valid messages are received the *InputThread* simply waits. On the other hand, a valid message is pushed to the *CallbackManager* as soon as it is received. If there are callbacks registered for this type of message, the corresponding callback function is executed.

The *VirtualFocusClient* registers callbacks at the *CallbackManager* by default when it starts up. Further, if a received message is of the type *VirtualFocusChange* then the *VirtualFocusClient* is called and the virtual focus is handled automatically - by either grabbing or releasing the virtual focus in question.

The *OutputThread* continuously polls the *SendQueue*. If there are messages available, they are pushed to the *OutputSocket* and transmitted.

Input devices are physically connected only to the *TileRenderer* application, the extension point in Figure 3b for this application is refined further in Figure 4. The *TileRenderer* based application continuously captures input data from all its connected input devices as *SDL_Events*. If an *SDL_Event* representing a special menu key is pressed, a message is put into the *SendQueue* to switch the focus to the *Menu View*. If the *SDL_Event* represents the



(a) Logical structure of the MessageHandler

(b) Typical behavior of a view during runtime

Figure 3 Structure and behavior of the MessageHandler.

special quit key, in this case the ESC key, a message of type *Quit* is put into the *SendQueue*. If the *TileRenderer* holds the virtual focus, it processes these *SDL_Events* locally. Alternatively, if it does not hold the virtual focus then the *SDL_Events* are encoded and put into the *SendQueue*, so that the remote application with the virtual focus can process it.

3.4 Virtual Input: Encoding/Decoding of SDL Events

The structure of an *SDL_Event* is shown in Figure 5. This is a streamlined structure as only the *SDL_Event* types which are used by our system are listed, further general SDL details are found on their Wiki documentation [20]. It is prominent from the above-mentioned diagram that each *SDL_Event* type has its own specific structure:

- *SDL_MouseButtonEvent*: this SDL event type signalizes a mouse button event and is followed by the fields: *button*, *state*, x, and y. These fields encode the following information: *button* encodes the mouse button index, *state* is either set to *SDL_PRESSED* or *SDL_RELEASED* indicating if the button is pressed or released, x holds the x coordinate relative to the window, and y holds the y coordinate relative to the window.
- *SDL_MouseMotionEvent*: this field signalizes a mouse motion event and is followed by the fields: *state*, x, y, *xrel*, and *yrel*. These fields encode the following information: for *state*, x, and y refer above, whereas *xrel* and *yrel* hold the relative motion in the x and y screen directions respectively.

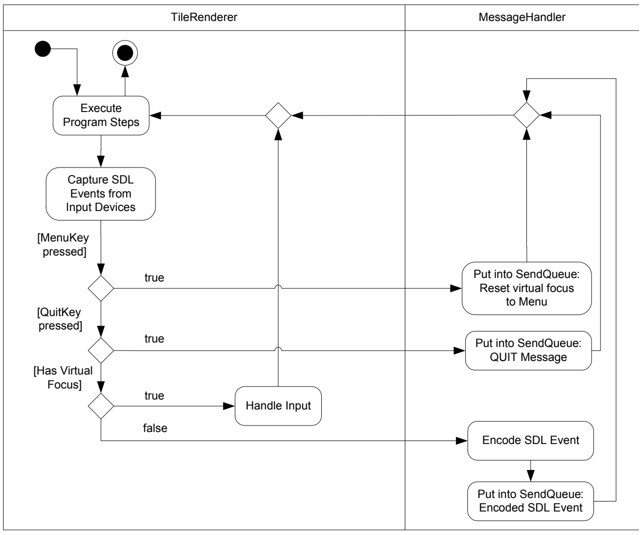
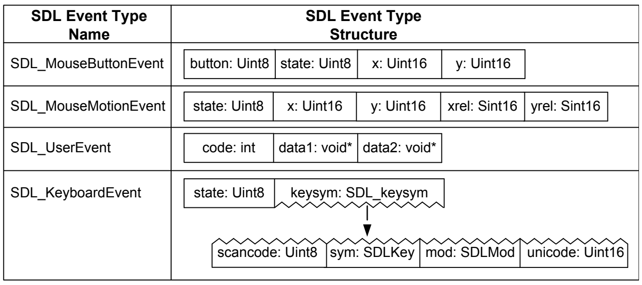

TileRenderer
MessageHandler
Execute Program Steps
Capture SDL Events from Input Devices
[MenuKey pressed]
true
[QuitKey pressed]
true
[Has Virtual Focus]
true
false
Handle Input
Put into SendQueue: Reset virtual focus to Menu
Put into SendQueue: QUIT Message
Encode SDL Event
Put into SendQueue: Encoded SDL Event

Figure 4 Local and remote processing of SDL_Events.



SDL Event Type Name	SDL Event Type Structure				
SDL_MouseButtonEvent	button: Uint8	state: Uint8	x: Uint16	y: Uint16	
SDL_MouseMotionEvent	state: Uint8	x: Uint16	y: Uint16	xrel: Sint16	yrel: Sint16
SDL_UserEvent	code: int	data1: void*	data2: void*		
SDL_KeyboardEvent	state: Uint8	keysym: SDL_keysym			
	scancode: Uint8	sym: SDLKey	mod: SDLMod	unicode: Uint16	

Figure 5 Structure of an SDL_Event.

- *SDL_UserEvent*: this field refers to a user defined input device and is followed by the fields: *code*, *data1*, and *data2*. These fields encode the following information: *code* is a user defined event code, whereas *data1* and *data2* are user defined data pointers that may be utilized as required.
- *SDL_KeyboardEvent*: this field refers to a keyboard event and is followed by the fields: *state* and *SDL_keysym*. The *state* field can hold either the value *SDL_PRESSED* or *SDL_RELEASED* indicating if a key is pressed or released. The *SDL_keysym* structure on the other hand is composed of several further fields: *scancode*, *sym*, *mod*, and *unicode*. The *scancode* field is normally not utilized; it contains a hardware-dependent scan-code returned by the keyboard. The *sym* field is the SDL-defined constant that represents the selected key and is often used while programming to inquire if a certain key has been pressed or released. The field *mod* stores the current state of the keyboard modifiers and *unicode* stores the unicode character corresponding to the key if it is enabled.

In the case of *TileRenderer* having virtual focus, these SDL events are processed locally. On the other hand, when another application has the virtual focus we encode these SDL events into a character string and send them to that application via the *Dispatcher*. It is possible to encode and then later decode these SDL events as knowledge of its datastructure is known beforehand. Once decoded, a new SDL event is created with the received values and pushed into the SDL event queue of the receiving application, hence the concept of

Virtual Input. The only exception to the above process was the *SDL_UserEvent*, where *data1* and *data2* are user defined data pointers. In our scenario (see Section 4), a space mouse was added as a user defined device. Here, knowledge of the *Rotation* and *Translation* datastructures utilized by the space mouse handler assisted in encoding and decoding *data1* and *data2* respectively.

4 Case Study: Safety and Security Analysis using CakES

The framework presented in this paper has been used to produce a safety visualization called CakES [1] that was designed to assist software engineers in the safety and security of embedded systems. This visualization system consists of multiple applications or views that visualize the physical model, the minimal cutsets, and the basic events of the fault tree. Our framework facilitates interaction amongst these different views, allowing the user to have different levels of focus and context simultaneously.

A brief introduction to these safety analysis topics is listed below with references for further readings:

- *Fault Trees (FTs)* are tools in system safety, reliability, and availability studies [13].
- *Basic Events (BEs)* are the lowest-level influence factors in the *FT* and they are represented as the leaves. The hazard that is examined in the fault tree is called the *top event* which is at its root [10].
- *Minimal Cut Sets (MCSs)* are unique combinations of *BEs* that can cause the top event to occur [2].

In addition to the *NetMessages* described in section 3.1, the following application specific messages were added to the framework:

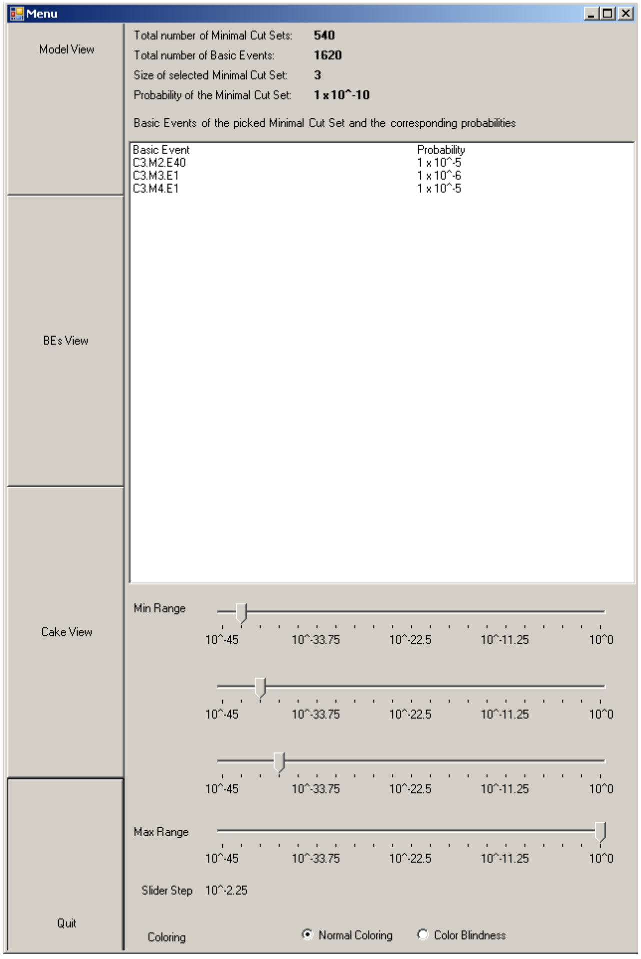
- *MCS_SELECTION:* A minimal cut set was selected.
- *MCS_BE_PROBABILITY:* A basic event's probability is sent.
- *HOLDER_BOUNDS_CHANGED:* Min and Max bounds changed.
- *HOLDER_VALUES_CHANGED:* Value of a holder changed.

4.1 Multiple Applications

The CakES system utilizes four distinct applications or views interconnected through the framework presented in this paper. The reason to employ these views as independent applications is two-folds: 1) give programming freedom to the developer, and 2) avoid different viewports in the same scene to avoid artifacts and interaction limitations. These views are described further in the following subsections.

4.1.1 Menu View

The menu (see Figure 6) functions as the primary interface between the other views that are discussed further in Sections 4.1.2, 4.1.3, and 4.1.4. It allows the user to switch the virtual focus between these different views. Additionally, it displays vital statistical data about the selected *MCS*: size, probability, and details of its *BEs*. Further, the user may reorganize the *MCSs* in the *MCS View* according to his preferred criteria by using either the sliders or radio buttons provided.



Total number of Minimal Cut Sets: **540**

Total number of Basic Events: **1620**

Size of selected Minimal Cut Set: **3**

Probability of the Minimal Cut Set: **1 x 10^-10**

Basic Events of the picked Minimal Cut Set and the corresponding probabilities

Basic Event	Probability
C3.M2.E40	1 x 10^-5
C3.M3.E1	1 x 10^-6
C3.M4.E1	1 x 10^-5

Slider Step 10^-2.25

Coloring ● Normal Coloring ○ Color Blindness

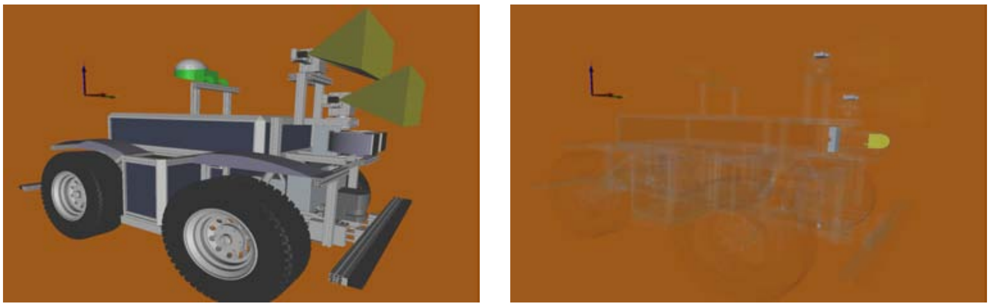
Figure 6 Menu View.

4.1.2 Model View

This view is employed to exhibit the physical parts of the RAVON model as shown in Figure 7a. The model depicted is of the Robot RAVON (Robust Autonomous Vehicle for Off-road Navigation); further details can be found on the AG Robotersysteme website [17]. The TileRenderer application drives this view as it is to be displayed on a stereo monitor in the configuration described in Section 4.2.1 and on the larger area of the tiled wall as in the configuration described in Section 4.2.2. Interaction mechanisms that allow the user to explore the model further are zooming, rotation, and translation. Further, it utilizes the framework by registering for an *MCS_SELECTION*, so that once an *MCS* is selected in the *MCS View* (see Section 4.1.3) appropriate data is received. As a consequence of this data exchange, the RAVON model is rendered transparent and the relevant *BEs* are rendered opaque (see Figure 7b).

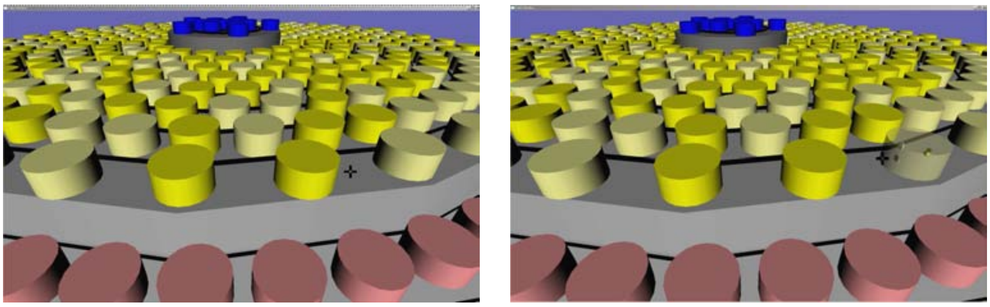
4.1.3 MCS View

This view uses a Cake metaphor [1] to visualize the *MCSs*, their probabilities, and their BEs (see Figure 8). A text file containing information about the *BE* distribution is generated



(a) Before (no MCS selected) **(b)** After (MCS selected)

Figure 7 Model View: Before and after receiving MCS data through Dispatcher.



(a) Before Virtual Picking **(b)** After Virtual Picking

Figure 8 MCS View: Before and after Virtual picking.

using the *ESSaRel* tool [7] and is used in the intial formation of the Cake (see Figure 8a). The Cake consists of three separate levels depicted by red, yellow, and blue cylinders. Each cylinder represents an *MCS* and within each *MCS* there are a certain number of *BEs*. These three levels correspond to a range of fault probabilities that may also be adjusted via the Menu sliders. Further, each level uses saturation to distinguish between probabilities that lie in the same range.

The user is provided interaction mechanisms quite similar to the *Model View* 4.1.2. In addition, the concept of *Virtual Picking* was utilized to handle virtual interactions within this view, as there was no real focus available - the user interacts through remote devices that send relevant data through our framework. *Virtual Picking* is accomplished by tracking relative mouse movements, drawing a ray through the current mouse position and the far clipping plane, and selecting the intersecting shape. When invoked, it makes an *MCS* transparent and one can see the *BEs* within (see Figure 8b). Information regarding the selected *MCS* and its *BEs* are then sent to the other views interested in them through the *Dispatcher*.

4.1.4 BEs View

The *BEs View* (see Figure 9) is directly related to the *Model* and *MCS Views* of Sections 4.1.2 and 4.1.3. In this view, the hardware components related to the *BEs* within the selected *MCS* in the *MCS View* (Figure 8b) are displayed in more detail. Currently, there are no interaction mechanisms in this view.

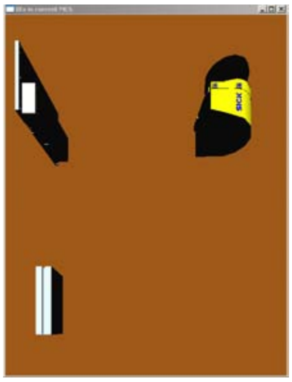


Figure 9 BEs View.

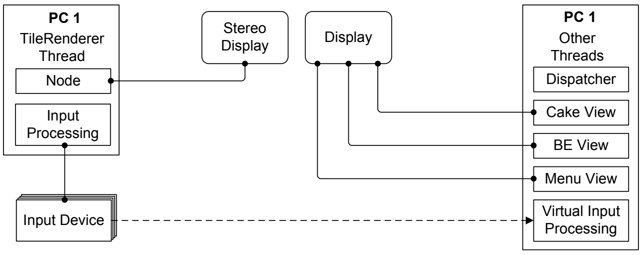


Figure 10 Desktop Configuration.

4.2 Hardware Configurations

The framework presented in this paper is highly flexible in terms of hardware configurations. A text file holds key information for each application such as: the name of the application, the port number associated to it, the PC host name, and a keyword indicating whether it is virtually focussable. Depending on the desired configuration, the developer would need to carry out the following two tasks for each application:

1. Provide the correct host name in the configuration file
2. Supply appropriate orientation and dimension for each view

For the CakES visualization, two different configurations were employed. The first is a two-monitor single-PC solution (see Section 4.2.1), while the second is a nine-monitor five-PC tiled-display solution (see Section 4.2.2).

4.2.1 Desktop

This configuration consisted of a standard monitor with 1920x1200 pixels and a Zalman Trimon passive-stereo monitor with a resolution of 1600x1050. The desktop PC ran on a Windows XP operating system and had the following key components: 2.60 GHz AMD Phenom™ 9950 Quad-Core Processor, 3.25 GB of RAM, and an NVIDIA GeForce GTX 280 graphics card. An additional user defined interaction device, the 3Dconnexion Space Navigator was added. This system was mainly used for development purposes and its physical layout is depicted in Figure 10.

The stereo monitor was used to display the physical parts of the RAVON model as shown in Figure 7. On the other hand, the standard monitor was used to tile the *Menu*, *BEs*, and *Cake* Views next to each other as in Figure 11.

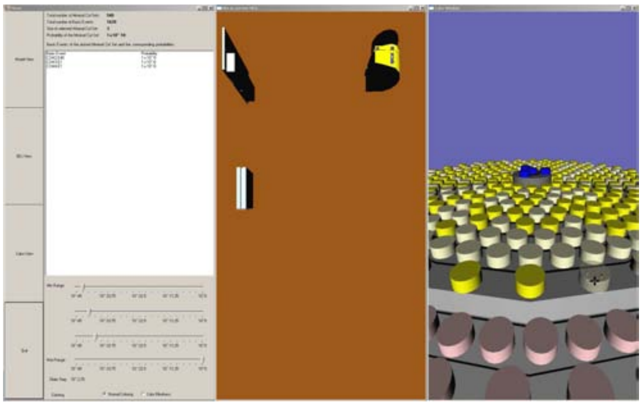


Figure 11 Menu, BEs, and MCS Views tiled on a standard PC.

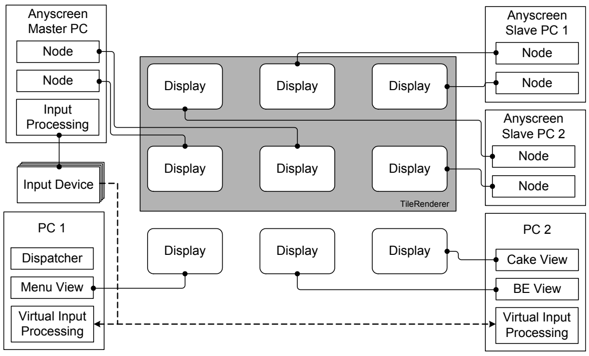


Figure 12 Tiled Wall Configuration.

4.2.2 Tiled Wall

A 3x3 tiled wall with five computers was implemented as depicted in Figure 12. This image is based on the 'schematic view on a typical 3x3 tiled display' figure presented by Deller et al. [5], it is modified to incorporate our dispatcher framework. As shown, the tiled wall consists of nine displays controlled by five network-connected PCs. Each computer ran on Windows XP and had the following hardware configuration: Intel® Core™ Dual Core @ 2.4 GHz, 4 GB of RAM, and two GeForce 7950 GX2 graphics cards. Each screen of the tiled wall has a resolution of 2500x1600 pixel, making the total resolution of the wall 7500x4800. Further, there is an upper and a lower separation of the Tiled-Wall:

- The upper six displays are managed by *TileRenderer* - it uses three PCs for this task, where one of them acts as the Master-PC and controls the other two Slave-PCs via the network.
- The lower three displays are controlled by the remaining two PCs. One of them runs the *Dispatcher* framework and displays the *Menu View* on the lower-left screen of the Tiled-Wall. The other is used to visualize the *BEs* and *MCS Views* on the remaining two screens.

All the above mentioned applications are controlled by the same input devices, which are physically connected to the Master-PC of the *TileRenderer*. This configuration allows the Master-PC to process input events locally when the *Model View* application has the *virtual focus*. On the other hand, it forwards input data via the network to PC1 and PC2

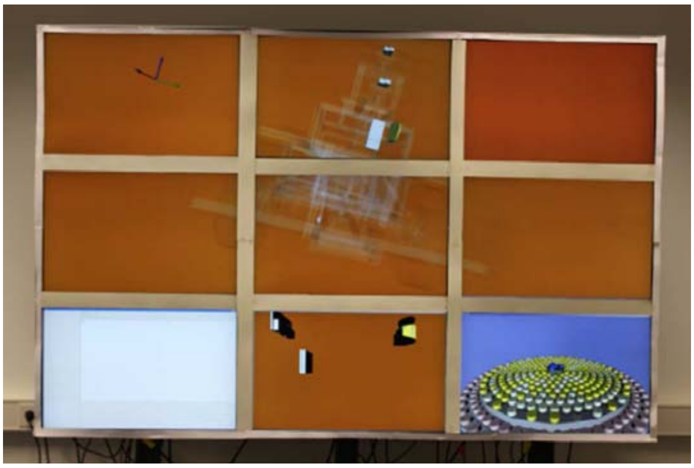


Figure 13 Real-Time Tiled-Wall Scenario.

when one of those two applications has the *virtual focus*. A screen shot of these applications interacting with one another is illustrated in Figure 13.

5 Conclusion and Future Work

In this paper we have presented a light-weight yet flexible *Dispatcher* framework that facilitates the simultaneous execution of multiple inter-communicative visualizations built on *Open Inventor* scene graphs.

TileRenderer was the preferred tiled-display-rendering framework due to its simple yet robust rendering framework. As seen in the case study, it was responsible for rendering the *Model View* across a large section of the display area. However, the approach was kept flexible enough to let developers employ their own preferred distributed rendering framework. Additional stand-alone visualizations such as the *Menu*, the *BEs View*, and the *MCS View* were rendered separately on three of the tiled-wall displays. All of the above-mentioned visual applications had to be registered with the *Dispatcher* in order to effectively communicate with one another and to appropriately handle the remote input devices via our *Virtual Input* implementation.

Through the *CakES* case study that has been realized using our framework, we tackled our primary goal of efficient tiled-display size and resolution utilization by rendering four different applications and interacting between them. Further, we have been able to develop a system to assist model designers and safety analysts in the context of analyzing embedded system. These applications are highly adaptive, as the developer can easily switch the scene graph database files and still use all our interaction mechanisms. Additionally, there is complete freedom to the number of applications that may be incorporated - it is a simple matter of adding them to the configuration file and providing appropriate display parameters.

The resulting *CakES* solution worked remarkably well, with the exception of an initial minor glitch - once blending was triggered in the *Model View*, interaction with the applications through the *Dispatcher* slowed down significantly. After studying this issue further we realized that this was an Open Inventor performance issue due to inadequate volatile memory. Once we increased the RAM from one gigabyte to four, interaction through the *Dispatcher* became virtually real-time.

In the future, we plan to extract input capture from *Model View* and have device capturing modules independent of the applications themselves. This would provide us with the ability to easily integrate suitable virtual reality interaction devices and for all the applications to handle interaction via our virtual input mechanism. Currently, only one application is integrated with a distributed rendering framework, we would like to add it to all of them, providing the operator a choice of which application he would like to be tiled. Alternatively, the *MCS View* may be tiled across a larger area and elements of the *BEs View* may be explored interactively. Also, projecting a low resolution image onto the bezels of the model view would improve the presentation of the model and the distinction between the different applications (see [5]).

Finally, another possible enhancement is to implement the *Message Handler* via interrupts or callbacks instead of polling threads to save CPU cycles. It is planned to conduct a quantitative measure of the framework's performance, once the *Dispatcher* is improved.

Acknowledgements

I would like to thank the members of both the Computer Graphics and Visualization Group and the Robotics Research Lab in Kaiserslautern, as well as the members of the International Research Training Group (IRTG) for their cooperation. The IRTG is supported by the German Research Foundation (DFG) under contract DFG GK 1131. Furthermore, the authors wish to acknowledge Daniel Steffen from the German Research Center for Artificial Intelligence for his constructive comments and assistance with the TileRenderer framework. This work was supported by the German Federal Ministry of Education and Research (BMBF), under contract number 01 IM 08003, through project ViERforES [23].

References

1 Y. Al-Zokari, T. Khan, D. Schneider, D. Zeckzer, and H. Hagen. CakES: Cake Metaphor for Analyzing Safety Issues of Embedded Systems. In Hans Hagen, editor, *Scientific Visualization: Interactions, Features, Metaphors.*, volume 2 of *Dagstuhl Follow-Ups*, pages 1–16, Wadern, Germany, 2011. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `http://dx.doi.org/10.4230/DFU.Vol2.SciViz.2011.1`.

2 T. Bedford and P. Gelder. *Safety and Reliability : Proceedings of the ESREL 2003 Conference, Maastricht the Netherlands, 15-18 June 2003.* Taylor & Francis, 2003.

3 Allen Bierbaum, Christopher Just, Patrick Hartling, Kevin Meinert, Albert Baker, and Carolina Cruz-Neira. VR Juggler: A Virtual Platform for Virtual Reality Application Development. In *VR '01: Proceedings of the Virtual Reality 2001 Conference (VR'01)*, page 89, Washington, DC, USA, 2001. IEEE Computer Society.

4 Judith S. Dahmann, Frederick S. Kuhl, and Richard M. Weatherly. Standards for Simulation: As Simple As Possible But Not Simpler The High Level Architecture For Simulation. *Simulation*, 71(6):378–387, 1998.

5 Matthias Deller, Sebastian Thelen, Daniel Steffen, Peter-Scott Olech, Achim Ebert, Jan Malburg, and Jörg Meyer. A Highly Scalable Rendering Framework for Arbitrary Display and Display-in-Display Configurations. In Hamid R. Arabnia and Leonidas Deligiannidis, editors, *CGVR*, pages 164–170. CSREA Press, 2009.

6 Kai-Uwe Doerr and Falko Kuester. CGLX: A Cross-Platform Cluster Graphics Library. `http://vis.ucsd.edu/mediawiki/index.php/Research_Projects:_CGLX`; Online; Accessed 23-January-2010.

7 ESSaRel. Background information — ESSaRel, 2002. `http://www.essarel.de/background/background.html`; Online; Accessed 30-December-2009.

8 Otmar Hilliges, Lucia Terrenghi, Sebastian Boring, David Kim, Hendrik Richter, and Andreas Butz. Designing for collaborative creative problem solving. In *C&C '07: Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, pages 137–146, New York, NY, USA, 2007. ACM.

9 Greg Humphreys, Mike Houston, Ren Ng, Randall Frank, Sean Ahern, Peter D. Kirchner, and James T. Klosowski. Chromium: a stream-processing framework for interactive rendering on clusters. *ACM Trans. Graph.*, 21(3):693–702, 2002.

10 B. Kaiser, C. Gramlich, and M. Foerster. State/event fault trees - safety analysis model for software-controlled systems. *Reliability engineering & systems safety*, 92:1521–1537, 2007.

11 KIST Imaging Media Research. IMRC Wiki: The NAVER framework. `http://www.imrc.kist.re.kr/wiki/NAVER_Framework`; Online; accessed 23-January-2010.

12 Kongsberg SIM AS. Coin3D: 3D Graphics Developer Kit. `http://www.coin3d.org/`; Online; Accessed 27-January-2010.

13 N. Limnios. *Fault Trees (Control Systems, Robotics & Manufacturing Series)*. Wiley, John & Sons, 2007.

14 Mechdyne. CAVELib Application programmer interface (API). `http://www.mechdyne.com/integratedSolutions/software/products/CAVELib/CAVELib.htm`; Online; Accessed 23-January-2010.

15 Tao Ni, Greg S. Schmidt, Oliver G. Staadt, Mark A. Livingston, Robert Ball, and Richard May. A Survey of Large High-Resolution Display Technologies, Techniques, and Applications. In *VR '06: Proceedings of the IEEE conference on Virtual Reality*, pages 223–236, Washington, DC, USA, 2006. IEEE Computer Society.

16 Bruno Raffin and Luciano Soares. PC Clusters for Virtual Reality. In *VR '06: Proceedings of the IEEE conference on Virtual Reality*, pages 215–222, Washington, DC, USA, 2006. IEEE Computer Society.

17 RAVON. AG Robotersysteme: Ravon, 2009. `http://agrosy.informatik.uni-kl.de/en/robots/ravon/`; Online; Accessed 30-December-2009.

18 RealityPrime. Scenegraphs: Past, present, and future. `http://www.realityprime.com/articles/scenegraphs-past-present-and-future`; Online; Accessed 29-January-2010.

19 Benjamin Schaeffer and Camille Goudeseune. Syzygy: Native PC Cluster VR. *Virtual Reality Conference, IEEE*, 0:15, 2003.

20 SDL: Simple DirectMedia Layer. A cross-platform multimedia library. `http://www.libsdl.org/cgi/docwiki.cgi/`; Online; Accessed 15-December-2009.

21 Luciano P. Soares and Marcelo K. Zuffo. JINX: an X3D browser for VR immersive simulation based on clusters of commodity computers. In *Web3D '04: Proceedings of the ninth international conference on 3D Web technology*, pages 79–86, New York, NY, USA, 2004. ACM.

22 Munjae Song, Seongwon Park, and Yongbin Kang. A Survey on Projector-Based PC Cluster Distributed Large Screen Displays and Shader Technologies. In Hamid R. Arabnia, editor, *CGVR*, pages 153–159. CSREA Press, 2007.

23 ViERforES. Virtuelle und Erweiterte Realität für höchste Sicherheit und Zuverlässigkeit von Eingebetteten Systemen, 2009. `http://www.vierfores.de/`; Online; Accessed 02-February-2010.

24 S. Vinoski. CORBA: integrating diverse applications within distributed heterogeneous environments. *Communications Magazine, IEEE*, 35(2):46–55, Feb 1997.

Salient Frame Detection for Molecular Dynamics Simulations

Youngmin Kim[1], Robert Patro[2], Cheuk Yiu Ip[3], Dianne P. O'Leary[4], Andriy Anishkin[5], Sergei Sukharev[6], and Amitabh Varshney[7]

1,2,3,4,7 Department of Computer Science and
University of Maryland Institute for Advanced Computer Studies
College Park, MD 20742
E-mail:
{ymkim,rob,ipcy,oleary,varshney}@cs.umd.edu

5,6 Department of Biology, University of Maryland
College Park, MD 20742
E-mail:
anishkin@icqmail.com
sukharev@umd.edu

Abstract

Recent advances in sophisticated computational techniques have facilitated simulation of incredibly-detailed time-varying trajectories and in the process have generated vast quantities of simulation data. The current tools to analyze and comprehend large-scale time-varying data, however, lag far behind our ability to produce such simulation data. Saliency-based analysis can be applied to time-varying 3D datasets for the purpose of summarization, abstraction, and motion analysis. As the sizes of time-varying datasets continue to grow, it becomes more and more difficult to comprehend vast amounts of data and information in a short period of time. In this paper, we use eigenanalysis to generate orthogonal basis functions over sliding windows to characterize regions of unusual deviations and significant trends. Our results show that motion subspaces provide an effective technique for summarization of large molecular dynamics trajectories.

1998 ACM Subject Classification J. Computer Applications, J.2 Physical Sciences and Engineering

Keywords and phrases Saliency based analysis, Molecular Dynamics, Simulation

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.160

1 Introduction

Recent advances in acquisition and simulation techniques have resulted in immense time-varying datasets. These datasets are being used to study and explore a wide variety of phenomena in a diverse set of disciplines spanning life sciences to earth and space sciences. As the number and complexity of these datasets increases exponentially [9], it is becoming impractical to expect a domain expert to be able to look at such datasets manually. Automatic or semi-automatic tools to help humans discover scientifically interesting features are especially important for this reason.

Many illustration-based techniques have been proposed by several researchers [3, 10, 21] to summarize time-varying datasets such as ocean flow, volume, and human skeletons. The basic step for these illustration techniques is automatic detection of salient frames which

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 160–175
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

have interesting features. In the method of image saliency by Itti [8] or mesh saliency by Lee [13], they use a center-surround operator to identify the uniqueness of a pixel or a vertex with respect to its neighborhood. In this paper, we have decided to use a similar approach and define saliency as the uniqueness of a single frame with respect to its neighboring frames both forwards and backwards in time. Our collaborator, Dr. Sergei Sukharev's group at Biology Department at the University of Maryland, was interested in identifying the frames in molecular dynamics simulations, where the anomalies (kinks) in the secondary structures happen in the opening and closing simulations of thechannel [1]. We validate the effectiveness of our salient frame detection algorithm in this molecular dynamics simulation.

The rest of this paper is organized as follows. A review of related work is provided in Section 2. In Section 3, we formulate the relationship between one residue and the neighboring residues in space, and present an algorithm to detect saliency in time. Results are presented in Section 4. Performance considerations to improve the scalability of our approach for larger simulations are given in Section 5. Section 6 concludes this paper and discusses future work.

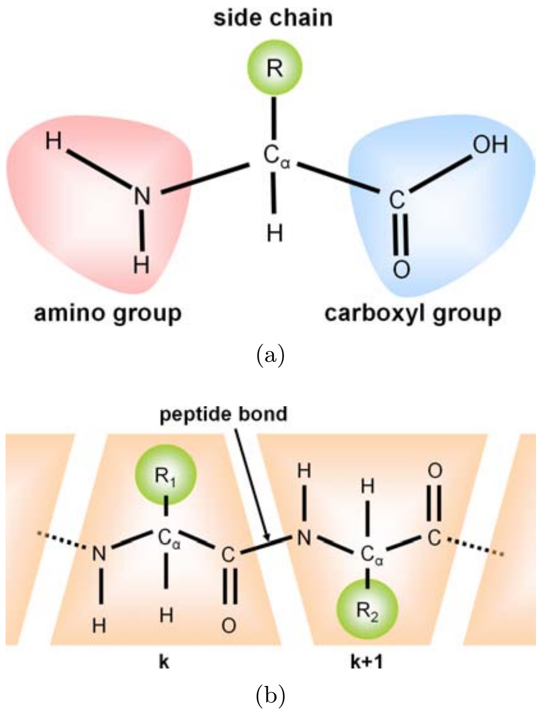
2 Background and Related Work

The goal of this paper is to detect salient frames in molecular dynamics simulations. This section briefly reviews some background in protein and ion channel structure and the related research in the area of motion analysis and visualization for time-varying 3D datasets.

2.1 Protein Structures

A protein structure is formed by a unique three-dimensional assembly of a specific polypeptide chain. Each polypeptide chain contains a particular sequence of serially linked amino acids. Figure 1(a) shows an amino acid which is composed of an amino group, a carboxyl group, and a side-chain, which are connected at the central C_α atom. When the carboxyl group of one amino acid reacts with the amino group of another amino acid, a peptide (i.e., amide) bond (Figure 1(b)) is formed by releasing a molecule of water (H_2O). This peptide bond is typically composed of four atoms (C, O, N, and H) which lie on a common plane due to the partial double bond characteristic at the CO-NH connection. Here, the recurring atomic array of N-C_α-C(=O) from each amino acid of a polypeptide chain constitutes the protein *backbone*. By definition, the specific amino acid sequence for each polypeptide chain is the *primary* structure of the protein. Segments of polypeptides often fold locally into stable structures such as α-helices or β-strands, each of which is called a *secondary* structure. An α-helix is a right-handed coiled conformation, resembling a spring. β-strands connected laterally by three or more hydrogen bonds, form a generally twisted, pleated sheet.

The angle between two planes is referred as their *dihedral* angle. Figure 2(a) and (b) shows how we can compute the dihedral angle when there are four atoms which are not co-linear in 3D space. We first align the atoms B and C as shown in Figure 2(b). Then the dihedral angle corresponds to the angle measured in clockwise direction between the atom A and the atom D. Similarly, for a sequence on a protein's polypeptide chain, backbone atoms (C, N, and C_α) allow for three different dihedral angles of proteins as depicted in Figure 2(c): ϕ involving the backbone atoms C-N-C_α-C, ψ involving the backbone atoms N-C_α-C-N, and ω involving the backbone atoms C_α-C-N-C_α. The planarity of the peptide bond usually restricts ω to be 180° or 0°. Thus the Ramachandran plot [19] considers two variable



(a)

(b)

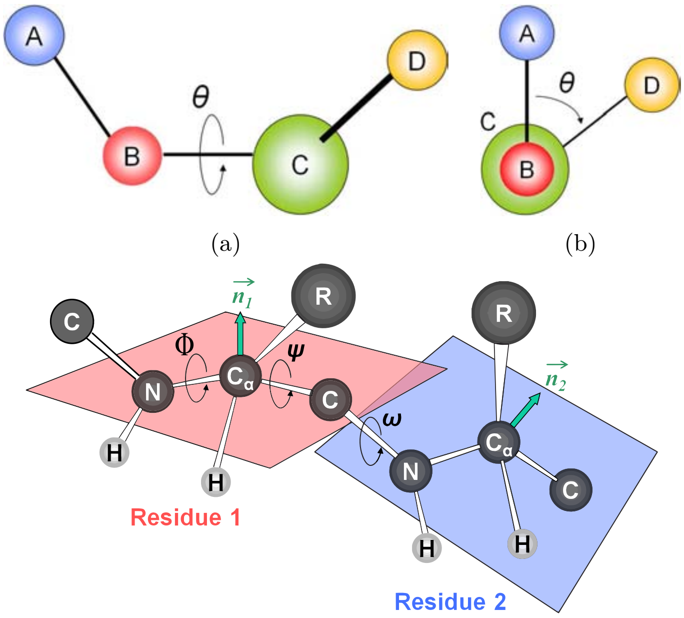
Figure 1 *Image (a) shows the structure of an amino acid. Image (b) shows a peptide bond formed by the reaction between a carboxyl group of one amino acid and an amino group of the other amino acid. Images are adapted from [6].*

dihedral (torsion) angles (ϕ and ψ) and shows possible combinations of these conformational angles of representative secondary structures in a polypeptide such as α-helices or β-sheets.

2.2 Ion Channels

Ion channels are proteins that regulate the flow of ions into and out of the cells. Ion channels enable a very rapid flow of ions. In physiological conditions, MscS can provide for the flow of about a billion ions per second. Ion channel transitions are very fast – some opening for less than a millisecond before they close. This rapid and highly specific gating of ion channels is necessary for survival of cells. The ion channel kinetics impacts the speed at which ions flow across the cell membrane and the reaction time of a nerve or a muscle cell, and thus dictates the response time of the animal to the possible environmental dangers. An accurate understanding of the structural changes and functioning of ion channels is vital for therapeutic drug design. Nearly a third of the top 100 pharmaceutical drugs target ion-channels.

The bacterial mechanosensitive channel MscS and its eukaryotic homologs are principal turgor regulators in many walled cells. In bacteria, both free-living and pathogenic, these channels play critical roles of tension-driven osmolyte release valves thus allowing the organisms to avoid osmotic rupture in the event of abrupt medium dilution. MscS opening is driven directly by tension in the surrounding lipid bilayer and is accompanied by tilting of the pore-lining helices (TM3) which assume a kink-free conformation [2, 4]. When tension is released, the TM3 helices may buckle at two different hinge points, which defines the progression toward the closed state, as is shown in Figure 3(a). Thus, helical flexibility appears to



(a) (b)

(c)

Figure 2 *Images (a) and (b) show the computation of a dihedral angle between 4 atoms (A, B, C, and D). When we align the atom B and the atom C as shown in Image (b), the dihedral angle θ is defined as the angle between the atom A and the atom D in clockwise direction. Image (c) shows the dihedral angles (ϕ between C-N-C_α-C, ψ between N-C_α-C-N, and ω between C_α-C-N-C_α) and the normal vectors ($\vec{n_1}$ and $\vec{n_2}$ on the planes defined using N-C_α-C in residue 1 and residue 2, respectively. Images are adapted from [6].*

define the functional cycle of E. *coli* MscS. The major dataset analyzed consisted of two trajectories of atomic coordinates obtained from 4 *ns* steered simulations representing opening of wild-type and F68S mutant of E. *coli* MscS. The major goal was identification of frames in which conformations of helices deviated from the typical alpha-helical conformations.

2.3 Saliency-based Motion Analysis

Designers and artists have long used a single static image or a few images to illustrate dynamics of scenes for motion. They have depicted dynamics to facilitate visual communication in comic books and storyboards [14]. Recently, several graphics researchers [10, 17, 18] have proposed illustration-based techniques to depict the dynamics of time-varying data in a compact way. They use principles of visual art such as glyphs, and generate an image (or a few images) to summarize the time-varying data to facilitate visual communication. For instance, Joshi and Rheingans [10] have used illustration-based techniques such as speedlines, flow ribbons, and strobe silhouettes to convey change over time for a time-varying dataset. Nienhaus and Dollner [17] have used dynamic glyphs such as directed acyclic graphs and behavior graphs to provide further information about dynamics in the 3D scene.

A very interesting beginning in detecting salient frames for human skeleton datasets has been made by Assa *et al.* [3]. They generate an action synopsis for presenting the motion of a single skeleton-based character. They represent motion in affinity matrices, constructed from various aspects of a pose such as joint positions, joint velocities, joint angles, and joint angular velocities. They first define a vector $\mathbf{x_a^k}$ which represents an aspect a of the pose at

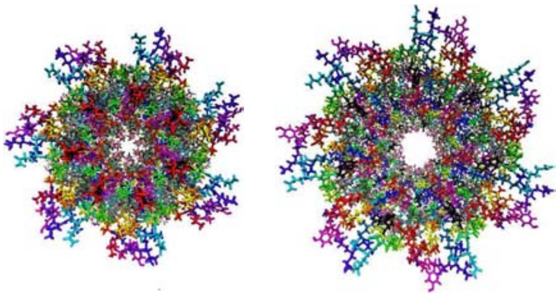


Figure 3 *The images above show the closed (left) and open (right) conformations of the heptameric E.* coli *mechanosensitive channel MscS.*

frame k. Then, they compute the dissimilarity of the aspect a between two given frames i and j by a simple distance measure to identify key poses. Finally, they compose these key poses into a single image by including the most significant poses.

There has been a significant increase in research activities related to the visualization of molecular dynamics simulations. Lampe *et al.* explore the use of a two-level hierarchical technique for the visualization of protein dynamics [12]. Recently, Krone *et al.* presented a method capable of visualizing molecular surface dynamics at interactive rates [11]. Tarini *et al.* [20] present a method to enhance shape perception in interactive molecular visualizations by employing ambient occlusion and edge cueing. Bidmon *et al.* present an informative and intuitive method for visualizing the motion of molecules around existing proteins using pathlines [5]. All of these papers [12, 11, 5] discuss methods for efficiently visualizing molecular dynamics, but do not detect key or salient frames in the simulation. Mehta *et al.* have explored approaches to the detection, classification and visualization of anomalous structures, such as defects in crystalline lattice structures [16, 15]. We are not aware of any research into the detection of salient frames for protein dynamics simulations.

3 Salient Frame Detection

In this paper, we characterize saliency as the uniqueness of a single frame with respect to its surrounding frames in time, and detect the salient frames for molecular dynamics simulations. A molecular dynamics simulation gives discrete samples of how a protein changes over a period of time. We are interested in identifying the time steps that highlight these changes by using subspace analysis. We analyze a particular time step k in a molecular dynamics simulation with the following approach:

1. Select the residues of interest, this could be a subunit of a molecule or residues corresponding to an α-helix of interest.
2. Model the angular relationship along the protein backbone for each time step k as an affinity matrix $\mathbf{A}_k$.
3. Decompose the affinity matrix using SVD, $\mathbf{A}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^T$
4. Analyze the salience, s_k, of the structure of the protein at time step k with respect to the surrounding time steps.
5. Peaks of salience s_k curve determine the best set of time steps.

Mechanosensitive ion channels play a critical role in transducing physical stresses at the cell membrane into an electrochemical response. The crystal structure of E. *coli* MscS has provided a starting point for detailed descriptions of its mechanism. Figure 3 shows the opening of the E. *coli* mechanosensitive ion channel that we will consider throughout this paper. There are 7 subunits in this ion channel, and all 7 subunits are topologically identical, but act relatively independently in the simulation. Out of 286 residues in each subunit, 175-residue N-terminal segments were included in simulations. To understand this mechanism, identifying the presence of kinks in α-helices is critical because they have functional importance. Kink detection, however, is a tricky question because there are many factors involved. These include the state (ruptured or not) of the H bonds, local geometric information such as Ramachandran angles (torsion angles), and more global information such as the angles among multiple atoms.

In this section, we formulate the relationship between one residue and the neighboring residues spatially, and present an algorithm to detect saliency in time. Our framework encompasses the global and local geometric properties of backbone residues in a molecular dynamics simulation.

3.1 Construction of Affinity Matrices in Space

We explore the relationship between one residue and the neighboring residues to detect the changes in α-helices. The straightening and buckling of α-helices are interesting because they appear in many simulations of ion channels and are believed to be correlated with conformational states of the whole channel. There are many ways to define the relationship among residues, but we believe the angles in backbone atoms would be one of the best ways since backbone atoms are much more stable in their positions than side-chains. As a Ramachandran plot suggests, we could have measured torsion (dihedral) angles and conjectured the changes of secondary structures for each residue. However, analysis of Ramachandran angles only considers very local properties inside a residue, and does not encompass the global geometric property among a sequence of residues. Instead, we use the relative angles between one C_α (α-carbon) and other α-carbons within a cutoff distance, r_s. The cutoff *distance* refers to the difference in index between the C_α currently being considered and its neighboring residues along the chain. A good choice is $r_s = 5$, because on average, α-helices turn once every 3.6 amino acids. Considering ±5 amino acids forwards and backwards should cover, in total, about 3 turns in α-helices, which is a sufficient scale for kink detection. Alternatively, instead of imposing a hard cutoff, we can use a Gaussian cutoff.

Molecular dynamics simulations give us a trajectory file which holds all the atom positions in 3D space for every frame k. Since three non-co-linear points in 3D space define a plane, the positions of N-C_α-C atoms in each residue can define a plane and its normal vector $\vec{n}$ as shown in Figure 2(c). We compute normal vectors ($\vec{n_i}$) to the planes formed by these N-C_α-C atoms in residues (R_i) for every frame k.

Specifically, we model the interactions amongst neighboring amino acids for time step k by an affinity matrix $\mathbf{A}_k$. Each entry a_{ij}, of the matrix $\mathbf{A}_k$ models the strength of interaction between amino acid residues i and j. As discussed above, each amino acid residue, i, is associated with the vector $\vec{n_i}$.

$$a_{ij} = \begin{cases} \frac{\langle \vec{n_i}, \vec{n_j} \rangle}{\frac{r_s}{2}\sqrt{2\pi}} e^{-\frac{(j-i)^2}{r_s^2}} & \text{if } |j-i| < r_s \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors. This allows us to generate the

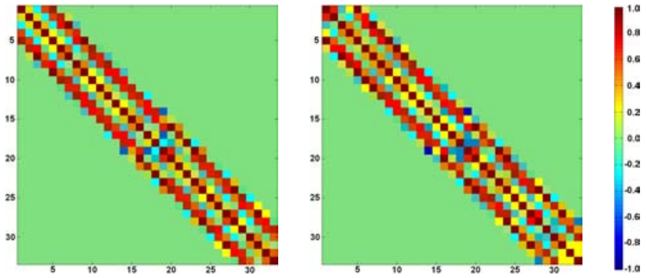


Figure 4 *Visualization of affinity matrices computed from the first and the second frames for* 33 *residues of the subunit* 1 *for E.* coli *MscS (shown in Figure 3) when the cutoff distance, $r_s = 5$ is used.*

affinity matrix $\mathbf{A}_k$ as:

$$A_k = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mm} \end{bmatrix} \tag{2}$$

Figure 4 visualizes the affinity matrices from the first and the second frames for 33 residues (from residue 94 to residue 126) of the subunit 1 for the molecule shown in Figure 3.

3.2 Saliency Detection among Neighboring Affinity Matrices

Our affinity matrix in equation 2 represents the geometric relationship among neighboring residues. This angular relationship cannot be represented by a single vector $\mathbf{x}$ as in [3]. Therefore, the dissimilarity between two given frames i and j should be computed as the difference between two affinity matrices A_i and A_j.

A naïve approach to comparing two affinity matrices A_i and A_j is to directly measure the difference. For example, one might compute $\|A_i - A_j\|_F$, where $\|\cdot\|_F$ is the Frobenius norm, and for a matrix M is defined as:

$$\| M \|_F = \sqrt{\sum_{a=1}^{m} \sum_{b=1}^{n} |m_{ab}|^2}$$

where m_{ab} is the entry in the a^{th} row and b^{th} column of M.

However, as the affinity matrices are constructed directly from the simulation data, they encode high frequency changes in atom positions that are the result of Brownian motion. Yet, we wish to identify large-scale conformational changes in the molecule. For our purposes, the high frequency information in the affinity matrix which results from Brownian motion is essentially noise. Thus, when comparing two affinity matrices, we wish to ignore such noise and to consider only more significant changes. For this purpose, we employ the Singular value decomposition (SVD), the computation of which can be found in the standard text by Golub and Van Loan [7]. The SVD factorizes a given $m \times n$ matrix A into three matrices: $A = U\Sigma V^T$, where U is an $m \times m$ orthogonal matrix ($UU^T = I$ and $U^TU = I$), Σ is $m \times n$

diagonal matrix with non-negative numbers, and V^T is the transpose of an $n \times n$ unitary matrix V. There are three properties of the SVD of which we shall take advantage:

1. U and V are a set of orthonormal basis vectors (singular vectors).
2. The diagonal entries in Σ (called singular values), are sorted in non-increasing order and indicate the importance of the corresponding basis vectors.
3. $\hat{A} = U\hat{\Sigma}V^T$, where $\hat{\Sigma}$ is the same as Σ except that all but the d largest singular values have been set to zero, is the best rank-d approximation to A in the sense that:

$$\hat{A} = \min_{\{M|\text{rank}(M)=d\}} \|A - M\|_F$$

In α-helices, backbone atoms (C, O, N, and H) are much more stable than side-chains because of the H bonds. However, there are still significant random vibrations in the positions of backbone atoms over time due to the effects of Brownian motion. By using SVD to obtain a lower-rank approximation of our original affinity matrices, we hope the to reduce or eliminate the effects of such vibrations by ignoring the contributions of the highest frequency singular vectors, where we expect their contributions to reside. This is why we use SVD as opposed to other methods for computing the uniqueness of the affinity matrix A_i relative to the affinity matrix A_j.

Uniqueness of the affinity matrix A_i relative to the affinity matrix A_j:

We use the following procedure to compare the affinity matrices A_i and A_j for frames i and j. We perform a singular value decomposition of A_i to obtain $A_i = U_i \times \Sigma_i \times V_i^T$. This returns the basis vectors as the column vectors in U_i. Since the basis vectors are sorted by their importance in SVD decomposition, we can obtain a reduced (rank d) matrix $\hat{U}_i$ by taking the first d basis vectors in U_i. For the j-th frame A_j, we use these d basis vectors to best approximate it. For this, we project A_j to the low-dimensional subspace spanned by the d basis vectors as: $W_{i,j} = \hat{U}_i^T \times A_j$. This gives us the weight matrix $W_{i,j}$ for the d basis vectors. We use this weight matrix to approximate A_j by: $\hat{A}_j = \hat{U}_i \times W_{i,j}$. Finally we compute the root mean square error (ϵ_{ij}) between $\hat{A}_j$ and A_j: $\| \hat{A}_j - A_j \|_F$.

To determine d, we use a user-specified parameter τ and choose d to be the largest integer such that $\delta_i = \frac{\|\hat{A}_i^d - A_i\|_F}{\|A_i\|_F} < \tau$; where $\hat{A}_i^d$ is the rank d approximation of A_i obtained using the truncated singular value decomposition. Note that δ_i^2 can be easily computed as $\sum_{j>d} \sigma_j^2 / \sum_j \sigma_j^2$, where σ_j is the jth largest singular value.

Saliency Value s_i for the frame i:

To compute the uniqueness of a frame i relative to other frames j, we avoid considering all possible pairs (i, j). Instead, we consider neighboring frames j where $|i-j| \le r_t$. Throughout this paper, we use $r_t = F/10$, where F is the total number of frames. The final saliency value s_i is the average of the errors ϵ_{ij} in neighboring frames of i:

$$s_i = \frac{\sum\limits_{|j-i| \le r_t} \epsilon_{ij}}{F_i} \tag{3}$$

where F_i is the number of frames whose distance from the frame i is less than or equal to r_t. Figure 5 shows the graph for these saliency values in blue.

4 Results

We have compared our detected salient frames with the ones identified independently by our collaborators (biology scientists) for molecular dynamics simulations. Figure 5 shows the five most salient frames detected by our method for the subunit 4 in the E. *coli* mechanosensitive ion channel in Figure 3. The frames 5, 26, 30, and 34 which have been detected by our method are the same or very close to the frames 3, 24, 26, 30, and 35 with changes in the kinks, which were detected manually by our collaborators. The frame 39 detected by our algorithm is not close to any frame detected manually by our collaborators, but it had the lowest saliency value among the five most salient frames. Generally, kinks change towards the end of this simulation, and our method successfully detects these important frames.

Figure 6 shows the five most salient frames detected by our method for the subunit 1 in the ion channel shown in Figure 3. This subunit is topologically identical to the subunit 4, but acts differently in the simulation. Therefore, it results in different salient frames (frames 11, 19, 21, 35, and 39) as shown in Figure 6. Our collaborators identified frames 2, 18, 20, 23, 35, 36, and 39 as being salient. Figure 7 shows the six most salient frames detected by our method for the subunit 4 in the symmetry annealing of MscS F68S mutant. In this molecular dynamics simulation, residue 68 was mutated to another, serine, which has very specific consequences for channel inactivation in real experiments. As changes in the kinks occur more frequently than the previous simulations, we observe a larger number of salient frames than in the previous cases. Our collaborators have manually identified frames 2, 4, 18, 34, and 38 as being salient. Among these, frames 2, 4, 18, and 38 are the same or close to the frames 1, 5, 18, and 39 detected by our algorithm, and the remaining frame 34 also exhibits a relatively high saliency value as shown in Figure 7.

5 Performance Considerations

Performance considerations need to be taken into account in order to make our approach feasible for large datasets. The analysis of a particular simulation may require us to consider thousands of amino acid residues, and thousands of frames. The running time of our algorithm is dominated by the computation of the SVD for each affinity matrix. If we consider r residues and F frames, this leads to a worst case complexity in $\mathrm{O}(r^3F)$. However, since our affinity matrices are highly sparse, and since we usually require use a rank d approximation of these matricies with $d \ll r$, our algorithm is far more efficient in practice. Below, we detail a number of optimizations we implemented to allow our approach to scale to problems of the size we need to consider.

5.1 Maintaining Only Local Data

For very small data sets, it is possible to first calculate the requisite information for every time step of the simulation and then to analyze the errors considering the window centered around each. Though this approach eases the implementation of our algorithm somewhat, it scales poorly to even moderately sized data. We typically expect molecular dynamics simulations to run for many time steps. Yet, we will not be able to store the necessary information for all time steps in random access memory.

To overcome this difficulty, the implementation of our algorithm stores only local data that is relevant to the analysis of a time step, k, about which the current window is centered. In particular, for time step k, it is necessary to compute $\mathbf{A}_k$, $\hat{\mathbf{U}}_k$ and for all other time steps ℓ in the window, it is necessary to compute $\mathbf{A}_\ell$. This information is sufficient to compute a saliency measure for time step k. Storing only the information necessary to analyze the current time step implies that the memory requirements can be made independent of the length of the simulation. Ideally, the size of a window is based on the physical timescale over which actions of interest are expect to occur in the molecule, and it varies independently of the number of time steps in the simulation.

5.2 Data Reuse

A second practical consideration we make in our implementation of our algorithm is the reuse of data to avoid redundant computation. If the algorithm is implemented in a naïve fashion (relevant analysis data for the entire simulation is computed at once) then this is trivially achieved. However, even when we consider only the local data, in the window W_k, relevant to the analysis of a time step k, it is possible to reuse many of the computed quantities when considering the window W_{k+1}, centered around the next time step, $k+1$. When the sliding window is moved forward by a single time step, from W_k to W_{k+1}, all but the leftmost of the affinity matrices from W_k remain relevant to the analysis of the new window. Further, since all but one of the affinity matrices from the previous window are reused, only a single new affinity matrix, corresponding to the rightmost time step in W_{k+1} need be computed. Finally, the basis vectors, $\hat{\mathbf{U}}_k$, may be discarded while $\hat{\mathbf{U}}_{k+1}$ will be computed. Thus, by reusing relevant data as the sliding window proceeds forward along the time steps of a simulation, we can ensure that, despite the fact that we only store window-local data, each affinity matrix and approximate basis is computed only once.

5.3 Exploiting Sparsity

Even if we only consider storing window-local data, memory requirements might still be exorbitant if we need to consider many amino acid residues for each time step. This is due to the fact that we will require the storage of an affinity matrix for each time step in the current window. However, since our affinity matrix considers only local interactions (residues within 5 units of each other along the protein backbone), the matrix itself is very sparse. In the affinity matrix of a given time step, k, each row will have, at most, 11 non-zero entries. Thus, by using a sparse matrix structure the memory requirements for storing an affinity matrix can be made linear, rather than quadratic, in the number of considered amino acid residues. This enables us to consider many residues for each time step while keeping feasible memory requirements.

5.4 Iterative SVD Transform Using Spectral Shift

The singular value decomposition is the most computationally intensive step of our algorithm. Yet, even this step of the algorithm can be optimized significantly by obtaining singular vectors iteratively. When obtaining an approximate basis, $\hat{\mathbf{U}}_k$, for the affinity matrix $\mathbf{A}_k$, we need enough singular vectors so that we can represent $\mathbf{A}_k$ with sufficient accuracy. However, the number of basis vectors required for the desired accuracy is often significantly less than the number of columns (or rows) of $\mathbf{A}_k$. Thus, it is wasteful and unnecessary to perform a full SVD of $\mathbf{A}_k$. Most SVD implementations allow the user to request only the D most significant singular vectors. Unfortunately, we do not know, a priori, the number of vectors that will be required to reach our desired error threshold. Additionally, most sparse SVD implementations exhibit another behavior that is undesirable. Namely, the running time of the algorithm is super-linear in the number of requested singular values/vectors. We adopt the approach suggested by Vallet and Levy [22] to overcome both of these difficulties simultaneously.

To overcome the aforementioned difficulties, we make use of the ability (available in most SVD implementations) to request the singular values and corresponding singular vectors that are closest to a particular spectral shift value, σ. We request a fixed number, $d = 50$, of singular values for each call to the SVD procedure. When we obtain the results, we find the largest (σ_u) and smallest (σ_l) singular values. Then, we compute a spectral shift $\sigma_s = \sigma_l + \lambda(\sigma_u - \sigma_l)$ for the next invocation of the SVD procedure. Here, λ is a small scalar value (we use $\lambda = 0.2$), and σ_s is computed so that there is overlap between the spectra returned by consecutive calls to SVD. Since we only request d singular vectors per invocation of the SVD procedure, we avoid the super-linear runtime in number of requested singular vectors. Furthermore, after each iteration, we obtain a more complete set of basis vectors for the affinity matrix; driving down the residual error. Thus, we can compute the residual error after each iteration, subsequently ensuring that the total number of singular vectors we obtain from the SVD is never more than $d-1$ in excess of the amount required to satisfy our error threshold.

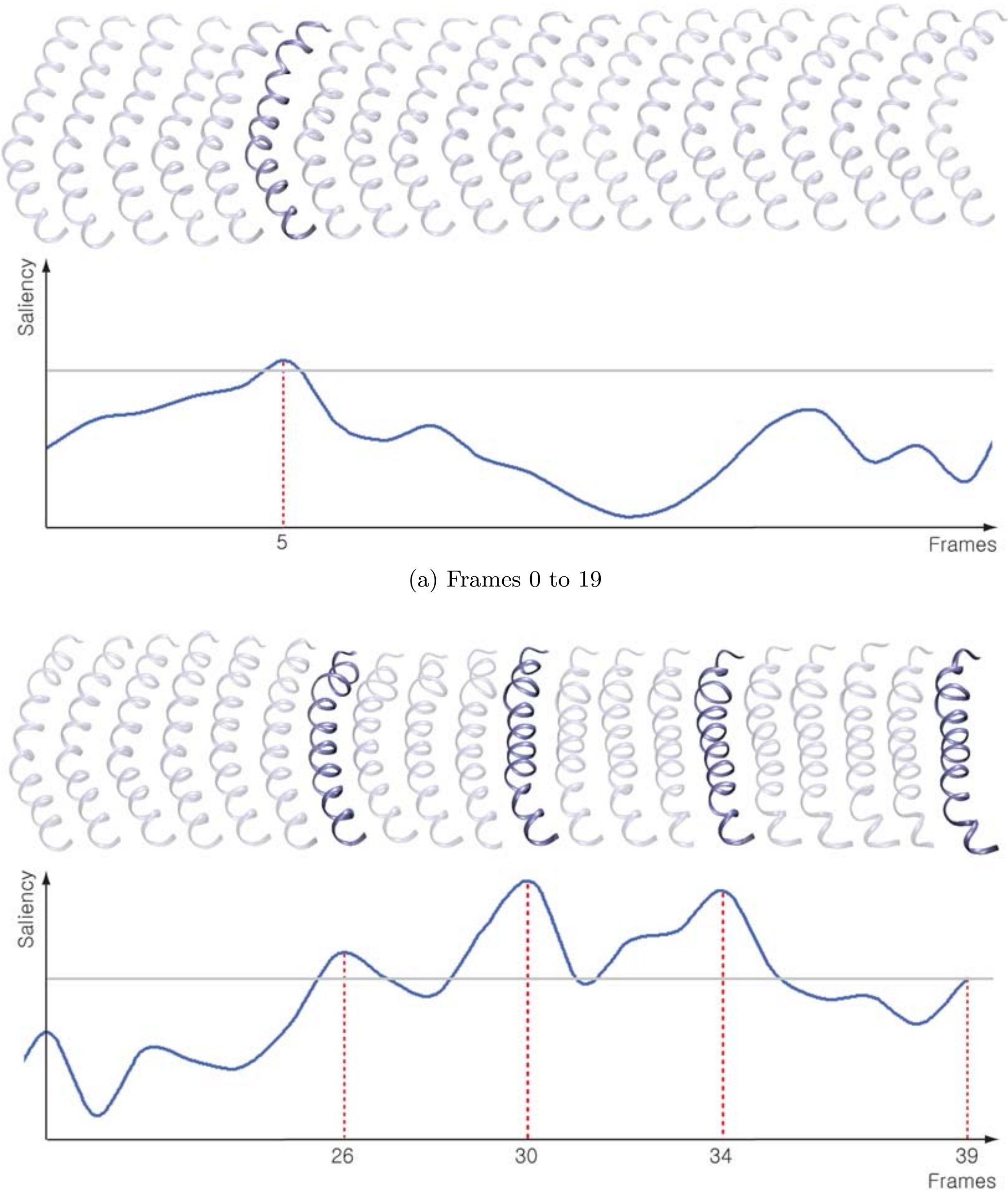
6 Conclusions and Future Work

In this paper, we have detected salient frames for molecular dynamics simulations. We have introduced the notion of saliency in time, and successfully identified most of the key frames which have changes in the kinks (i.e. appearance or disappearance of a kink) for E. *coli* channel. We believe that our method can enable researchers to focus on the important frames for further analysis of the dataset.

We currently consider the angles between the normal vectors defined by the planes of different residues in α-helices, and identify the anomalies (kinks) in the secondary structures for an E. *coli* channel. However, it makes sense to explore other structural properties of the molecule as well. For example, we could consider quantities like the rotational angles between residues or the derivatives of such quantities over time. It will be interesting to see how affinity matrices based on other quantities compare to the ones we have chosen for this work. Also, it will be interesting to explore how the approach detailed in this work might be generalized to other types of time-varying data. We believe this framework can be easily extended to encompass salient features in other time-varying simulations by changing the way we construct affinity matrices to address other needs by scientists or domain experts.

Acknowledgment

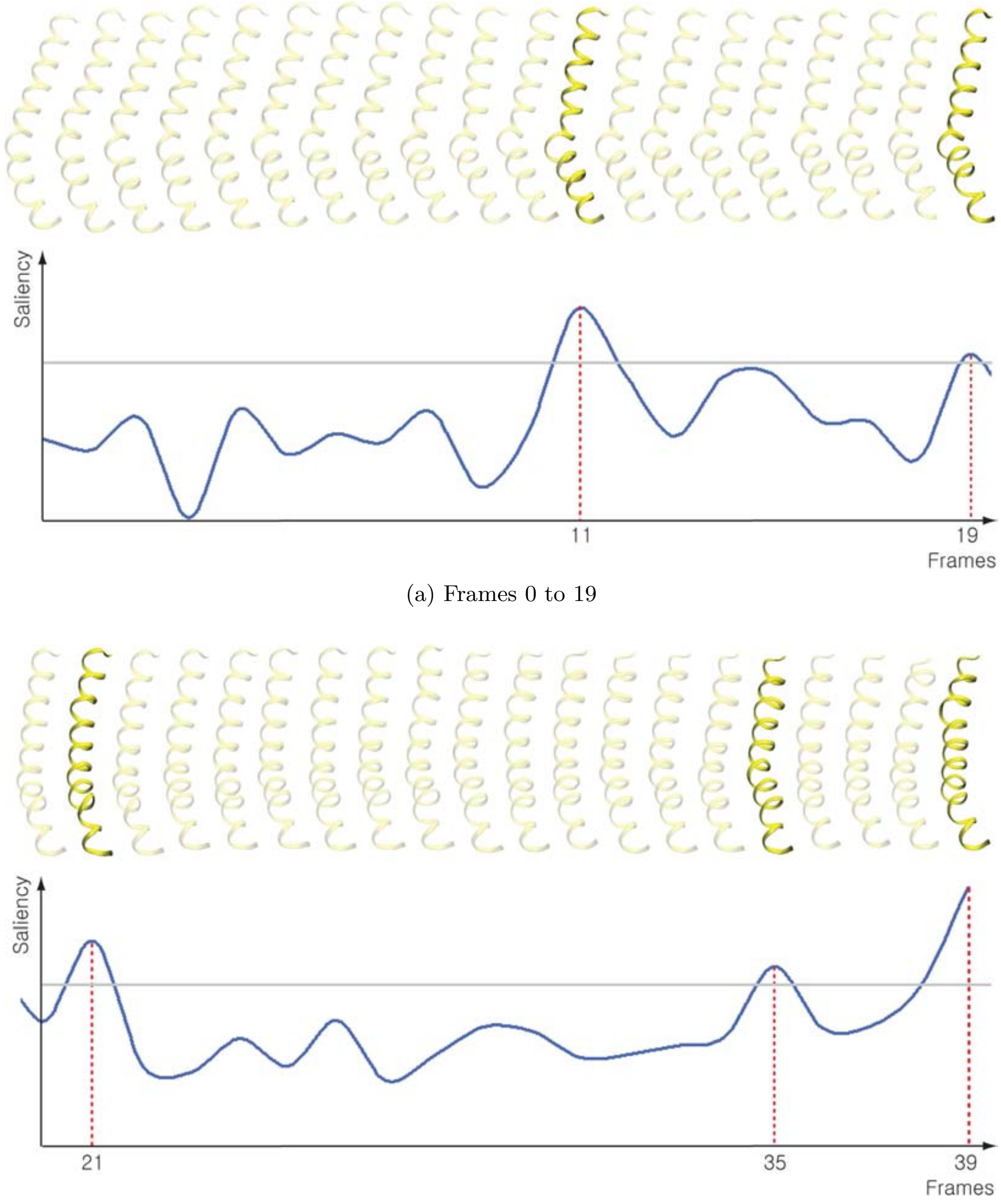
We gratefully acknowledge David Jacobs, David Mount, and Yoonkyung Kim for several valuable discussions. This work was supported in part by the NSF grants: CCF 05-41120, CCF04-29753, CNS 04-03313, and IIS 04-14699 and the NIH grants R01GM075225 and 2R01 NS03931405A. We would also like to thank the anonymous reviewers for their detailed and highly constructive comments that were extremely helpful.



(a) Frames 0 to 19

(b) Frames 20 to 39

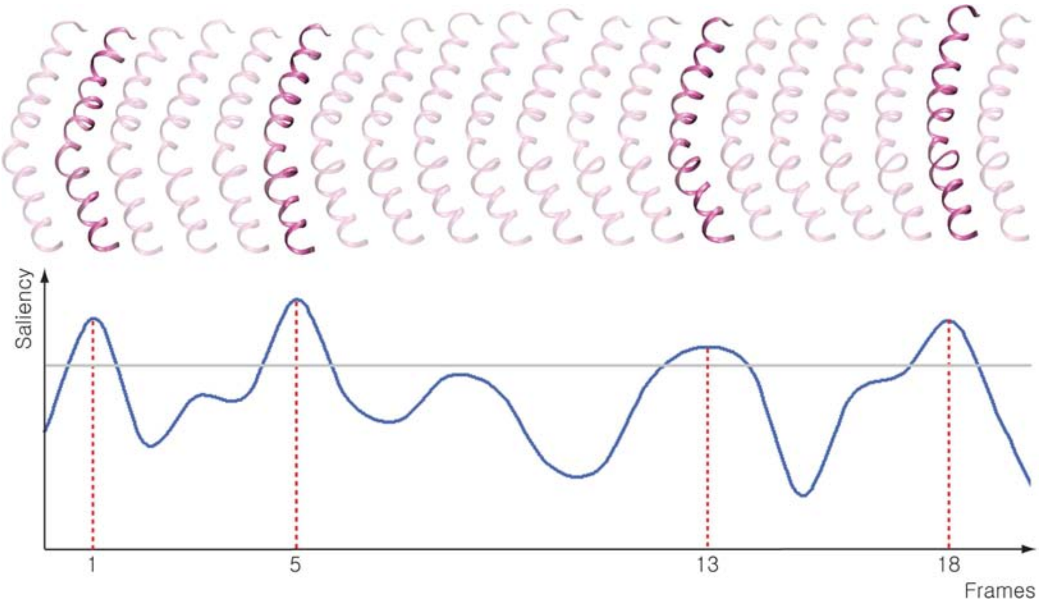
Figure 5 *Five most salient frames detected by our method for the subunit 4 in the E.* coli *ion channel (MscS) in Figure 3. The changes in the kinks are detected towards the end of this simulation, and our method successfully detects some of the most important frames.*



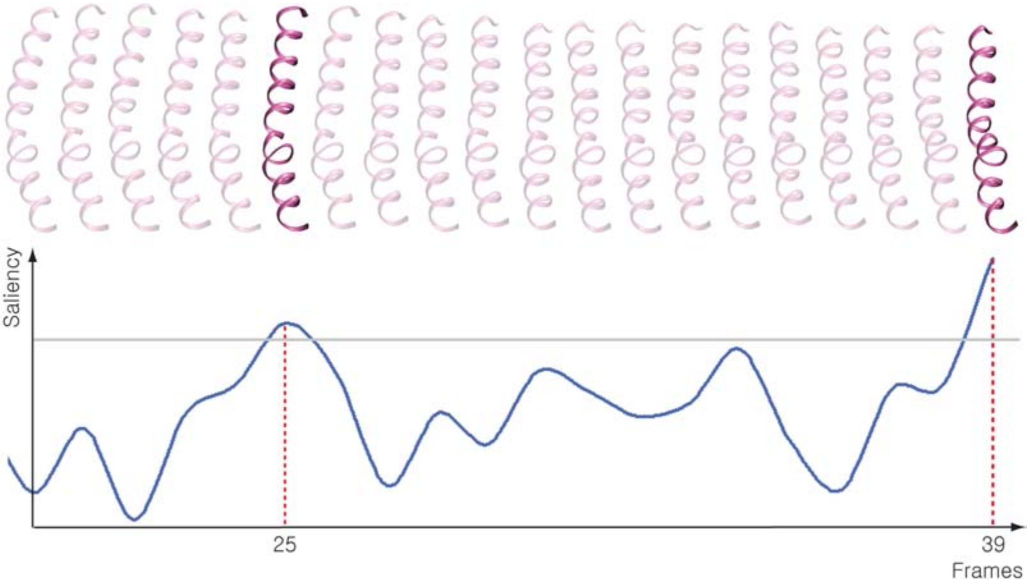
(a) Frames 0 to 19

(b) Frames 20 to 39

Figure 6 *Five most salient frames detected by our method for the subunit* 1 *in the E.* coli *ion channel (MscS) in Figure 3. This subunit is topologically identical to the subunit* 1, *but acts differently in the simulation.*



(a) Frames 0 to 19



(b) Frames 20 to 39

Figure 7 *Six most salient frames detected by our method for the subunit* 4 *in the other molecular dynamics simulation, showing the symmetry annealing of MscS F68S mutant – the residue* 68 *was mutated to another, serine, which has very specific consequences for channel inactivation in real experiments.*

References

1 B. Akitake, A. Anishkin, N. Liu, and S. Sukharev. Straightening and sequential buckling of the pore-lining helices define the gating cycle of mscs. *Nature Structural and Molecular Biology*, 14(12):1141–1149, 2007.

2 Andriy Anishkin and Sergei Sukharev. State-stabilizing interactions in bacterial mechanosensitive channel gating and adaptation. *The Journal of Biological Chemistry*, 284(29):19153–19157, Jul 2009.

3 J. Assa, Y. Caspi, and D. Cohen-Or. Action synopsis: pose selection and illustration. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)*, 24(3):667–676, 2005.

4 V. Belyy, A. Anishkin, K. Kamaraju, N. Liu, and S. Sukharev. The tension-transmitting 'clutch' in the mechanosensitive channel mscs. *Nat Struct Mol Biol*, 2010.

5 Katrin Bidmon, Sebastian Grottel, Fabian Bös, Jürgen Pleiss, and Thomas Ertl. Visual abstractions of solvent pathlines near protein cavities. *Computer Graphics Forum (EuroVis 2008 Special Issue)*, 27(3):935–942, 2008.

6 C. Branden and J. Tooze. *Introduction to Protein Structure.* Garland Publishing, Inc., second edition, 1999.

7 Gene H. Golub and Charles F. Van Loan. *Matrix computations (3rd ed.).* Johns Hopkins University Press, Baltimore, MD, USA, 1996.

8 L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine intelligence*, 20(11):1254–1259, 1998.

9 C. Johnson, R. Moorhead, T. Munzner, H. Pfister, P. Rheingans, and T. S. Yoo. NIH-NSF visualization research challenges report. Technical report, Mitsubishi Electric Research Laboratories, 2006. Computing in Science and Engineering,.

10 A. Joshi and P. Rheingans. Illustration-inspired techniques for visualizing time-varying data. In *IEEE Visualization*, pages 679–686, 2005.

11 Michael Krone, Katrin Bidmon, and Thomas Ertl. Interactive visualization of molecular surface dynamics. *IEEE Trans. on Visualization and Computer Graphics*, 15(6), 2009.

12 Ove Daae Lampe, Ivan Viola, Nathalie Reuter, and Helwig Hauser. Two-level approach to efficient visualization of protein dynamics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1616–1623, 2007.

13 C. H. Lee, A. Varshney, and D. Jacobs. Mesh saliency. *ACM Trans. on Graphics (Procs. ACM SIGGRAPH)*, 24, No. 3:659 – 666, 2005.

14 S. McCloud. *Understanding Comics – The Invisible Art.* Harper Perennial, 1994.

15 Sameep Mehta, Steve Barr, Alex Choy, Hui Yang, Srinivasan Parthasarathy, Raghu Machiraju, and John Wilkins. Dynamic classification of defect structures in molecular dynamics simulation data. In *Proceedings of SIAM on Data Mining*, 2005.

16 Sameep Mehta, Kaden Hazzard, Raghu Machiraju, Srinivasan Parthasarathy, and John Wilkins. Detection and visualization of anomalous structures in molecular dynamics simulation data. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 465–472, Washington, DC, USA, 2004. IEEE Computer Society.

17 M. Nienhaus and J. Dollner. Depicting dynamics using principles of visual art and narrations. *IEEE Computer Graphics and Applications*, 25(3):40–51, 2005.

18 G. Pingali, A. Opalach, Y. Jean, and I. Carlbom. Visualization of sports using motion trajectories: Providing insights into performance, style, and strategy. In *Proceedings Visualization 2001*, pages 75–82, 2001.

19 G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. Stereochemistry of polypeptide chain configurations. *Journal of Mol. Biol.*, 7:95–99, 1963.

20 Marco Tarini, Paolo Cignoni, and Claudio Montani. Ambient occlusion and edge cueing to enhance real time molecular visualization. *IEEE Transaction on Visualization and Computer Graphics*, 12(6), sep/oct 2006.

21 G. Turk and D. Banks. Image-guided streamline placement. In *Proceedings of SIGGRAPH 1996*, pages 453–459, 1996.

22 Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum (Proceedings Eurographics)*, 2008.

3D Reconstruction of Human Ribcage and Lungs and Improved Visualization of Lung X-ray Images Through Removal of the Ribcage

Christopher Koehler[1] and Thomas Wischgoll[2]

1 **Wright State University**
3640 Col. Glenn Hwy, Dayton OH 45431
koehler.11@wright.edu
2 **Wright State University**
3640 Col. Glenn Hwy, Dayton OH 45431
thomas.wischgoll@wright.edu

Abstract

The analysis of X-ray imagery is the standard pre-screening approach for lung cancer. Unlike CT-scans, X-ray images only provide a 2D projection of the patient's body. As a result occlusions, i.e. some body parts covering other areas of the body within this projected X-ray image, can make the analysis more difficult. For example, the ribs, a predominant feature within the X-ray image, can cover up cancerous nodules, making it difficult for the Computer Aided Diagnostic (CAD) systems or even a doctor to detect such nodules. Hence, this paper describes a methodology for reconstructing a patient-specific 3D model of the ribs and lungs based on a set of lateral and PA X-ray images, which allows the system to calculate simulated X-ray images of just the ribs. The simulated X-ray images can then be subtracted from the original PA X-ray image resulting in an image where most of the cross hatching pattern caused by the ribs is removed to improve on automated diagnostic processes.

1998 ACM Subject Classification J.3 Life and medical sciences, I.3 Computer graphics, I.3.7 Three-Dimensional Graphics and Realism

Keywords and phrases 3D Reconstruction, X-ray images

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.176

1 Background

Lung cancer accounts for more cancer deaths in the United States than any other form of cancer [5]. Unfortunately, the chances of surviving lung cancer drop drastically if it is not detected early enough. 3D CT scans are a more effective visualization technique for detecting lung cancer than X-ray images [6], however they – besides a significantly higher dose of radiation, are very expensive in comparison to use for early screening of patients who do not know if they have the disease. For this reason, traditional PA and lateral X-ray images are still the most used method for lung cancer early detection. A PA X-ray image is taken from behind a patient and shows the posterior (rear) and anterior (front) portion of the ribs, whereas a lateral X-ray image is taken from the side of a patient.

Computer aided disease detection (CAD) software is an effective tool to aid medical professionals in the difficult task of detecting lung cancer at its early stages in X-ray images. Typically CAD software focuses on the PA X-ray images, and uses image processing and soft computation techniques to identify potentially diseased areas. However, the anatomy of the upper body, such as the ribs, can obscure some of the cancerous nodules making it

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 176–187
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

difficult for the CAD software to detect such nodules. Hence, removing the ribs beforehand can improve the overall result.

It would be ideal to be able to use the available X-ray images that are already being collected to generate an approximate 3D reconstruction of the ribs and lungs and then use the final reconstructed geometry to remove the ribs from the original X-ray image. This paper presents a unique approach for generating such a reconstruction that combines the user's knowledge of the general shape of human ribs and lungs with X-ray image segmentations and 3D geometric templates to generate patient specific ribcage and lung reconstructions. Furthermore, the final reconstructions are voxelized and subtracted from the PA X-ray image in order to remove the ribs, thereby allowing a user to see the lungs more clearly.

2 Related Work

There have been several techniques presented in the literature for creating 3D reconstructions based on X-ray image data. Statistical shape models and atlases were used to reconstruct a 3D pelvis based on two X-ray images [8]. Several techniques focusing specifically on creating ribcage reconstructions to aid in treatment of patients with scoliosis have also been presented. Patient specific scoliotic 3D bone models were generated based on free form deformation, one lateral X-ray image and two PA X-ray images taken at different angles [3]. Also, X-ray image based reconstructions were done for scoliosis patients by using an energy minimization function to select from a database of probabilistic prior models. While the application is entirely different, the techniques to reconstruct buildings in 3D based on architectural drawings described by Yin et al. [16] also have a lot in common with reconstructing anatomical structures based on X-ray images. More recently, Dworzak et al. reconstructed the human ribcage from a set of two images based on a statistical shape model [4].

For segmenting the rib borders in PA X-ray images, several techniques have been proposed by other researchers. Multiple vertical portions of X-ray images have been mapped to sinusoidal functions to detect rib edges [11]. Approximate rib borders have been found with the Hough transform and then refined with R-Snakes [17]. Statistical ribcage models have been aligned to patient specific data [15]. Iterative pixel classification has also been used to segment rib borders [9]. Semi-automatic segmentation of both the anterior and posterior ribs has been achieved with directional filtering and parallel edge following [10]. Since it is the only published algorithm capable of segmenting entire ribs in PA X-ray images, the technique described in [10] was used to generate segmentation data to test our reconstruction algorithms.

Our lung reconstruction algorithm is based on interactively refining the Boolean intersection of the convex hull of a reconstructed ribcage and a 3D extrusion of the lung fields detected in the PA X-ray image. Thus, a reliable method for segmenting the lungs is a necessary part of the reconstruction process. There have been several methods to segment lung fields presented in the literature in the last several years. Adaptive-sized hybrid neural networks were used to classify pixels as being part of a lung or not based on measurements taken at that pixel [13]. A combination of pixel classification and rule-based segmentation has also been effectively used to segment lung fields in X-ray images [14].

3 Rib Subtraction

One promising application of a 3D reconstruction of the ribs and lung fields is to subtract the ribs from the original X-ray images. The rib subtraction algorithm utilizes knowledge gained during both the segmentation and reconstruction processes. The goal of the algorithm is to reduce intensities at pixels which correspond to ribs in the original PA X-ray image in order to significantly reduce the confusing cross-hatching pattern that the ribs leave in the images without altering any other data contained in the image. The algorithm consists of several steps. First, the 3D reconstructions of both the ribs and the lung fields have to be generated based on a patient specific set of X-ray images. For this, the ribs and lung fields have to be segmented within the X-ray images. Based on these results, a 3D geometric representation of the ribs and the lung fields can be derived. Using these geometric representations, an X-ray image can be simulated that only contains a projection of the ribs by emulating the way an X-ray would transverse the ribs and the lung fields. This simulated X-ray image is then subtracted from the original PA X-ray image, thereby removing the majority of the rib silhouettes from that image. This cleaned up image can then be used for further diagnostic processing, such as analysis using CAD software. The individual steps of the algorithm are discussed in more detail in the following sub-sections.

3.1 Segmentation

The algorithms to reconstruct both the ribs and the lungs involve alignment of 3D models to 2D segmentation data from the X-ray images and also generation of 3D meshes by extruding certain 2D features found in the images. Hence, it is important to start with accurate segmentations of the X-ray images. In particular it is necessary to segment the lung fields and the rib borders in the PA X-ray image, and the outer border of the ribcage in the lateral X-ray image. Segmenting anatomical features in 2D X-ray images is a difficult task due to the large genetic variations between patients as well as the subtle intensity variations resulting from bone, organs and soft tissue being superimposed onto each other during the X-ray process. Our solution is to use a combination of both automated and interactive segmentation techniques that integrate some of the best features of previously published segmentation work with our own unique algorithms to detect all necessary features from a pair of X-ray images.

The segmentation of the rib borders is based on the approach by Plourde et al. [10]. The 3D reconstruction process also requires some data about the rib structure from the lateral X-ray image. Lateral X-ray images can be extremely difficult to interpret due to ribs from both sides of the patient as well as the vertebrae all being superimposed on top of each other, so a completely automated segmentation is not feasible due to the high amount of intuition required for this task. However, since the general shape of a rib is known ahead of time, it is only necessary for a technician to interactively identify one point from each rib along the rear border of the ribcage and one point on the front border in the lateral X-ray image. A b-spline curve is used to interpolate between the marked points, which forms an approximate segmentation of the ribcage boundary. A local window of histogram equalization and contrast adjustment follows the user's mouse cursor throughout the process to make it easier to identify ribs in the noisiest portions of the image. The results of the rib segmentations are shown in Figure 1.

In order to eliminate the ribs from the PA X-ray image, an accurate reconstruction of the lung fields is needed in addition to the segmented ribs. The lung fields appear as two large dark patches in the center of the images, surrounded by a lighter area due to the lateral ribs, shoulder bones, organs and soft tissue. The corners of PA X-Ray images are often dark due to the patient's bodies not extending to the image corners. The first step of the lung segmentation algorithm is to identify pixels near the center of both lungs, separate approximate lung field boundaries from any other dark patches in the image and then later

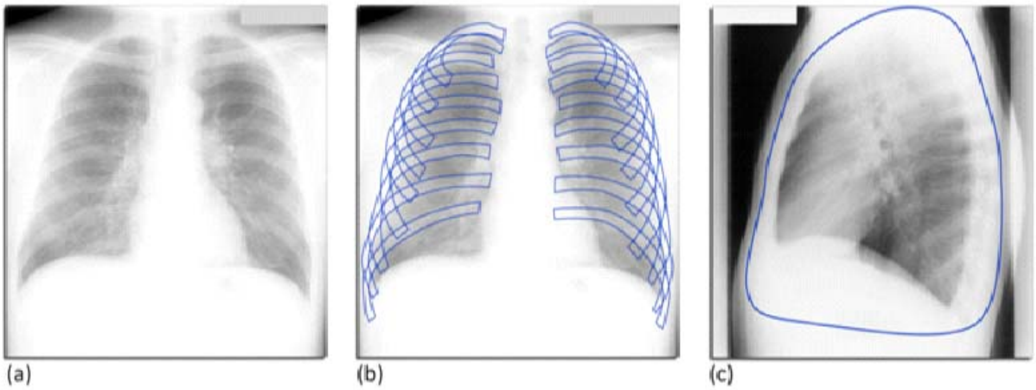


Figure 1 Rib segmentation process and results. (a) Initial PA X-ray image. (b) Segmentation done based on Plourde's edge following method. (c) Ribcage boundary segmentation.

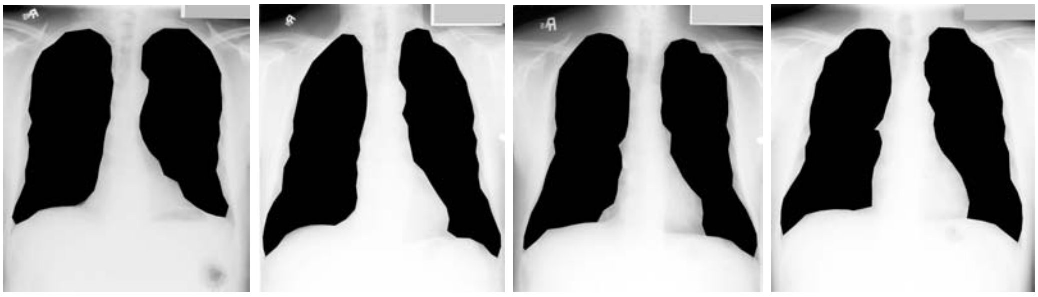


Figure 2 Lung segmentation for four different patient data sets.

attempt to refine the segmentation border.

As a starting point, the average pixel intensity in the PA X-ray image is taken as a threshold value. All pixels below this threshold are marked black and all those above are marked white. Then, image intensities are examined along every scan line of the image; white pixels closest to its center are assumed to be on the spinal cord and black pixels to the left and right of this white area are assumed to belong to the lung fields. The threshold value is gradually reduced until there are no longer any paths of black pixels from the lung field centers to the image borders. All black pixels that are connected to the image borders are then set to white. To smooth the lung borders and to remove any remaining noisy spots in the image, Gaussian smoothing is performed with a large value of sigma, and thresholding is done again resulting in another binary image. Any remaining black patches except the two largest are then removed.

Based on this segmentation, a contour line is generated consisting of a series of connected vertices which are refined to more closely match the true lung boundaries using Canny edge detection [2]. The contours are then recursively traced and discretized into a series of vertices. The bottom inner portions of the discretized lung segmentations are refined with an active contour based technique. The slope of the tangent at each discretized vertex is taken as the slope of the line between the next and previous vertices. The negative inverse of the slope is then used to find endpoints of a line segment orthogonal to the current pixel and the endpoints are connected. The intensities of the original PA X-ray image underlying each point on this new line segment are found resulting in an intensity histogram, which is smoothed with a Gaussian kernel. The first derivatives of the intensity histogram are computed. The pixel with the maximum rate of intensity change is assumed to be part of the new discretized boundary of the lung field if the intensity change at that pixel exceeds a threshold. Parameters such as this threshold, the length of the orthogonal line segments, and the smoothing filter kernel size were determined experimentally in order to achieve the closest match with expert segmentations. Figure 2 shows results of the lung segmentation and discretization process for four different sets of X-ray images.

3.2 3D Reconstruction

The ribcage is reconstructed first based on our previous work [7]. Using a rib template of the shape of a half-torus, the geometry of the individual ribs is reconstructed by moving the vertices of the template to line up with the projections as defined by the segmentation results of the lateral and PA X-ray images. The basic idea is to exploit the fact that the overall shape of a rib is known. Hence, this shape, represented by the rib template, has to be matched with the segmentation found in the X-ray images. Specifically, the template is split up into clusters of vertices, where each of these clusters forms a ring around the rib. These clusters are first moved to match the location of the corresponding rib within the PA X-ray image. In a second step, the clusters of vertices are aligned with the segmentation results of the lateral X-ray image. This then results in a geometric representation of each rib as shown in the first row of Figure 4.

The geometric lung reconstruction utilizes both the rib reconstruction data that has already been generated and the lung segmentation data. Essentially it works by creating a mesh that encompasses the inside of the reconstructed ribcage, and then carves out the portion of that mesh which corresponds to the lungs by extruding the segmented lungs up through it. The resulting mesh is then refined with polygon reduction and subdivision smoothing so that its surface is smooth like that of a real lung.

Earlier in the section on lung segmentation, it was mentioned that the segmentation boundary was discretized into a series of vertices. The reason for doing this is that those vertices can now be extruded into 3D to generate a shape that is the basis for reconstructing the lungs. The result is two tube-like structures that resemble lungs from the front, but are completely flat along the sides. The reason for choosing a discretization interval that is coarser than the available accuracy of the segmentation is that it would create a mesh with many extremely skinny triangular faces later when it is merged with another mesh whose accuracy is already limited by a lack of data between ribs.

True lungs do not have flat sides, but rather they curve along with the interior of the ribs. The extrusion of the lung fields alone does not capture this fact, and there is no data that can be taken directly from the X-ray images to generate this structure. What can be used is the reconstructed ribcage that has already been generated at this point. Vertices along the interior of each reconstructed rib are found and a new mesh is generated that fills the interior of the ribcage by computing the Delaunay tetrahedrization [12] [1] of these vertices.

The new mesh resulting from the Delaunay tetrahedrization captures approximately how the outer border of the lungs follow the inside of the ribcage, but it is solid in the center and near the bottom where the trachea, diaphragm, vertebrae and heart should be located. The silhouette of these features shows up white in the PA X-ray image so they are not included in the lung segmentation. Thus, the next step of the lung reconstruction is to carve the lung fields out by taking the Boolean intersection of the two meshes. The lung reconstruction process up to this point is illustrated in Figure 3.

The initial reconstructed lung mesh contains hard edges at the boundaries of where the ribcage interior and lung extrusion meshes intersected each other. This can be fixed through

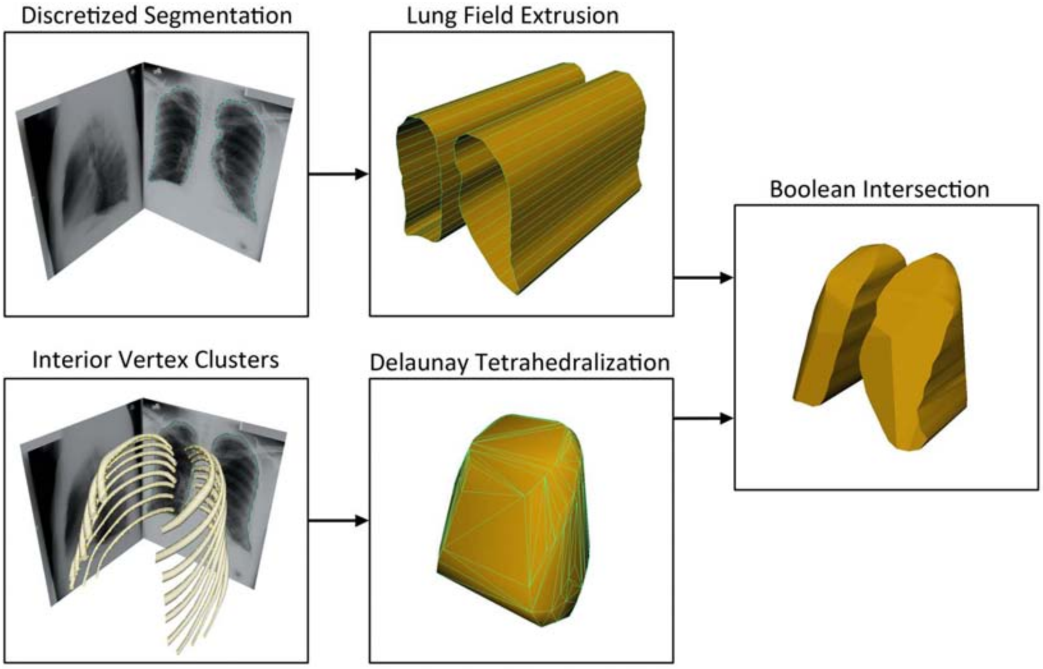


Figure 3 Initial lung reconstruction process. Top: The discretized lung segmentation is extruded into two hollow lung fields. Bottom: A polygon mesh is generated by taking the Delaunay tetrahedrization of the interior vertex clusters of each rib after the rib reconstruction is complete. Right: The initial reconstructed lung meshes are created by taking the Boolean intersection of the extruded lung fields and the Delaunay tetrahedrization.

a combination of polygon reduction and subdivision smoothing. This process is not entirely straightforward though because despite our best efforts to limit it at several steps in the reconstruction, there will inevitably be a variable number of strangely shaped polygons in the initial mesh of the reconstructed lungs. Subdivision smoothing alone produces undesirable results due to the high variation in the faces of the mesh. Even with polygon reduction performed first to make the faces more uniform, artifacts of the reconstruction process will manifest themselves in the final smoothed model for variable percentages of polygon reduction, making it difficult to hardcode a set reduction percentage. Hence, the current software allows for some user intervention at this point ensuring an optimal choice to yield an appropriate reconstruction of the lung volume.

The rib and lung reconstructions as well as the segmentation methods were tested on 19 pairs of PA and lateral X-ray images. Results from four of the patients with significantly different bone structures are shown from several views in Figure 4. In the future we hope to further validate the results of the lung reconstruction algorithm against a CT scan of a human torso by creating simulated X-ray images through volume renderings, performing rib and lung reconstructions and then comparing the results to the original CT.

3.3 X-Ray Simulation

To get an estimate of how much intensity should be subtracted from each pixel in the original X-ray images, simulated X-ray images of both the ribs and the lungs are generated. In order to create X-ray simulations, the polygon based meshes created by the rib and lung

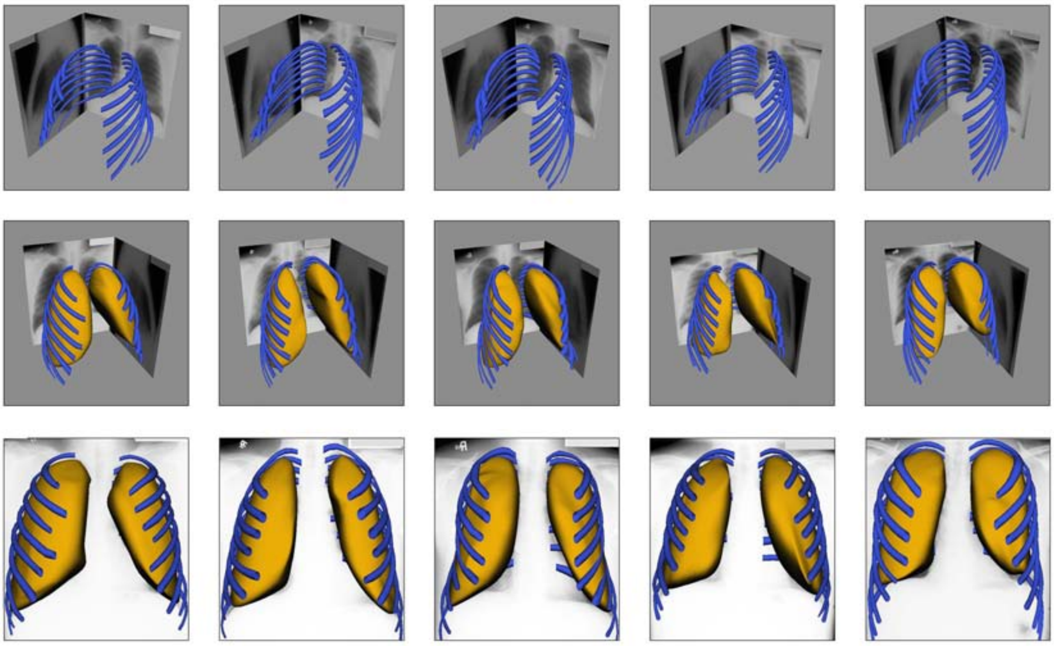


Figure 4 Renderings of final output from the rib and lung reconstruction algorithms when used on four different patients with significantly different bone structures rendered from the front, back, and two other angles.

reconstruction are first converted to voxels. For this, the geometric description of the ribs is discretized into voxels at the same resolution as governed by the input X-ray image dimensions such that every triangle in the reconstructed mesh is covered by a voxel in the resulting volumetric space. Volume renderings are then created from the voxelized reconstruction to simulate X-raying the reconstructed ribs and lungs. In order to remove any artifacts of the reconstruction, voxelization and volume rendering processes, Gaussian blurring is performed on the renderings. Then they are clipped against the original rib segmentations. These cleaned renderings are then weighted and subtracted from the original X-ray images, which effectively removes most of the ribs from the X-ray images.

The interior of the voxelized lungs was filled prior to rendering them however, the ribs are intentionally not filled after they are converted to voxels so that the edges of the rendered ribs will be lighter in order to capture the effects of real bones when they are X-rayed. Bone is porous on the inside, but much denser toward its surface, thus more intensity needs to be subtracted from the bone edges to remove them from the images. This effect can be seen in Figure 5.

A rib removal scheme needs to ensure minimal alterations of the original lung data in the X-ray images and not just softening the effect of the ribs. Intensity subtraction based on the rendering of the voxelized rib and lung reconstruction data helps achieve both of these goals. Rib removal is only done inside the original lung segmentations because the lungs are the most important part of the PA X-ray images for applications like disease detection, so it is not important to remove the lateral ribs as nothing of interesting is located behind them as far as detecting cancerous nodules is concerned.

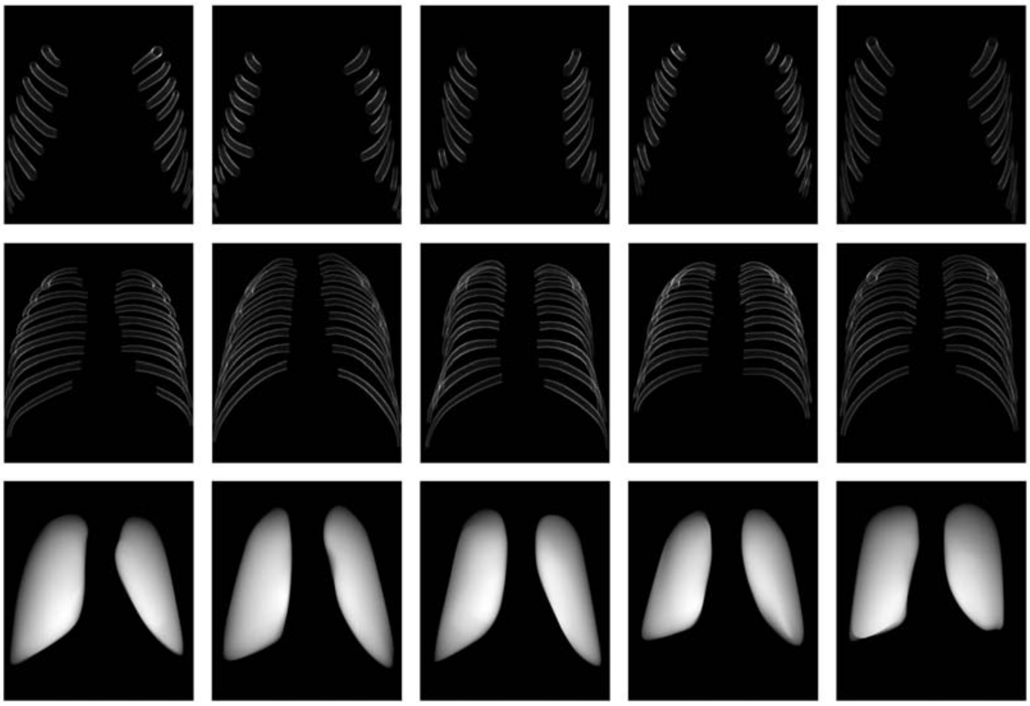


Figure 5 Simulated X-ray renderings of the anterior (top), posterior (center), and lungs (bottom).

3.4 Rib Splitting

Since the effect of the anterior portion of the ribs in the original X-ray images is very subtle, they are rendered separately from the posterior ribs by splitting the reconstructed rib meshes into posterior and anterior portions before they are voxelized (Figure 5). In order to reduce the intensities of each pixel in the anterior rib rendering, it is desirable to know how thick the lung would have been at that pixel. This is what the simulated lung X-ray image mentioned in the previous section is used for. The anterior rib subtraction process is the same as what is done for the posterior ribs; however, their effect is reduced proportionally to the intensity of the lung X-ray simulation at each pixel and a different user controlled weight is applied to them. These renderings are weighted and subtracted from the original X-ray images, which effectively removes the ribs from the X-ray images.

One side effect from both the posterior and anterior rib removal is a slight dark boundary around where the ribs used to be. This is due to a partial pixel effect combined with slight inaccuracies in the original segmentation of the ribs. Despite this side effect, the overall result reduces the effect of the ribs in the PA X-ray image and makes it easier to get a better look at the lungs. Pairs of PA X-ray images for four patients are shown before and after rib subtraction in Figure 6.

3.5 Automation

To get the best results in the rib removal, a technician adjusts two sliders controlling the weights applied to the posterior and anterior rib renderings before subtraction. This process can be done in under a minute so it is not a hindrance, but preliminary experiments have been done on how to automatically generate the best rib subtraction images.

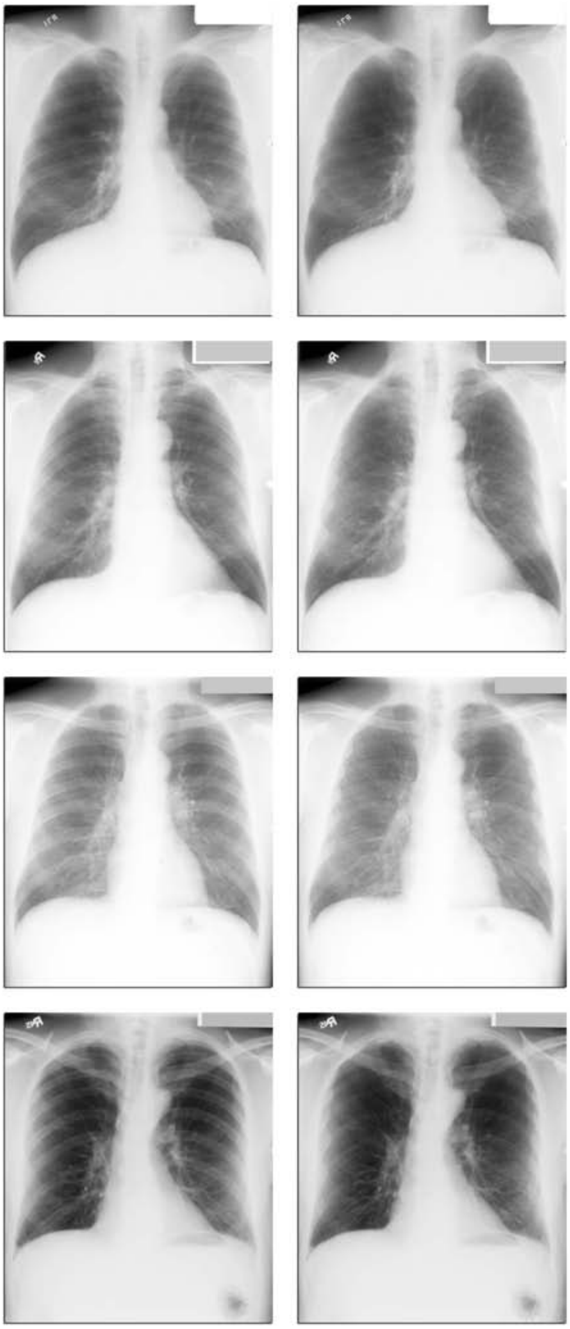


Figure 6 Original (left) and X-ray images with ribs subtracted (right).

Second-order gray-level statistics were chosen as a starting point to automatically discern the quality of a rib subtraction due to the fact that they can look at pairs of pixels that are both inside and outside of a rib. In order to compare second-order gray-level statistics before and after the rib subtraction, the co-occurrence matrix of each image must be generated. The co-occurrence matrix $H(m, n)$ is defined as the number of image locations (r, c) for which the corresponding pixel has intensity m and the pixel at location $(r + \Delta r, c + \Delta c)$ has intensity n.

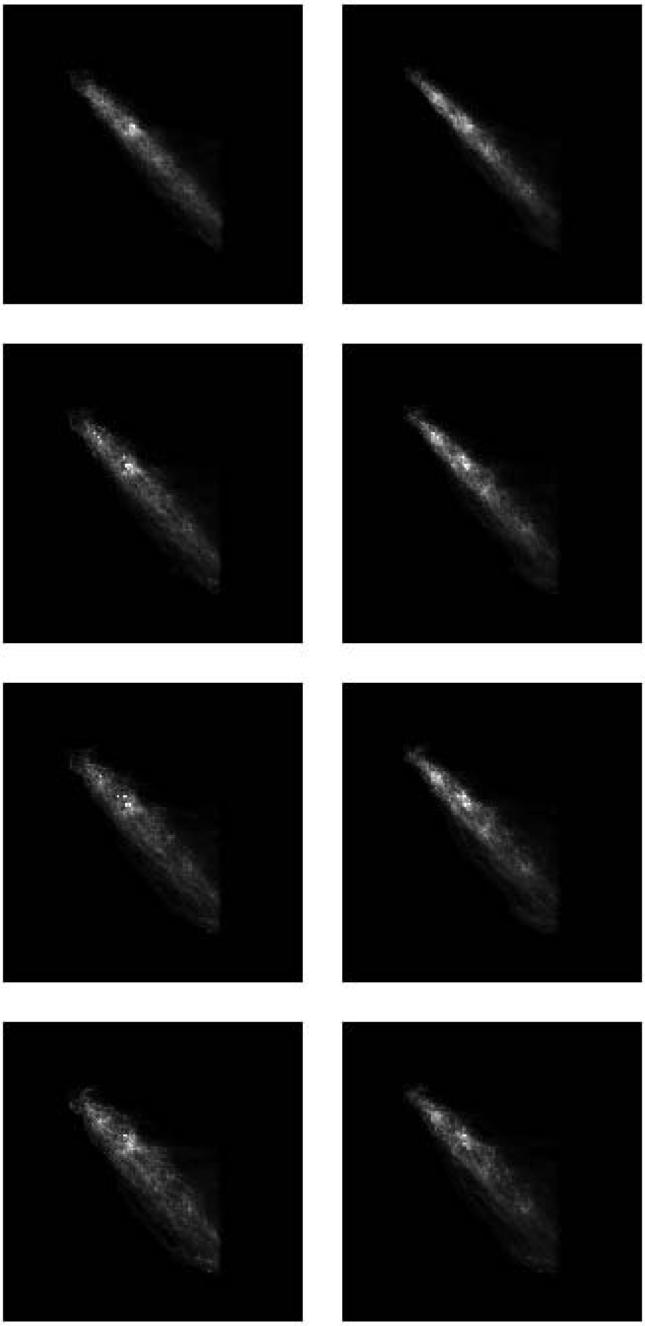


Figure 7 Intensity co-occurrence matrices before (left) and after (right) rib-removal for four different values of Δr.

The co-occurrence matrices generated from before and after rib subtraction for one patient are shown in Figure 7. The individual images show the matrix before (left) and after the removal of the ribs (right). In each pair Δc was set to 0 while Δr is incremented, because the rows occupied by a single rib does not change greatly at any portion of the rib. The values of Δr were chosen in hopes of maximizing the number of pixel pairs where one pixel was inside a rib and one pixel was not. Only intensities that were inside the segmented lungs were used in this step to prevent the very light lateral ribs and the very dark background from throwing off measurements. The co-occurrence matrix should depict mainly diagonal entries when there is little intensity variation in the measured image. As can be seen from the images, there is a slight improvement in that the light diagonal strip shown in the images becomes slightly thinner and more distinct in the matrices on the right side of Figure 7. However, all the second-order grey level statistics based on these co-occurrence matrices tested thus far, such as energy, entropy, correlation and contrast, have been insufficient to automatically discern the quality of a rib removal, so more work is needed in this area to achieve a fully automated approach.

4 Conclusion

We have developed an algorithm that generates geometric reconstructions of the ribs and lungs based on a pair of PA and lateral chest X-ray images by combining segmentation and geometric templates. Using the geometric reconstructions of the ribs and lungs, simulated X-ray images can be generated and subtracted from the original PA X-ray images to eliminate the ribs. This then can help in the diagnostic process where ribs overlaying cancerous nodules may result in misdiagnosis when analyzed by a physician or using computer aided detection software.

Medical professionals will ultimately evaluate whether such reconstruction techniques are useful as a supplement to increase the accuracy of current disease detection methods. Any technique that can add even a small percentage decrease in the number of false positive and false negative diagnoses is of great value, which is where the removal of the ribs can help by reducing the effect of obstructing anatomic elements on the detection of cancerous nodules.

Acknowledgments

The authors thank Riverain Medical for providing the data sets used in this paper as well as the Ohio Department of Development and the Dayton Area Graduate Studies Institute (DAGSI) for providing funding for the Early Lung Disease Detection Alliance to pursue the presented research agenda.

References

1 Joe Barry. Construction of three-dimensional Delaunay triangulations using local transformations. *Comput.Aided Geom.Des.*, 8(2):123–142, 1991.

2 John Canny. A Computational Approach to Edge Detection. *IEEE Transactions onPattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.

3 S. Delorme, Y. Petit, J. A. de Guise, H. Labelle, C. E Aubin, and J. Dansereau. Assessment of the 3-D reconstruction and high-resolution geometrical modeling of the human skeletal trunk from 2-D radiographic images. *IEEE Transactions on Biomedical Engineering*, 50(8):989–998, 2003.

4 J. Dworzak, H. Lamecker, J. von Berg, T. Klinder, C. Lorenz, D. Kainmu"ller, H. Seim, H. C. Hege, and S. Zachow. 3D reconstruction of the human rib cage from 3D projection images using a statistical shape model. *International Journal of Computer Assisted Radiology and Surgery*, 2009.

5 Centers for Disease Control and Prevention. Cancer – Lung Cancer Statistics, January 7 2009.

6 M. Kaneko, K. Eguchi, H. Ohmatsu, R. Kakinuma, T. Naruke, K. Suemasu, and N. Moriyama. Peripheral lung cancer: screening and detection with low-dose spiral CT versus radiography. *Radiology*, 201(3):798–802, December 1 1996.

7 C. Koehler and T. Wischgoll. 3-d reconstruction of the human ribcage based on chest x-ray images and template models. *IEEE MultiMedia*, 17(3):46–53, 2010.

8 Hans Lamecker, Thomas H. Wenckebach, and Hans-Christian Hege. Atlas-based 3d-shape reconstruction from x-ray images. *International Conference on Pattern Recognition*, 1:371–374, 2006.

9 M. Loog and B. Ginneken. Segmentation of the posterior ribs in chest radiographs using iterated contextual pixel classification. *IEEE Transactions on Medical Imaging*, 25(5):602–611, 2006.

10 F. Plourde, F. Cheriet, and J. Dansereau. Semi-automatic detection of scoliotic rib borders using chest radiographs. *Studies in Health Technology & Informatics*, 123:533–537, 2006.

11 Shigeru Sanada, Kunio Doi, and Heber MacMahon. Image feature analysis and computer-aided diagnosis in digital radiography: Automated delineation of posterior ribs in chest images. *Med.Phys.*, 18(5):964–971, September 1991 1991.

12 Hang Si. TetGen: A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator. howpublished: http://tetgen.berlios.de/.

13 Osamu Tsujii, Matthew T. Freedman, and Seong K. Mun. Automated segmentation of anatomic regions in chest radiographs using an adaptive-sized hybrid neural network. *Medical physics*, 25(6):998–1007, 1998.

14 B. van Ginneken and B. M. ter Haar Romeny. Automatic delineation of ribs in frontal chest radiographs. *Proceedings of SPIE – The International Society for Optical Engineering*, 3979:825–836, 2000.

15 Bram van Ginneken and Bart M. ter Haar Romeny. Automatic segmentation of lung fields in chest radiographs. *Medical physics*, 27(10):2445–2455, 2000.

16 Xuetao Yin, P. Wonka, and A. Razdan. Generating 3D Building Models from Architectural Drawings: A Survey. *Computer Graphics and Applications, IEEE*, 29(1):20–30, 2009.

17 Zhanjun Yue, A. Goshtasby, and L. V. Ackerman. Automatic detection of rib borders in chest radiographs. *IEEE Transactions on Medical Imaging*, 14(3):525–536, 1995.

A Visual Approach to Analysis of Stress Tensor Fields*

Andrea Kratz[1], Björn Meyer[1], and Ingrid Hotz[1]

1 **Zuse Institute Berlin (ZIB)**
Department Visualization and Data Analysis
`{kratz, bjoern.meyer, hotz}@zib.de`

Abstract

We present a visual approach for the exploration of stress tensor fields. In contrast to common tensor visualization methods that only provide a single view to the tensor field, we pursue the idea of providing various perspectives onto the data in attribute and object space. Especially in the context of stress tensors, advanced tensor visualization methods have a young tradition. Thus, we propose a combination of visualization techniques domain experts are used to with statistical views of tensor attributes. The application of this concept to tensor fields was achieved by extending the notion of shape space. It provides an intuitive way of finding tensor invariants that represent relevant physical properties. Using brushing techniques, the user can select features in attribute space, which are mapped to displayable entities in a three-dimensional hybrid visualization in object space. Volume rendering serves as context, while glyphs encode the whole tensor information in focus regions. Tensorlines can be included to emphasize directionally coherent features in the tensor field. We show that the benefit of such a multi-perspective approach is manifold. Foremost, it provides easy access to the complexity of tensor data. Moreover, including well-known analysis tools, such as Mohr diagrams, users can familiarize themselves gradually with novel visualization methods. Finally, by employing a focus-driven hybrid rendering, we significantly reduce clutter, which was a major problem of other three-dimensional tensor visualization methods.

1998 ACM Subject Classification I.3.3 Picture/Image Generation

Keywords and phrases Tensor Field, Visualization and Analysis

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.188

1 Introduction

The focus of this work is the analysis and visualization of 3D stress tensor fields, which express the response of a material to applied forces. Important application areas and their interest in such data are: In material science, a material's behavior under pressure is observed to examine its stability. Similar questions also arise in astrophysics. Rock fractures caused by tension or compression, for example, are analyzed in geosciences. A medical example is the simulation of an implant design's impact on the distribution of physiological stress inside a bone [12]. Common to most of these areas is the goal of finding regions where the inspected material tends to crack. Various failure models exist, but in general they are based on the analysis of large shear stresses. Besides understanding a physical phenomenon, tensor analysis can help to detect failures in simulations where tensors appear as intermediate product. In

* This work was funded by the Emmy Noether Program of the German Research Foundation (DFG).

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 188–211
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

all these application areas, regions of interest are not necessarily known in advance. For this reason, powerful visual exploration and analysis tools are of high importance.

The complexity of tensor data makes them hard to visualize and interpret. Therefore, users tend to analyze tensor data via two-dimensional plots of derived scalars (data reduction). Although these plots simplify the analysis at first glance, they do not communicate the evolution of tensors over the whole field [19]. They might even fail to convey all information given by a single tensor. From a visualization point of view, the difficulty lies in depicting each tensor's complex information, especially for three-dimensional tensor fields. Often, visualizations are restricted to two-dimensional slices (data projection), as three-dimensional visualizations tend to result in cluttered images. However, data reduction and data projection both reduce the complex information of the tensor field to a small subset. Thus, the richness of the data is not communicated.

A further challenge, for example in contrast to vector field visualization, is the young tradition of advanced tensor visualization methods in the considered application areas. Users need to get used to the advantages of modern visualization techniques, and therefore need tools to *explore* the data so they can develop an intuition and construct new hypotheses. Therefore, it is important to link methods domain experts are already used to with novel techniques. The main challenges and the resulting goals of our work are:

- Tensor data are hard to interpret. Thus, we provide an intuitive approach to the analysis of tensor data.
- Tensor visualization methods do not have a long tradition in the application areas that are considered in this paper. Thus, we provide well-known perspectives onto these data and link them with novel visualization methods.
- A lack of a-priori feature definitions prevents the use of automatic segmentation algorithms. Thus, we allow users to *find the unknown* and let them steer the visualization process.
- The stress tensors we are dealing with are symmetric 3D tensors described by six independent variables. Thus, effectively capturing all of this information with a single visualization method is practically not feasible. We therefore employ a feature-dependent hybrid visualization combined with diagram views.

To meet these goals, we present a new access to stress tensor fields. We consider our major contributions as below:

- *Extending the notion of shape space as basic means for feature designation in attribute space.* We introduce an intuitive way of finding tensor invariants that reflect relevant features. Building upon the idea of shape space, the challenging task of translating questions into appropriate invariants boils down to a basis change of shape space. Using concepts from stress analysis and including failure models, we present invariants for stress tensor fields together with common and new visualization techniques (Figure 2). However, our approach is extendable to the analysis of various types of symmetric second-order tensors.
- *Introduction of multiple linked views to stress tensor visualization.* Previous work mostly concentrated on only two dimensions and/or one particular visualization technique. We pursue the idea of providing various perspectives onto the data and propose visual exploration in attribute and object space. The concept of shape space serves as link between the abstract tensor and its visualization in attribute space. In object space, features are mapped to displayable entities and are explored in a three-dimensional hybrid visualization.

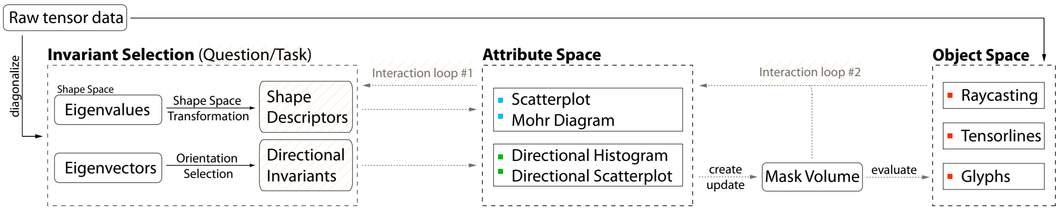


Figure 1 Tensor analysis and visualization pipeline. The basis builds the diagonalization of the tensor into its eigenvalues and eigenvectors. The first step then is the choice of appropriate shape descriptors and directional invariants steered by a specific question or task. These are visualized in attribute space. Within this space, features are selected using brushing techniques and are then encoded in a mask volume. In object space, various tensor visualization techniques are combined in a feature-driven hybrid visualization. Volume rendering provides context, and glyphs or tensorlines are placed in focus regions. The user has a variety of options to adjust the visualization (interaction loops): Shape descriptors and directional invariants can be adapted (#1) and focus/context regions can be interactively refined (#2).

2 Related Work

Besides work from tensor field visualization [18], our work is based on publications from multiple view systems [13] as well as from the visualization of multivariate data [7]. This review is structured according to our main contributions focusing on second-order stress tensors and their visualization in attribute (diagram views) and object space (spatial views).

Tensor Invariants: Central to our work is the finding that tensor visualization methods can be designed and parametrized by a specific choice of invariants, which are scalar quantities that do not change under orthogonal coordinate transformation. Considering and analyzing important invariants is common in many physical applications [9]. For analysis of diffusion tensors, [9] has been transferred to visualization [15]. In the same context, Bahn [1] came up with the definition of *eigenvalue space*, where the eigenvalues are considered to be coordinates of a point in Euclidean space. In this work, we use the term *shape space* referring to application areas such as vision and geometric modeling. Coordinates within this space describe a set of tensor invariants and are called *shape descriptor*.

Diagram Views: Only few visualization papers are related to using diagram views for tensors [24, 8]. Mohr's circle [6] is a common tool in material mechanics, being used to compute coordinate transformations. In visualization, it has been applied to diffusion tensors to depict the tensor's diffusivity [4] as well as to stress tensors [10]. Being a known technique for domain experts, Mohr diagrams can ease the access to novel visualization methods. Directional histograms have been used to visualize the distribution of fiber orientations in sprayed concrete [17] and for diffusion tensors in terms of rose diagrams and 3D scatterplots of the major eigenvector angles [40]. Despite [6], combined views for stress tensors have not been presented previously.

Spatial Views: A common classification of spatial visualization methods for second-order tensors is to distinguish between local, global and feature-based methods.

Local methods use geometries (glyphs) to depict single tensors at discrete points. Shape, size, color and transparency are used to encode tensor invariants. Dense glyph visualizations use less complex geometries together with placement algorithms [28, 21, 16]. When only selected locations are examined (probing), more complex geometries can be used. A variety of glyph types have been presented, focusing on stress tensors [19], higher-order tensors [30] and perceptual issues [25, 39]. Although, local methods have the potential to depict the whole tensor information, they generally fail in giving an overview of the complete 3D tensor field.

In contrast, global methods present an overview and emphasize regional coherence. They can be classified into methods based on scalar and vector visualization, as well as hybrid methods. Scalar visualization methods that are used to visualize tensors are ray-casting [26, 27, 22] and splatting [2, 3]. The main challenge is the design of an appropriate transfer function. Kindlmann et al. [26, 27] define an opacity transfer function based on the isotropic behavior of the tensor field. Color and shading are defined by tensor properties such as orientation and shape. Inspired by this work, Hlawitschka et al. [22] focus on directional information for transfer function design to emphasize fiber bundle boundaries. Recently, Dick et al. [12] presented a colormapping for stress tensors in order to distinguish between compressive and tensile forces.

Vector visualization methods are used to depict the behavior of the eigenvectors. We distinguish line tracing algorithms like tensorlines [38], texture-based approaches such as Line Integral Convolution (LIC) [42] and reaction-diffusion textures [33, 27]. Hotz et al. [23] presented a LIC-like method for the visualization of two-dimensional slices of a stress tensor field. They introduce a mapping of the indefinite stress tensor to a positive-definite metric. The mapped eigenvalues then are used to define input parameters used for LIC.

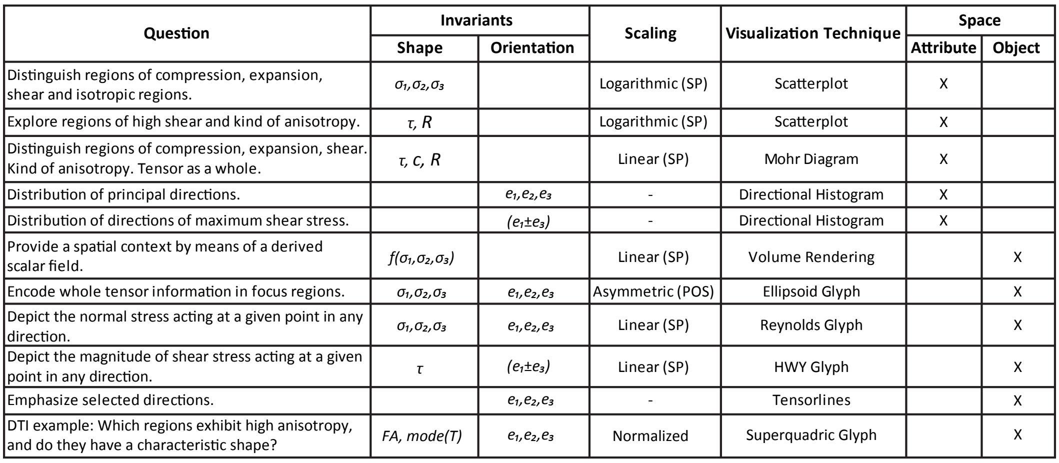
Whereas scalar-related visualization techniques are able to cover aspects of the whole 3D field, vector-related methods are mostly restricted to two dimensions. Hybrid approaches combine global and local methods [31, 34] as well as scalar- and vector-related techniques [12, 41]. Dick et al. [12] proposed hybrid visualization for 3D stress tensor fields. They combine ray-casting of the three eigenvalues with tensorlines to depict selected directions. To account for clutter, tensorlines are only seeded on a surface mesh. Although some hybrid approaches try to combine complex focus with non-disruptive context visualization, none of the existing methods allows the analysis and visualization of a complete 3D field both in detail and at large.

Feature-based methods comprise topological methods [20, 43, 44, 37] and tensor segmentation algorithms [35, 11]. Regions of similar behavior are merged, which helps to handle the complex information within a tensor field. However, automatic segmentation algorithms can only be used, if the characteristics of interesting structures can be defined in advance. They fail in describing new features and might even remove important aspects of the data [7]. Furthermore, they are hard to extend to three-dimensional tensor data.

3 Tensor Visualization and Analysis Pipeline

Multiple linked views are used to explore three-dimensional stress tensor fields. We distinguish between diagram views in *attribute space* (see Section 5) and three-dimensional spatial views in *object space* (Section 6). Both are linked over a *mask volume*, i.e., a three-dimensional data structure of the same size as the input data storing a binary value (0 or 1). The mask is created and modified by brushing tensor properties in attribute space; it is evaluated for rendering in the spatial domain. For an overview of the proposed pipeline see Figure 1.

The basis of the pipeline is the diagonalization of the tensor (Section 4.1.1). Thus, the tensor is decomposed into *shape* and *orientation*, whereas shape refers to the eigenvalues and orientation to the eigenvectors. The first step then is the choice of appropriate shape descriptors and directional invariants (Section 4). We conceive this process as translating a question into a mathematical description (Figure 2). Being supported by various views in attribute space, the user can select and substitute tensor properties (interaction loop #1)



Question	Invariants		Scaling	Visualization Technique	Space	
	Shape	Orientation			Attribute	Object
Distinguish regions of compression, expansion, shear and isotropic regions.	$\sigma_1,\sigma_2,\sigma_3$		Logarithmic (SP)	Scatterplot	X	
Explore regions of high shear and kind of anisotropy.	τ, R		Logarithmic (SP)	Scatterplot	X	
Distinguish regions of compression, expansion, shear. Kind of anisotropy. Tensor as a whole.	τ, c, R		Linear (SP)	Mohr Diagram	X	
Distribution of principal directions.		e_1,e_2,e_3	-	Directional Histogram	X	
Distribution of directions of maximum shear stress.		$(e_1 \pm e_3)$	-	Directional Histogram	X	
Provide a spatial context by means of a derived scalar field.	$f(\sigma_1,\sigma_2,\sigma_3)$		Linear (SP)	Volume Rendering		X
Encode whole tensor information in focus regions.	$\sigma_1,\sigma_2,\sigma_3$	e_1,e_2,e_3	Asymmetric (POS)	Ellipsoid Glyph		X
Depict the normal stress acting at a given point in any direction.	$\sigma_1,\sigma_2,\sigma_3$	e_1,e_2,e_3	Linear (SP)	Reynolds Glyph		X
Depict the magnitude of shear stress acting at a given point in any direction.	τ	$(e_1 \pm e_3)$	Linear (SP)	HWY Glyph		X
Emphasize selected directions.		e_1,e_2,e_3	-	Tensorlines		X
DTI example: Which regions exhibit high anisotropy, and do they have a characteristic shape?	FA, $mode(T)$	e_1,e_2,e_3	Normalized	Superquadric Glyph		X

Figure 2 Invariant Selection. The table gives examples for shape descriptors and directional invariants that correspond to a specific task or question. We mainly present invariants for stress tensors. However, our approach is extendable to various types of symmetric second-order tensors. Besides convertible invariants, the analysis of tensors from diverse application areas requires variable scalings. The abbreviations SP and POS refer to sign-preserving mappings (SP) and mappings into a positive-definite metric (POS), respectively. Furthermore, the table lists possible visualization techniques in attribute and object space.

until a set is found for being explored in more detail. Multiple views are possible at the same time, so different parameter choices and selections can be visually compared. Within these views, features are selected and highlighted using brushing-and-linking techniques (interaction loop #2). In this work, we propose the following diagrams: Shape space scatterplots can be understood as a cut through three-dimensional shape space and, thus, deliver insight into the distribution of tensor properties (Section 5.1). Mohr diagrams [10] represent the most important invariants for stress tensors (Section 4.4). Directional histograms are used to analyze the distribution of principal directions (Section 5.3), and directional scatterplots to inspect shape properties together with directions (Section 5.4).

Hybrid object space rendering (Section 6) allows the inspection of the selected features in a spatial context. The mask defines in which regions glyphs are displayed and/or tensorlines are started. Volume rendering of scalar invariants serves as a context view. If the final image does not show all relevant features, users may refine their selections in attribute space, changing the mask volume and the rendering accordingly. Selections in object space, for example of single glyphs, are part of our future work.

4 Tensor Invariants and Shape Space

In this section we formulate the task of finding relevant features in the language of shape space. Then we discuss our particular choice of shape descriptors, directions (Section 4.4) and shape space scaling (Section 4.3) for stress tensor fields.

4.1 Foundations

For the three-dimensional Euclidean space, a tensor T with respect to a basis (b_1, b_2, b_3), denoted by T_b, can be described by a matrix $M \in \mathbb{R}^{3\times3}$. That is, $T_b = M = (m_{ij})$ with i,j = 1,2,3. A tensor field over some domain D assigns a tensor $T(x)$ to every point $x \in D$.

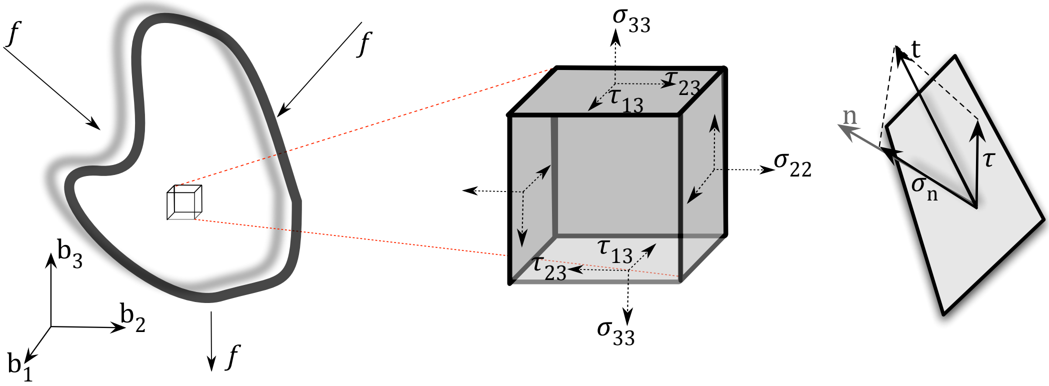


Figure 3 External forces f that are applied to a material (left), stress measured on an infinitesimally small volume element (middle), and force (traction t) acting on an cutting plane with normal vector n (right).

Tensor invariants are scalar quantities that do not change under orthogonal coordinate transformation. In general, any scalar function $f(\lambda_1, \lambda_2, \lambda_3)$ again is an invariant. Most common examples are the tensor's eigenvalues, determinant and trace.

4.1.1 Tensor Diagonalization

Tensors are *invariant* under coordinate transformation, which distinguishes them from matrices. That is, the characteristics of the tensor stay the same, independent from the choice of basis. Consequently, a tensor can be analyzed using any convenient coordinate system.

In the following, we only consider symmetric tensors, i.e., $m_{ij} = m_{ji}$, being defined by six independent components. They can be transformed into a principal coordinate system using the concept of eigenanalysis

$$U\,T\,U^T = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}. \tag{1}$$

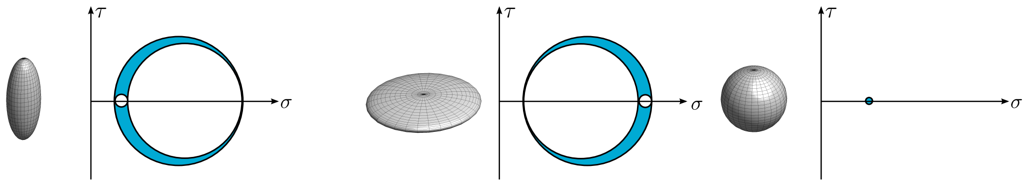
The diagonal elements λ_i are the eigenvalues and the transformation matrix U is composed of the eigenvectors e_i. For symmetric tensors, the eigenvalues are all real, and the eigenvectors constitute an orthonormal basis. They are ordered such that $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

4.1.2 Stress Tensor

A stress tensor conveys information about the stress acting on cutting planes through a material (Figure 3). It is given as

$$\sigma = \begin{pmatrix} \sigma_{11} & \tau_{12} & \tau_{13} \\ \tau_{12} & \sigma_{22} & \tau_{23} \\ \tau_{13} & \tau_{23} & \sigma_{33} \end{pmatrix}, \tag{2}$$

with the diagonal components σ_{ij} being the *normal stress* components and the off-diagonal components τ_{ij} the *shear stress* components respective to cutting planes normal to the coordinate axis. The sign of the normal stress components encodes if they are compressive



(a) Linear $\sigma_3 \approx \sigma_2 < \sigma_1$; $R \approx 1$ **(b)** Planar $\sigma_3 < \sigma_2 \approx \sigma_1$; $R \approx 0$ **(c)** Isotropic $\sigma_3 \approx \sigma_2 \approx \sigma_1$

Figure 4 Lame's stress ellipsoid (displaying all possible traction vectors) and Mohr's circle in comparison. The ellipsoid's axis are aligned with the three eigenvectors, which are scaled by the eigenvalues. For three-dimensional tensors, Mohr's circle consists of three circles drawn between the three eigenvalues [10]. The horizontal axis depicts the normal stress and the vertical axis the shear stress. The outer circle gives an impression of the maximum shear stress, i.e., the larger the circle, the greater the shear stress acting on that plane. The blue shaded area represents all possible combinations of normal and shear forces for a given cutting plane. A point within this region then corresponds to the orientation of the plane's normal.

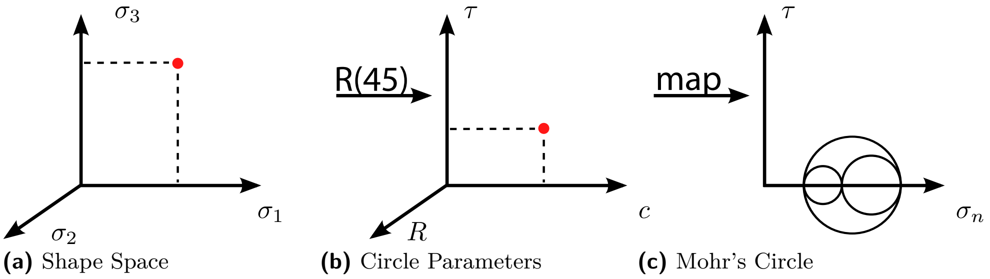
or tensile. In this paper, we interpret negative eigenvalues as compressive forces (making the volume smaller) and positive eigenvalues as tensile forces (expanding the volume). It is worth noting that in some application areas the sign is interpreted in a reverse way. If forces are balanced and there is no rotation (which is, in general, fulfilled for infinitesimally small volume elements), the tensor is *symmetric* and uniquely described by its three eigenvalues and eigenvectors (Equation (1)). In this context, the eigenvectors are called principal stress axes, and the eigenvalues are called principal stresses. As principal stresses may be positive or negative, the tensor is indefinite. The force (traction vector) t acting on a cutting plane with normal vector n is given by

$$t = \sigma \cdot n = \tau + \sigma_n. \tag{3}$$

It can be decomposed into its normal stress σ_n and shear stress component τ (Figure 3, right). In cutting planes orthogonal to the principal directions the shear stress vanishes. For planes with normals bisecting the minimum and maximum principal direction, the shear stress takes its maximum value and is called *maximum shear stress* $\tau_{\max}$. The corresponding directions are called *direction of maximum shear stress.*

4.2 Shape Space

We use the term shape space for the vector space spanned by the three eigenvalues. In this space, tensor shape [15] is represented by a point, whose coordinates are called shape descriptors. Finding shape descriptors, suiting the initial question, then corresponds to finding an appropriate reference frame (Figure 5). Common orthogonal reference frames correspond to Cartesian, spherical, and cylindrical coordinates, respectively. An example for a complete orthogonal spherical invariant set commonly used in the context of diffusion tensor imaging (DTI) is [15]: tensor norm (radius), fractional anisotropy (polar angle) and tensor mode (azimuthal angle). These descriptors represent central physiological properties (Figure 2). It is worth noting that all angular coordinates correspond to relative entities and are not defined in the origin ($\text{norm}(T)^2 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2 = 0$). For tensors with small norm, these values are unstable and sensitive to small changes. As a consequence, such coordinate systems are not optimal for indefinite tensors, for which the characteristic invariants may be positive, negative, or equal to zero (Section 4.1.2).



(a) Shape Space (b) Circle Parameters (c) Mohr's Circle

Figure 5 Shape Space Transformation using the example of Mohr's Circle. The shape space (a) is spanned by the major (x-axis), minor (y-axis), and medium eigenvalue (z-axis). The tensor's shape is represented by a point. A circle is described by its center c and its radius, which corresponds to the maximum shear τ. These shape descriptors are computed by a change of basis, which corresponds to a rotation around the σ_2-axis by 45 degrees (b). A final step corresponds to a mapping of (τ, c, R) to glyph geometry (c), whereas R distinguishes planar and linear stresses.

An additional useful property of a reference frame is orthogonality. Orthogonal invariants exhibit maximum independence of the shape descriptors by isolating changes of one invariant from variations of the others.

Which shape descriptors to use may depend on a variety of criteria. We propose the use of descriptors that give answers to specific questions and that are familiar to domain-experts. These criteria do not necessarily coincide with the mathematically most appealing choices.

4.3 Shape Space Scaling

The scale of the shape space's coordinate axes has a high impact on the visualization result. Therefore, it plays a crucial role in the diagram views (Section 5), as well as for rendering in the spatial domain (Section 6), where tensor invariants define color, transparency, glyph shape and glyph size. Most common visualization methods require positive values, which is challenging for indefinite tensors, where the sign of the invariants reveals important physical characteristics. On the other hand, most diagram views are based on positive as well as negative eigenvalues. An optimal mapping depends on the given dataset and the desired visualization. It is possible to apply the mapping before choosing appropriate shape descriptors (holds for relative entities), or afterwards. We distinguish between the following mappings:

- Sign-preserving mappings (SP): Examples are linear and logarithmic mappings (Equation 6) as well as histogram equalizations.
- Mapping to $\mathbb{R}^+$ (POS) [23, 29]: Values are mapped to the positive domain in a way that keeps the distinction between positive and negative values (Equation 7).

These mappings are further discussed in Sections 5 and 6 in the context of the specific visualization methods. The eigenvectors are already normalized and, therefore, do not need a mapping.

4.4 Shape Descriptors and Directions for Stress Tensors

Typical questions related to stress tensors are concerned with stability and failure analysis. Therefore, most failure models build on the analysis of the maximum shear stress. An example is the *Coulomb-Mohr failure criterion* [32]. Assuming no internal friction ($\mu = 0$), it

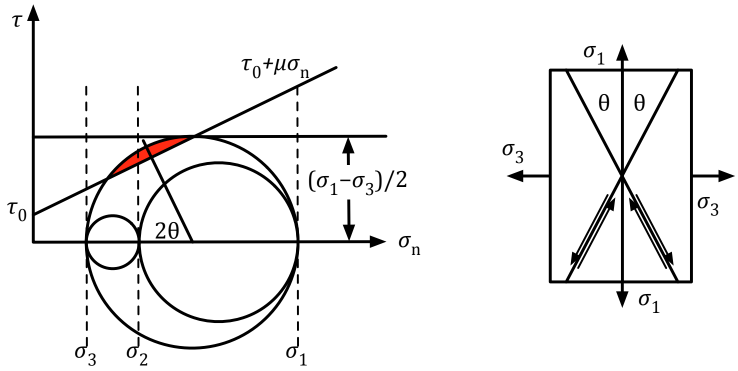


Figure 6 Coulomb-Mohr failure criterion: The red area indicates normal-shear force combinations leading to material failure (left). The relation between the principal stress directions σ_1 and σ_3 and the predicted fracture plane are given by the angle θ.

states that a material yields as long as the maximum shear stress τ falls below the *intrinsic shear strength* τ_0 of the material. Figure 6 depicts this failure criterion graphically. As long as Mohr's circle (Section 5.2) does not intersect the failure line, the inspected material does not fracture. The normal of the corresponding fracture plane is the angle bisector of the principal directions of σ_1 and σ_3: the direction of maximum shear stress. The material parameters μ and τ_0 are measured in experiments.

Shape descriptors corresponding to the Coulomb criterion are [32]:

$$\begin{aligned} \tau &= \frac{\sigma_1-\sigma_3}{2} \quad \text{maximum shear stress} \\ c &= \frac{\sigma_1+\sigma_3}{2} \\ R &= \frac{\sigma_1-\sigma_2}{\sigma_1-\sigma_3} \quad \text{shape factor.} \end{aligned} \tag{4}$$

For other failure models, other shape descriptors exist. In general, these sets are not simple orthogonal coordinate frames, but represent important physical quantities.

Considering Mohr's circle, c represents its center, and τ its radius (Figure 5). The shape factor $R \in [0,1]$ reveals the kind of anisotropy. Similar to the terminology used in DTI, stresses with $R = 0$ are called *planar* and $R = 1$ are called *linear* (Figure 4). It is a relative value and undefined for small values of τ (isotropic stresses).

An example for another common anisotropy measure considering all principal stresses is the *von Mises stress*

$$\sigma_v = \sqrt{0.5 \cdot ((\sigma_1-\sigma_2)^2 + (\sigma_2-\sigma_3)^2 + (\sigma_1-\sigma_3)^2)}. \tag{5}$$

5 Diagram Views

We propose several diagram views, presenting various perspectives onto (stress) tensor characteristics (Figure 2). The views abstract from the tensor volume's spatial representation, and give insight into the statistical distribution of tensor properties. All attribute-space views are linked and can be used side-by-side. Brushing in the views creates and updates a mask volume that is used to assign visualization methods in the spatial view (Section 6). The diagram views are parameterized by:

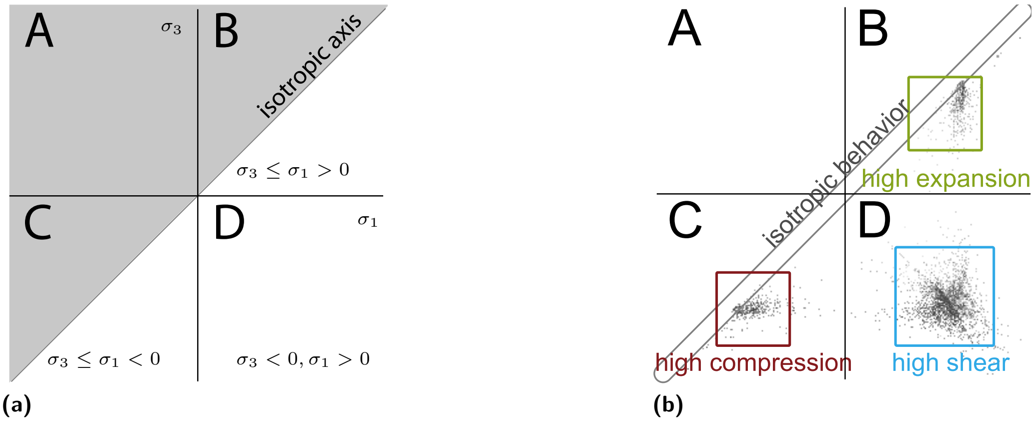


Figure 7 Schematic illustration of the (σ_1, σ_3)-scatterplot (left). The x-axis corresponds to the major eigenvalue and the y-axis the minor eigenvalue. Example scatterplot for the two-force dataset (right); a simulation of a cube affected by a pushing and a pulling force, which results in compressive as well as tensile stresses. The eigenvalues were logarithmically mapped (Equation 6).

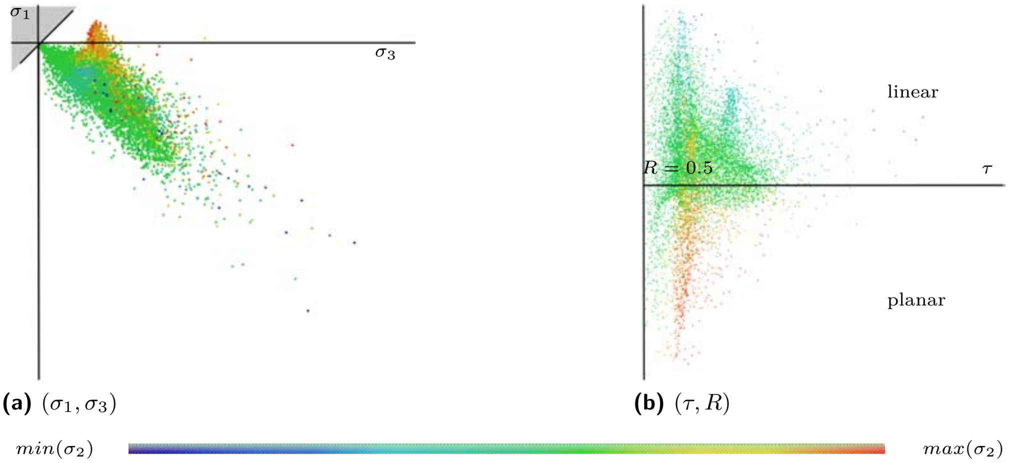
- Choice of shape descriptors.
- Choice of directions.
- Choice of shape space scaling.

In this paper, we offer a default selection of views and parameterizations specific to the failure analysis of stress tensors. Of course, our concept offers a wide range of other parameter choices, too. Statistical views (e.g. scatterplots, histograms) are especially suitable to quantify tensor characteristics. We have adapted scatterplots to fit scalar (Section 5.1) as well as directional tensor invariants (see Section 5.4). Directional histograms quantify selected directions, as eigenvectors or the direction of maximum shear. Furthermore, we present Mohr diagrams (Section 5.2) as additional perspective on the tensor data. They are a common tool in engineering, and therefore familiar to a large group of users.

5.1 Shape Space Scatterplot

Scatterplots are used to depict the relation between two scalar invariants. Figure 7 illustrates a scatterplot that is used to quantify normal (compressive or tensile) and shear stresses. The input, therefore, are the three principal stresses sorted in descending order, i.e., $\sigma_1 \geq \sigma_2 \geq \sigma_3$. The plot is divided into four quadrants (A,B,C,D). Due to the ordering, there never will be any points in the upper left quadrant (A). Points in the upper right quadrant (B) correspond to eigenvalues that are all positive, characterizing tensors of high tensile stresses. Accordingly, points in the lower left quadrant (C) correspond to high compressive stresses. The most interesting region is the lower right quadrant (D), which shows tensors with tensile and compressive stresses. To summarize, we can deduce the following tensor field characteristics from the (σ_1, σ_3)-scatterplot:

- The more points in quadrant B, the higher the level of expansion.
- The more points in quadrant C, the higher the level of compression.
- The more points in quadrant D, the higher the level of mixed stresses.
- Points that have a large distance to the isotropic axis exhibit a high level of shear.
- Points that are located near the isotropic axis exhibit no shear at all; they describe tensors with isotropic behavior.



(a) (σ_1, σ_3) **(b)** (τ, R)

Figure 8 Scatterplot for the slit-cube dataset (Section 7) with varying shape descriptors as input. (a) Considering (σ_1, σ_3) as input, regions of compression, expansion and shear can be distinguished. The inspected dataset exhibits mostly high shear stresses, no compressive forces and marginally tensile forces. Therefore, we analyze the shear region (Quadrant D) in more detail (b) considering (τ, R). Plotting the shape factor R reveals that within this region more linear ($R \approx 1$) than planar ($R \approx 0$) behavior happens.

For the scatterplot, there is no need for a mapping into the positive domain. In contrast, an explicit distinction between positive and negative scalar invariants is desirable. Therefore, we propose two sign-preserving mappings: Logarithmic and histogram equalization [5]. As a standard logarithmic mapping has a singularity in zero, we use

$$f(\sigma_i) = \begin{cases} \log(\sigma_i + 1), & \text{for } \sigma_i \geq 0 \\ -\log(1 - \sigma_i), & \text{for } \sigma_i < 0. \end{cases} \tag{6}$$

The results are then linearly mapped to the range of -1 to 1. In order to see as many tensor characteristics as possible, often a logarithmic mapping is sufficient. For some datasets, however, the data remain cluttered after the mapping. In this case, a histogram equalization is useful. Our modular approach allows an interactive adjustment of the mapping to the needs of the underlying dataset.

5.2 Mohr Diagram

Figure 9 illustrates the Mohr diagram, which is used to analyze selected tensors in more detail. It consists of Mohr circles (Figures 4 and 6), which give an impression of the relationship between the three eigenvalues and their relative strength. The circle's position on the x-axis indicates whether the respective tensor exhibits tensile or compressive forces. Its radius expresses the level of shear. In the original diagram [10], most circles would be located around the origin. This is a region of high interest as it represents high shear and sudden changes from tensile to compressive stresses. To equalize the circles' distribution, we exploit that, in general, a Mohr diagram is only one-dimensional; all circles are centered at the x-axis. We categorize the circles according to the tensor's anisotropic behavior (isotropic, linear, planar), and divide the Mohr diagram into three separate diagrams (Figure 9). Thus, clutter around the origin is reduced significantly. By drawing semi-circles in context regions,

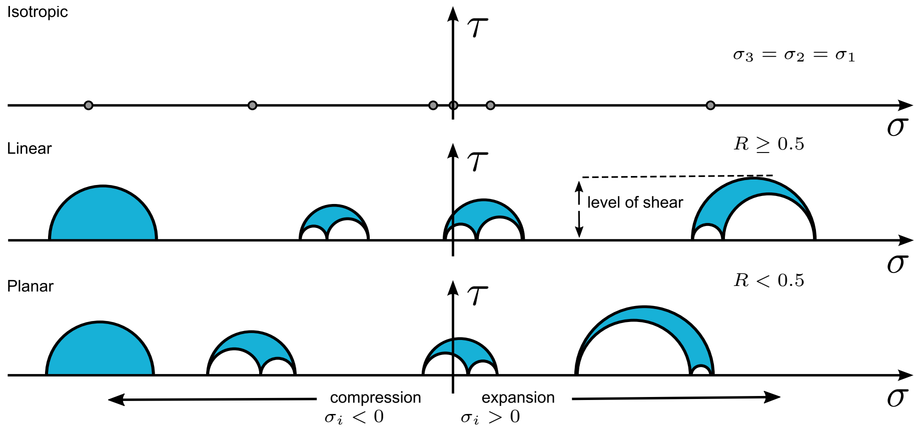


Figure 9 We extended the Mohr diagram proposed by [10] as depicted above. The circle's position on the x-axis represents whether the corresponding tensor is in compression (left) or tension (right). The vertical position corresponds to their anisotropic behavior. For a better overview, we only draw semi-circles.

we achieve a more compact visualization without losing information or clarity. To summarize, we can deduce the following conditions from the Mohr Diagram:

- The more circles on the left, the higher the level of compression.
- The more circles on the right, the higher the level of expansion.
- Circles around the origin exhibit both: compressive and tensile forces.
- The greater the circle's radius, the higher the level of shear.
- Circles degenerating to a single point exhibit no shear at all; they describe tensors with isotropic behavior.
- A high number of circles on one of the three categorization axes represents a high number of isotropic/linear/planar tensors.

Due to its usability for the analysis and depiction of a single tensor or only a few tensors, the Mohr diagram is best used after a selection has been specified in the other diagram views. We achieve a further reduction of Mohr circles to be displayed, by clustering tensors with similar eigenvalue behavior. As similarity measure we use the Euclidean distance between two points in shape space. In the Mohr diagram, we encode the number of occurrences by color (Figure 13).

5.3 Directional Histogram

Figure 10 illustrates the directional histogram, which is used to analyze the distribution of principal shear directions. Of course, other directions of interest can be inspected, too. An example is the directions of maximum shear stress. The spherical diagram projects each direction, for example the major eigenvector, onto the surface of a unit sphere. Due to the non-oriented nature of a symmetric tensor's directional components only half of the sphere's surface needs to be considered. Therefore, all vectors are flipped to the positive half space of a user-selected axis (x, y, z). To create the histogram, either a binning or a splatting approach [17] can be followed. We use the former. The number of intersections between vectors and a given surface patch on the sphere are counted, thus performing a region-dependent binning. For accurate results, a uniform subdivision of the surface is crucial. To account for patch size variations, we normalize the counted frequencies by the respective patch's surface area. Given a triangulation of the unit sphere, we either bin by triangle or by the Voronoi cell of

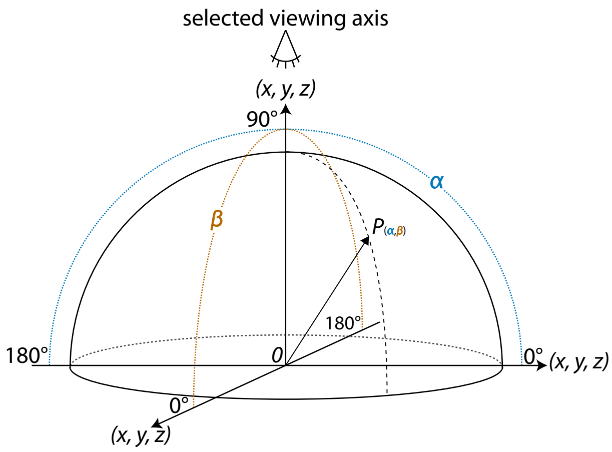


Figure 10 The input directions for directional histogram and directional scatterplot are projected on a hemisphere, flipping all vectors to the half space defined by the selected viewing axis. Each point $P_{(\alpha,\beta)}$ on the hemisphere represents all vectors spanning the angles α and β.

each vertex. Triangle binning results in a discrete visualization of the counted frequencies, where each triangle is colored uniformly. Mapping the frequencies to vertex colors produces a continuous diagram, as the values are interpolated between neighboring vertices. The interpretation of the final plot depends on the selected viewing direction. In the 2D plot, the diagram's center corresponds to all vectors that are collinear with this viewing direction. An arbitrary point on the sphere's surface represents all vectors that span the angles α and β with respect to the two axes orthogonal to the selected viewing direction (Figure 10).

We use two representations of the directional histogram (Figure 14):

- Hemisphere
- Mapping of the hemisphere into a planar representation for a better depiction in 2D [36]

5.4 Directional Scatterplot

The directional scatterplot uses the same setup of unit sphere and projected vectors as the directional histogram, but instead of binning the directions each vector is represented by an individual point on the sphere's surface. This direct representation of the vectors allows using the point's size, color, and transparency to represent tensor properties. Due to numeric instabilities in simulations, isotropic tensors exhibit an increased noise ratio (Figure 11, a). To reduce the noise level in the plot and emphasize pronounced directions, we map the shear stress to transparency. Thus, nearly isotropic tensors do not contribute to the final plot (Figure 11, b). Reasonable quantities to be mapped to colors are normal and shear stresses.

6 Spatial Views

The spatial views represent the tensor field in its original three-dimensional coordinates. The most basic method to display tensors in a spatial context is to use graphical icons (glyphs), e.g. ellipsoids, that are placed at discrete points within the volume. Although glyphs have the potential to show the entire tensor information, they fail to give a continuous view of the tensor field. Such a global view, however, is important to identify regions of compression and expansion, respectively. Volume rendering methods give a global view of the tensor field. However, in general they only work on derived scalar values and thus do not contain

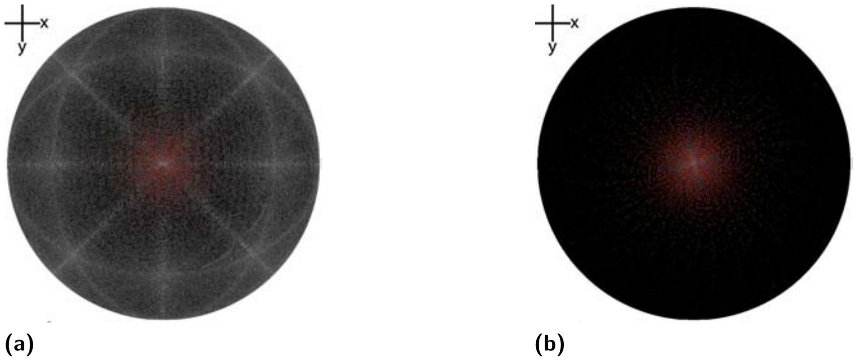


Figure 11 Directional scatterplot for the rotating-star dataset. In Figure (a), all points have the same transparency, which reveals artifacts due to isotropic tensors. In Figure (b) the shear stress is mapped to transparency, i.e., low transparency for low shear stresses and nearly opaque points for high shear stresses. Thus, nearly isotropic tensors do not contribute to the plot.

directional information. We use a hybrid rendering approach, combining volume rendering with glyphs and tensorlines. The visualization is interactively steered by a mask volume that is created and updated through user selections in the diagram views.

The basic idea is to use various visualization methods to separate focus and context regions in the dataset. Therefore, we evaluate the mask volume and map the selected features to geometrical tensor representations (glyphs, tensorlines). The context is visualized by a volume rendering of the remaining dataset, using a scalar invariant chosen by the user.

Volume Rendering: Volume rendering serves as context view with decreased opacity in focus regions, allowing to analyze glyphs and tensorlines in more detail. We use standard GPU ray-casting of scalar invariants, for example, the von Mises stress. As tensorlines and glyphs are explicit geometries, we have to account for correct intersections between volume and opaque scene geometry. As proposed by [14], we use a depth image of the geometry. During volume traversal, rays are stopped as soon as they hit geometry positions.

Tensorlines: To add directional information, tensorlines can be drawn in focus regions, i.e., seeds are randomly placed inside the masked volume. Starting at these seed points, the line is integrated using a fourth-order Runge-Kutta scheme. The integration is stopped as soon as the line runs into an isotropic region.

Tensor Glyphs: Alternatively, glyphs can be drawn in focus regions, encoding the whole tensor information locally. Currently, we use ellipsoids. In order to distinguish between positive and negative eigenvalues, we map the tensor to a positive-definite metric using an antisymmetric-mapping [23]:

$$f(\sigma_i, c, \alpha) = exp(\alpha arctan(c \cdot \sigma_i)). \tag{7}$$

The parameter c determines the slope of the function in the origin, α is a scaling parameter. The glyph's size can be adapted using a global scale parameter.

7 Results

We describe two visual analysis sessions by means of two datasets with diverse characteristics (Sections 7.1, 7.2). Whereas the slit cube simulation is an example where domain experts have clear questions, the rotating-star dataset demonstrates a case with less specific questions.

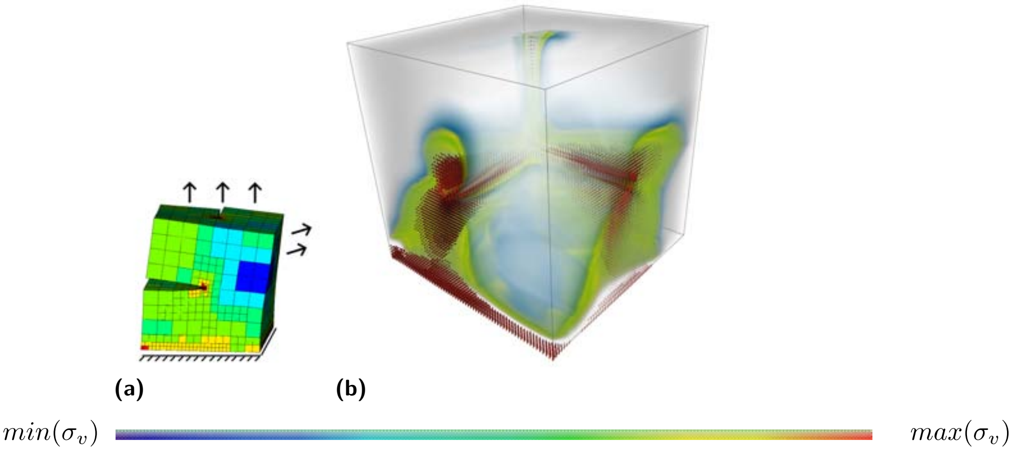


Figure 12 Slit Cube. The dataset is based on a finite element simulation of the deformation of a clamped cube with two slits (a). It is fixed at the bottom. Surfaces forces act on the top and the side of the cube. (b) Shows a hybrid rendering of the slit-cube dataset. Volume rendering of the von Mises stress serves as context, while ellipsoids oriented by the eigenvectors and scaled by the eigenvalues are positioned in focus regions to emphasize high shear stresses.

All analyses were performed on a standard desktop PC, equipped with an Intel Core 2 Duo CPU with 3.0 GHz and a NVIDIA GeForce 8800GT GPU.

7.1 Exploring the Slit-Cube Dataset

The slit-cube dataset is generated via a finite element simulation of the deformation of a clamped cube with two slits. Surface forces are applied to the top and the side of the cube, which is fixed at the bottom. Figure 12 (a) illustrates this process. The images are rendered based on a uniform resampling of the dataset. The resolution of the tensor field is $256 \times 256 \times 256$. In this context, the stress tensor expresses the cube's response to the applied forces. Questions are:

- How does the material respond to the applied forces?
- Which forces act in the material?

In general, the von Mises stress σ_v, a scalar value that is derived from the stress tensor (Equation (5)), is used to predict yielding of materials. Regions where σ_v is high, are prone to material failure.

Figure 12 (b) shows a spatial view of the dataset using a hybrid rendering to visualize focus and context. A volume rendering of the von Mises stress σ_v gives an impression of the whole field, while glyphs are positioned in focus regions and highlight areas of extremely high stresses. It can be seen that, due to the applied forces, the cube's slits increase. Large stresses are concentrated close to the edges of the slits and at the bottom where the cube is fixed, while large areas of the cube are hardly affected by the applied forces.

The von Mises stress is easy to interpret, however, important information of the stress tensor is ignored. That is, we cannot say which forces are prevalent in the material and we cannot say anything about the direction of maximum stresses.

The (σ_1, σ_3)-scatterplot (Figure 8) allows a distinction between compressive, tensile and mixed stresses as well as isotropic and high shear stresses. It shows that the inspected dataset exhibits mostly indefinite stresses, no compressive forces and marginally tensile forces.

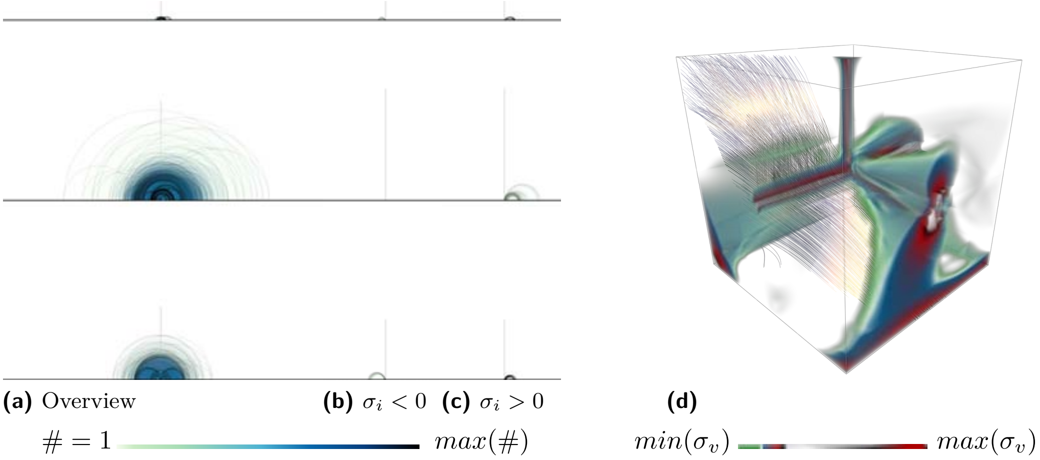

(a) Overview **(b)** $\sigma_i < 0$ **(c)** $\sigma_i > 0$ **(d)**

Figure 13 Slit Cube. The Mohr diagram (a) mainly reveals mixed stresses (circles around the origin) as already shown in the scatterplot. Figures (b) and (c) show only those circles that correspond to compressive (b) and tensile forces (c), respectively. The circles are colored according to their frequency (#). Hybrid rendering (d) that combines volume rendering of the von Mises stress σ_v and tensorlines. Seedpoints for the computation of tensorlines were only seeded in regions of high stress and integrated along the major eigenvector direction.

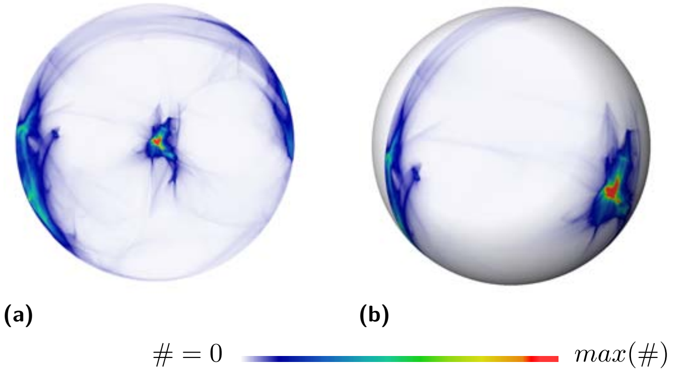

(a) **(b)**

Figure 14 Slit Cube. Directional histogram to examine the distribution of the directions of maximum shear stress. We mapped the hemisphere (front part of the sphere in (b)) into a planar representation (a) for a better depiction in the paper.

Therefore, in the next step of the analysis, we switch the shape descriptors from (σ_1, σ_3) to (τ, R) using a logarithmical mapping (interaction loop #1) for the display (Figure 8). Thus, the shear region (Quadrant D) can be analyzed in more detail. Plotting the maximum shear stress against the shape factor R reveals that this region exhibits more linear than planar behavior. Linear, planar and isotropic stresses are further explored in the Mohr diagram (Figure 13). The circles are color-coded according to their frequency. Using the Mohr diagram as overview, we can reveal the physical behavior over the whole field. It is clearly visible that the slit-cube mainly exhibits indefinite stresses resulting in Mohr circles centered around the origin. However, looking at Figures 13 (b), (c) we can also deduce marginally compressive and tensile forces. As compressive forces are only small outliers in the scatterplot, we have not seen them before.

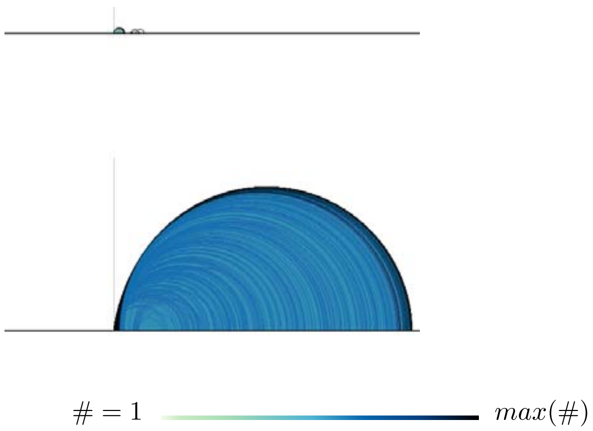


Figure 15 Rotating star. The Mohr diagram reveals only compressive forces, which are perfectly linear. According to our sign convention this means that all stresses are negative. In such a case it is common in the respective application areas to consider only the absolute value of the stresses. The principal stresses are ordered according to their magnitude, i.e., $|\sigma_1| \geq |\sigma_2| \geq |\sigma_3|$ and, thus, restricted to the positive $x-$axis. In the selected region we draw full colored circles.

As we are especially interested in regions of high shear stress, we next examine the directions of maximum shear stress (Figure 14). The directional histogram reveals one strongly expressed peak aligned with the z-direction. A second, minor accumulation is smeared over a larger angle in x, y-plane, approximately 90 degrees to the main stress direction. Figure 13 (d) shows a hybrid rendering, where additional tensorlines are seeded in regions of high stress following the major principal stress direction.

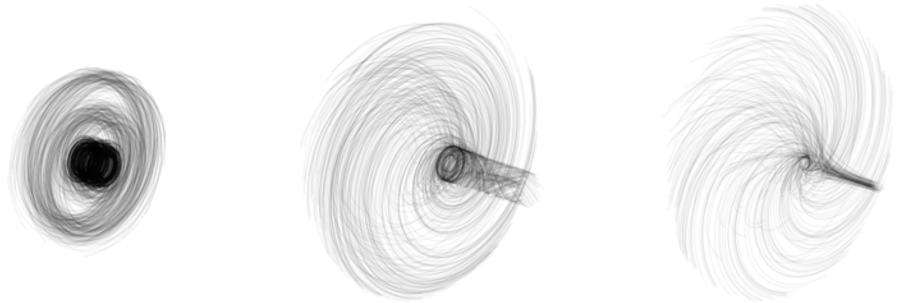
7.2 Exploring the Rotating-Star Dataset

Our second example shows data from an astrophysical simulation of a rotating neutron star's dynamics. Analyzing the evolution of such systems plays a major role for the understanding of the fundamental processes involved in core collapse supernovae and gravitational wave production.

The simulation results consist of a variety of data types, i.e., (complex) scalars, vector fields and tensors. The data is usually three-dimensional and time-dependent, given on a grid with spatially varying resolution (AMR). In this work, we focus on the second-order stress tensor field. The data is resampled on a uniform grid with a resolution of $128 \times 128 \times 128$ samples.

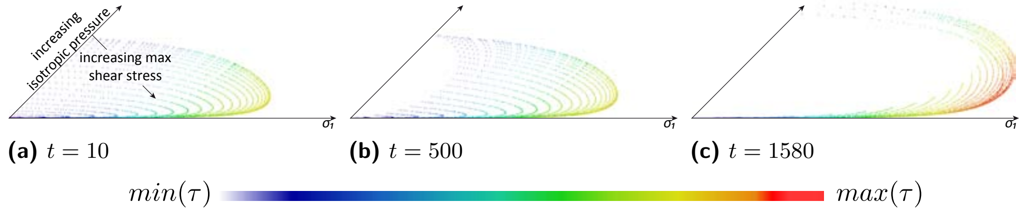
Until now, the domain experts' examination focused on the scalar fields (e.g. magnetic-, velocity- and density fields), which give insight into the evolution of the star formation. Other data types arise as intermediate product of the simulation. The additional analysis of the stress tensor field could support a deeper understanding of the physical processes that cause this specific formation. Investigations are, for example, related to the forces that participate in the star's collapse. However, contrary to our first example (Section 7.1), questions are much more basic. Since the users are not used to look at the tensor data, they do not have any specific expectations. Therefore, the first goal of the visual exploration is to get an initial idea of the information that is contained in the data. Besides the physical interpretation, a thorough data analysis is of high importance to validate the quality of the simulated data. Often, even simple visualizations can reveal failures in the simulations.

For a first impression of the dataset, Figure 16 displays tensorlines following the major



(a) $t = 10$ **(b)** $t = 500$ **(c)** $t = 1580$

Figure 16 Rotating Star. The tensorlines are integrated along the major eigenvector direction. Following the lines shows that the star's rotation lags around its perturbation. The seed points were placed at the star's center using a simple random seeding.



(a) $t = 10$ **(b)** $t = 500$ **(c)** $t = 1580$

Figure 17 Rotating Star. The scatterplot shows that, due to the perturbation, the forces get stronger with increasing time steps. The shape of the plot stays the same, which leads to the assumption that the eigenvalues correlate to each other. The color represents the maximum shear stress τ.

eigenvector for three time steps t. The lines are seeded close to the center of the star.

Figure 15 shows the Mohr diagram of the rotating-star dataset. Due to the high gravitational forces inside the star only compressive stresses occur. According to our sign convention this means that all stresses are negative. In such a case it is common in the respective application areas to consider only the absolute value of the stresses. The principal stresses are ordered according to their magnitude, i.e., $|\sigma_1| \geq |\sigma_2| \geq |\sigma_3|$. As a consequence, the dataset reveals positive-definite behavior, which can be clearly seen in the Mohr diagram. An interesting observation that can be made when zooming into the focus regions, is that the stresses exhibit perfectly linear behavior, i.e., the shape factor is $R = 1$.

Figure 17 shows scatterplots for three time steps. As $|\sigma_1| \geq |\sigma_2| \geq |\sigma_3|$, only positive stresses occur. Therefore, only quadrant B is displayed. It can be observed that after the initial perturbation the principal stresses $|\sigma_1|$, $|\sigma_3|$ as well as the shear forces get stronger with increasing time, i.e., the distance to the isotropic axis increases (Figure 17 (c)). Interestingly the characteristic shape of the scatterplot stays the same over a long period of time. This might lead to the assumption that the major and minor eigenvalue correlate to each other. As we have seen in the Mohr diagram, all tensors exhibit perfect linear behavior. Therefore, we can deduce that the major eigenvalue $|\sigma_1|$ grows almost quadratically with respect to $|\sigma_2|$ and $|\sigma_3|$.

A temporal analysis of the dominant shear directions can be performed based on the directional histograms given in Figure 18. All time steps clearly reflect the symmetry inherent to the data set. At the beginning of the simulation ($t = 10$) all shear directions exhibit a

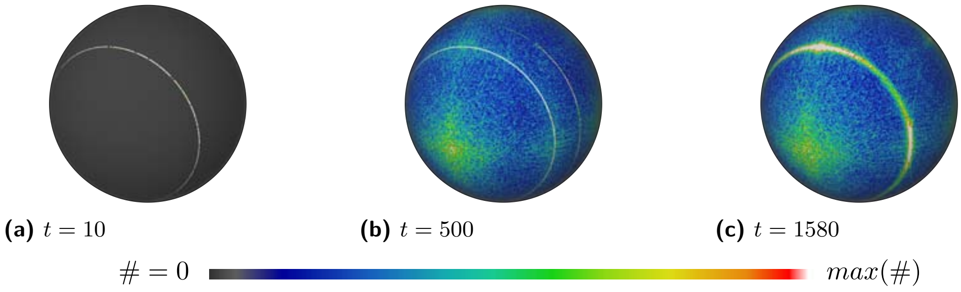


Figure 18 Rotating Star. Directional histogram for the shear vectors. The colored triangles represent the number of data points (#) exhibiting a maximum shear direction falling into the triangle. At the beginning of the simulation (a) all shear directions exhibit a specific angle, which is nicely depicted by a single circle in the diagram. In later time steps, the shear directions become more scattered and the strongly expressed direction splits into two maxima rings (b). With further increasing time, these two maxima merge again resulting in one dominant ring (c). According to our domain experts, the splitting is not physical. It possibly reveals discretization artifacts.

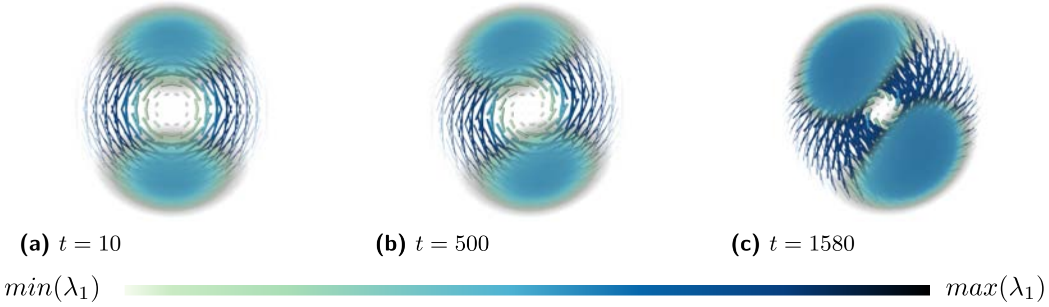


Figure 19 Rotating Star. Hybrid rendering that combines volume rendering of the first tensor component with glyphs. Superquadrics are calculated only on the equatorial plane and colored according to the major eigenvalue λ_1. As deduced from the scatterplot, the forces get stronger with increasing time steps resulting in larger ellipsoids revealing linear forces.

specific angle, which is nicely depicted by a single circle in the diagram. In later time steps, the shear directions become more scattered and the strongly expressed direction splits into two maxima rings ($t = 500$). With further increasing time, these two maxima merge again resulting in one dominant ring ($t = 1580$). According to our domain experts, the splitting is *not* physical. As a consequence, the visualization triggered a discussion about the possible reasons for this development. First ideas included discretization artifacts and problems with the resolution of the star's surface.

Figure 19 shows a hybrid rendering of the dataset. The volume rendering uses the σ_{11} component of the tensor. Even though this is not an invariant it expresses a characteristic tensor behavior, due to the high symmetry of the data. The rendering is combined with a glyph representation seeded in the equatorial plane.

8 Discussion

Our results demonstrate the application of the presented pipeline (Figure 1) and shape space as basic means for feature designation in attribute space (Section 4). Previous work mostly concentrated on one particular type of tensor and visualization technique. Analyzing tensors from diverse application areas, which exhibit different properties, requires convertible invariants and variable mapping techniques. Figure 2 shows invariants for stress tensors, however, the underlying concept can be used for other tensors, too. As motivated by our co-operation partners, future research will include the comparative visualization of different tensor types that occur during the same simulation (e.g. gravitational field tensor and stress tensor).

Our results and the discussion with domain experts further confirm the need for powerful visual exploration and analysis tools. The concurrent use of well known and new visualization methods provides an access to both, the data and modern visualization techniques. In material science and astrophysics, tensors are simulated solely to investigate scalar quantities (von Mises stress, density). Tensors mainly appear as intermediate product of simulations. Although, experts know that the tensor contains important information, which may help to answer their questions (What makes the material crack?, Which forces make the star collapse?), they avoid looking at the complex data as they do not know how to interpret it.

Until now, domain experts are mainly used to two-dimensional plots. Therefore, attribute-space plots are of high importance. Our experience is that domain experts favor simple visualization techniques like scatterplots and icons that are familiar to them (e.g. Mohr's circle). In object space, a sparse usage of lines and glyphs at specific locations is preferred, which motivates the use of a binary mask volume to determine focus and context regions. Moreover, all physicists and engineers rated the brushing-and-linking as extremely helpful to ease the interpretation of the data. That way, the visual exploration leads to new questions, which encourages both, the curiosity to look at the whole tensor and, as a consequence, the development and usage of more complex visualization techniques. Another aspect that arose during discussions was the usefulness of our methods to detect failures in simulations.

Our material science partners are familiar with the Mohr diagram so that it motivated them to use our tools. For astrophysicists the representation was new but considered as interesting. A limitation is that the diagram suffers from clutter. A clear distinction between linear and planar anisotropy in the tensor field is still difficult. Therefore, we will integrate clustering algorithms in the future to reveal more insight into important tensor properties.

Another subject that remains to be investigated is volume rendering in the context of tensor fields. Users like this visualization technique as they know how to interpret it. This work presents renderings of scalar measures (e.g., von Mises stress). Recently, Dick et al. [12] presented a colormapping of the three eigenvalues to distinguish between compressive and tensile forces. More advanced colormappings based on other tensor invariants may be interesting, too.

Our results demonstrate the application of the presented pipeline (Figure 1) and shape space as basic means for feature designation in attribute space (Section 4). Previous work mostly concentrated on one particular type of tensor and visualization technique. Analyzing tensors from diverse application areas, which exhibit different properties, requires convertible invariants and variable mapping techniques. Figure 2 shows invariants for stress tensors, however, the underlying concept can be used for other tensors, too. As motivated by our co-operation partners, future research will include the comparative visualization of different tensor types that occur during the same simulation (e.g. gravitational field tensor and stress tensor).

Our results and the discussion with domain experts further confirm the need for powerful visual exploration and analysis tools. The concurrent use of well known and new visualization methods provides an access to both, the data and modern visualization techniques. In material science and astrophysics, tensors are simulated solely to investigate scalar quantities (von Mises stress, density). Tensors mainly appear as intermediate product of simulations. Although, experts know that the tensor contains important information, which may help to answer their questions (What makes the material crack?, Which forces make the star collapse?), they avoid looking at the complex data as they do not know how to interpret it.

Until now, domain experts are mainly used to two-dimensional plots. Therefore, attribute-space plots are of high importance. Our experience is that domain experts favor simple visualization techniques like scatterplots and icons that are familiar to them (e.g. Mohr's circle). In object space, a sparse usage of lines and glyphs at specific locations is preferred, which motivates the use of a binary mask volume to determine focus and context regions. Moreover, all physicists and engineers rated the brushing-and-linking as extremely helpful to ease the interpretation of the data. That way, the visual exploration leads to new questions, which encourages both, the curiosity to look at the whole tensor and, as a consequence, the development and usage of more complex visualization techniques. Another aspect that arose during discussions was the usefulness of our methods to detect failures in simulations.

Our material science partners are familiar with the Mohr diagram so that it motivated them to use our tools. For astrophysicists the representation was new but considered as interesting. A limitation is that the diagram suffers from clutter. A clear distinction between linear and planar anisotropy in the tensor field is still difficult. Therefore, we will integrate clustering algorithms in the future to reveal more insight into important tensor properties.

Another subject that remains to be investigated is volume rendering in the context of tensor fields. Users like this visualization technique as they know how to interpret it. This work presents renderings of scalar measures (e.g., von Mises stress). Recently, Dick et al. [12] presented a colormapping of the three eigenvalues to distinguish between compressive and tensile forces. More advanced colormappings based on other tensor invariants may be interesting, too.

9 Conclusion

We presented an approach that tackles the challenging problem of visualizing three-dimensional tensor fields by combining multiple views. A solid theoretical basis was provided by extending the notion of shape space, which serves as a link between the abstract tensor and its visualization in attribute space. This theory allows the application of the multiple-views concept to tensors.

In the considered application areas, visual tensor analysis and exploration are still in their infancy. Domain experts are often used to the analysis of derived scalar fields, although they know that the tensor field contains important information to their questions. Especially attribute plots help them to familiarize themselves with more advanced visualization techniques, explore the data, to construct new hypotheses and to detect possible failures in their simulations.

Acknowledgments

This work was funded by the German Research Foundation (DFG) through a Junior Research Group Leader award (Emmy Noether Program). Rotating-star data supplied courtesy of Luca Baiotti from the Albert Einstein Insitute (AEI). The slit-cube dataset was provided by Andreas Schroeder, Humboldt University Berlin. We would further like to thank the reviewers for their valuable comments as well as Markus Hadwiger, Aaryn Tonita, Michael Koppitz, Steffen Prohaska and our students David Bressler and Nino Kettlitz.

References

1 Mark M. Bahn. Invariant and orthonormal scalar measures derived from magnetic resonance diffusion tensor imaging. *Journal of Magnetic Resonance*, 141(1):68–77, 1999.

2 Werner Benger, Hauke Bartsch, Hans-Christian Hege, and Hagen Kitzler. Visualizing neuronal structure in the human brain via diffusion tensor MRI. *Intern. Journal of Neuroscience*, 116:461–514, 2006.

3 Abhir Bhalerao and Carl-Fredrik Westin. Tensor splats: Visualising tensor fields by texture mapped volume rendering. In *MICCAI (2)*, pages 294–302, 2003.

4 Mehmet Bilgen, Ibrahim Elshafiey, and Ponnada A. Narayana. Mohr diagram representation of anisotropic diffusion tensor in MRI. *Magn Reson Med*, 47:823–827, Apr 2002.

5 J. Blaas, C.P. Botha, and F.H. Post. Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1436–1451, Nov.-Dec. 2008.

6 Rebecca Brannon. Mohr's circle and more circles, 2003. `http://www.mech.utah.edu/~brannon/public/Mohrs_Circle.pdf`.

7 Raphael Bürger and Helwig Hauser. Visualization of multi-variate scientific data. In *EuroGraphics 2007 State of the Art Reports (STARs)*, pages 117–134, 2007.

8 Wei Chen, Zi'ang Ding, Song Zhang, Anna MacKay-Brandt, Stephen Correia, Huamin Qu, John Allen Crow, David F. Tate, Zhicheng Yan, and Qunsheng Peng. A novel interface for interactive exploration of dti fibers. *IEEE Transactions on Visualization and Computer Graphics*, 15:1433–1440, November 2009.

9 John C. Criscione, Jay D. Humphrey, Andrew S. Douglas, and William C. Hunter. An invariant basis for natural strain which yields orthogonal stress response terms in isotropic hyperelasticity. *Journal of the Mechanics and Physics of Solids*, 48:2445–2465, 2000.

10 Patricia Crossno, David H. Rogers, Rebecca M. Brannon, David Coblentz, and Joanne T. Fredrich. Visualization of geologic stress perturbations using mohr diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 11(5):508–518, 2005.

11 Rodrigo de Luis-García, Carlos Alberola-López, and Carl-Frederik Westin. Segmentation of tensor fields: Recent advances and perspectives. In Santiago Aja-Fernández, Rodrigo de Luis García, Dacheng Tao, and Xuelong Li, editors, *Tensors in Image Processing and Computer Vision*. Springer London, 2009.

12 Christian Dick, Joachim Georgii, Rainer Burgkart, and Rüdiger Westermann. Stress tensor field visualization for implant planning in orthopedics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1399–1406, 2009.

13 Helmut Doleisch, Martin Gasser, and Helwig Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, pages 239–248. Eurographics Association, 2003.

14 Klaus Engel, Markus Hadwiger, Joe M. Kniss, Christof Rezk Salama, and Daniel Weiskopf. *Real-Time Volume Graphics*. AK Peters, Ltd., 2006.

15 Daniel B. Ennis and Gordon Kindlmann. Orthogonal tensor invariants and the analysis of diffusion tensor magnetic resonance images. *Magnetic Resonance in Medicine*, 55(1):136–146, 2006.

16 Louis Feng, Ingrid Hotz, Bernd Hamann, and Kenneth Joy. Anisotropic noise samples. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):342–354, 2008.

17 Laura Fritz, Markus Hadwiger, Georg Geier, Gerhard Pittino, and Meister Eduard Gröller. A visual approach to efficient analysis and quantification of ductile iron and reinforced sprayed concrete. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1343–1350, October 2009.

18 Hans Hagen and Christoph Garth. An introduction to tensors. In *Visualization and Processing of Tensor Fields*. Springer, 2006.

19 Youssef M. A. Hashash, John I-Chiang Yao, and Donald C. Wotring. Glyph and hyperstreamline representation of stress and strain tensors and material constitutive response. *International Journal for Numerical and Analytical Methods in Geomechanics*, 27(7):603–626, 2003.

20 Lambertus Hesselink, Yuval Levy, and Yingmei Lavin. The topology of symmetric, second-order 3D tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):1–11, January/March 1997. diss.

21 Mario Hlawitschka, Gerik Scheuermann, and Bernd Hamann. Interactive glyph placement for tensor fields. In *ISVC (1)*, pages 331–340, 2007.

22 Mario Hlawitschka, Gerik Scheuermann, Gunther H. Weber, Owen T. Carmichael, Bernd Hamann, and Alfred Anwander. Interactive volume rendering of diffusion tensor data. In D. H. Laidlaw and J. Weickert, editors, *Visualization and Processing of Tensor Fields: Advances and Perspectives*, Mathematics and Visualization. Springer, 2009.

23 Ingrid Hotz, Louis Feng, Hans Hagen, Bernd Hamann, Kenneth Joy, and Boris Jeremic. Physically based methods for tensor field visualization. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 123–130. IEEE Computer Society, 2004.

24 Radu Jianu, Cagatay Demiralp, and David Laidlaw. Exploring 3d dti fiber tracts with linked 2d representations. *IEEE Transactions on Visualization and Computer Graphics*, 15:1449–1456, November 2009.

25 Gordon Kindlmann. Superquadric tensor glyphs. In *Proceedings of IEEE TVCG/EG Symposium on Visualization '04*, pages 147–154, 2004.

26 Gordon Kindlmann and David Weinstein. Hue-balls and lit-tensors for direct volume rendering of diffusion tensor fields. In *VIS '99: Proceedings of the conference on Visualization '99*. IEEE Computer Society, 1999.

27 Gordon Kindlmann, David Weinstein, and David Hart. Strategies for direct volume rendering of diffusion tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):124–138, 2000.

28 Gordon Kindlmann and Carl-Fredrik Westin. Diffusion tensor visualization with glyph packing. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1329–1336, 2006.

29 R. M. Kirby, H. Marmanis, and D. H. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *Proc. Visualization*, pages 333–340, 1999.

30 Ron D. Kriz, Edward H. Glaessgen, and J.D. MacRae. Eigenvalue-eigenvector glyphs: Visualizing zeroth, second, fourth and higher order tensors in a continuum. *NCSA Workshop on Modeling the Development of Residual Stresses During Thermoset Composite Curing*, 1995.

31 David H. Laidlaw, Eric T. Ahrens, David Kremers, Matthew J. Avalos, Russell E. Jacobs, and Carol Readhead. Visualizing diffusion tensor images of the mouse spinal cord. In *VIS '98: Proceedings of the conference on Visualization '98*, pages 127–134. IEEE Computer Society Press, 1998.

32 Björn Lund. *Crustal Stress Studies Using Microearthquakes and Boreholes.* PhD thesis, Department of Earth Sciences, Uppsala University, 2000.

33 Allen R. Sanderson, Mike Kirby, Chris R. Johnson, and Lingfa Yang. Advanced reaction-diffusion models for texture synthesis. *Journal of Graphics Tools*, 11(3):47–71, 2006.

34 Andreas Sigfridsson, Tino Ebbers, Einar Heiberg, and Lars Wigström. Tensor field visualisation using adaptive filtering of noise fields combined with glyph rendering. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 371–378. IEEE Computer Society, 2002.

35 Jaya Sreevalsan-Nair, Cornelia Auer, Bernd Hamann, and Ingrid Hotz. Eigenvector-based interpolation and segmentation of 2d tensor fields. In *Proceedings of Topological Methods in Visualization (TopoInVis'09)*, to appear 2010.

36 Detlev Stalling and Hans-Christian Hege. Fast and intuitive generation of geometric shape transitions. *The Visual Computer*, 16:241–253, 2000.

37 Xavier Tricoche, Gordon L. Kindlmann, and Carl-Fredrik Westin. Invariant crease lines for topological and structural analysis of tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1627–1634, 2008.

38 D Weinstein, G Kindlmann, and E Lundberg. Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. In *Proceedings of IEEE Visualization 1999*, pages 249–253, 1999.

39 C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis. Processing and visualization of diffusion tensor MRI. *Medical Image Analysis*, 6(2):93–108, 2002.

40 Yu-Chien Wu, Aaron S. Field, Moo K. Chung, Benham Badie, and Andrew L. Alex. Quantitative analysis of diffusion tensor orientation: Theoretical framework. In *Magnetic Resonance in Medicine*, pages 1146–1155, 2004.

41 Björn Zehner. Interactive exploration of tensor fields in geosciences using volume rendering. *Computers and Geosciences*, 32:73–84, 2006.

42 Xiaoqiang Zheng and Alex Pang. Hyperlic. In *VIS '03: Proceedings of the conference on Visualization '03*, pages 249–256. IEEE Computer Society Press, 2003.

43 Xiaoqiang Zheng and Alex Pang. Topological lines in 3d tensor fields. In *VIS '04: Proceedings of the conference on Visualization '04*. IEEE Computer Society Press, 2004.

44 Xiaoqiang Zheng, Xavier Tricoche, and Alex Pang. Degenerate 3d tensors. Technical Report UCSC-CRL-04-09, University of California, Santa Cruz, 2004.

Abstract Feature Space Representation for Volumetric Transfer Function Exploration

Ross Maciejewski[1], Yun Jang[2], David S. Ebert[3], and Kelly P. Gaither[4]

1,3 **Purdue University Visual Analytics Center – West Lafayette, IN, USA**
`{rmacieje|ebertd}@purdue.edu`

2 **ETH Zurich – Zurich, Switzerland**
`jangy@inf.ethz.ch`

4 **University of Texas – Austin, TX, USA**
`kelly@tacc.utexas.edu`

Abstract

The application of n-dimensional transfer functions for feature segmentation has become increasingly popular in volume rendering. Recent work has focused on the utilization of higher order dimensional transfer functions incorporating spatial dimensions (x, y, and z) along with traditional feature space dimensions (value and value gradient). However, as the dimensionality increases, it becomes exceedingly difficult to abstract the transfer function into an intuitive and interactive workspace. In this work we focus on populating the traditional two-dimensional histogram with a set of derived metrics from the spatial (x, y and z) and feature space (value, value gradient, etc.) domain to create a set of abstract feature space transfer function domains. Current two-dimensional transfer function widgets typically consist of a two-dimensional histogram where each entry in the histogram represents the number of voxels that maps to that entry. In the case of an abstract transfer function design, the amount of spatial variance at that histogram coordinate is mapped instead, thereby relating additional information about the data abstraction in the projected space. Finally, a non-parametric kernel density estimation approach for feature space clustering is applied in the abstracted space, and the resultant transfer functions are discussed with respect to the space abstraction.

1998 ACM Subject Classification I.3 Computer graphics, I.3.5 Computational Geometry and Object Modeling, I.3.6 Methodology and Techniques, I.3.7 Three-Dimensional Graphics and Realism

Keywords and phrases Volumetric Transfer Function, Abstract Feature Space

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.212

1 Introduction

A common method for direct volume rendering is to employ the use of interactive transfer functions as a means of assigning color and opacity to the voxel data. One of the most popular transfer function design tools is the interactive 2D histogram widget introduced by Kniss et al. [8]. In this widget, the user is presented with a 2D histogram (the axes of which represent a feature space of the data) and various selection tools are used to assign color and opacity to the voxels through an interactive brushing of the feature space. This widget typically displays each entry in the histogram as a gray scale color with white representing the entry that maps to the largest number of voxels within a given data set, Figure 1 (a). While such a tool has been shown to be extremely effective at advanced transfer function

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 212–221
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

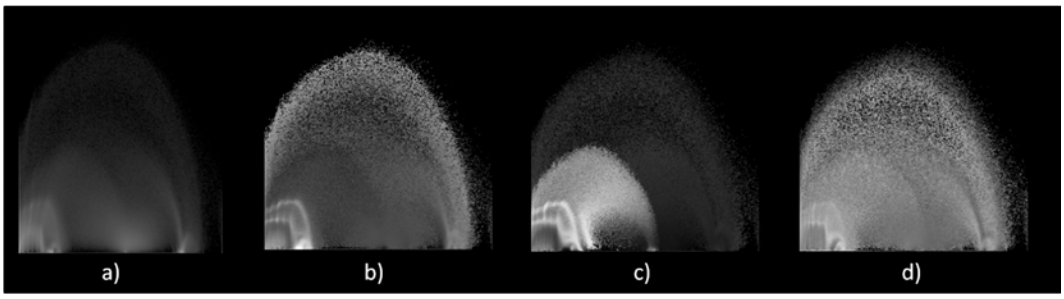


Figure 1 The value vs. value gradient magnitude feature space with the intensity as a function of a) the number of voxels, b) the magnitude of spatial variance of x, c) the magnitude of spatial variance of y, and d) the magnitude of spatial variance of z for the CT bonsai dataset.

creation, this type of histogram provides no information about the spatial relationships between the voxels at the histogram entry.

In this work we propose to enhance the conventional 2D histogram transfer function by mapping the entries of the histogram to statistical properties related to the spatial locations of the voxels, specifically, the magnitude of the spatial variance, Figure 1 (b-d). By doing so, this new abstracted feature space now illustrates the areas in which the voxels have a higher spatial relationship as opposed to simply providing the user with a view of where in the feature space the majority of their voxels lie. Users are able to toggle between the *conventional transfer function view*, Figure 1 (a), (in which the entries in the 2D histogram are colored by the number of voxels that map to a location) and the *abstracted transfer function views*, Figure 1 (b-d), (in which the entries are colored by the spatial variance in the voxels that map to a location).

Traditionally, the appropriate selection of features in multi-dimensional transfer functions is a difficult task, often requiring the user to have an underlying knowledge of the data set under exploration. By providing users with information about the spatial domain (x, y, and z) of their data in an abstracted feature space (e.g., value versus value gradient magnitude) we are able to enhance the exploration, allowing users to better discover features within their dataset. Finally, we utilize the non-parametric transfer function design method proposed by Maciejewski et al. [12] clustering the feature space in both the conventional and abstracted transfer function views. In this manner, we explore the usefulness of abstracting statistical properties into the transfer function widget and illustrate the effects on the exploration of the feature space.

2 Related Work

Interactive transfer function design has been addressed with many different approaches, ranging from simple (yet intuitive) one-dimensional (1D) transfer functions (e.g., [5, 13]) in which a scalar data value is mapped to color and opacity, to more complex multi-dimensional transfer functions in which color and opacity are mapped across multiple variables. Early work by Kindlmann et al. [6] and Kniss et al. [8] applied the idea of a multi-dimensional transfer function [9] to volume rendering. This work identified key problems in transfer function design, noting that many interactive transfer function widgets lack the information needed to guide users to appropriate selections, making the creation of an appropriate transfer function essentially trial-and-error which is further complicated by the large degrees of freedom available in transfer function editing. While many volume rendering systems have

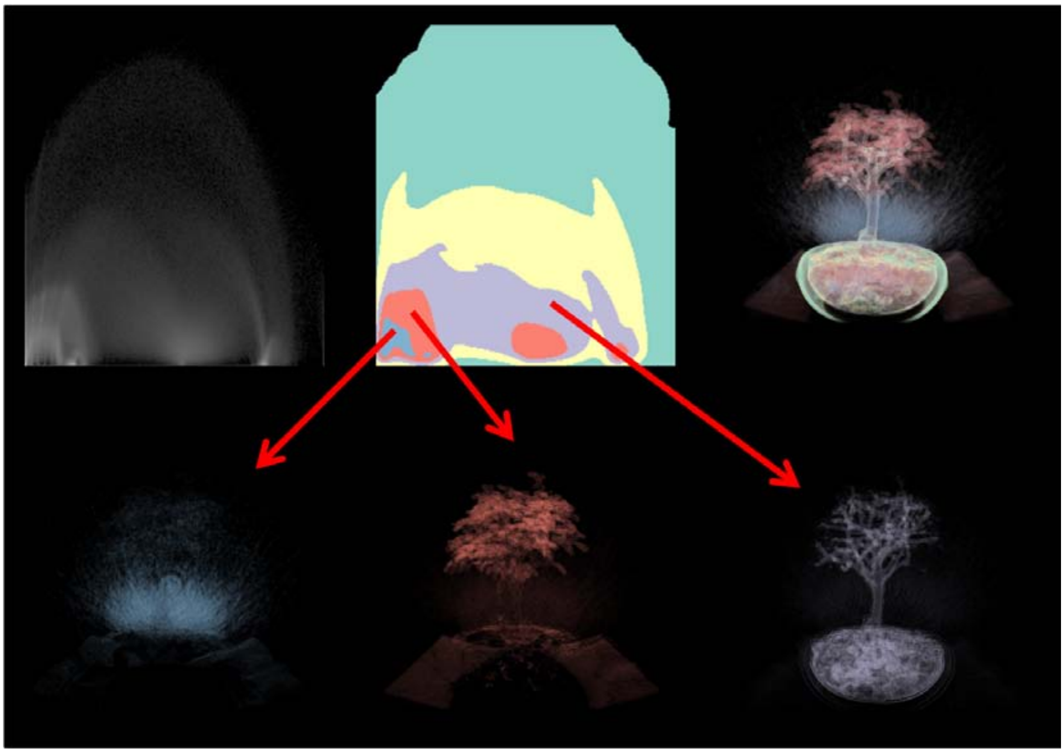


Figure 2 The value vs. value gradient magnitude feature space as a function of the number of voxels and the rendering from the denoted segments of the transfer function for the CT bonsai dataset.

adopted multi-dimensional transfer function editing tools, the creation of an appropriate transfer function is still difficult as the user must understand the dimensionalities of the feature space that they are interacting with.

Recent works on transfer function design have proposed higher-dimensional transfer functions based on mathematical properties of the volume. Examples include work by Kindlmann et al. [7], which employed the use of curvature information to enhance multi-dimensional transfer functions, and Tzeng et al. [18], which focused on higher dimensional transfer functions which use a voxel's scalar value, gradient magnitude, neighborhood information and the voxel's spatial location. Work by Potts et. al [15] suggested visualizing transfer functions on a log scale in order to better enhance feature visibility. Lundstrom et al. introduced the partial range histogram [10] and the α-histogram [11] as means for incorporating spatial relations into the transfer function design. Correa et al. introduced size based transfer functions [3] which incorporate the magnitude of spatial extents of volume features into the color and opacity channels and visibility based transfer functions [4] where the opacity transfer function is modified to provide better visibility of features.

While such extensions enhance the volume rendering and provide a larger separability of volumetric features, they still fail to provide users with information about feature space structures. In fact, the addition of more dimensionality into the transfer function is often automatically incorporated into the rendering parameters, obscuring the relationship between the volumetric properties and the volume rendering. Work by Roettger et al. [16] incorporates similar ideas of using spatial features of the volume to enhance transfer function design. They enable the automatic setup of multi-dimensional transfer functions by adding spatial

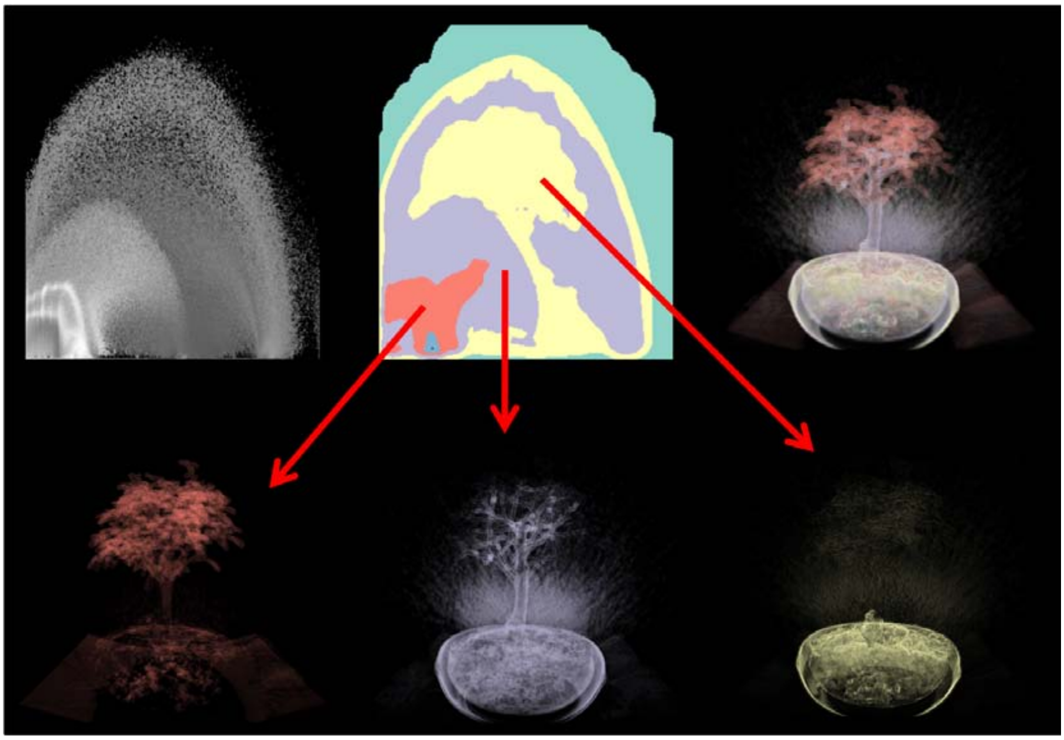


Figure 3 The value vs. value gradient magnitude feature space as a function of the magnitude of the spatial variance and the rendering from the denoted segments of the transfer function for the CT bonsai dataset.

information to the histogram of the underlying dataset, where as our work proposes a means for the direct analysis of spatial properties. Other work focuses on the addition of more dimensions to provide coherency between transfer functions. Recent work by Akiba et al. [1, 2] utilized parallel coordinate plots to create a volume rendering interface for exploring multivariate time-varying datasets. Muelder and Ma [14] proposed a prediction-correction process to aid in creating coherent feature tracking.

In all of these related works, one can note that various statistical properties of the volumes are being used in order to extract features of interest and segment properties of the volume. Unfortunately, as the number of dimensions increases, interaction in n-dimensional space becomes cumbersome to the point that few systems exceed two dimensional transfer functions; instead, the extra dimensionality is incorporated automatically, somewhat limiting the user's control. In order to enhance the information provided in the transfer function histogram widget, our work incorporates some statistical properties of the spatial domain (x, y and z) into the projected feature space domain (e.g., value versus value gradient magnitude).

3 Abstract Feature Space Representation

Given any two-dimensional feature space for a given volume data set, the user is presented with a representation illustrating the number of voxels at a given location within that feature space. In order to enhance this feature space, we propose the use of an *abstract feature space* representation. By this, we mean that the 2D histogram feature space is no longer representing the number of voxels as a given location. Instead, the 2D histogram feature space is representing statistical properties of the volumetric (x, y and z) space with relationship to a particular feature. The key property we have chosen to look at is spatial variance with respect to a given feature set. Thus, given any two feature properties of the volumetric data set (e.g., value vs. value gradient norm, temperature vs. pressure, x vs. pressure) we compute the spatial variance of all the points that map to a given entry in the feature space histogram.

As such, at location (i, j) in the feature space histogram, we have N voxels, V_k, that map to this given feature space pair. For all the N voxels that map to the feature space pair (i, j) we calculate the mean position within the (x, y, z) space of the volume.

$$\bar{V}_{(i,j)} = \frac{1}{N} \sum_{k=1}^{N} V_k \tag{1}$$

Once the mean is calculated, we can then determine the magnitude of the standard deviation within the volumetric space at the feature space pair (i, j).

$$|\sigma_{i,j}| = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (V_k - \bar{V}_{(i,j)})^2} \tag{2}$$

Figure 1 illustrates the difference between the traditional histogram mapping the magnitude of the voxels found at a histogram entry (a) to the magnitude of the spatial variance found at a histogram entry (b-d). Furthermore, we can see that the magnitude of the spatial variance in the z-direction is relatively constant; however, there is an obvious clustering of high spatial variance in the y-direction. The magnitude of the standard deviation vector can also be employed for analyzing the spatial variance found within a given feature space, and the utility of the creation of this abstract feature space is further discussed in Section 5.

4 Non-Parametric Transfer Function Generation

Once the abstract feature space is defined in the 2D histogram, we can use the non-parametric transfer function generation approach described in Maciejewski et al. [12] to provide users with a means to group data in the 2D histogram by areas of similar value (in terms of the number of voxels or in terms of the abstracted standard deviation metric described in Section 3). In order to create these clusters we employ the use of a variable kernel method [17, 19] formed in Equation 3. Furthermore, we utilize an adaptive kernel which scales the parameter of the density estimation by allowing the kernel radius to vary based upon the distance from each point, X_i, to the kth nearest neighbor in the set comprising the $N - 1$ data points of the histogram feature space.

$$\hat{f}_h(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{d_{i,k}} K\left(\frac{\mathbf{x} - X_i}{d_{i,k}}\right) \tag{3}$$

Here, $\hat{f}_h(\mathbf{x})$ is the probability density estimate of the histogram (h) at a given location, $\mathbf{x}$, in the feature space, $d_{i,k}$ represents the multi-dimensional smoothing parameter and N is the total number of samples in the histogram (i.e., the number of voxels in the volume). The window width of the kernel placed on the point X_i is proportional to $d_{i,k}$ (where $d_{i,k}$ is the distance from the i-th sample to the k-th nearest neighbor) so that data points in regions where the data is sparse will have flatter kernels. We choose $k = \lfloor \sqrt{N} \rfloor$ as this tends to approximate the optimal density estimation fitting (this is a rule of thumb approximation

[17]). Such a method groups the data based on their neighborhood information, allowing us to visualize the underlying structure of the data.

In order to reduce the calculation time, we have chosen to employ the Epanechnikov kernel, Equation 4.

$$K(\mathbf{u}) = \frac{3}{4}(1 - \mathbf{u}^2)1_{(||\mathbf{u}|| \leq 1)} \tag{4}$$

The function $1_{(||\mathbf{u}|| \leq 1)}$ evaluates to 1 if the inequality is true and zero for all other cases.

5 Exploring Abstracted Feature Spaces

The focus of this work is to determine if statistical properties derived from the make-up of a given feature space can be utilized to extract more information and aid in transfer function creation and data exploration. In this section we describe the impact of utilizing the proposed abstract feature space for feature segmentation and data exploration.

5.1 Feature Segmentation

Figure 2 illustrates the application of the non-parametric transfer function generation technique on the conventional transfer function view of the CT bonsai data set, and Figure 3 illustrates the application of the non-parametric transfer function generation technique on the abstracted transfer function. Here we can see which portions of the clustered transfer function map to regions of the volume. Note that the cluster pattern is completely different between Figure 2 and Figure 3. Here we can see that the clustering in the conventional transfer function space winds up with a larger portion of the background noise in each feature segmentation than the abstract transfer function view.

Furthermore, in Figure 2 we see that in segmenting out the leaves (the red cluster), values representing air are still being included with the transfer function; however, by utilizing the spatial variance, those values are not included in the segmentation of the leaves as seen in Figure 3 (again the red cluster). However, other features wind up being combined when using the spatial variance. We can see that the tree trunk and root ball has more separation when utilizing the voxel magnitude for clustering than when using the spatial deviation.

We can further explore the effects of using spatial deviation in transfer function generation by utilizing the x, y and z spatial components separately, thus creating an abstract transfer function space representing the standard deviation . Figure 4 (middle) illustrates the same transfer function generation technique on each of the components of the spatial deviation mapped to the transfer function. What we find in the case of the CT bonsai data is that the area representing air (the red circled area in Figure 4 (middle)) in the feature space is not found as a separate cluster portion in any application of the non-parametric transfer function generation; however, when utilizing the y spatial variance component in the transfer function, the area representing air in the transfer function is associated with another data grouping within the transfer function. This grouping allows us to segment out the air component (as seen in the resultant volume rendering (Figure 4 (bottom)) where as this component is directly related to other important features in the volume when utilizing the x and z spatial components. The segmentation of the air component is also what leads to the noise in Figures 2 and 3

Based on these observations, it seems likely that utilizing information on the spatial variance can help in transfer function design; however, it seems unlikely that this feature alone can create the desired segmentation. Another option for creating abstract feature space

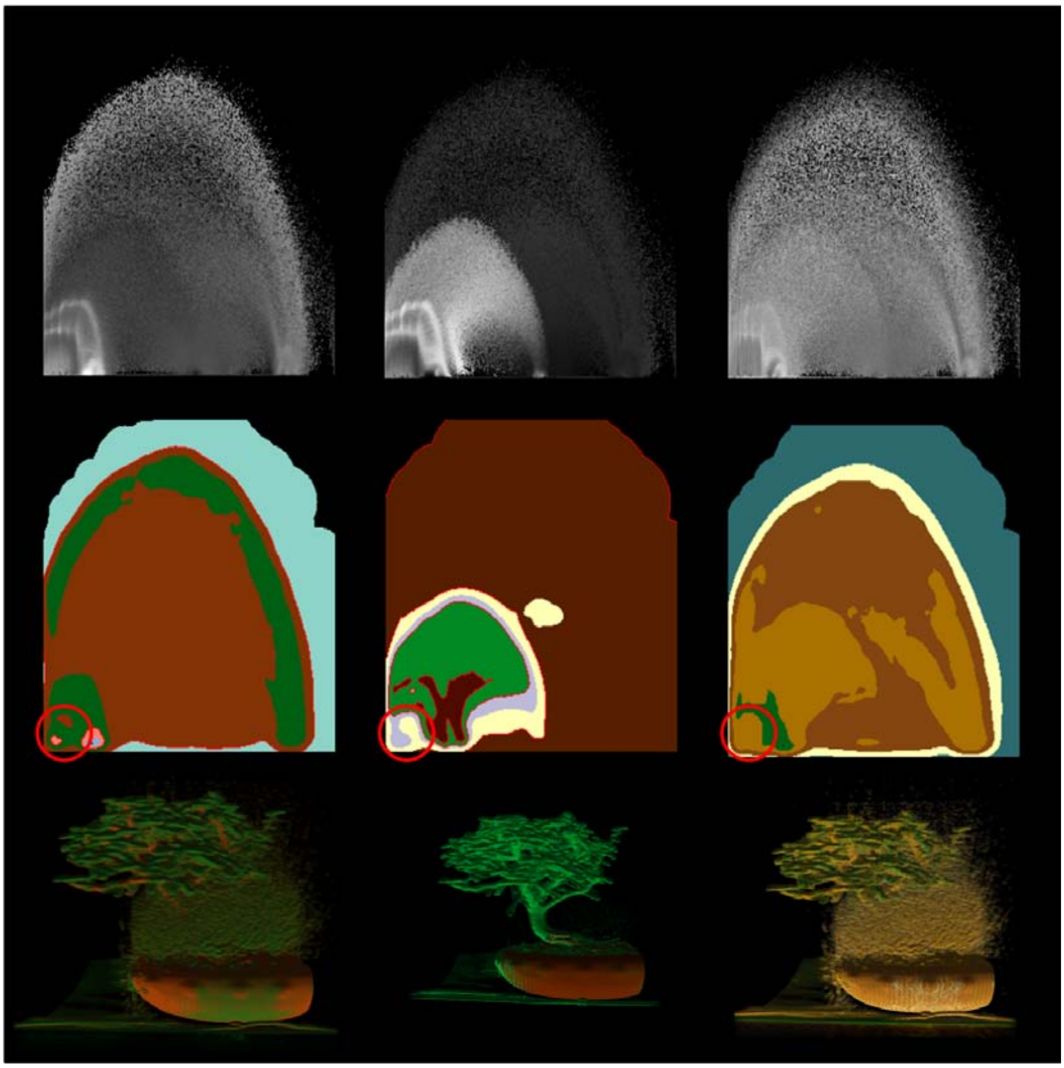


Figure 4 (Top) The value vs. value gradient magnitude feature space as a function of the spatial variance in the x (left), y (middle) and z (right) component direction. (Middle) The resultant non-parametric transfer function. (Bottom) The resultant volume rendering from applying various segments of the transfer function. Note in the x and z spatial variance histograms, segmentation of the noise can not be accomplished.

representations could be utilizing bivariate color mapping schemes to simultaneously relay information about the magnitude of change at a transfer function location and the number of voxels that map to that location.

5.2 Data Exploration

While such tools have their use in traditional volumetric datasets (CT and MRI), the application of these novel analytic methods to more complex simulation data also proves to be very interesting. With the maturation of computational power, simulations are capable of modeling physical phenomena at increasingly more realistic scales. Analysis tools, however, are struggling to keep up with the explosion of data that has resulted from the increase in computational horse-power. This is particularly evident in computational fluid dynamic

(CFD) simulations modeling time-dependent complex flow phenomena. Because the contents of the simulation are not known exactly, and because computational simulations are still evolving and improving their ability to correctly and accurately model physical behavior, the output from these simulations must be analyzed using data mining, feature detection, and feature extraction techniques to provide useful, pertinent information.

Analytical 3D feature detection methods employ well-formed mathematical descriptions of features often unique to a given domain, thus demanding *a priori* knowledge of the nature and location of any potential "areas of interest." This is a cumbersome and tedious task, resulting in a set of methods incapable of scaling with the size of the data. Statistical feature detection tools, on the other hand, provide an automatic way to provide general characteristics.

Flow data sets are ideal candidates for statistically based feature detection techniques because we can leverage physical properties to apply segmentation algorithms. For example, using similarity allows us to segment those regions of the flow data that contain like properties for a given variable. Conversely, using dissimilarity allows us to pinpoint regions in which the flow changes drastically or perhaps discontinuously. This discontinuity is of interest in all flow data sets and often indicates an important region of interest. As such, the utilization of abstract feature space for exploring CFD data should allow researchers to discover interesting regions hidden within their datasets. By utilizing a mapping of the spatial deviation, analysts should be able to quickly explore areas in which flow feature properties (temperature, pressure, velocity, etc.) quickly change from a consistent amount of spatial deviation to a (relatively) larger or smaller amount.

6 Conclusions

From our preliminary work on creating abstract feature space, it seems like utilizing the underlying statistical properties of the spatial volumetric features that map to a given feature space histogram pair will provide analysts with another tool in which to explore data. Based on our exploration of the abstract feature space, it seems likely that enhancing transfer function design through the use of statistical information about the spatial relationships between voxels will aid in feature segmentation and exploration. Future work will focus on utilizing the spatial variation as an uncertainty metric, and looking at other derived data properties (entropy, etc.) as a means for enhancing transfer function exploration and design. Furthermore, we plan to utilize abstracted feature space measures for novel volume rendering parameters in order to reduce the burden of transfer function design on the user. In this manner, we hope to match statistical properties of the data to visual properties thereby being able to semi-automatically create effective transfer functions. Finally, we plan to extend this to the analysis of complex fluid dynamic simulations and map a variety of statistical properties to the transfer function domain, thereby providing researchers with a tools that provide both a fast statistical analysis of data properties and a means in which to filter the data on said properties.

Acknowledgements

This work is supported by the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0001 and by the National Science Foundation OCI-0906379. Yun Jang is supported by the Swiss National Science Foundation under grant 200021_124642.

References

1 Hiroshi Akiba and Kwan-Liu Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization*, pages 115–122, May 2007.

2 Hiroshi Akiba, Kwan-Liu Ma, Jacqueline H. Chen, and Evatt R. Hawkes. Visualizing multivariate volume data from turbulent combustion simulations. *Computing in Science and Engineering*, 9(2):76–83, 2007.

3 Carlos Correa and Kwan-Liu Ma. Size-based transfer functions: A new volume exploration technique. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1380–1387, October 2008.

4 Carlos Correa and Kwan-Liu Ma. Visibility-driven transfer functions. In *Proceedings IEEE-VGTC Pacific Visualization Symposium*, Beijing, China, April 2009.

5 Taosong He, Lichan Hong, Arie Kaufman, and Hanspeter Pfister. Generation of transfer functions with stochastic search techniques. In *Proceedings of the IEEE Conference on Visualization*, pages 227–234, 1996.

6 Gordon Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of the IEEE Symposium on Volume Visualization*, pages 79–86, 1998.

7 Gordon Kindlmann, Ross Whitaker, Tolga Tasdizen, and Torsten Moller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of the IEEE Conference on Visualization*, pages 513–520, 2003.

8 Joe Kniss, Gordon Kindlmann, and Charles Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of the IEEE Conference on Visualization*, pages 255–262, 2001.

9 Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics & Applications*, 8(3):29–37, 1988.

10 Claes Lundstrom Patric Ljung and Anders Ynnerman. Local histograms for design of transfer functions in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1570–1579, 2006.

11 C. Lundström, A. Ynnerman, P Ljung, A. Persson, and H. Knutsson. The alpha-histogram: Using spatial coherence to enhance histograms and transfer function design. In *Proceedings Eurographics/IEEE-VGTC Symposium on Visualization 2006*, May 2006.

12 Ross Maciejewski, Insoo Woo, Wei Chen, and David S. Ebert. Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization and Computer Graphics*, (To appear) 2009.

13 J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. In *Proceedings of the ACM SIGGRAPH Conference on Computer graphics and Interactive Techniques*, pages 389–400, 1997.

14 Chris Muelder and Kwan-Liu Ma. Interactive feature extraction and tracking by utilizing region coherency. In *Proceedings of IEEE Pacific Visualization Symposium*, April 2009.

15 S. Potts and Torsten Moller. Transfer functions on a logarithmic scale for volume rendering. In *Proceedings of Graphics Interface*, pages 57–63, 2004.

16 Stefan Roettger, Michael Bauer, and Marc Stamminger. Spatialized transfer functions. In *Proceedings Eurographics/IEEE-VGTC Symposium on Visualization 2005*, 2005.

17 B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall/CRC, 1986.

18 Fan-Yin Tzeng, Eric B. Lum, and Kwan-Liu Ma. A novel interface for higher-dimensional classification of volume data. In *Proceedings of the IEEE Conference on Visualization*, pages 505–512, 2003.

19 M. P. Wand. *Kernel Smoothing*. Chapman & Hall/CRC, 1995.

Variational Level-Set Detection of Local Isosurfaces from Unstructured Point-based Volume Data

Vladimir Molchanov[1], Paul Rosenthal[1], and Lars Linsen[1]

[1] Visualization and Computer Graphics Laboratory
Jacobs University Bremen
Campus Ring 1, Bremen, Germany

Abstract

A standard approach for visualizing scalar volume data is the extraction of isosurfaces. The most efficient methods for surface extraction operate on regular grids. When data is given on unstructured point-based samples, regularization can be applied but may introduce interpolation errors. We propose a method for smooth isosurface visualization that operates directly on unstructured point-based volume data avoiding any resampling. We derive a variational formulation for smooth local isosurface extraction using an implicit surface representation in form of a level-set approach, deploying Moving Least Squares (MLS) approximation, and operating on a kd-tree. The locality of our approach has two aspects: first, our algorithm extracts only those components of the isosurface, which intersect a subdomain of interest; second, the action of the main term in the governing equation is concentrated near the current isosurface position. Both aspects reduce the computation times per level-set iteration. As for most level-set methods a reinitialization procedure is needed, but we also consider a modified algorithm where this step is eliminated. The final isosurface is extracted in form of a point cloud representation. We present a novel point completion scheme that allows us to handle highly adaptive point sample distributions. Subsequently, splat-based or mere (shaded) point rendering is applied. We apply our method to several synthetic and real-world data sets to demonstrate its validity and efficiency.

1998 ACM Subject Classification Algorithm/Technique

Keywords and phrases Level-set, isosurface extraction, visualization in astrophysics, particle simulations.

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.222

1 Introduction

Many applications in modern engineering and science provide techniques that produce unstructured point-based volume data. Such data are, for instance, generated by measurements with sensors that are irregularly placed in space. An even larger class of unstructured data generation is given by simulations with Lagrangian methods in engineering, hydrodynamics, astrophysics, etc. A classical representative of such methods is the approach of Smoothed Particle Hydrodynamics (SPH) developed by Lucy [16] and Gingold and Monaghan [8].

SPH is currently used for numerous astrophysical simulations of galaxy mergers, star formation, supernova explosions, and other processes. The method involves no regular spatial grid. Objects are represented as an ensemble of interacting particles, which permanently change their positions according to some governing equations. The particles carry all quantities of interest, which may remain unchanged (mass of a particle) or may vary with time (velocity, density, temperature, etc.). The number of particles in modern simulations

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 222–239
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

is in the order of several millions. Thus, a large amount of information needs to be saved, transformed, analyzed, and rendered. As a consequence, there exists a constantly growing need in fast, robust, and effective visualization techniques.

A standard approach to visualize scalar volume data is to extract and render isosurfaces. A huge variety of algorithms are known for gridded data. To detect an isosurface from unstructured data, a common procedure is to first interpolate the scalar field to vertices of a regular (rectangular) grid. Such a regularization simplifies the extraction algorithm but inevitably introduces an interpolation error. This error can significantly affect the result, especially in regions with low sample density or highly adaptive sample distribution. The particular strength of Lagrangian simulations such as SPH is its adaptivity. Therefore, visualization methods should make an effort to avoid interpolation to a regular grid whenever possible [23]. Our intent is to perform all operations exclusively on scattered samples, i.e., on unstructured point-based volume data without any given connectivity information.

A high-quality smooth isosurface visualization requires accurate computation of gradients to the isosurface. This becomes a problem for some real-world data sets, as the sampled scalar function may contain noise and may not be sufficiently smooth. There exist two principle approaches to extract smooth isosurfaces:

- The original scalar function f is regularized in a neighborhood of the isosurface and the surface position and the gradient field are found with respect to the modified function;
- An auxiliary smooth (signed distance) function ψ is initialized at the samples and iteratively updated to approach f maintaining desired regularity. Subsequently, an isosurface is extracted due to the final values of ψ.

The first scheme works well, if an a priory information on noise parameters is available. Moreover, the behavior of the scalar function close to the isosurface may be non-uniform: for the real astrophysical data we work with, the ratio of the largest to the smallest gradient norm can be of order 10^3, which makes the first approach hardly applicable. The competitive second scheme being a flexible multitasking tool is referred to as a level-set approach and is widely used in edge detection, image segmentation problems, etc.

We propose a novel method for smooth local isosurface extraction from point-based volume data using an implicit surface representation. Locality in the context means that we are only interested in extracting those components of the isosurface that have a non-empty intersection with a given subdomain (region of interest). This feature is highly desirable when operating on many-object astrophysical data, where different components of an isosurface may occlude each other from a viewer. To derive the governing equation we introduce a cost functional, that is minimized during level-set evolution. After convergence, we extract the detected isosurface in point-cloud representation, which can be made arbitrarily dense for point or splat-based rendering.

The main contributions of this work are:

- a new main term (governing equation) of the cost function, which significantly softens the dependence of parameters on given data;
- a variational approach to the minimization problem;
- an accurate and fast estimation of differential operators based on the Moving Least Squares (MLS) algorithm;
- investigation of a reinitialization-free formulation using the functional proposed by Li et al. [14];
- iterative extraction of isopoints using MLS approximation and Newton's method;

- a new procedure for point-cloud completion resulting in dense, accurate, and smooth point-based rendering of the isosurface; and
- numerical tests involving both synthetical and real-world data sets indicating the behavior of several quantities measuring the quality of the solution.

Our method is designed for but not limited to the visualization of data coming from astrophysical simulations using Smoothed Particle Hydrodynamics.

2 Related Work

Level-set methods are successfully used in various areas of research including visualization. They find their application in image segmentation [5, 17], object detection [6], shape reconstruction [28], isosurface generation [7], multiphase motion [27], etc. The methods are applied to data coming from scanners [17], satellites [25], or tomography [11, 12].

Most level-set algorithms are designed to operate on a grid, which may be uniform, adaptive, octree-based, or of other form. If scattered data are allowed as input, they are typically interpolated to an auxiliary underlying mesh. We refer to recent results in the area of scattered data interpolation that allow for efficient large-scale data processing [20, 21].

MLS is a well-studied tool for reconstructing smooth function from scattered data. The approach finds many applications in numerical simulations (especially, particle methods) and computer graphics. Recently, MLS was applied to volume rendering of shaded isosurfaces from regular and irregular volume data [10]. The high-quality shading requires accurate approximation of derivatives. There exist two methods for gradient computation using MLS: the faster one was proposed by Nayroles et al. [18] and was used in the Diffuse Element Method (DEM) to solve compressible fluid equations in $\mathbb{R}^d$. The more accurate but time-consuming method proposed by Belytschko et al. [3] was used in the Element-Free Galerkin Method (EFG).

A PDE-based technique for smooth isosurface extraction from volume data sampled on particles was developed by Rosenthal and Linsen [23, 24, 15]. Our method also uses original irregularly sampled data to propagate an implicit surface according to a level-set equation. However, in contrast to the approach proposed by Rosenthal and Linsen, we suggest to minimize a cost functional by variational methods. Also, we use the non-morphological formulation to make the method local and achieve better efficiency [19]. The advantages of local level-set methods are reduction of computational efforts, simplicity of implementation, and generality [22, 26]. The data sets we work with represent systems of interacting astrophysical objects. Thus, we prefer to use a local method to be able to extract isosurfaces associated with one object or a group of objects.

We extract isosurfaces in point cloud representation, which we render using point-based rendering methods. The main problem we face is that the point cloud representation of the surface is inhomogeneously dense due to the irregularity of the data samples. Thus, a sampling of additional isosurface points is required. We refer to recent approaches [13, 1, 2] and references therein for some algorithms. For instance, adaptive splatting is a well-known rendering method allowing for high-quality images from point-set surfaces [29]. Our approach to produce additional samples is not based on the extracted point cloud, but is derived from the underlying data field, which allows for a more precise positioning.

3 Problem Setting and Data Structure

In astrophysical simulations, when using a particle-based method like SPH, all quantities of interest are periodically recorded in files. The data usually contains vector fields (current particles positions, velocities, pressure, magnetic field) and scalar functions (mass density, particles' interaction radii, temperature, chemical component concentrations, etc.), typically ordered by particle indices. Required information should then be extracted from the data, transformed, processed, and displayed to facilitate visual exploration in order to obtain a deeper understanding of the modeled processes. In our work, we focus on the detection and rendering of isosurfaces — surfaces corresponding to some prescribed value of a scalar field.

Given a set of N irregularly distributed points $\mathbf{x}_i \in D \subset \mathbb{R}^3$ with associated scalar values $f_i \approx f(\mathbf{x}_i) \in \mathbb{R}$ representing some smooth field $f : D \to \mathbb{R}$ in a bounded domain D, we want to detect and render all components of a smooth isosurface $\Gamma_{\text{iso}} = \{\mathbf{x} \in D : f(\mathbf{x}) = f_{\text{iso}}\}$ with respect to a given real isovalue f_{iso} that have non-empty intersections with a subdomain $D' \subset D$. We assume $\min_i f_i < f_{\text{iso}} < \max_i f_i$. Our method produces a finite set of points $\mathbf{p}_k \in \Gamma_{\text{iso}}$, which are used for surface rendering. From now on, we refer to these points as *isopoints*.

A three-dimensional kd-tree T is used to store space coordinates of points together with pointers to the scalar functions used in the surface extraction algorithm. This data structure is an efficient and flexible tool for fast handling of irregular data and search queries [23, 10]. In a pre-processing step, an indexing of the samples is applied. Subsequently, a major part of the required operations on the data is reduced to a fast search over the indices. In particular, we use the generated kd-tree to compute and store the n nearest neighbors for each sample point. In all our tests, we used $n = 26$.

Although it should be obvious, we still want to point out that the proposed methods can be applied to structured data sets, as well. For structured data, the neighborhood information can be retrieved much easier, i.e., without using the kd-tree structure, which also allows for a more accurate estimation of differential operators on samples and a more precise evaluation of isopoint positions.

4 Variational Formulation and Level-Set Equation

The problem of detecting isosurface locations has much in common with problems in image segmentation. In fact, a simple transformation $f \to \text{sgn}(f - f_{\text{iso}})$ of the data converts the surface defined by a prescribed isovalue into a boundary with a sharp gradient. Then, one can apply any known edge-detection algorithm to capture the surface. However, our goal is to develop an algorithm for isosurface detection that is more efficient, simple, and robust for this task than the transformed image segmentation algorithms.

4.1 Cost Function

We start with a construction of a functional E, which is the target function that we want to minimize. E depends on the given data f, the constant f_{iso} representing the isovalue, and a level-set function φ together with its derivatives of first order, i.e., $E = E(\varphi, \nabla\varphi; f, f_{\text{iso}})$. The total functional consists of two weighted terms

$$E = E_1 + \lambda E_2, \tag{1}$$

responsible for accuracy and smoothness of the solution, respectively. We propose to use

$$E_1 = \frac{1}{4}\int_D (\mathrm{sgn}(\varphi(\mathbf{x})) - \mathrm{sgn}(f(\mathbf{x}) - f_{\mathrm{iso}}))^2 \, \mathrm{d}\mathbf{x} \tag{2}$$

and

$$E_2 = \int_D \delta(\varphi(\mathbf{x}))\, |\nabla\varphi(\mathbf{x})| \, \mathrm{d}\mathbf{x}. \tag{3}$$

Here, we use the standard definitions of the sign function

$$\mathrm{sgn}(x) = \begin{cases} 0 & \text{for} \quad x = 0 \\ x/|x| & \text{for} \quad x \neq 0 \end{cases}$$

and the Dirac function

$$\delta(x) = \frac{\mathrm{d}}{\mathrm{d}x} H(x), \qquad H(x) = \begin{cases} 1 & \text{for} \quad x \geq 0 \\ 0 & \text{for} \quad x < 0 \end{cases} .$$

$H(x)$ denotes the Heaviside function and the derivative is used in the sense of distributions. A discussion of the proposed functional E in comparison with previous work is provided in Section 7.

A function φ_∞ minimizing a functional of the form $\int L(\varphi, \nabla\varphi)\, \mathrm{d}\mathbf{x}$ satisfies the Euler-Lagrange equation

$$\left(\frac{\partial L}{\partial \varphi} - \sum_i \frac{\partial}{\partial x_i} \frac{\partial L}{\partial \varphi_i} \right) \Bigg|_{\varphi = \varphi_\infty} = 0, \tag{4}$$

where φ_i is the i-th component of $\nabla\varphi$. We derive the Euler-Lagrange equations for E_1 by

$$\frac{1}{2}(\mathrm{sgn}(\varphi(\mathbf{x})) - \mathrm{sgn}(f(\mathbf{x}) - f_{\mathrm{iso}}))\ \mathrm{sgn}'(\varphi(\mathbf{x})) = 0 \tag{5}$$

and for E_2 by

$$\begin{aligned} 0 &= \delta'(\varphi(\mathbf{x}))|\nabla\varphi(\mathbf{x})| - \nabla \cdot \left[\delta(\varphi(\mathbf{x}))\, \frac{\nabla\varphi(\mathbf{x})}{|\nabla\varphi(\mathbf{x})|} \right] \\ &= \delta'(\varphi)|\nabla\varphi| - \delta'(\varphi)\nabla\varphi \cdot \frac{\nabla\varphi}{|\nabla\varphi|} - \delta(\varphi)\, \nabla \cdot \left[\frac{\nabla\varphi}{|\nabla\varphi|} \right] = -\ \delta(\varphi(\mathbf{x}))\, \nabla \cdot \left[\frac{\nabla\varphi(\mathbf{x})}{|\nabla\varphi(\mathbf{x})|} \right] . \end{aligned} \tag{6}$$

The idea of a level-set approach is to reach φ_∞ being a fixed point of an evolution equation for $\varphi = \varphi(\mathbf{x}, t)$ minimizing E. Here, t is an artificial time parameterizing the minimization process $\varphi(\mathbf{x}, t) \to \varphi_\infty(\mathbf{x})$ as $t \to \infty$. To construct the PDE, one equates the left-hand side of Equation (4) with $-\partial\varphi/\partial t$. For the functional E, it reads [6, 9]

$$\frac{\partial \varphi}{\partial t} = \delta(\varphi)\ (\mathrm{sgn}(f - f_{\mathrm{iso}}) - \mathrm{sgn}(\varphi)) + \lambda\, \delta(\varphi)\ \nabla \cdot \left(\frac{\nabla\varphi}{|\nabla\varphi|} \right) . \tag{7}$$

In this derivation, we used

$$\mathrm{sgn}(x) = 2H(x) - 1, \qquad \mathrm{sgn}'(x) = 2H'(x) = 2\delta(x) \tag{8}$$

in the sense of distributions. In the following, Equation (7) will be regularized and discretized in space and time, leading to an iterative process for the value of φ at each sample point.

4.2 Spatial and Temporal Discretization

In practice, the Dirac function $\delta(x)$ in Equation (7) is smoothed on a small scale ϵ. Recall that $\delta(\varphi)$ in the first term of Equation (7) appears as a derivative of the sgn-function. Therefore, in order to be consistent with the differentiation, one should also smooth $\mathrm{sgn}(\varphi)$. It is worth to note that in many papers (e.g., [6, 14, 19]) the smoothed functions contain in their definitions trigonometric expressions, which are expensive to compute. Our choice is to use polynomial splines

$$\delta_\epsilon(x) = \frac{1}{\epsilon}\hat{\delta}\left(\frac{x}{\epsilon}\right), \qquad \hat{\delta}(x) = \frac{1}{6}\begin{cases} 4 - 6x^2 + 3x^3 & \text{for} \quad |x| \le 1 \\ (2-x)^3 & \text{for} \quad 1 < |x| \le 2 \\ 0 & \text{for} \quad |x| > 2 \end{cases},$$

$$\mathrm{sgn}_\epsilon(\epsilon x) = \frac{\mathrm{sgn}(x)}{12}\begin{cases} 16|x| - 8|x|^3 + 3x^4 & \text{for} \quad |x| \le 1 \\ 12 - (|x|-2)^4 & \text{for} \quad 1 < |x| \le 2 \\ 12 & \text{for} \quad |x| > 2 \end{cases}.$$

Here $\mathrm{sgn}'_\epsilon(x) = 2\delta_\epsilon(x)$ similar to Equation (8).

The quantity 4ϵ characterizes the width of a band around the current zero-level surface, in which the terms of Equation (7) operate. The smaller the value of parameter ϵ is, the stronger is the localization of the update. We want to mention that Equation (7) contains two delta-functions, which could be smoothed on different scales ϵ and ϵ'. In grid-based methods, the smoothing parameter is chosen to be in the order of several grid steps. In our case, there is no such characteristic, since the samples are scattered. Thus, the optimal ϵ and ϵ' depend on local samples distribution and the parameters can be corrected during simulations.

To discretize Equation (7) in space we need a technique to approximate the differential operators (i.e., gradient and divergence) on the unstructured data. For each point of interest $\mathbf{x}_i$, we use the kd-tree to find its n nearest neighbors from the whole set of samples. Let V_i be the set of the n nearest neighbors of $\mathbf{x}_i$ including the point itself. Then, for any scalar function g and sample $\mathbf{x}_i$ we can determine the parameters of the linear function $g_i(\mathbf{x}) = \mathbf{a}_i \cdot \mathbf{x} + b_i$, $g_i : \mathbb{R}^3 \to \mathbb{R}$ from the condition

$$\sum_{\mathbf{x}_j \in V_i} w_i^2(\mathbf{x}_i - \mathbf{x}_j)(g(\mathbf{x}_j) - g_i(\mathbf{x}_j))^2 \to \min$$

with some weight function $w_i(\mathbf{x})$. In other words, g_i is the local MLS approximation of the scalar field g in the neighborhood of $\mathbf{x}_i$. Practically, the unknown coefficients are found as the solution of the overdetermined system of linear equations

$$w_i(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{a}_i \cdot \mathbf{x}_j + b_i) = w_i(\mathbf{x}_i - \mathbf{x}_j)g(\mathbf{x}_j), \qquad \mathbf{x}_j \in V_i,$$

using QR-decomposition [4] or by explicit formulae that are analogous to the ones proposed by Rosenthal and Linsen [24] for least-squares approximation. The least-squares approximation used by Rosenthal and Linsen can be viewed as a particular case of MLS with the weight function $w(\mathbf{x}) \equiv 1$. A simple one-dimensional test comparing the two approaches is presented in Figure 1. It is shown that MLS produces a smoother interpolation when compared to least squares.

For all our tests we use

$$w_i(\mathbf{x}) = \begin{cases} (1 - \|\mathbf{x}\|^2/h_i^2) & \text{for} \quad |x| \le h_i \\ 0 & \text{else} \end{cases},$$

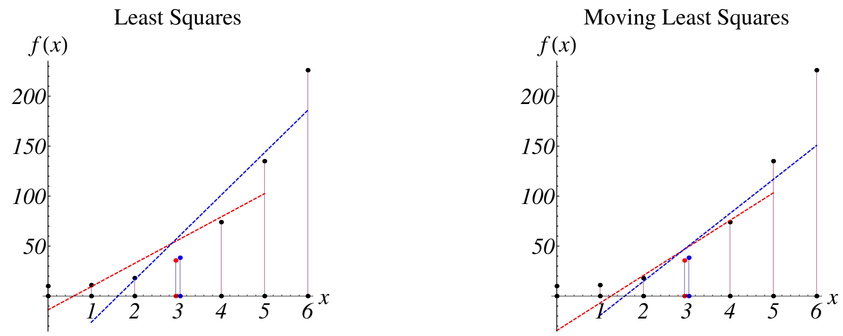


Figure 1 Comparison of least squares and MLS approximation for two close samples in one spatial dimension. Samples are located at $\{0, 1, 2, 2.95, 3.05, 4, 5, 6\}$ with values due to the function $f(x) = x^3 + 10$. We used 6 nearest neighbors in both cases and the weight function $w(x) = (1 - (x/3.5)^2)$ for MLS. MLS produces a smoother interpolation.

where h_i is the support size chosen to be equal to the distance between the ith sample and the farthest of the 26 nearest neighbors.

After having determined an MLS linear approximation in the neighborhood of the ith sample, we can set

$$\nabla g(\mathbf{x}_i) \approx \mathbf{a}_i.$$

We note that we stay rather in the frame of the DEM than the EFG formulation. Similar to Ledergerber et al. [10] we prefer the faster method, since the improvement by more accurate derivative estimation is negligible. To estimate the divergence of a vector field (g^1, g^2, g^3) at a sample point $\mathbf{x}_i$ we construct the linear least-square approximations g_i^k for each k-th component and then sum up the coefficients as

$$\nabla \cdot (g^1,\ g^2,\ g^3)(\mathbf{x}_i) = (\mathbf{a}_i^1)_1 + (\mathbf{a}_i^2)_2 + (\mathbf{a}_i^3)_3.$$

For the time integration we use an explicit Euler scheme. Hence, the iteration step

$$\varphi^{k+1} = \varphi^k + \Delta t \left[\delta_\epsilon(\varphi^k)\ (\operatorname{sgn}(f - f_{\text{iso}}) - \operatorname{sgn}_\epsilon(\varphi^k)) + \lambda\, \delta_{\epsilon'}(\varphi^k)\ \nabla \cdot \left(\frac{\nabla \varphi^k}{|\nabla \varphi^k|}\right)\right] \tag{9}$$

is applied to all sample points $\mathbf{x}_i$. The upper index k stands for the k-th layer in time. The use of the explicit first-order scheme ensures that the iterations are stable for a sufficiently small Δt depending on the spatial resolution. However, since the sampling is irregular, the choice of a globally stable Δt can be small. We refer the reader to the discussion about the choice of Δt and asynchronous time integration by Rosenthal and Linsen [24].

4.3 Reinitialization

Stability of evolution of φ according to a level-set equation requires closeness of φ to a signed distance function [22]. The width of the band involved in computations due to Equation (9) is determined by the size of the support of the smeared-out delta function $\delta_\epsilon(\varphi)$. In the neighborhood of an isopoint $\mathbf{p} \in \Gamma_{\text{iso}}$, it can be roughly estimated as $4\epsilon|\nabla\varphi(\mathbf{p})|$. Thus, a high gradient of φ makes the band narrow and slows down propagation of Γ_{iso}. However, small values of the gradient norm make the computations in Equation (9) unstable due to the division by $|\nabla\varphi|$. This observation explains the need for keeping $|\nabla\varphi| \approx 1$ during computations.

An exact calculation of the signed distance function is extremely time consuming. A standard approach is to trigger a reinitialization routine after several evolution steps [19]. This procedure drives the auxiliary function ψ towards a signed distance function of the zero-isosurface as in the Eikonal equation

$$\frac{\partial\psi(\mathbf{x},\tau)}{\partial\tau} = \operatorname{sgn}(\varphi(\mathbf{x},t))\ (1 - |\nabla\psi(\mathbf{x},\tau)|)\ . \tag{10}$$

We discretize and solve Equation (10) for $\tau > 0$ with the initial condition $\psi(\mathbf{x}, 0) = \varphi(\mathbf{x}, t)$. A few iterations suffice. After the reinitialization procedure, we set $\varphi = \psi$ and continue the computations with respect to the level-set equation.

If the level-set function remains close to the signed distance function during evolution, the governing equation could be simplified [19]. For example, the mean curvature term could be reduced to the Laplacian of φ. In our approach, no simplified equation is used. Therefore, we may moderate the requirement for φ to $c < |\nabla\varphi(\mathbf{x}_i, t)| < C$ with some constants c and $C \in \mathbb{R}^+$ for all $1 \le i \le N$ and $t \ge 0$. Hence, it is enough to have a mechanism that suppresses large and low gradients. Such a tool was proposed by Li et al. [14] in form of an additional diffusion term. This term minimizes a functional controlling the closeness of the level-set function to the signed distance function. It eliminates the need to solve the Eikonal equation. We discuss this enhancement of our level-set method in Section 5.

4.4 Choice of Parameters and Level-Set Function Initialization

The parameters ϵ and ϵ' define a band around the isosurface, where the level-set function values are to be updated. Usually, they are chosen to be comparable with the cell size. In our experiments, we set $\epsilon = \epsilon' \in [1.0l, 1.5l]$, where l is the local average distance between samples. This choice results in a width of the band of 4ϵ due to the definition of the δ_ϵ function. Small values prevent isosurface propagation, large smoothing parameters decrease efficiency of the method.

Next, we choose a sufficiently small time step Δt to ensure that the main term in Equation (9) does not destroy local monotonicity of φ and provides stable computations. In Figure 2, we show how the choice of Δt influences the shape of $\varphi^1(x)$ initialized as $\varphi^0(x) = -x$. The one-dimensional examples show that large time steps cause instabilities, whereas small steps Δt slow down the propagation of the profile. Scaling analysis gives $\Delta t \sim \epsilon^2$. In our tests we use $\Delta t \in [0.3\epsilon^2,\ 0.6\epsilon^2]$. It is possible to derive an exact estimation for the time stepping as it is shown for simple one-dimensional regularized functions in the appendix.

The last parameter to be set is λ. We refer the reader to Rosenthal and Linsen [24] for the stability analysis of the smoothing term in Equation (9) for regular and unstructured grids. Small λ may result in non-smooth surface, large values make computations unstable. For our tests we use $\Delta t\lambda < 0.5$.

Then we should specify an initial condition for Equation (9) to obtain a Cauchy problem. For the reasons discussed in the previous section, it is desirable to set $\varphi(\mathbf{x}, 0)$ as a signed distance function, i.e., $|\nabla\varphi(\mathbf{x}, 0)| = 1$. In our tests, we use the scalar function f representing mass density or temperature distribution in an astrophysical object. These quantities necessarily vanish on the boundary of the domain D. Thus, their isosurfaces are closed hypersurfaces with gradients pointing inside the closed volume. Therefore, it is preferable to initialize the level-set function with a closed zero-level surface embedded in D and containing the subdomain of interest D' inside the bounded volume. A simple way to achieve this, is to use the ansatz

$$\varphi(\mathbf{x}, 0) = r_0 - |\mathbf{x} - \mathbf{x}_0|$$

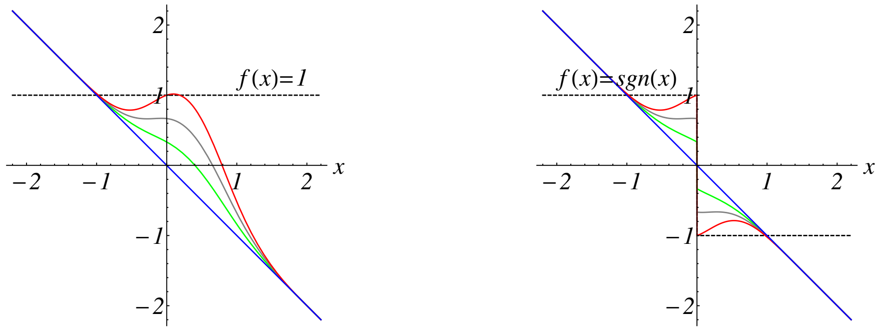


Figure 2 Dependance of the profile of the level-set function after one time step on the choice of Δt using the one-dimensional underlying scalar fields $f(x) = 1$ (left) and $f(x) = sgn(x)$ (right). Initial level-set function $\varphi^0(x) = -x$ (blue line) and level-set function $\varphi^1(x)$ after first time step for $\Delta t = 0.5,\ 1.0,\ 1.5$ (green, gray, and red lines, respectively) are shown with parameter $\epsilon = 1$. Large time steps destroy the local monotonicity of φ.

with some parameters $r_0 > 0$ and $\mathbf{x}_0 \in D$ representing the radius and center of a sphere, respectively. Let $\Omega^+(g) = \{\mathbf{x} \in D :\ g(\mathbf{x}) > 0\}$ and $\Omega^-(g) = \Omega^+(-g)$ for any scalar function g. Let the region $\Omega^+(f - f_{\text{iso}})$ have several components. Our method recovers all isosurfaces, for which the intersection of $\Omega^+(f - f_{\text{iso}})$ with $\Omega^+(\varphi)$ is non-empty.

5 Level-Set Method without Reinitialization

A new variational formulation for active contours was proposed by Li et al. [14]. The main idea is to add the term

$$E_3 = \frac{1}{2}\int_D (|\nabla\varphi(\mathbf{x})| - 1)^2\, \mathrm{d}\mathbf{x} \tag{11}$$

added with some weight μ to the functional E. This term penalizes deviation of the level-set function from a signed distance function. It aims to make the algorithm more efficient by eliminating the costly reinitialization routine.

The associated Euler-Lagrange equation reads

$$0 = -\ \nabla \cdot \left[(|\nabla\varphi(\mathbf{x})| - 1)\frac{\nabla\varphi(\mathbf{x})}{|\nabla\varphi(\mathbf{x})|}\right]. \tag{12}$$

We add the weighted term to Equation (7), discretize it in space and time, and derive the formulation

$$\begin{aligned}\varphi^{k+1} = \varphi^k + \Delta t\ \Big[&\delta_\epsilon(\varphi^k)\ (\operatorname{sgn}(f - f_{\text{iso}}) - \operatorname{sgn}_\epsilon(\varphi^k)) \\ &+\lambda\, \delta_{\epsilon'}(\varphi^k)\ \nabla \cdot \left(\frac{\nabla\varphi^k}{|\nabla\varphi^k|}\right) + \mu\, \nabla \cdot \left((|\nabla\varphi^k| - 1)\frac{\nabla\varphi^k}{|\nabla\varphi^k|}\right)\Big].\end{aligned} \tag{13}$$

One of our tasks was to test the algorithm without reinitialization on the problems of smooth isosurface extraction in comparison to the standard variant with the Eikonal equation.

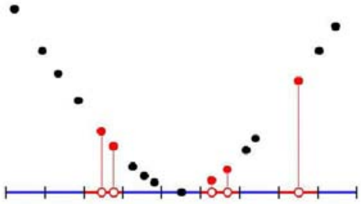


Figure 3 Scheme of the local completion of the point-based surface. The sample of interest and its neighbors (dark), added rough estimates of additional samples (red non-filled), and corrected additional points (red filled) are shown.

6 Isopoints Computation

Following steps are performed after finishing an iteration due to the level-set equation:

1. a rough estimate for a set of isopoints $\mathbf{p}_k$ is computed;
2. positions of $\mathbf{p}_k$ are improved;
3. the set of isopoints is completed and rendered.

In the first step, we find all pairs of neighboring samples $\mathbf{x}_i$ and $\mathbf{x}_j$ such that $\operatorname{sgn}(\varphi_i) \neq \operatorname{sgn}(\varphi_j)$. Using an angle criterion [23], some pairs are rejected. Isopoints are found by means of linear interpolation between the pairs of detected samples.

In the second step, we refine the position of each isopoint $\mathbf{p}_k$ using MLS and Newton's method as follows:

$$\mathbf{p}_k^{i+1} = \mathbf{p}_k^i - \frac{\varphi(\mathbf{p}_k^i)}{|\nabla\varphi(\mathbf{p}_k^i)|^2}\nabla\varphi(\mathbf{p}_k^i), \qquad \mathbf{p}_k^0 = \mathbf{p}_k. \tag{14}$$

Iterations are performed until $|\mathbf{p}_k^{i+1} - \mathbf{p}_k^i|$ is sufficiently small.

The resulting point set is quite accurate but in most cases has low (and non-uniform) density for rendering. We complete the set as it is shown in Figure 3. In the neighborhood of each $\mathbf{p}_k$ we compute a linear approximation to the isosurface. The plane is divided in cells of small size. Then, neighbors of $\mathbf{p}_k$ are orthogonally projected to the plane. We add new samples in empty cells and refine them by Newton's method (14). Normals at isopoints are computed during the refinement procedure. Assuring high sampling rate, the final completed point set can be rendered using mere point renderings (with local shading). Alternatively, one can use a somewhat lower sampling rate and use the standard splat-based rendering approach

7 Comparison with Literature

There are three aspects that distinguish our approach from the one proposed by Rosenthal and Linsen [24], namely: the main part of the cost function, the variational derivation of the level-set equation, and the non-morphological approach. In addition, we consider an algorithm with a proper diffusive term to make the reinitialization procedure obsolete.

The functional

$$\bar{E} = \int_D |\varphi(\mathbf{x}) - (f(\mathbf{x}) - f_{\text{iso}})|\,\mathrm{d}\mathbf{x} \tag{15}$$

was proposed by Rosenthal and Linsen [23] to measure the accuracy of isosurface extraction. The proposed evolution level-set equation has the form

$$\frac{\partial\varphi(\mathbf{x},t)}{\partial t} + (\varphi(\mathbf{x},t) - (f(\mathbf{x}) - f_{\text{iso}}))\,|\nabla\varphi(\mathbf{x},t)| = 0\,. \tag{16}$$

This equation pushes φ towards $f(\mathbf{x}) - f_{\text{iso}}$ on the global scope, whereas the reinitialization step tries to restore the signed distance function. As a result, φ converges to some compromise, possibly being far away from both. Moreover, if we scale the field $f(\mathbf{x}) - f_{\text{iso}}$ by a large constant factor k, the isosurface obviously remains unchanged, but the discretized Equation (16) requires a much smaller time step Δt. When nevertheless choosing larger time steps, the reinitialization procedure needs to be called after each iteration and needs to perform many iterations on the whole data set. Obviously, the computational costs increase significantly.

It is absolutely not necessary to achieve $\varphi(\mathbf{x},t) \to f(\mathbf{x}) - f_{\text{iso}}$ to be able to detect the isosurface Γ_{iso}. It suffices to make the sets $\Omega^{\pm}(\varphi)$ and $\Omega^{\pm}(f - f_{\text{iso}})$ coincide. The closeness of the sets can be measured with the functional E_1. In contrast to $\bar{E}$, our functional and level-set equation parameters are not sensible to the magnitude of f, which makes our approach preferable. In practice, the minimization of E_1 on a discrete set of nodes results in the detection of the isosurface that is accurate up to the local internodal scale. Correction of the isosurface on the subgrid scale is then dominated by the smoothing and reinitialization procedures.

Finally, we underline the fact that our goal is to extract local isosurfaces. The method automatically detects those parts of a multi-component isosurface, which intersect with the subdomain of interest. This property will be demonstrated in experiments below.

8 Results and Discussion

8.1 Measurement Functionals

The following functionals are computed and recorded during our tests to measure the quality of the solution.

$$P(t) = \frac{1}{2N}\sum_{i=1}^{N} |\text{sgn}(\varphi(\mathbf{x}_i,t)) - \text{sgn}(f(\mathbf{x}_i) - f_{\text{iso}})|$$

computes the relative number of samples, at which the sign of the level-set function is different to the sign of the underlying scalar field. P can remain large, if not all isosurface components are extracted. Constant P indicates no propagation of the isosurface.

$$MG(t) = \max_j \left\{|\nabla\varphi(\mathbf{p}_j,t)|,\ |\nabla\varphi(\mathbf{p}_j,t)|^{-1}\right\}$$

and

$$AG(t) = \frac{1}{M}\sum_{j=1}^{M} \max\left\{|\nabla\varphi(\mathbf{p}_j,t)|,\ |\nabla\varphi(\mathbf{p}_j,t)|^{-1}\right\}$$

measure the maximal and the average deviation of φ from the signed distance function at M isopoints. If the analytical form of f is known, we can estimate the average distance from isopoints to the exact isosurface as follows

$$L(t) = \frac{1}{M}\sum_{j=1}^{M} \frac{|f(\mathbf{p}_j) - f_{\text{iso}}|}{|\nabla f(\mathbf{p}_j)|}.$$

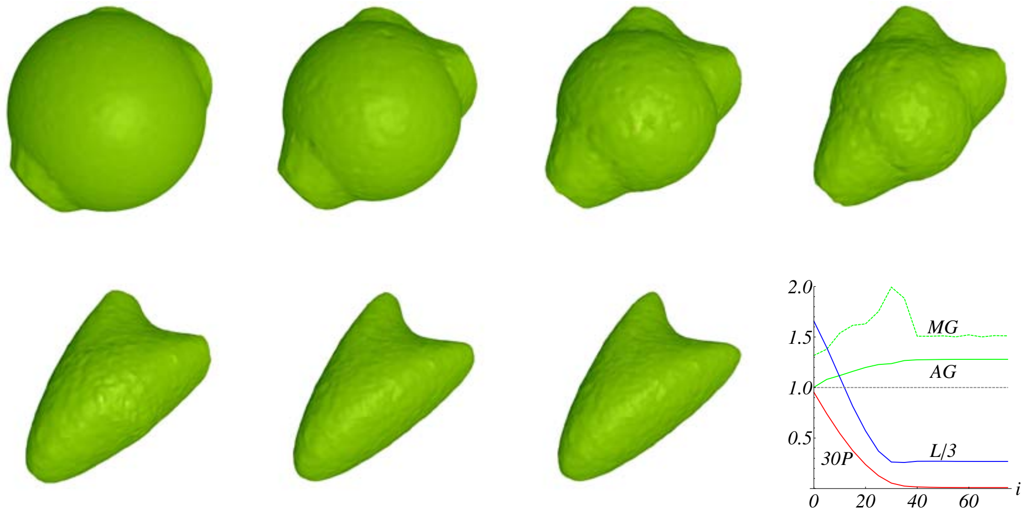


Figure 4 Level-set evolution of the isosurface for a one-component synthetic data set: steps 5, 10, 15, 20, 30, 50, and 75 are presented. Splat-based rendering of refined isopoint sets is performed. Parameters: total number of samples in the box $[0, 128]^3$ is 2M, $\triangle t = 1.0$, $\epsilon = 4.0$, $\epsilon' = 5.0$, and $\lambda = 0.15$. Two reinitialization steps are performed after each iteration.

Here, we used

$$|\mathbf{p}_j - \mathbf{y}| = \frac{|f(\mathbf{p}_j) - f(\mathbf{y})|}{|f(\mathbf{p}_j) - f(\mathbf{y})|} \|\mathbf{p}_j - \mathbf{y}\| \approx \frac{|f(\mathbf{p}_j) - f_{\mathrm{iso}}|}{|\nabla f(\mathbf{p}_j)|},$$

where $\mathbf{y}$ is the point closest to $\mathbf{p}_j$ on the exact isosurface: $f(\mathbf{y}) = f_{\mathrm{iso}}$. The approximation is accurate for small $|\mathbf{p}_j - \mathbf{y}|$.

Beside the quantities above, we are interested in the efficiency of the methods, which is related to the amount of time needed to perform one temporal iteration. We have performed several tests based on synthetic and real-world data presented in the following.

8.2 One-component Synthetic Data

In our first experiment, we investigated the accuracy and convergence of our method. We use a synthetic volume data given by the analytic scalar field $f(\mathbf{x}) = -((ax+0.05(ay)^3)^2+0.7(ay)^2+0.3(az - 0.7(ya)^2)^2 - 4.0)/a^2$ with $a = 0.1$. Extracted is the isosurface $\Gamma = \{\mathbf{x} : \ f(\mathbf{x}) = 0\}$. There are 2M samples randomly distributed within a cube of edge size 128. The isosurface is initialized as a sphere. Positions of the isosurface during the level-set iterations as well as the evolution of the main process characteristics are presented in Figure 4. One iteration of our level-set approach takes about 15s on a standard PC with an Intel Xeon (3.2GHz, quad-core) processor. The final average distance between isopoints and the exact isosurface is of order 0.7, which is less than the average sample distance (~ 1) and the characteristic length of the isosurface (~ 100). Here, as well as in the next experiments we intentionally perform more iterations than necessary to document convergence of our method.

8.3 Multi-component Synthetic Data

The second experiment shows the extraction of a subset of isosurface components and illustrates the flexibility of the method with respect to the initial surface setting. The results are shown in Figure 5. The initial isosurface represents two intersecting spheres.

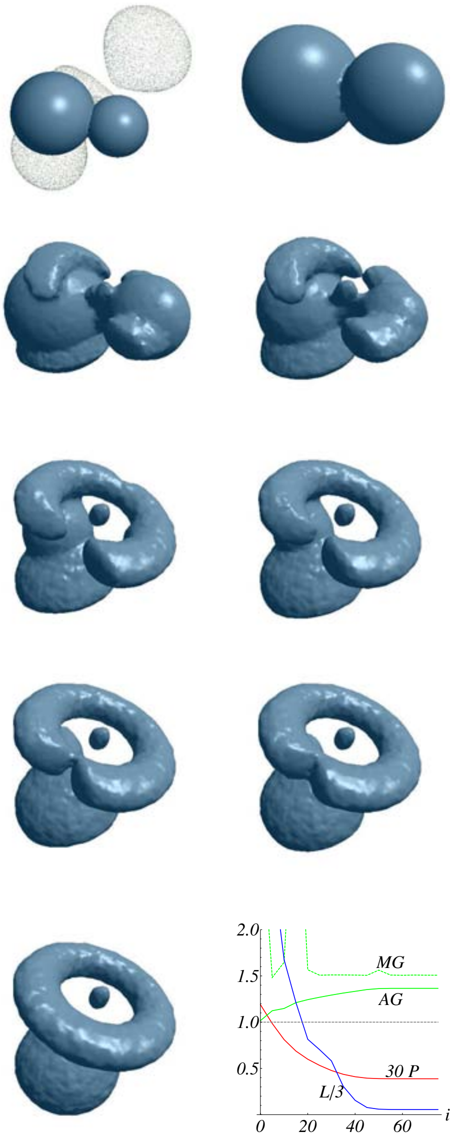


Figure 5 Level-set evolution of the isosurface for a multi-component synthetic data set: steps 0, 3, 5, 8, 10, 20, 30, 50 and 100 are shown. Point rendering is used. Parameters: total number of samples in the box $[0, 128]^3$ is 2M, $\triangle t = 1.25$, $\epsilon = 3.5$, $\epsilon' = 5.0$, and $\lambda = 0.01$. Two reinitialization steps are performed after each iteration.

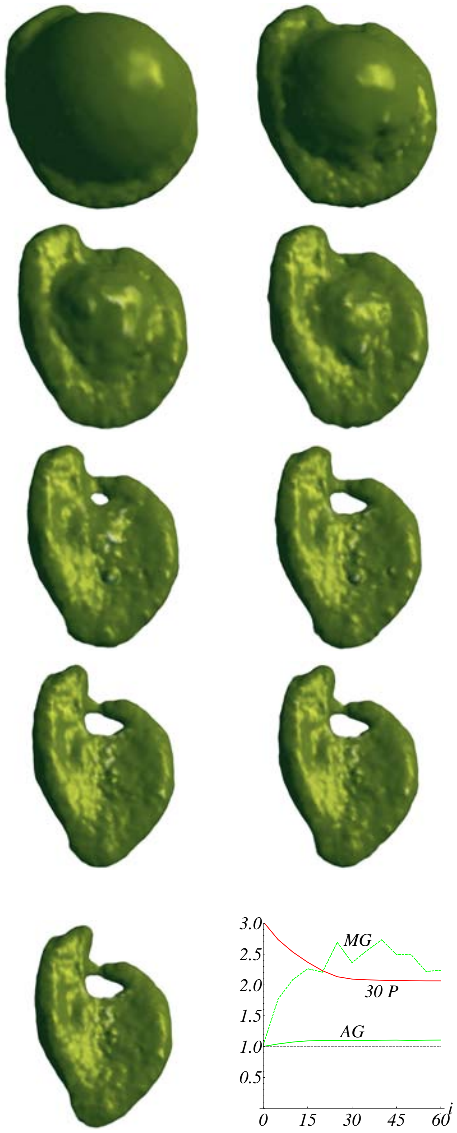


Figure 6 Level-set evolution of the isosurface for the White Dwarf data set: steps 0, 3, 5, 8, 10, 20, 30, 50 and 100 are shown. Point rendering is used. Parameters: total number of samples in the box $[0, 128]^3$ is 2M, $\triangle t = 1.25$, $\epsilon = 3.5$, $\epsilon' = 5.0$, and $\lambda = 0.01$. Four reinitialization steps are performed after each iteration. Data set courtesy of Stephan Rosswog, Jacobs University, Bremen, Germany.

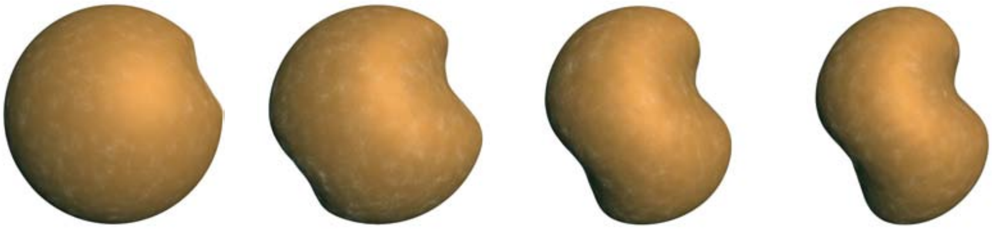


Figure 7 Level-set evolution of the isosurface without reinitialization for the data set with two White Dwarfs: steps 10, 50, 100 and 150 are shown. Splat-based rendering is used. Parameters: total number of samples is 2.5M, $\triangle t = 0.0025$, $\epsilon = 1.0$, $\epsilon' = 0.2$, $\lambda = 10^{-8}$, and $\mu = 10^{-6}$. Data set courtesy of Stephan Rosswog, Jacobs University, Bremen, Germany.

One iteration takes about 15s. The final average distance between isopoints and the exact isosurface is ≈ 0.14 (the average sample distance: ~ 1, object size: ~ 90). The method is stable to the isosurface's topology change. Larger values of λ lead to disappearance of the central spherical isosurface component due to the high curvature values. High values of $MG(t)$ in the first steps are due to non-smooth behavior of the initial level-set function: the gradient degenerates at the samples close to the spheres' intersection line.

8.4 Real Volume Data

In the next test we extract one of two isosurface components by applying our method to a region of interest of a real-world astrophysical data set. The data set consists of 0.5M samples and represents a White Dwarf passing close to a Black Hole (not represented in data). The isosurface of the star corresponding to $f_{\text{iso}} = 0.001$ consists of two parts (a head and a tail). We extract one of them placed in the core of the White Dwarf. The results are shown in Figure 6. One iteration takes about 12s.

8.5 Method without Reinitialization

We applied the reinitialization-free method (13) to all data sets from previous tests. We found that the propagation of the isosurface is too slow for various parameters combinations we tried. The reason could be that the diffusion-like term may push the isosurface in the direction opposite to its movement. The larger μ, the more obvious is this action. Low μ prevents the suppression of high gradients, which also leads to slow propagation of the surface. In contrast, this is not the case for the method with reinitialization, since we do not change the isosurface location while solving the Eikonal equation (10).

Results of one experiment for the reinitialization-free algorithm are presented in Figure 7. The data set consist of 2.5M samples and represents two White Dwarfs passing close to each other with a mass accretion developed. We extract the isosurface $\Gamma = \{\mathbf{x} : f(\mathbf{x}) = 0.055\}$ of one star only. The level-set function was initialized as a sphere. One iteration took $\sim 12s$ computational time. In Figure 8 we compare the evolution of gradient norms (minimal, maximal, and average) with the diffusion term enabled (green curves) and disabled (blue curves). The diffusion term moves the average gradient norm toward the value 1 and therefore helps the surface propagation. However, since the coefficient μ was chosen to be small, the effect is minimal. It is impossible to increase μ, as it leads to unstable computations.

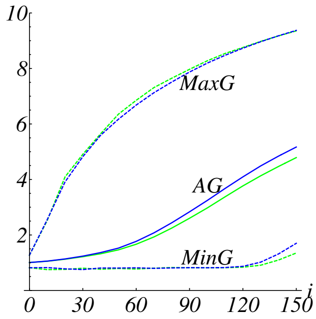


Figure 8 Influence of the diffusion term from Equation (5) on the behavior of the gradient norm. Green lines represent the experiments of Figure 7 with $\mu = 10^{-6}$, while blue lines show the same experiments with $\mu = 0$. The improvement of the gradient norm evolution is minimal, but larger values of μ lead to instable computations.

9 Conclusion and Future Work

We have proposed a novel level-set approach for smooth isosurface extraction that works for both structured and unstructured point-based volume data. A variational approach was used to derive a level-set equation that determines the evolution of the interface towards its steady state. The variational approach provides fast convergence. The action of the main terms of the governing equation at each time step was localized to a narrow neighborhood around the zero level surface of the auxiliary function. This led to a speed-up of our method when compared to algorithms acting on all levels. The locality allowed us to apply our method to subdomains of interest and to extract a subset of isocomponents. Finally, we tested a specific diffusion term in the level-set equation to avoid costly reinitialization.

We compared our algorithm to the state-of-the-art approach proposed by Rosenthal and Linsen. We showed that in their algorithm the efficiency was strongly affected by properties of the underlying scalar field f. In particular, large gradients required small time steps and frequent reinitialization. We removed this issue by a proper choice of the cost functional. We tested the proposed method on synthetic and real-world data sets. Localization of the update reduced the computational time of the isosurface extraction in the White Dwarf data by factor 2 when compared to the method by Rosenthal and Linsen.

In the proposed method, all computations (except those due to the signed distance property) are concentrated in a small neighborhood around the zero-isosurface of the level-set function. It provides a gradual smooth evolution of the surface during computations, but still requires significant computational efforts to maintain the signed distance property. An improvement in the efficiency of the method can be obtained by incorporating a local reinitialization procedure in the algorithm.

There are possible improvements in the algorithm which would increase the accuracy at the cost of efficiency:

- use of a higher-order basis in MLS;
- more accurate estimation of MLS gradients [3]; and
- more accurate time-integration scheme (provided that the spatial approximation is accurate).

To increase the efficiency of the method, one can use individual time steps for samples and smoothing parameters that are adaptive in space and time.

Acknowledgements

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under project grant LI-1530/6-1.

Appendix

▶ **Theorem 1.** *Let*

$$\varphi^{k+1}(x) = \varphi^k(x) + \Delta t\, \delta_\alpha(\varphi^k(x))\, (\operatorname{sgn}(f(x) - f_{\text{iso}}) - \operatorname{sgn}_\alpha(\varphi^k(x)))$$

for all natural $k \geq 1$ with some $\varphi^0(x)$, $f(x)$, and a constant f_{iso}. Here,

$$\delta_\alpha(x) = \begin{cases} \dfrac{(\alpha - |x|)}{\alpha^2} & |x| < \alpha \\ 0 & \textit{else} \end{cases}, \qquad \operatorname{sgn}_\alpha(x) = \begin{cases} \dfrac{x\,(2\alpha - |x|)}{\alpha^2} & |x| < \alpha \\ \operatorname{sgn}(x) & \textit{else} \end{cases}.$$

Then, $\varphi^k(x) \to \varphi^\infty(x)$ as $k \to \infty$ pointwise for $\Delta t < \alpha^2$, where

$$\varphi^\infty(x) = \begin{cases} \alpha \operatorname{sgn}(f(x) - f_{\text{iso}}) & \textit{for } |\varphi^0(x)| < \alpha \\ \varphi^0(x) & \textit{else} \end{cases}.$$

Moreover, $\varphi^k(x)$ inherits the monotonicity of $\varphi^0(x)$ if $\Delta t < \alpha^2/3$.

Proof. For all x with $|\varphi^0(x)| \geq \alpha$, the main equation reads $\varphi^{k+1}(x) \equiv \varphi^k(x)$. Thus, the statement is trivial. Now, we assume $\operatorname{sgn}(f(x) - f_{\text{iso}}) \equiv 1$ for those x at which $0 \leq \varphi^0(x) < \alpha$. Other cases can be proven by analogy. The simplified equation reads

$$\varphi^{k+1}(x) = \varphi^k(x) + \frac{\Delta t}{\alpha^4}\,(\alpha - \varphi^k(x))\,(\alpha^2 - \varphi^k(x)\,(2\alpha - \varphi^k(x))) = \varphi^k(x) + \frac{\Delta t}{\alpha^4}\,(\alpha - \varphi^k(x))^3.$$

The following estimations for $\Delta t \leq \alpha^2$ and $0 \leq \varphi^k(x) < \alpha$ hold:

$$\varphi^{k+1}(x) - \varphi^k(x) = \frac{\Delta t}{\alpha^4}\,(\alpha - \varphi^k(x))^3 > 0;$$

$$\alpha - \varphi^{k+1}(x) = \alpha - \varphi^k(x) - \frac{\Delta t}{\alpha^4}\,(\alpha - \varphi^k(x))^3 = (\alpha - \varphi^k(x))\left[1 - \frac{\Delta t}{\alpha^4}\,(\alpha - \varphi^k(x))^2\right] \geq 0.$$

This shows that $\{\varphi^k(x)\}$ is a strictly monotonely increasing sequence bounded from above. Clearly, α is the minimal upper bound. Thus, the first part of the theorem is proven.

To show the monotonicity of $\varphi^k(x)$ with respect to x, we observe that for the function $f(x) = x + \Delta t(\alpha - x)^3/\alpha^4$

$$f'(x) = 1 - 3\Delta t(\alpha - x)^2/\alpha^4 \geq 0$$

if $\Delta t \leq \alpha^2/3$ and $0 \leq x < \alpha$. Therefore, $\varphi^{k+1}(x) \geq \varphi^{k+1}(y)$ if $\varphi^k(x) \geq \varphi^k(y)$. ◀

The regularized functions used in our experiments have a support that is two times larger than the support of the functions in the theorem statement. The substitution $\alpha = 2\epsilon$ leads to the estimation $\Delta t \leq 4\epsilon^2/3$, which is close to what one finds numerically.

References

1 M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1), 2003.

2 Nina Amenta and Yong Joo Kil. Defining point-set surfaces. *ACM Trans. Graph.*, 23(3):264–270, 2004.

3 T. Belytschko, Y. Y. Lu, and L. Gu. Element-free galerkin methods. *Int. J. Numer. Meth. Engng*, 37:229–256, 1994.

4 Wolfgang Boehm and Hartmut Prautzsch. *Numerical methods.* VIEWEG, 1993.

5 D. Breen, R. Whitaker, K. Museth, and L. Zhukov. Level set segmentation of biological volume data sets. In *Handbook of Medical Image Analysis, Volume I: Segmentation Part A*, pages 415–478, New York, 2005. Kluwer.

6 T. F. Chan and L. A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, 2001.

7 C. S. Co and K. I. Joy. Isosurface Generation for Large-Scale Scattered Data Visualization. In G. Greiner, J. Hornegger, H. Niemann, and M. Stamminger, editors, *Proceedings of Vision, Modeling, and Visualization 2005*, pages 233–240. Akademische Verlagsgesellschaft Aka GmbH, November 16–18 2005.

8 R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics — theory and application to non-spherical stars. *Mon. Not. Roy. Astron. Soc.*, 181:375–389, November 1977.

9 Y. Jung, K. T. Chu, and S. Torquato. A variational level set approach for surface area minimization of triply-periodic surfaces. *J. Comput. Phys.*, 223(2):711–730, 2007.

10 Christian Ledergerber, Gaël Guennebaud, Miriah Meyer, Moritz Bächer, and Hanspeter Pfister. Volume mls ray casting. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1372–1379, 2008.

11 Aaron E. Lefohn, Joe M. Kniss, Charles D. Hansen, and Ross T. Whitaker. Interactive deformation and visualization of level set surfaces using graphics hardware. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 11, Washington, DC, USA, 2003. IEEE Computer Society.

12 Aaron E. Lefohn, Joe M. Kniss, Charles D. Hansen, and Ross T. Whitaker. A streaming narrow-band algorithm: interactive computation and visualization of level sets. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 243, New York, NY, USA, 2005. ACM.

13 D. Levin. Mesh-independent surface interpolation. *Geometric Modeling for Scientifc Visualization*, 2003.

14 Chunming Li, Chenyang Xu, Changfeng Gui, and M. D. Fox. Level set evolution without re-initialization: a new variational formulation. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, pages 430–436 vol. 1, 2005.

15 Lars Linsen, Tran Van Long, Paul Rosenthal, and Stephan Rosswog. Surface extraction from multi-field particle volume data using multi-dimensional cluster visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1483–1490, 2008.

16 L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024, December 1977.

17 Ken Museth, David E. Breen, Leonid Zhukov, and Ross T. Whitaker. Level-set segmentation from multiple non-uniform volume datasets. *Visualization Conference, IEEE*, 2002.

18 B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Comput. Mech.*, 10:307–318, 1992.

19 Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces.* Springer, 2003.

20 Sung Park, Lars Linsen, Oliver Kreylos, John D. Owens, and Bernd Hamann. A framework for real-time volume visualization of streaming scattered data. In Marc Stamminger and Joachim Hornegger, editors, *Proceedings of Tenth International Fall Workshop on Vision, Modeling, and Visualization 2005*, pages 225–232,507. DFG Collaborative Research Center, 2005.

21 Sung W. Park, Lars Linsen, Oliver Kreylos, John D. Owens, and Bernd Hamann. Discrete Sibson interpolation. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):243–253, 2006.

22 Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. A pde-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, November 1999.

23 Paul Rosenthal and Lars Linsen. Direct isosurface extraction from scattered volume data. In *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization*, pages 99–106, 2006.

24 Paul Rosenthal and Lars Linsen. Smooth surface extraction from unstructured point-based volume data using PDEs. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1531–1546, 2008.

25 Rudiger Westermann, Christopher Johnson, and Thomas Ertl. A level-set method for flow visualization. In *In Proceedings of Viz2000, IEEE Visualization*, pages 147–154, 2000.

26 R.T. Whitaker. Isosurfaces and level-sets. In C.D. Hansen and C.R. Johnson, editors, *The Visualization Handbook*, pages 97–123. Elsevier, 2005.

27 Hong-Kai Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127(1):179–195, 1996.

28 Hong-Kai Zhao, Stanley Oshery, Barry Merrimany, and Myungjoo Kangy. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding*, 80:295–319, 2000.

29 Matthias Zwicker, Jussi Räsänen, Mario Botsch, Carsten Dachsbacher, and Mark Pauly. Perspective accurate splatting. In *GI '04: Proceedings of Graphics Interface 2004*, pages 247–254, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human-Computer Communications Society.

Reflections on QuestVis: A Visualization System for an Environmental Sustainability Model *

Tamara Munzner[1], Aaron Barsky[1], and Matt Williams[1]

1 Department of Computer Science, University of British Columbia
201-2366 Main Mall, Vancouver BC V6T 1Z4, Canada
{tmm,barskya,mwill}@cs.ubc.ca

Abstract

We present lessons learned from the iterative design of QuestVis, a visualization interface for the QUEST environmental sustainability model. The QUEST model predicts the effects of policy choices in the present using scenarios of future outcomes that consist of several hundred indicators. QuestVis treats this information as a high-dimensional dataset, and shows the relationship between input choices and output indicators using linked views and a compact multilevel browser for indicator values. A first prototype also featured an overview of the space of all possible scenarios based on dimensionality reduction, but this representation was deemed to be be inappropriate for a target audience of people unfamiliar with data analysis. A second prototype with a considerably simplified and streamlined interface was created that supported comparison between multiple scenarios using a flexible approach to aggregation. However, QuestVis was not deployed because of a mismatch between the design goals of the project and the true needs of the target user community, who did not need to carry out detailed analysis of the high-dimensional dataset. We discuss this breakdown in the context of a nested model for visualization design and evaluation.

1998 ACM Subject Classification I.3.6 Computer Graphics, Methodology and Techniques,Graphics data structures and data types

Keywords and phrases high-dimensional visualization, dimensionality reduction, linked views, simulation visualization, design study

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.240

1 Introduction

Most people have a vision of the future they desire. A desired future might have low unemployment, or less traffic, or clean air, or maybe all of the above. However, people are often unaware of how the interplay between regional policy choices made in the present could either bring about or prevent desired aspects of these futures.

The Georgia Basins Futures Project created an interactive software system called QUEST (Quite Useful Ecosystem Scenario Tool) to engage the public in dialogue about sustainability choices and outcomes [4]. QUEST contains computational models that generate future scenarios by predicting living conditions over the next forty years for the Georgia Basin region. These models are informed by expert understanding of ecological, social and economic systems. They have free parameters that users can set to reflect assumptions about how the future will unfold, governmental policies that will be implemented at a local level, and individual behaviors. QUEST is designed for use by lay people interested in learning about

* This work was supported by the GEOIDE NCE.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 240–259
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

sustainability issues in a facilitated workshop format lasting less than a single day. The goal of the project is to increase a participant's understanding of how current decisions affect the sustainability for the region. Its underlying philosophy is that the real purpose of modeling is not to describe the world, but to change the thinking of users. After its initial development at the Sustainable Development Research Institute at the University of British Columbia, QUEST has been further refined by the spinoff company Envision Sustainability Tools, Inc.

A single QUEST prediction has 294 indicators that characterize a future scenario. Example indicators include the amount of coal burned for electricity generation, the average amount of time spent in a car commuting each day, and the health of the sockeye salmon fisheries. Any change to a model parameter will generates a new scenario, where some or all of these 294 indicators may have changed.

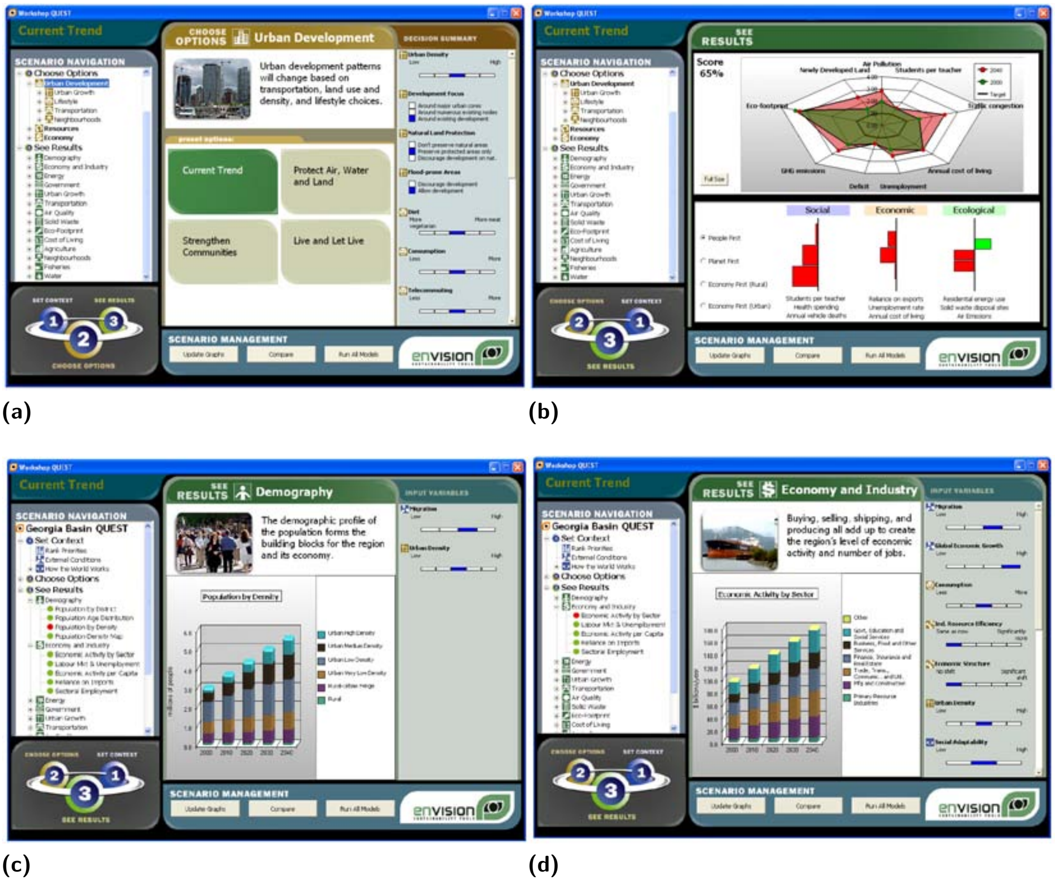
From an information visualization perspective, we can consider these QUEST predictions as a high-dimensional data set with 294 dimensions. The goal of the QuestVis project was to use information visualization techniques to allow individual users to explore this rich data set of scenarios more extensively, both within and between scenarios. In this paper, we will present the design of the two interactive prototypes that we iteratively developed. The main contribution of this paper is the reflection on the lessons learned in this process.

We begin in Section 2 with background about the usage of the current QUEST system in workshops. Section 3 presents the design choices of the first prototype, and Section 4 discusses its limitations. We continue with the design of the second prototype in Section 5, and then reflections on its strengths and limitations in Section 6. We then discuss the related work in Section 7, and conclude in Section 8.

2 QUEST Usage

The QUEST software was designed for use in a large-group workshop format with a trained facilitator, rather than for individual exploratory use. A QUEST session has three sequential stages: an input stage where the group makes policy decisions by specifying the choices for a scenario, a model computation stage to which generates scenario indicators as output, and an analysis stage where future scenarios are understood through inspection of these indicators. Optimizing for some indicators may force others to have less desirable values. QUEST is intended to help lay people understand these tradeoffs at a deeper level than they did before participating in a workshop. These people are often citizens in a particular community that are wrestling with thorny policy choices, and QUEST workshops are used as part of a public engagement process.

In the input stage, up to 49 policy choices can be made. For example, the waste reduction policy choice controls which of five levels the scenario will use, from a maximum of `significant reduction` to a minimum of `same as now`. The facilitator leads the group through these choices, with significant discussion of possible consequences of their decision making. These choices are grouped into categories, and QUEST supports *presets* that can specify many input choices at once within a category. For example, Figure 1a shows that the `Urban Development` options has four presets: `Current Trend`; `Protect Air, Water and Land`; `Strengthen Communities`; and `Live and Let Live`. Choosing one of these presets sets all seven of the input choices, although any of them can be explicitly overridden after that choice is made. At a higher level, there is a two-phase structure, with a division into the `Set Context` and `Choose Options` phases. After all input decisions are made, the facilitator explicitly starts the model computation phase, which takes approximately two minutes. That time is typically spent in continued group discussion.



Urban development patterns will change based on transportation, land use and density, and lifestyle choices.

The demographic profile of the population forms the building blocks for the region and its economy.

Buying, selling, shipping, and producing all add up to create the region's level of economic activity and number of jobs.

Figure 1 The QUEST user interface. **(a)** In the input stage, policy decisions are made by using the `Scenario Navigation` menu on the left to select an option category, and then either using one of the `preset options` in the middle to specify an entire set of choices at once, or making detailed choices for individual options on the right. **(b)** The analysis stage begins with the overview page, showing a carefully chosen subset of 9 of the 294 output indicators using a radial chart. **(c)** Picking picking one of the 88 possible windows from the hierarchical `Scenario Navigation` menu on the left allows detailed analysis of the 294 QUEST output indicators for the chosen scenario. The `Population by Density` view shows six of the indicators as stacked bar charts. The input choices related to each view are shown on the right as read-only displays that cannot be changed at this stage. **(d)** The `Economic Activity by Sector` view shows seven other indicators.

The third stage of QUEST usage is the analysis phase. Figure 1b shows the overview screen for the scenario that presents nine of the indicators in radial chart form, with the values for today overlaid with the values forty years from now as a radar graph. The 294 output indicators are categorized into a two-level hierarchy with a total of 88 detailed views featuring bar charts, which can be selected by drilling down through the `Scenario Navigation` result menus. Figures 1c and 1d show two of these views, which are used to understand specific aspects of the chosen future scenario. The input variables that could affect the indicators in question are shown on the right, but they cannot be changed during this analysis stage.

The entire facilitated process of choosing the input decisions, computing the future, and analyzing a single chosen future scenario typically takes over an hour. The facilitator does not typically lead the group through an exhaustive analysis of every single indicator, but focuses on the most relevant ones. In a workshop, this process is typically repeated a few times to analyze a small number of scenarios.

2.1 QUEST Strengths and Limitations

Many design decisions in QUEST were made with careful attention to the difficult nuances of how to use technology to engage citizens in an informed sustainability discussion, with the ultimate goal of not only furthering their understanding but changing their behavior. The user interface has a very polished graphic design, a very appropriate choice for its intended audience of the nontechnical lay public. The software is well suited for a detailed investigation of a few key indicators, as part of an in-depth facilitated group discussion.

However, when considering the capabilities of the software as an interface that could allow the exploration of a rich information space, we identified a significant limitation. The tool does not provide support for synthesizing a high-level understanding of how specific input choices allow or preclude whole sets of scenarios. The reason for this limitation is two-fold.

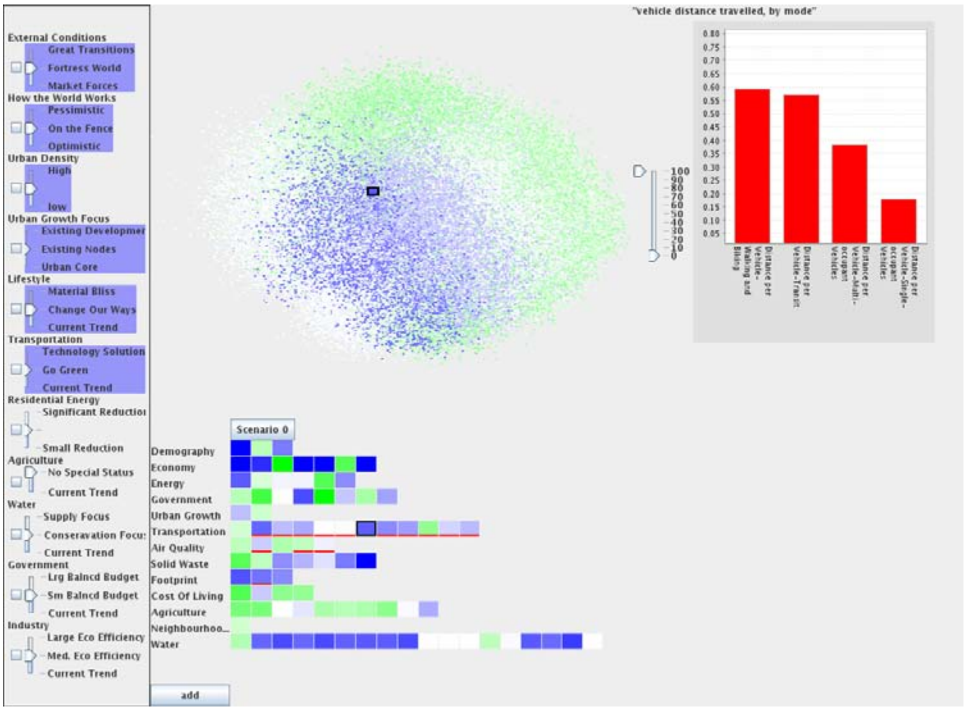
First, it is difficult to compare indicators from different categories within a single scenario because they are presented on separate screens. Individual screens have a low *information density*, using a large number of pixels to show only a few variables. To comprehend the relationships between the highly interdependent output indicators, for example whether an attempt to improve water quality will always result in an economic loss as measured across several possible indicators, people must internally synthesize results from multiple screens. However, research has shown strict limits on the capacity for people to conduct a detailed comparison of remembered images with their current view [12, 17]. Moreover, the overview only presents information about nine specific indicators. If the user is interested in any of the 285 other indicators, the overview screen is not useful.

Second, comparing indicator values between different scenarios is even more difficult, so it is difficult to make a judgement about the quality of a particular scenario compared to other possible alternatives. The central problem is the three-stage architecture where interaction is separated into two distinct phases, setting input choices versus analyzing output indicators, with a wait of multiple minutes between the two. A fundamental theory of interaction design separates human cognitive response to systems based on response time, with major differences at three threshold limits: multisecond, one second, and subsecond wait times [3]. The multi-minute gap precludes any kind of tight coupling where the result of changing an input choice on the output indicators is immediately visible. QUEST displays the specific inputs that might affect the output indicators on a given screen in the analysis phase, but they are read-only and cannot be changed. Moreover, setting the inputs a relatively cumbersome multi-step operation, because the user must use a menu to navigate to the multiple views to make input choices across many categories.

Finally, the need for a trained facilitator is a major resource constraint. An earlier version of QUEST did try to support individual rather than facilitated group exploration, but it was deemed too difficult for lay users to understand the complexities of the problem without a trained facilitator.

3 First Prototype

We created QuestVis, shown in Figure 2, a fully interactive prototype with an interface designed for interactive exploration of data generated by the existing QUEST computational models. It was designed primarily to support individual exploration, although it could also be used by a facilitator in a group setting. Our hope was that powerful interactive exploration



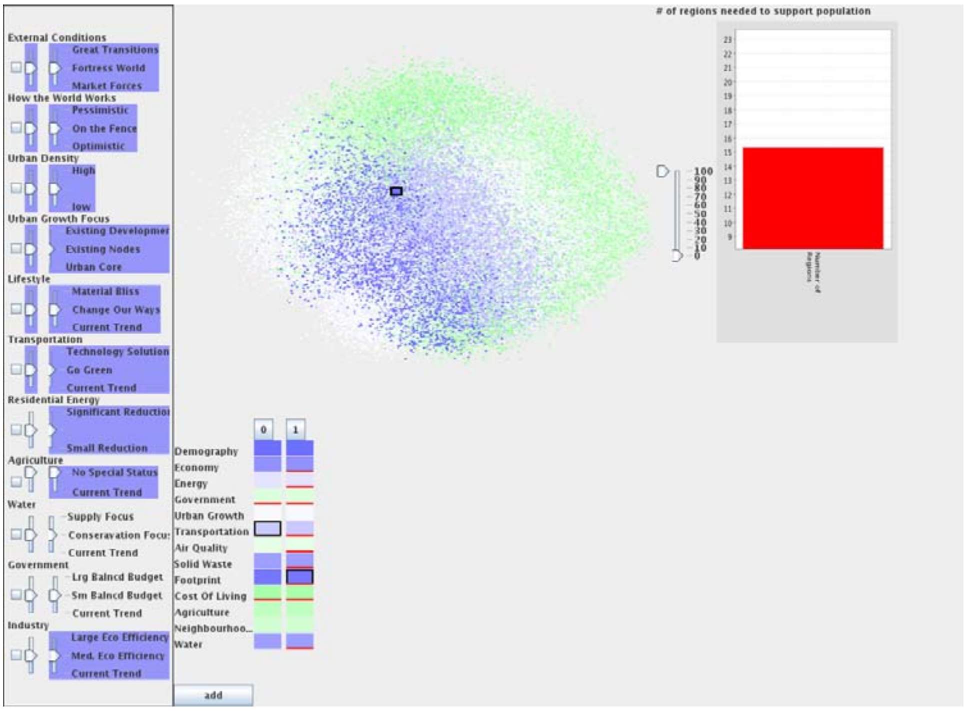


Figure 2 The first QuestVis prototype. **Top:** Exploring a single scenario, with the multilevel indicator expanded to the middle level. **Bottom:** Comparing two scenarios at the highest level of aggregation.

capabilities would allow nontechnical users to directly synthesize an understanding of the linkages between current policy choices and future outcomes, without the need for a facilitator.

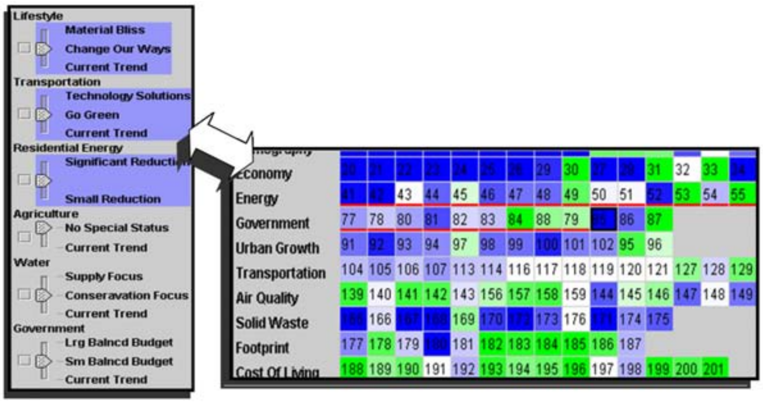
We made three major decisions in the design of the first QuestVis prototype. First, we provided linked views of the input choices and the output indicators that were always visible, with the ability to change any input choices at any time and see immediate visual feedback of how the indicators changed in response. Second, we created a multilevel overview of the indicators that was compact enough to see all indicators within a scenario simultaneously when fully expanded, or to compare between scenarios at higher levels of aggregation. Third, we showed where chosen scenarios fell within a visual representation of the space of all possible scenarios.

3.1 Linking Input Choices With Output Indicators

The goal of enabling immediate changes for output indicators in response to input choice changes required us to use a very different system architecture than the original QUEST. We generated a database of precomputed scenarios, removing the need for a model computation stage and for a separation between specifying input and analyzing output stages. The set of input choices is used as the key for the database, which returns the 294 output indicators associated with that particular scenario. The database was generated by running the computational model in batch mode, and saving the computed indicators. This architecture would not be feasible with the full set of 49 input decisions that each had four possible choices, yielding the intractable number of 10^{30} scenarios. The model creators at SDRI and Envision chose the 11 most influential input decisions to use in creating the database. Ten of these inputs had three possible choices, while one input had two choices. The resulting database was a manageable size, containing $3^{10} * 2^1 = 118{,}098$ scenarios.

QuestVis shows input choices and output indicators side by side on the same screen. The full set of 11 input choices are displayed in a panel on the left, with sliders that visually indicate the current choices and can be changed at any time. When an input choice is changed, the output indicators immediately update. QuestVis is architected as a front-end Java client that connects to a back-end database running on a separate machine. Subsecond response time for an update is typical when a high-bandwidth connection between these two machines is available.

In addition to real-time responsiveness, we create an explicit visual linkage between the input and output displays using linked highlighting. In addition to the large data set that is the mappings from input to output choices for each scenario, the database also contains a list of which input choices can affect each indicator, and which indicators can change when a particular input choice is changed. The linked mouseover highlighting makes this information visually apparent to the user at all times, as shown in Figure 3. When the user moves the cursor over an output indicator, the input sliders that can affect it are highlighted by changing their background color to light blue. Conversely, when the user moves the cursor over an input slider, the output indicators that could change when it is moved are highlighted with a red underline. Our conjecture was that the combination of real-time response and linked highlighting would allow users to understand the consequences of input choices through direct experimentation and exploration, without the need for guidance from a facilitator to explain the consequences of environmental choices during an extensive and separate input decision phase. In QuestVis, exploration can take place in a tight loop that allows users to see the impact of a few policy changes per second, as opposed to the time scale of a few choices per hour in the previous paradigm.



Economy	20	21	22	23	24	25	26	29	30	27	28	31	32	33	34
Energy	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
Government	77	78	80	81	82	83	84	88	79	85	86	87			
Urban Growth	91	92	93	94	97	98	99	100	101	102	95	96			
Transportation	104	105	106	107	113	114	116	117	118	119	120	121	127	128	129
Air Quality	139	140	141	142	143	156	157	158	159	144	145	146	147	148	149
Solid Waste	[illegible]	166	167	168	169	170	[illegible]	173	176	[illegible]	174	175			
Footprint	177	178	179	180	181	182	183	184	185	186	187				
Cost Of Living	188	189	190	191	192	193	194	195	196	197	198	199	200	201	

Figure 3 Bidirectional linked highlighting between input choices and output indicators. **Left**: The input choice sliders that can affect an output indicator are highlighted on mouseover with blue backgrounds. **Right**: All output indicators that an input slider can affect are highlighted on mouseover by underlining them in red.

3.2 Multilevel Indicator Browsing

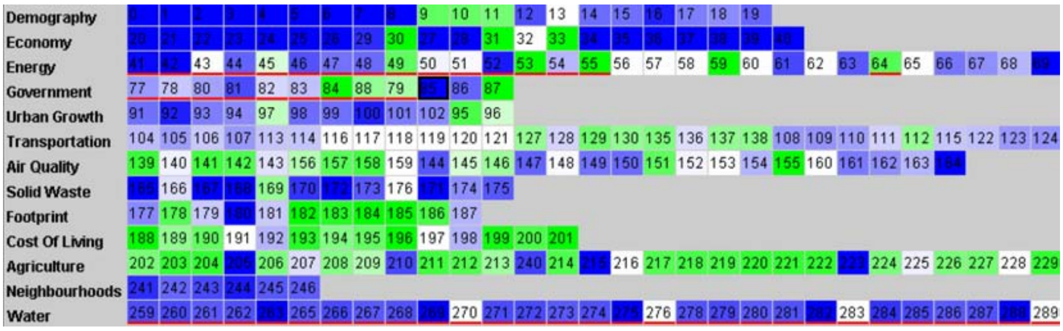
QuestVis provides indicator browsing in two ways. There is a detail view very similar to QUEST, with a dedicated window for showing a small number of indicators in a single bar chart, using spatial position to encode the indicator values. However, considerably less screen area is devoted to this window than in QUEST.

QuestVis also has a large overview region that shows information about indicators at three possible levels. The multilevel indicator browsing is supported using a compact visual encoding of colored boxes, as in the heatmap views that have become popular in bioinformatics [19], rather than the radar graph of Figure 1b that requires a lot of display area because the information is encoded using spatial position.

Each colored box showing an indicator value is several pixels on a side: large enough to avoid difficulties with small-field color perception, but much smaller than would be required to encode the data spatially. We use a diverging color scale, where white represents no change compared to the present-day value. Saturation represents the size of the increase or decrease from this value, with fully saturated blue for the maximum increase and fully saturated green for the maximum decrease. These values are normalized for each indicator relative to the its minimum and maximum values across all scenarios in the database.

Figure 4 shows a zoomed-in view of this overview region in the fully expanded state, when the full set of 294 output indicators for a scenario are shown simultaneously. The indicators are organized into thirteen categories, such as `Energy`, `Transportation`, `Water`, and `Air Quality`, and each is shown in a separate row. The fully expanded state allows easy comparison between all of the indicators for a scenario. Clicking on an indicator box in the fully expanded view changes the detailed bar chart shown in the upper right corner. The full overview can be collapsed to a mid-level view showing only the most important indicators for each category, or to a high-level view showing only one colored box for each category with an aggregate value that is the average of all indicators in the category, as shown in Figure 5. Clicking on the scenario name at the top of the column triggers the expanding or collapsing behavior. The organization of indicators into categories and the choice of which indicators to use for the mid-level overview was provided by the model creators.

The small footprint of the highest-level aggregate overview, requiring only a single column



Demography	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19									
Economy	20	21	22	23	24	25	26	29	30	27	28	31	32	33	34	35	36	37	38	39	40								
Energy	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
Government	77	78	80	81	82	83	84	88	79	85	86	87																	
Urban Growth	91	92	93	94	97	98	99	100	101	102	95	96																	
Transportation	104	105	106	107	113	114	116	117	118	119	120	121	127	128	129	130	135	136	137	138	108	109	110	111	112	115	122	123	124
Air Quality	139	140	141	142	143	156	157	158	159	144	145	146	147	148	149	150	151	152	153	154	155	160	161	162	163	164			
Solid Waste	165	166	167	168	169	170	172	173	176	171	174	175																	
Footprint	177	178	179	180	181	182	183	184	185	186	187																		
Cost Of Living	188	189	190	191	192	193	194	195	196	197	198	199	200	201															
Agriculture	202	203	204	205	206	207	208	209	210	211	212	213	240	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229
Neighbourhoods	241	242	243	244	245	246																							
Water	259	260	261	262	263	265	266	267	268	269	270	271	272	273	274	275	276	278	279	280	281	282	283	284	285	286	287	288	289

Figure 4 The fully expanded indicator overview shows all 294 dimensions simultaneously using color encoding rather than spatial position. In this scenario, the saturated blue for most of the `Demography` indicators show a dramatic population increase. A few `Cost of Living` indicators are white, showing no change from the present day, while the rest are green, showing a decrease. (The numbers presented within each cell identify the indicator number, for development purposes.)

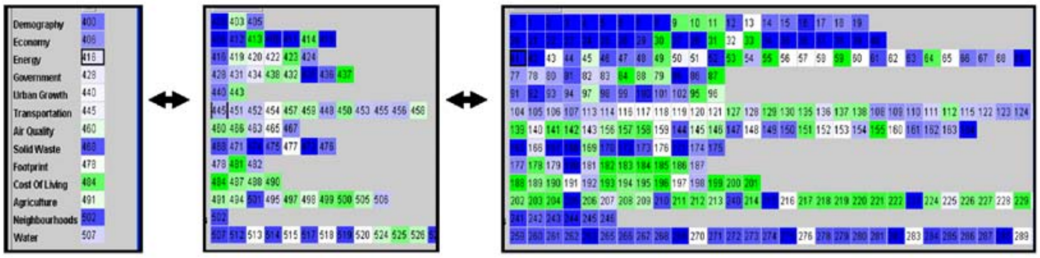


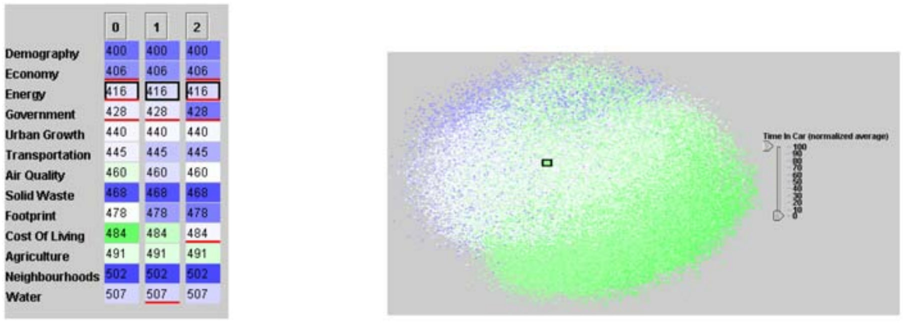
Figure 5 Multilevel indicator browsing. **Left:** The highest level shows aggregate values for each category. **Middle:** The middle level shows only the most important indicators within a category. **Right:** The fully expanded overview shows all 294 indicators simultaneously. The user can drill down to the lowest level by clicking on an indicator box to see detailed bar charts.

of boxes, enables side-by-side comparison between multiple scenarios. Figure 6a illustrates a comparison between three scenarios. We can see in a single glance that many categories are unchanged with the same colors across each row, including `Demography`, `Solid Waste`, `Neighbourhoods`, and `Water`; the leftmost scenario has lower `Cost of Living`, in green; and the rightmost scenario has more `Government`, in blue.

3.3 Dimensionality Reduction for Context

A QUEST scenario can be considered as a point in a high-dimensional space of 294 dimensions, one for each indicator, which we will call *scenario space*. Although scenario space has far too many dimensions to inspect directly, we wanted to show how a scenario fit into the context of other scenarios by using a dimensionality reduction approach to create a two-dimensional overview. Dimensionality reduction relies on the idea that some data sets measured in a high-dimensional space have a true structure that can be closely approximated in a space of much lower dimension.

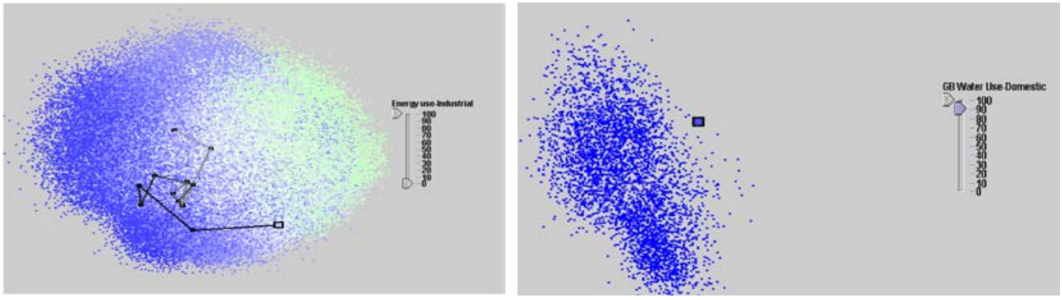
In QuestVis, we used multidimensional scaling to create a two-dimensional overview of scenario space, as shown in the upper middle panes of Figure 2 and in Figures 6b and 7. In multidimensional scaling, points are placed by minimizing the difference in distances between pairs of low-dimensional points versus their high-dimensional counterparts. This computation is time-consuming, so we pre-computed a single layout for scenario space offline



	0	1	2
Demography	400	400	400
Economy	406	406	406
Energy	416	416	416
Government	428	428	428
Urban Growth	440	440	440
Transportation	445	445	445
Air Quality	460	460	460
Solid Waste	468	468	468
Footprint	478	478	478
Cost Of Living	484	484	484
Agriculture	491	491	491
Neighbourhoods	502	502	502
Water	507	507	507

(a) **(b)**

Figure 6 **(a)** The highest-level indicator view has a small footprint, allowing side-by-side comparison between columns of multiple scenarios. It aggregates information for all 294 indicators, as opposed to the radar graph in Figure 1b that only shows 9 of the indicators. **(b)** Dimensionality reduction to show where a particular scenario, marked by the rectangle, fits within the context of scenario space. Here, the points are colored by the value of the `Time in Car` indicator.



(a) **(b)**

Figure 7 Scenario space overviews. **(a)** Colored by the `Industrial Energy Use` indicator, with trail showing the history of previously selected scenarios. **(b)** Colored by the `Water Use` indicator, and filtered to only show scenarios with the greatest increase for this indicator.

using an existing program, and saved the 2D location values for each scenario point in a database. We used Hive [14], which could compute the layout for the nearly 120,000 points in 294-dimensional space in under three hours.

Each scenario point is always shown in the same spatial location to create a stable spatial substrate, and the user can interactively change select which one of the 294 indicator values should be used for color coding the points. Figures 6b and 7a show close-up views of the scenario space overview pane, where the `Time in Car` indicator was chosen to color the former and `Industrial Energy Use` was selected for the latter. Figure 7a also shows the visual history mechanism, a trail marking the twelve most recently selected scenarios in the interactive session.

The layout of points in the overview pane is dense enough that many points may be overlapped in the same screen pixel. The overview window has a slider that filters the number of points shown based on the value of the currently selected indicator that is used to color the points. Figure 7b shows an overview colored by `Domestic Water Use`, filtered to show only scenarios with the highest values for this indicator.

4 Reflections on the First Prototype

We obtained feedback on the first interactive prototype from the expert modellers and workshop facilitators at SDRI and Envision.

4.1 Immediate Response

Providing an immediate response to input changes at any time allowed exploratory "What if?" queries and exploration of many different future scenarios. The architectural change of using a precomputed database was considered highly successful. This idea was immediately incorporated by Envision into their main product line. Their subsequent MetroQuest system was built on a precomputed database.

4.2 Dimensionality Reduction

The use of dimensionality reduction was deemed to cause far more confusion than insight. The scenario space overview was an incomprehensible cloud of dots. The failure came from two quite distinct reasons: a mismatch with the characteristics of the target audience and task, and the mathematical properties of this particular data set.

First, the idea of a low dimensional embedding of a high dimensional space required far too much time to communicate to a nontechnical user during a brief session. The target audience does not necessarily have a scientific or mathematical background and were only intending to use our software for a short time, from a few minutes to a few hours. The amount of time available for explaining this aspect of the software was thus only a few minutes at best. Following an explanation on the meaning of the overview from a facilitator required multiple conceptual leaps on the part of the audience: defining an indicator of future life as an abstract dimension, considering a high-dimensional space beyond the familiar three dimensions of direct perception, and embedding that high-dimensional space to a two-dimensional plane. Moreover, one of the project goals was to allow the software to be used without a facilitator. The experts considered it highly unlikely that the audience would understand the meaning of the scenario space overview through interactive exploration with the software alone, without explanation from a facilitator.

The past successes of dimensionality reduction in visualization have been in very different circumstances: target audiences of scientists, who would be doing analysis for extended periods of time. This project began before the publication of an intriguing definition of *casual* information visualization in opposition to expert use for deep analysis tasks [13]. Using that vocabulary, our usage scenario is a better match with casual infovis than traditional infovis.

Second, the amount of information lost in the reduction from 294 to two dimensions was so great that very little information could be reliably inferred from the scenario space overview, even for an observer who completely understands the mathematical ideas behind the visual representation. The intent of the display was that similar scenarios would be clustered together, with dissimilar scenarios far apart. Dimensionality reduction is typically useful when points fall into spatially distinct clusters in the low-dimensional embedding. However, in the scenario space overview, the points were fairly evenly distributed in an oval region. A scenario's location in 2D space was very difficult to predict, and the trail of explored scenarios was close to a random walk. We concluded from these results that the true dimensionality of this data set was considerably higher than two. Later analysis using MATLAB showed that five dimensions were required to characterize most of the variance in the data set, and 18 dimensions were enough to characterize nearly all of it. Several techniques, including using a matrix of scatterplots rather than a single scatterplot, have been proposed for analyzing dataset of higher true dimensionality than two. However, all these would require even more explanation, and so would also be inappropriate for a casual use scenario. We thus decided to abandon a dimensionality reduction approach.

4.3 Multilevel Indicator Browsing

The multilevel indicator browsing had mixed reviews. While the ability to compare indicators side by side between multiple future scenarios seemed promising, the representation was so compact that comprehensibility suffered. The labels for output indicators were tooltips available on mouseover for an individual cell. Examining indicators one a time slowed down detailed exploration and provided little inspiration for further examination. Although the relationship between the most compact high-level view and the fully expanded view was clear, the mid-level view was confusing because the meaning of the aggregation was unclear.

4.4 Aggregation and Normalization

Averaging multiple low-level cells into a single aggregate higher-level cell often did not reflect a valid summarization of a category. We had carried out a first level of normalization separately for each indicator, in hopes of allowing sensible aggregation despite differences in the original scales or units of the indicators. The normalized scale ranged from the maximum possible decrease between the present and future to the maximum possible increase.

However, this method of aggregation treats all indicators neutrally. The aggregate value shows whether the contained items increased or decreased on average. After using the interactive prototype, our collaborators noted that many indicators had conflicting values in the eyes of any particular person, with some positive and some negative. Aggregation that did not reflect those personal values was not very useful.

For example, the `Cost of Living` category shown in Figure 8a contains the indicators `Average Wages` and `Tax`. When average wages increased and tax decreased, the aggregate cost of living indicator showed a neutral value because the signs of the indicators are opposite. Intuitively, most users would consider both of these changes an improvement, and expect that the scenario should show improvement in the aggregate value.

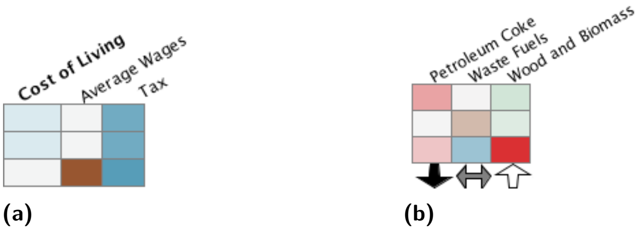


Figure 8 Aggregation. **(a)** In the first prototype, unintuitive displays resulted from averaging across indicators that are ascribed opposite values by users. **(b)** In the second prototype, clickable arrows beneath indicators allow users to indicate deteriorating, neutral, and improving valences for each value.

4.5 Aesthetics

The look and feel of the first prototype was extremely primitive compared to the polished interface of QUEST. Although research prototypes are typically held to a lower standard of graphic design than commercial software, interfaces designed for the general public do need a certain level of friendliness and visual appeal. We deemed it important to pay closer attention to aesthetics in the next attempt.

5 The Second Prototype

The first prototype was designed to dramatically increase the density of information simultaneously visible to the user, but the abstraction was so extreme that the tool was deemed difficult to engage with by casual users. For the second prototype, we wanted to keep the interactive and exploratory nature of the first prototype, while addressing its limitations. We completely eliminated the scenario space overview. For multilevel browsing, we kept the high-level view and eliminated the mid-level view. We changed the low-level view to show one category at a time, with visible labels, and be visible simultaneously with the high-level view to support examining details of a scenario while still comparing the summary information with other scenarios. Finally, we improved the overall look and feel to provide a friendlier interface suitable for a casual user.

Figures 9, 10, and 11 show the second prototype when exploring one, two, and five scenarios. At the top, the axes allow the user to change input choices, and trails showing the selections for all active scenarios are simultaneously visible. The middle layer shows high-level summarized scenario views side by side for easy comparison. The bottom layer shows drilled-down views of a chosen indicator category in detail, again for every active scenario.

5.1 Scenario Generation

The input region at the top of the window acts both as a control of and a display for multiple active scenarios. A query is formed by choosing one option for each input choice. The radio button design of the previous prototype could show only one active scenario, with an external mechanism to show history only as trails in the abstract scenario space. In order to establish a stronger link between input choices and future scenarios, this prototype was designed to display the choices for multiple scenarios at once, providing a rich history. The design was motivated by parallel coordinates, which are normally used to represent points in continuous multidimensional space [7, 18]. Unlike Cartesian coordinates, where each new axis is placed orthogonally to all other axes, each axis is placed parallel to all other axes. By discretizing our axes into regions representing mutually exclusive input choices, we lose many of the consistent mathematical properties of a parallel coordinate space. However, if we consider a query to be a point in multidimensional space, our discretized parallel coordinates retain their ability to visualize many high dimensional points at once.

The interface design was inspired by the work of Tory *et al.* [16]. Each axis consists of active buttons placed side by side horizontally. All of the choices for an individual scenario are linked by a colored line that zigzags through the input axes, and a colored dot marks each of the choices. When creating a new scenario the dots appear after clicking on a button, or the user can modify an existing scenario by dragging the dot to a new location. The scenario line continues past the bottom axis to a block marked in the same color at the top of the high-level output indicator summary for that scenario, in the middle of the window. The

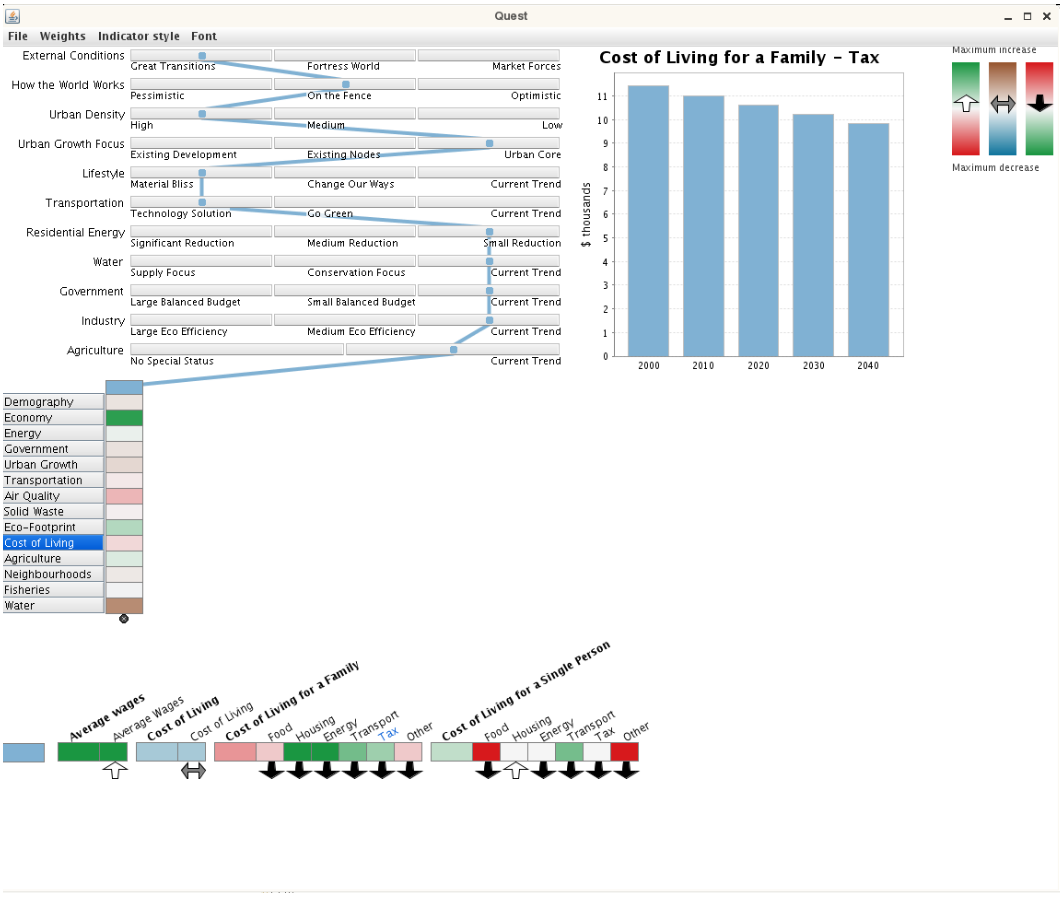


Figure 9 Browsing one scenario in the second prototype.

combination of the joining line and the common color creates a strong perceptual linkage between inputs and outputs.

Our use of color is informed by the human perceptual limitations of the small number of distinguishable colors that can be used to represent nominal values in small spatially separate regions. The prototype has a carefully chosen palette of eleven unique colors. If more than eleven scenarios are active at once, the colors are reused in the same order. Confusion is unlikely because the lines connecting the dots creates a strong perceptual grouping, and the dots are ordered horizontally on the buttons so there will always be ten other dots separating a repeated color.

5.2 Two-Level Indicator Browsing

In the middle of the window, each scenario is summarized in a column showing an aggregate view of the fourteen high-level categories. Each row has a label on the far left, and a colored box representing the aggregate values for the category in each scenario. Summary columns can be rearranged with drag and drop, with the scenario lines from above automatically following to the new positions, so that any two scenarios can be placed side by side for easy pairwise comparison.

Clicking on a labelled category button highlights it in blue, and changes the expanded view at the bottom of the window to show that category. In the expanded view, all of the

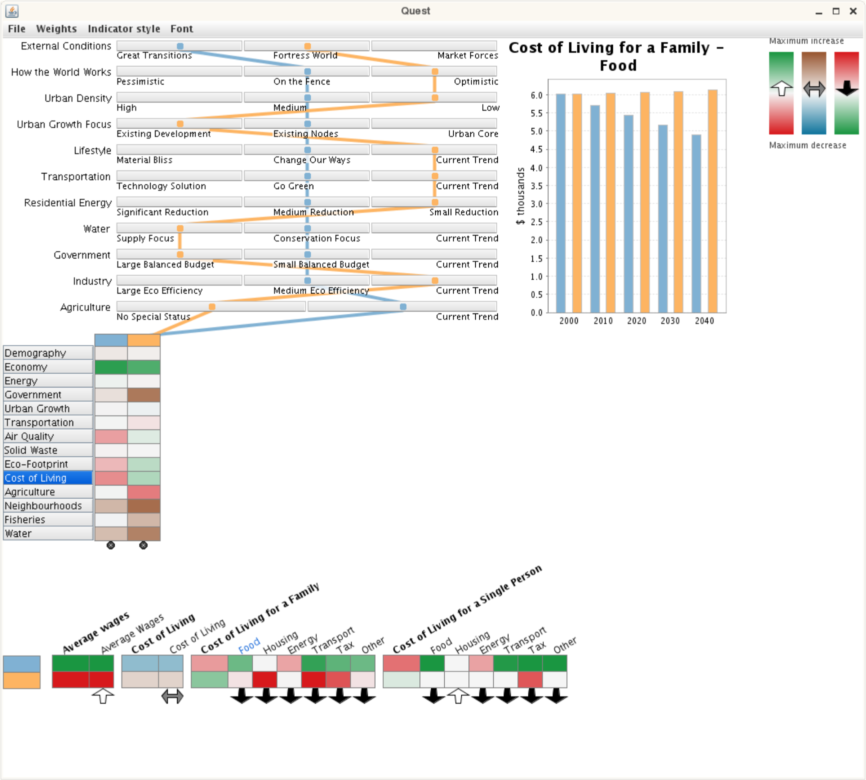


Figure 10 Comparing between two scenarios in the second prototype.

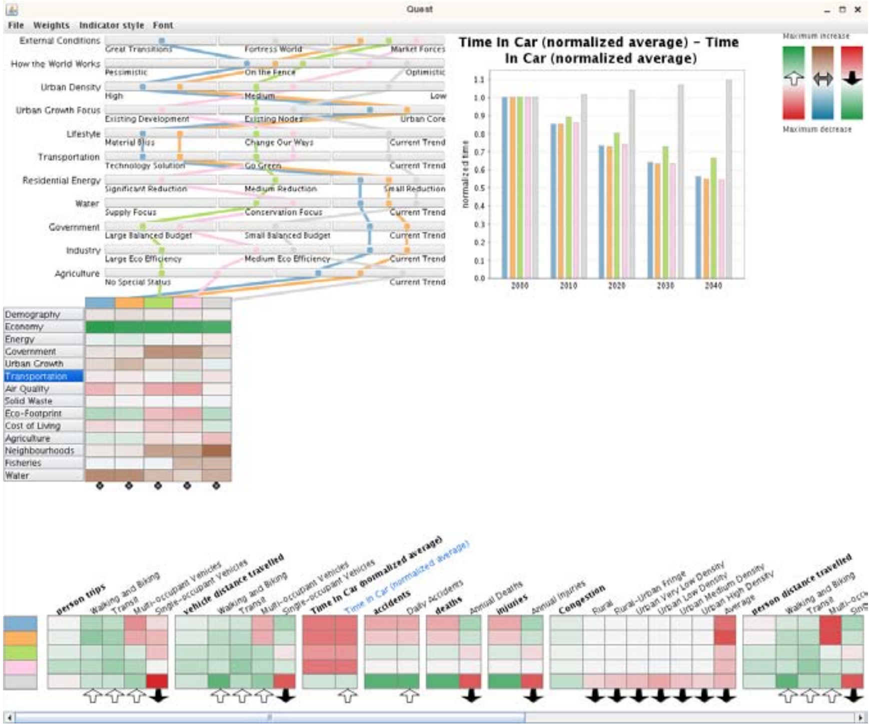


Figure 11 Comparing between five scenarios in the second prototype.

indicators for the chosen category are shown as colored boxes, with a row for each active scenario. These rows are visually linked with the scenario lines above using a header box of the same color on the far left. The vertical order of the scenario rows in the detail view is the same as the horizontal order of the draggable scenario columns in the summary view, to maintain context.

In the expanded view, a column shows the same indicator across multiple scenarios, and a text label for the indicator appears at the top. In order to have densely packed yet readable text, the labels are drawn on a diagonal rotated 30° from the horizontal. With this layout, it is easy to scan the labels for measures of interest and then make comparisons down the columns. We chose to arrange the scenarios in rows for the detail view to take advantage of the aspect ratio of most monitors, assuming higher resolution horizontally than vertically.

Clicking on the colored box for any indicator in the expanded view shows its full details in a bar chart in the top right corner, with color-coded bars for each active scenario.

5.3 Aggregation

As discussed above, for some indicators an increase is an improvement, while for others a decrease is an improvement. A technically straightforward solution would be simply record a choice for the polarity of each indicator in the master database. For some indicators, such as automobile deaths per year, it is trivial to assign the meaning that increased traffic deaths are bad. However, for many indicators, assigning a good or bad value is controversial. Is increased nuclear power good or bad? What about high-density housing, social services spending, or bus trips to work? The goal of the QUEST project is to promote discussions about values amongst the users of the tool, rather than advocating the specific answers dictated by the tool creators. Furthermore, there are several indicators, such as population age demographics, that can be considered neutral.

We therefore allow the users to interactively make and change personal value judgments of positive, negative, and neutral for each indicator in the tool interface. In the bottom detail pane, each indicator has a simple arrow below it to indicate whether the indicator is currently considered good, bad, or neutral. A simple mouse click will change the mapping, cycling through the three choices of the white upward good arrow, the black downward bad arrow, or the grey double-sided horizontal neutral arrow. As soon as an indicator mapping is changed, there is an immediate update of the color scale used and any upstream aggregations involving that value. Figure 8b shows a close-up example of indicators and their arrows.

The second prototype uses one of three color scales, depending on the personal value mapping for the indicator. We use a culturally familiar red-green divergent scale for the positive indicator, and the same color range inverted for the negative indicator, so that good indicators are green and bad are red. We use a visually distinguishable blue-brown divergent scale for neutral indicators. A legend is visible at all times in the upper right corner of the window to help users remember the meaning of the color coding. When aggregating a mix of both neutral and good/bad indicators, we use the majority to choose which kind of arrow to show.

The entire set of current values can be saved for later reloading. Thus, users do not have to explicitly make choices for values before they start using the program. Using a pre-loaded set of values does not prevent the user from customizing individual values while exploring future scenarios. A starting set of values could be provided by a facilitator in a workshop setting, or be automatically generated from the results of a separate values quiz taken by a user before using the prototype.

6 Reflections on the Second Prototype

We again obtained feedback from modellers and facilitators at SDRI and Envision on the second prototype.

6.1 Successes

QuestVis2 was considered to be much more understandable than the first prototype. The combination of the scenario generation display and the two-level overviews supported both browsing through the indicators for a single scenario and comparing between multiple scenarios. Allowing users to express their own values in order to see meaningful aggregation in the overviews was considered to be a notable success. The look and feel were much cleaner in the second prototype.

6.2 Scalability

The second prototype works well for browsing through a single scenario and comparing a limited number of scenarios. When too many scenarios are active at once, the scenario lines are hard to follow and scrolling is required to browse the detailed category outputs. Figure 12 shows that viewing 12 scenarios simultaneously nears the limit of manageability, and Figure 13 shows that viewing 20 scenarios once is clearly unwieldy.

The display has a lot of unused screen space when browsing a single scenario or a small number of scenarios. If we use a target of 10 or fewer scenarios, it would be straightforward to improve the information density by simply rearranging the major components within the window and redistributing the space allocated to them: the scenario space generator, the summary columns, the category view, the detail view, and the color legend.

6.3 Problem Mismatch

Although the design goals for the second prototype as we understood them were met, the QUEST designers were still uneasy about deploying the tool. After further discussion, we came to understand that there was a fundamental mismatch between our view of the problem and theirs. We had approached the problem with the intention of creating software that would help its users develop a mental model of the complex relationships between the input policy choices and the output indicators. However, the intuition of the facilitators after using the second prototype was that it exposed far *too much* of the complexity of the model to users. Their practice in leading groups through the QUEST software was to deliberately limit the complexity by focusing on just a few key choices and indicators. We had thought that this limitation was part of the problem that our software should help solve, and designed the prototype to help users explore the rich information space of the high-dimensional data. It was only after the post-QuestVis2 discussions that we realized that the designers of QUEST did *not* want users to focus too much on the internal details of the model. The high-level goal of the QUEST project is to promote environmental sustainability and behavior change via community engagement in environmental policy decisions. Teaching the general public about the information space underlying their model is thus a sidetrack and a diversion from this main goal.

The idea that exploring an information space is a central user goal is a typical bias of an information visualization point of view. In this case, it was not the right match with the problem at hand. After we understood this, we halted the project and did not deploy or further refine the QuestVis prototype.

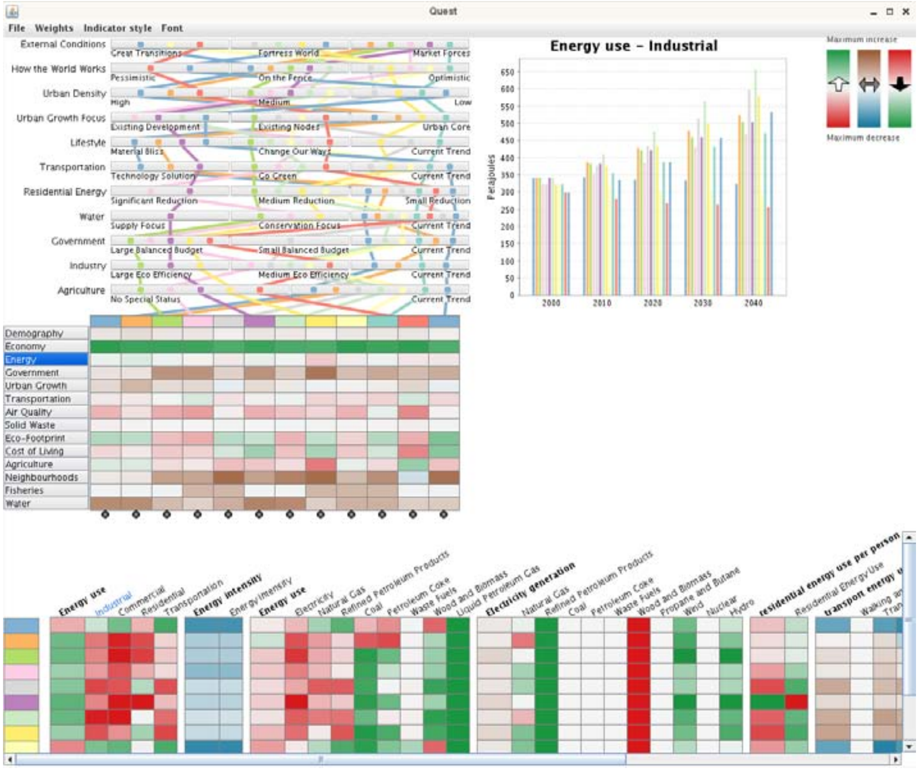


Figure 12 At 12 scenarios, the second prototype nears the limit of manageability.

6.4 Visualization Design Process

We recently introduced a model of the visualization design process with four nested stages: domain problem characterization, data and operation abstraction design, visual encoding and interaction design, and algorithm design [11]. Some aspects of that model were inspired by reflection on the evolution of this project.

Using the language of that model, we see in retrospect that this project ran aground because of a misunderstanding at the first stage of domain problem characterization. The cascading implications of that misunderstanding are that we were unable to choose the right abstraction at the next stage, and thus that refinement at the third stage of visual encoding could never fix these fundamental upstream flaws. For the first prototype, we chose dimensionality reduction as a visual encoding technique. For the second prototype, we backed away from dimensionality reduction but still framed the underlying abstraction as exploring a high-dimensional information space. The visual encoding we designed did successfully support that abstraction, but that abstraction did not address a valid user need.

7 Related Work

The first QuestVis prototype is discussed in more detail by Williams in his thesis [20]. The two prototypes make use of several information visualization techniques previously introduced in the literature. The utility of linked views has long been established [2], and previous systems in many application domains have been built around this approach [1, 6, 9, 10, 15].

The multilevel indicator browser was inspired in part by previous interfaces that use color as a compact way to represent data, such as the heatmaps currently popular in the

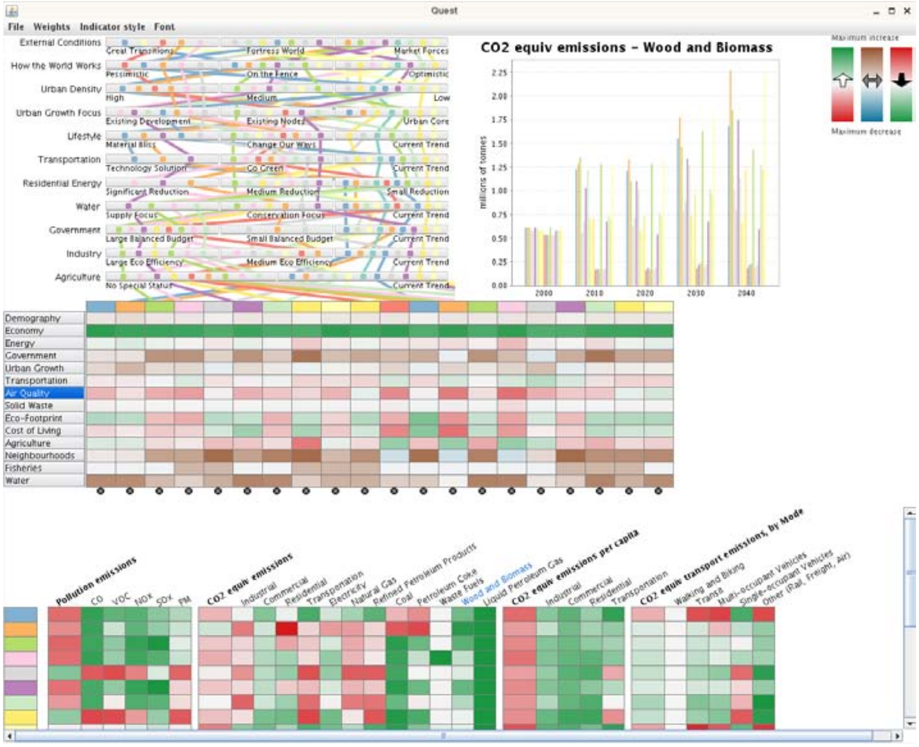


Figure 13 With 20 scenarios, the second prototype is clearly unwieldy.

bioinformatics literature [19]. The perceptual and cognitive tradeoffs between a compact representation using color and a more traditional representation using spatial positions have since been studied, both in the context of this particular application [5] and in more general settings [8].

We did not break new algorithmic ground in dimensionality reduction techniques, since we use an existing algorithm for layout [14]. However, our experience may shed some light on the question of when dimensionality reduction methods are appropriate, a topic that is still largely unexplored in the previous literature.

8 Conclusion

We have discussed the lessons learned in building a visualization tool intended for exploration of a dataset generated by an environmental sustainability model. The use of linked views and aggregation to create a compact overview were appropriate techniques for visual encoding and interaction given the design goals of the project. However, those goals were based on a mischaracterization of the problem at hand: providing the target users with the analysis tools for a high-dimensional dataset was in fact counterproductive in the larger context of the sustainability project. This lesson was a motivating factor in the creation of a new nested model for the design and validation of visualization system.

Acknowledgements

We are very grateful to the many people involved with QUEST for the time and energy they devoted to this project, especially John Robinson of SDRI, Mike Walsh and Dave Biggs of

Envision, and Jeff Carmichael of Envision and SDRI. This work was supported by grants from the NSERC-funded GEOIDE NCE: “CIRCUITS: Collaborative Interdisciplinary Research on Communities Using Information Technology for Sustainability”, and “GeoCognito: Connecting People with Ideas and Ideas with Place”.

References

1 Aaron Barsky, Tamara Munzner, Jennifer Gardy, and Robert Kincaid. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 08)*, 14(6):1253–1260, 2008.

2 Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

3 Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The information visualizer, an information workspace. In *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI 91)*, pages 181–186, 1991.

4 Jeff Carmichael, James Tansey, and John Robinson. Georgia Basin Quest: an integrated assessment modelling tool. *Global Environmental Change*, 14(2):171–183, 2004.

5 Cristina Conati and Heather Maclaren. Exploring the role of individual differences in information visualization. In *Proc. Advanced Visual Interfaces (AVI)*, pages 199–206, 2008.

6 Paul Craig, Jessie Kennedy, and Andrew Cumming. Animated interval scatter-plot views for the exploratory analysis of large-scale microarray time-course data. *Information Visualization*, 4(3):149–163, 2005.

7 Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: A tool for visualizing multidimensional geometry. In *Proc. IEEE Visualization (Vis 90)*, 1990.

8 Heidi Lam, Robert Kincaid, and Tamara Munzner. Overview use in multiple visual information resolution interfaces. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 07)*, 13(6):1278–1285, 2007.

9 Heidi Lam, Daniel Russell, Diane Tang, and Tamara Munzner. Session Viewer: Visual exploratory analysis of web session logs. In *Proc. IEEE Symp. Visual Analytics Science and Technology (VAST)*, pages 147–154, 2007.

10 Alan MacEachren, Xiping Dai, Frank Hardisty, Diansheng Guo, and Gene Lengerich. Exploring high-D spaces with multiform matrices and small multiples. In *Proc. IEEE Symp. Information Visualization (InfoVis 03)*, pages 31–38, 2003.

11 Tamara Munzner. A nested process model for visualization design and validation. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis 09)*, 15(6):921–928, 2009.

12 M. Plumlee and C. Ware. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *Proc. ACM Trans. on Computer-Human Interaction (ToCHI)*, 13(2):179–209, 2006.

13 Zachary Pousman, John T. Stasko, and Michael Mateas. Casual information visualization: Depictions of data in everyday life. *IEEE Trans. Visualization and Computer Graphics (TVCG) (Proc. InfoVis 07)*, 13(6):1145–1152, 2007.

14 Greg Ross and Matthew Chalmers. A visual workspace for hybrid multidimensional scaling algorithms. In *Proc. IEEE Symp. Information Visualization (InfoVis)*, pages 91–96, 2003.

15 Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4(2):96–113, 2005.

16 Melanie Tory, Simeon Potts, and Torsten Möller. A parallel coordinates style interface for exploratory volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11:71–80, Jan./Feb. 2005.

17 Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262, 2002.

18 Edward J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, September 1990.

19 Leland Wilkinson and Michael Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, 2009.

20 Matt Williams. QuestVis and MDSteer: The visualization of high-dimensional environmental sustainability data. Master's thesis, University of British Columbia, Department of Computer Science, July 2004.

Generation of Adaptive Streak Surfaces Using Moving Least Squares

Harald Obermaier[1,2], Martin Hering-Bertram[2], Jörg Kuhnert[2], and Hans Hagen[1]

1 University of Kaiserslautern, Germany
2 Fraunhofer ITWM Kaiserslautern, Germany

Abstract

We introduce a novel method for the generation of fully adaptive streak surfaces in time-varying flow fields based on particle advection and adaptive mesh refinement. Moving least squares approximation plays an important role in multiple stages of the proposed algorithm, which adaptively refines the surface based on curvature approximation and circumradius properties of the underlying Delaunay mesh. We utilize the grid-less Moving Least Squares approximation method for both curvature and surface estimation as well as vector field evaluation during particle advection. Delaunay properties of the surface triangulation are guaranteed by edge flipping operations on the progressive surface mesh. The results of this work illustrate the benefit of adaptivity techniques to streak surface generation and provide the means for a qualitative analysis of the presented approach.

1998 ACM Subject Classification I.6.6 Simulation Output Analysis

Keywords and phrases scattered, flow, streak surface, adaptivity, moving least squares

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.260

1 Introduction

Providing data for the evaluation of aerospace prototypes, industrial mixing processes, and many other applications of flowing liquids or gases is the result of more and more accurate simulations in *Computational Fluid Dynamics* (CFD). Analysis of the vector fields of these flow simulations is heavily dependent on the range of available visualization methods, as the information gathered from standard methods like direct volume rendering is limited and of neglectable expressive power when applied globally. This emphasizes the need for sophisticated feature-based visualization techniques, which has been the topic of active research for a number of years, leading to the definition of integral curves and surfaces. Generation and rendering of these surfaces in simulated flow fields is a well-established technique in the field of vector field visualization, whose homogeneous visual properties allow an in-depth analysis of the behavior of connected components of flow fields. While efficient methods to adaptively generate such surfaces in stationary vector fields and some generalizations like adaptive time surfaces are state-of-the-art, there are no known methods for fully adaptive streak surface generation as presented in this work. An important application of surface feature extraction is the visualization of separation topology in three-dimensional data sets what, due to the absence of adaptive streak surfaces, has been limited to the stationary vector field case so far. In comparison to surface definitions where merely the trace of a specific single particle or curve is tracked, as is the case in path surfaces, the definition of streak surfaces is capable of visualizing phenomena such as smoke and dye-advection, efficiently showing the movements of distinct, continuously seeded regions over time. In

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 260–275
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

practice, they may be used to visualize material boundaries in mixing processes or, more general, a time-varying analogon of the stationary three-dimensional separatrix definition, leading to a non-stationary form of three-dimensional vector field topology. Therefore, streak surfaces form the basis for topological volume segmentation in time-dependent vector fields. The major challenge of adaptive streak surface construction is the high complexity of mesh refinement arising from the new time dimension of the well-known advancing front definition introduced by Hultquist [11]. In every time step of a time-varying velocity field, not only a front of a few stream lines or particle traces needs to be examined, but the whole surface has to be analyzed with respect to adaptivity measures. In this work, we solve the problem of adaptive streak surface integration with the help of *Moving Least Squares* (MLS) approximation for particle advection as well as surface estimation and refinement based on Delaunay meshes. MLS facilitates accurate surface integration independent of any underlying computational meshes that may be created by the specific tool used for CFD simulation. In fact, the presented test data sets are mesh-less and obtained from a grid-less *Finite Pointset Method* (FPM) [18], whose interpolant is also based on MLS. Delaunay meshing helps to approximate surface particle densities and provides a basic triangulation for visualization. The challenge of time coherent surface generation and rendering is overcome by surface particle integration, adaptive particle tracing, and look back methods.

The work presented in this paper has the following main contributions to the community of vector field visualization:

- Fully adaptive grid-less streak surface generation
- Introduction of MLS into the generation of time-varying integral surfaces
- Integral surface refinement by time-dependent Delaunay based particle density adaptation

In section 2 we give an overview of the related work that has been published in the field of surface generation and Delaunay meshing. We present the concept of streak surfaces in section 3. The mathematical definition of the Moving Least Squares approximation method is given in section 4. Section 5 describes the steps of our algorithm for adaptive streak surface generation in detail. Numerical examples are provided in section 6. Section 7 concludes this paper and gives an outlook on future work.

2 Related Work

Adaptive (stream-) surface generation in stationary flow fields was introduced by the work of Hultquist [11] in 1992, whose well-known advancing-front concept has led, among other things, to the development of sophisticated methods for path surface integration in time-varying flow fields [9, 17]. The work by Krüger et al. [13] represents the most direct way of streak generation and visualization, namely by the use of large particle systems being influenced by the surrounding flow field. The absence of a triangulated mesh does however limit these point sets to discontinuous representations. Operations such as surface intersections are not possible without further effort. Funck et al. [8], Cuntz et al. [6], and Weiskopf et al. [19] introduced work on smoke surfaces and particle level set advection, focusing on the visualization of non-adaptive streak surface like structures. The first of these papers triangulates the particle system for visualization purposes and is therefore closer to our method, as far as visualization techniques are concerned.

Examples for MLS based surface approximation in the context of surface mesh reconstruction and refinement from point clouds are given by the work of Alexa et al. [1] and Mederos et al. [14].

Delaunay type mesh refinement of static point sets has been the topic of multiple papers such

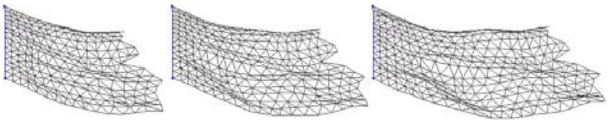


Figure 1 Triangulations of a non-adaptive streak surface in three consecutive time steps.

as the work of Chew et al. [5] and Chen et al. [4]. Both approaches use incremental mesh construction by either building a constrained Delaunay triangulation or by transforming common Delaunay algorithms to a new parametric space. Contrary to this work, the nature of our problem allows us to make use of an existing time-varying triangulation, thus eliminating the need for complete mesh reconstruction and facilitating the incorporation of multiple Delaunay triangulations into the particle insertion process, as discussed in section 5.4.

3 Streak Surface Definition

A streak surface is the locus of a set of (connected) particles that are advected by a time-dependent flow field $f : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R}^3$. The integral surface (1) defines a streak surface S at time t with particles emerging from points $c(s)$ at an univariate seeding curve $c : [0,1] \rightarrow \mathbb{R}^3$. Individual instances of particles are identified by their age parameter $r \in [0,t]$.

$$S(r,s,t) = c(s) + \int_{t-r}^{t} f(S(r-(t-x),s,x),x)dx\,. \tag{1}$$

In contrast to stream surfaces, streak surfaces are generally no longer tangential to the flow field and need to be updated or refined at their whole range during integration. Due to the additional parameter r, streak surfaces describe a truly three-dimensional complex in time and space, whereas stream surfaces are only two-dimensional in space. This increase of complexity prevents the use of classic approaches to adaptivity such as the concept introduced by Hultquist.

Figure 1 illustrates a simple example of (1) with a sequence of triangulations of three consecutive time steps of a streak surface. At the curve c shown in blue, 15 particles are seeded at equidistant positions. The problem exhibited by non-adaptive approaches can be identified as the far too uniform particle distribution, preventing the accurate and smooth representation of folds. One has to note, that in the discrete case a single streak surface consists of a number of consecutive static surfaces obtained in different time steps of the surface. Depending of the time resolution of the data set, coherency between consecutive surfaces might be low, producing a rough, jagged animation during rendering of the streak surface.

4 Moving Least Squares Approximation

Least Squares fitting is a common approach to data approximation, providing a method to construct functions that minimize the squared distance to a given set of data points (x_i, f_i). If a local approximation of the data is desired, the classic Least Squares scheme may be generalized to the Weighted Least Squares method by the introduction of a weighting function ω, see [16]. A polynomial function f with given degree at a point of evaluation x in a Weighted Least Squares sense is defined by (2).

$$\sum_i \omega(x,x_i)||f(x_i) - f_i||^2 \rightarrow \min\,. \tag{2}$$

The scheme obtained from moving the Weighted Least Squares over the domain of the data set to yield a continuous approximation of the field, is called *Moving Least Squares* approximation. Solving (2) for the coefficient vector a_x of a polynomial $f(x) = a_x^T \cdot b(x)$ with a given base vector $b(x)$ leads to a linear system of equations (LSE). For a linear, two-dimensional base vector $b(x) = (1 \text{ x } \text{y})^T$ it takes the following form:

$$\left(\sum_i \omega(x, x_i) \begin{pmatrix} 1 & \text{x} & \text{y} \\ \text{x} & \text{x}^2 & \text{xy} \\ \text{y} & \text{xy} & \text{y}^2 \end{pmatrix} \right) \cdot a_x = \sum_i \omega(x, x_i) \begin{pmatrix} 1 \\ \text{x} \\ \text{y} \end{pmatrix} \text{f}_i \, . \tag{3}$$

Computational complexity of this LSE is drastically increased by degree or dimension elevation of domain or range of the data set. So does quadratic three-dimensional approximation already require solving a 10×10 system once per dimension of f_i.

Accuracy and level of detail in the reconstructions created by MLS are strongly dependent on the properties of the weighting function used. Common exponential weighting functions of the general form

$$\omega(x, x_i) = a \cdot e^{\frac{||x - x_i||^2}{r^2}} + b$$

for example, yield different results for a varying *smoothing length* r. Hereby, an increased smoothing length leads to less detailed but smoother reconstructions. The appropriate choice of r is detailed in the following sections whenever we make use of MLS.

5 Surface Generation

5.1 Algorithm Outline

Our algorithm to generate adaptive streak surfaces consists of five basic stages that are repeated for every time step of surface integration:

1. Generate new particles at the rake and insert them into the existing mesh
2. Concurrently advect all particles of the surface to their new positions
3. Restore the Delaunay property of the surface mesh by edge flipping
4. Determine curvature of the surface at every surface particle by MLS approximation
5. Adapt resolution by insertion and advection of new particles in current and previous time steps

These steps are described in detail in the following.

5.2 Particle Birth

Let $t \in [t_0, t_1]$ be the current time step. The basic representation of a streak surface usually consists of a set of particles that are advected through the flow field, being seeded at a predefined curve known as *rake*. Based on a user-defined resolution, we release a set of particles at equidistant positions along the rake. While the given resolution does directly influence the number of seed positions at the rake, the magnitude of the velocity field indirectly governs the number of particle rows that are generated and advected at the rake. If n particles with a seed distance of d are seeded at the rake, where the distance traveled in one time step at an arbitrary position on the rake is $l := ||f(.)|| \cdot \Delta t$, we release $m = \frac{l}{d}$ particles at every seeding position, leading to a total of $n \times m$ new particles.

Let T be a given triangulation in time step t of the particles released in $[t_0, t)$. We link the $n \times m$ new particles based on their neighborhood in parameter space (s, t) and connect this

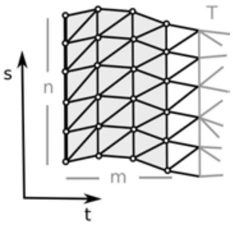


Figure 2 The triangulation of the recently seeded particles is linked to the old mesh. Note that the old mesh is still located at its position in time step $i-1$ until corresponding particles are advected.

new triangulation to the first row of particles in T, as shown in Figure 2.
Consequently we obtain a fully connected particle-based streak surface, whose triangular mesh is used for surface approximation, refinement and visualization of the surface, as explained in the following sections. As a result, particles on a streak surface do not only carry spatial information, but provide data about the (s,t) parametrization of the surface, that can be used for the generation of texture coordinates, as well as normal and neighborhood information. It is important to point out, that the basic definition, computation and visualization of a streak surface does not require the availability of connectivity information between particles. As described in the next sections, the triangulation created during particle birth is an auxiliary construct to reduce the computational effort for surface approximation and particle density calculations. While it only represents a linear approximation of the true streak surface, it can additionally be used for basic surface visualization.

5.3 Particle Advection

Particle data of time step $t-1$ has to be propagated to the new time step t and particles need to be advected to their respective new positions in t according to the appropriate velocity values obtained from vector field evaluation. To compensate possible artifacts in poorly time resolved data sets or data that has a high curvature in time, we take special care during vector field approximation. While adaptive Runge-Kutta approximation schemes can determine the step size or order of integration k used during particle advection, the necessary interpolation between adjacent time steps that needs to be performed, if $k>1$ often introduces artifacts if the degree of interpolation is too low, as in (4). Higher order integration such as cubic Hermite interpolation, taking into account $f(p,t-2)$, $f(p,t-1)$, $f(p,t)$, and $f(p,t+1)$ for particle advection from $t-1$ to t double the number of required field evaluations but tend to yield more accurate surfaces.

$$f(p,t-1+\frac{j}{k}) = \left(1-\frac{j}{k}\right) f(p,t-1) + \frac{j}{k} f(p,t) \tag{4}$$

where $f(p,t-1+\frac{j}{k})$ is the velocity of a particle with position p and integration order k during the j-th step of advection from $t-1$ to t. This scheme subdivides the time interval $[t-1,t]$ into k time intervals with linearly interpolated velocity fields. To reach a sufficient accuracy, we determine k for every particle individually by comparison of angular deviation between velocity vectors resulting from consecutive vector field evaluations. This particle advection scheme reduces path deviations in data sets with large time steps and improves visual coherence. A mapping of the order of integration k onto a streak surface is shown in Figure 3. As individual particles from surfaces of consecutive time steps are matched in our

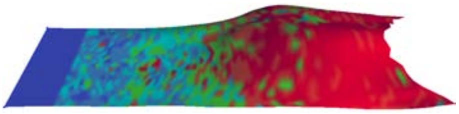


Figure 3 A simple streak surface is color-mapped with the integration order $k \in [1, 16]$. Order of integration is mapped to the hue spectrum from 0° to 240°, with red being the maximum. As can be seen, even adjacent particles might need a highly different number of vector field evaluations.

data structure, storing not only the final position of an advected particle, but reusing the k intermediate ones obtained from field integration during streak surface visualization facilitates the rendering of smooth surface animations even in data sets with low time resolution. In these computations, the magnitude of the smoothing length r of the weighting function used in MLS during vector field evaluation is inversely related to the point density of the data set. This point density is usually either an output value of the CFD simulation itself, or has to be determined on the fly by k-nearest neighbor computations or similar methods. We use MLS for vector field approximation because of its independence of a computational grid and because it is used as interpolant by the simulation that generated our test data sets.

To speed up the advection process, particle locality is used for efficient data set caching and parallel particle advection. We impose a rectangular grid on the surface clustering particles into independent sets and delegate the according computations from (4) to different CPU cores.

If a particle of the surface leaves the boundaries of the data set during advection, the particle itself and adjacent mesh elements are deleted, efficiently trimming the streak surface.

5.4 Delaunay Meshing

As mentioned before, the increased dimensionality of streak surfaces has a direct influence on the adaptivity methods that can be used. Hultquist's approach of inserting stream lines at a single curve-like consistent front cannot be generalized to the insertion of streak lines at a one-dimensional front of the streak surface. The two-dimensional character of the generalized front definition for streak lines requires the insertion of refinement structures on arbitrary positions of the surface. This fact virtually rules out the exclusive refinement along time- and streak lines of the surface, as most interior refinement points would miss these grid lines. Thus, we use a Delaunay type progressive mesh to define criteria for particle insertion.

Various work on the construction of Delaunay type surfaces from point clouds has been published over the years. For example, Gopi et al. [10] use a projective local Delaunay mesh for surface reconstruction. The underlying particle concept of streak surfaces suggests the use of point cloud reconstruction methods to obtain a surface mesh. However, the evolving property of the streak-surface mesh facilitates topologically correct (re-) meshing in a certain time step based on connectivity information given by prior time steps as well as triangle, edge, and node matching over multiple time steps.

The mesh structure of the previous time step generally loses its Delaunay properties as particles are advected, if corners of adjacent triangles describe different paths through the flow field, as illustrated in Figure 4. Since we want to estimate particle distributions using circumcircle properties for mesh refinement in a later step of the algorithm, it is important to have a well conditioned triangulation avoiding skinny triangles. Therefore, we choose to impose Delaunay's mesh properties on the triangulation of our surface, as the minimal circumradius property is a direct indicator of particle density. The availability of a mesh on the current particle set greatly simplifies construction of a curved Delaunay mesh, since

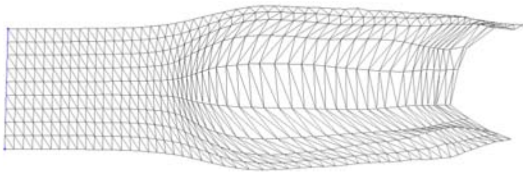


Figure 4 A uniform grid of a non-adaptive streak surface gets deformed, producing ill-conditioned triangles of bad aspect ratios.

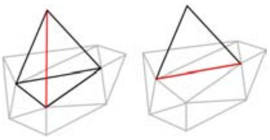


Figure 5 Edges of a tetrahedral structure on the curved surface mesh cannot be flipped without changing the underlying topology and creating holes.

this fact makes it possible to use local edge flipping instead of global mesh construction algorithms such as line-sweep. Moreover, reusing the previous particle connectivity allows matching of non-flipped edges as well as triangles of different time steps.

An edge of a mesh is *flippable*, if it is shared by two triangles and its flipped counterpart is not already part of the mesh , see also [7]. Let $e = (b, c)$ be a flippable edge shared by the two triangles $\Delta_1 = (a, b, c)$ and $\Delta_2 = (d, c, b)$. We flip e, if the sum of the angles α at a and β at d exceeds π, resulting in $\Delta'_1 = (a, d, b)$ and $\Delta'_2 = (c, d, a)$, satisfying local Delaunay properties. This flipping procedure, having been shown to converge for curved surfaces by Dyer et al. [7], is repeated until the mesh contains no more flippable edges. If non-flippable edges of tetrahedral structures of the curved triangle mesh remain after all flippable edges have been swapped, see Figure 5, we collapse the corresponding tetrahedra by removing the particle at its tip. Such tetrahedra generally represent noise in the form that they indicate a particle that evades the path of the streak surface and has earlier been inserted at a wrong position. After this step, our surface mesh is usually Delaunay conform, with exceptions to rare special cases of non-flippable edges. We avoid insertion of additional particles on the surface to obtain a mesh that fully satisfies Delaunay properties as proposed in [7] to reduce the resulting computational overhead that is needed to advect the new particles through the vector field. In these special cases we allow our mesh to be locally non Delaunay, as the according triangles are commonly not badly shaped due to the fact of mesh deformation of large parts of the surface in every step of integration.

5.5 Curvature Approximation

We determine local geometric complexity of the streak surface by measuring its local curvature. Curvature at a point p of a bivariate surface S is described by the maximal and minimal curvature of the curves on S that result from intersecting S with planes through p containing the surface-normal vector at p. These curvature values are called *principal curvatures* and will be used in the following as an indicator of whether to refine the mesh of the surface.
For adaptive particle insertion we need to know both a parametric form of the surface, as well as its local curvature at every particle of the surface. Instead of handling these tasks by two different approximation techniques, we use one weighted Least Squares approximation to both obtain a valid surface representation as well as to calculate the according local

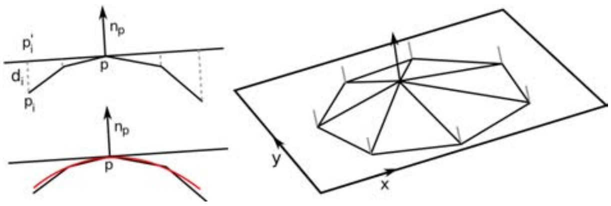


Figure 6 Two-dimensional illustration of a MLS curve approximation, MLS curve is shown in red (left) and three-dimensional analogon with a particle neighborhood-level of one (right).

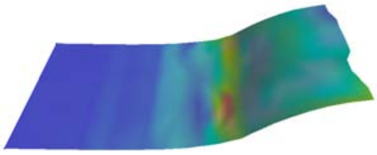


Figure 7 Maximum absolute curvature as hue color-map on a streak surface. Red color shows regions of high curvature, blue indicates planar sections.

curvature. We compute local surface approximation at a particle p based on particle offsets from a local tangential plane with vertex normal n_p as shown in Figure 6, resulting in a new set of projected data points $(p'_i = (x_i, y_i), d_i)$ that are approximated by a scalar-valued bivariate quadratic MLS. If the surface is to be approximated at a particle p, we assemble surrounding particles p_i by a neighborhood search along the edges of the mesh up to a neighborhood distance of two, as the bivariate quadratic LSE that has to be solved for a MLS approximation requires the data of at least six non-collinear points. The smoothing length r of the weighting function used during MLS approximation is chosen in a way, such that $w(p, p'_j) = \epsilon$ with p'_j being the neighboring particle that is farthest away from p. This MLS surface representation defines every point on a surface by its orthogonal distance to the tangential plane.

The eigenvalues of the Hessian of this polynomial approximation represent the principal curvatures c_1 and c_2. The Hessian of a bivariate scalar function takes the form of a 2×2 matrix, which can easily be calculated using a parametric form of the approximating MLS polynomial [12]. We use the *maximum absolute curvature* $c = \max(||c_1||, ||c_2||)$ as a measure of local feature size, which has proven to lead to good results in other applications, see [2]. Figure 7 shows a simple streak surface colored according to maximum absolute curvature. While Moving Least Squares for surface approximation is computationally more expensive than to simply use the piecewise linear representation of the surface described by the mesh itself, it both reduces errors in the particle insertion step, as explained in the next section and gives a more accurate notion of the surface curvature, by filtering small noise due to its approximating behavior.

5.6 Particle Insertion

The general notion of adaptivity is to sample a surface according to its geometric complexity, meaning that regions of high curvature need more samples to be represented accurately than regions that are almost planar. In the context of parametric or implicit surface modeling, curvature-dependent particle density control is often handled by minimization of an energy function [15]. In our case the availability of a correct coarse mesh as well as the absence of a gradient-definition requires different adaptivity measures.

The crucial step of streak surface adaptivity is the correct insertion of new particles at positions throughout the surface. The most accurate way of adaptively inserting particles is the insertion of a particle in the appropriate first time-step t_0, whenever an ill-conditioned particle-density is detected at an arbitrary later time step t_1. This method does however require the computationally inefficient advection of all newly inserted particles from t_0 til t_1. We therefore prefer to insert new particles directly into the evaluated time step or few of its predecessors.

Given the computed curvature at a particle p, see section 5.5, we are able to decide whether to insert new particles in the immediate neighborhood of p. Figure 4 demonstrates, why refinement purely along time- and path lines is not desired in the context of streak surfaces, since right angles between such lines are not maintained over time and such a skewed global coordinate frame is not suitable for balanced control of particle densities. We therefore use the local feature size in form of the curvature at a position p to directly describe the desired maximal allowed distance between two neighboring particles:

$$r_c = \frac{b}{\max(||c_1||, ||c_2||)}$$

where b controls the impact of curvature on the degree of mesh refinement and is commonly chosen be a value between zero and one. Circumradii of all triangles adjacent to p are compared to r_c, if a circumradius exceeds this threshold, and is larger than a pre-defined minimum triangle size, a new particle is inserted on the according triangle. We commonly limit the minimum triangle size to a size a few magnitudes smaller than the data set resolution to avoid numerical instabilities and oversampling. As our underlying mesh is mostly Delaunay conform and therefore satisfies the smallest circumcircle property, the circumradius of triangles is a valid measurement of particle density at a specific region.

Particle insertion itself poses the question of how to find the optimal position of insertion. On planar Delaunay meshes, one would choose the circumcenter of a triangle as location and handle point insertion in a simple way: Remove all triangles, whose current circumcircle includes the newly placed point and connect vertices of the resulting polygon with the new point. The resulting mesh again satisfies the Delaunay property. However, on curved surfaces this method yields several problems. Location of the circumcenter of obtuse triangles is not trivial, as the circumcenter is located outside of the triangle, thus requiring a search on neighboring triangles and even performing intersection operations, if the surface bends. Moreover, the point-in-circumcircle property does not work as expected from the planar case for both the common projected-circumcircle and the circumsphere definition on curved surfaces, if the point is not inserted on the piecewise linear representation of the surface. These considerations motivate our approach of particle insertion.

Once a triangle with a too large circumradius is detected, the according new particle is either inserted at the centroid of this triangle or on one of its edges. More precisely, we subdivide the longest edge of the triangle if the circumcenter is located outside of the triangle or the triangle itself is a boundary element of the surface. The problem caused by an invalid insertion of particles at the centroid of a boundary triangle is depicted in Figure 8, where the quality of the aspect-ratio of the boundary triangle is degraded significantly. These two types of particle insertion lead to well conditioned triangles with smaller circumcircle size as soon as the Delaunay property of the mesh is restored in the mesh at the next time step.

This insertion scheme yields adaptive streak surfaces but tends to introduce errors in the particle insertion positions that lead to further surface deviations during advection, even if newly inserted particles are offset according to the local MLS approximation of the surface as computed in the previous step of our algorithm. To avoid these artifacts, we

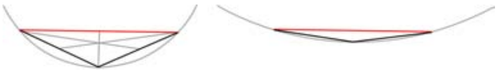


Figure 8 Circumradius of the triangle adjacent to the boundary edge (red) increases after particle insertion, as the boundary edge cannot be flipped to produce better conditioned triangles. Such cases are avoided by splitting of boundary edges.

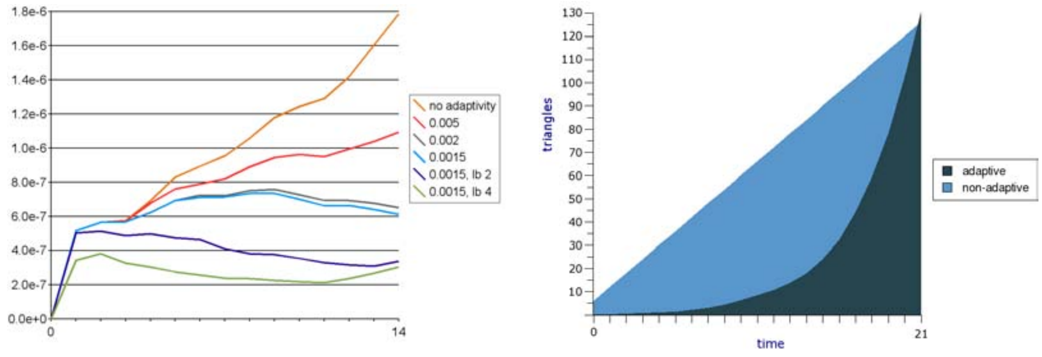
introduce the concept of looking back at a history of $lb > 0$ previous instances of concerned triangles or edges. If a triangle (edge) tri_1 is bound to be refined in time step t_1, we find matching triangles (edges) in time steps $t_i \in [t_1 - lb + 1, t_1 - 1]$ and insert particles into according triangles in t_i, until either step $t_1 - lb + 1$ is reached, the curvature of the surface at the corresponding position in t_i falls below a given minimum, or no matched preceding triangle exists and cannot be created by edge flips. These m newly inserted particles describe an ordered sequential set of points $[p_0, p_{m-1}]$ located on the linear representation of the individual surfaces $[S_{t_1-m}, S_{t_1}]$. As these particles need to lie on the same path line, positions of particles p_i, with $0 < i <= m - 1$ are offset according to the position of p_{i-1} after one full time step of particle advection.

Our tests have shown, that this leads to a great reduction of noise artifacts in later time steps, that are caused by deviating paths of particles that were inserted at poorly approximated positions.

Curvature as well as circumradius properties can be used for the removal of particles as well. In the case, that a particle p has a sufficiently small curvature, i.e. the surface is almost planar, and all adjacent triangles have small circumradii, it is valid to remove the concerned particle without losing any details in the surface representation. In realistic applications such as the test data sets shown in section 6 such situations do hardly occur.

6 Results

In the following we present numerical examples of adaptive streak surface integration in different data sets generated by a point based, grid-less CFD simulation [18]. The first data set consists of about 25.000 particles simulating flow around a cylindrical obstacle with ellipsoidal profile. Dimensions of the obstacle as well as velocity of the fluid were chosen specifically to yield a high Reynolds number, leading to a three-dimensional Von Kármán vortex street and a low time resolution. The resulting disturbed flow structures have optimal properties to observe the quality of our adaptive streak surface integration approach. Figure 10 depicts six non-consecutive time steps of adaptive streak surface integration in this first data set. Near the vortex structures, the surface mesh has a high curvature and needs to be sampled accordingly, as described in sections 5.5 and 5.6. Highly twisted and folded surface structures in the Von Kármán vortex street demonstrate the robust refinement of our approach. The surface in the last time step shown consists of around 110.000 particles. The second data set contains a spherical obstacle, approx. 26.000 flow particles and fluid flow with a Reynolds number close to the one from data set one. Both data sets have a low resolution in the time dimension to demonstrate the robustness of artifact avoidance by utilization of our look back strategy. In contrast to the first data set, absence of a distinguished direction on the obstacle leads to aperiodic, intensively folded flow structures, see Figure 11. To give an impression of how accurately surfaces generated by the adaptive integration scheme and the Delaunay mesh refinement method introduced in this work represent the "real" surface, we show comparisons between high-resolution non-adaptive meshes and adaptive surfaces in the two Figures 14 and 15. The former illustrates how the



(a) **(b)**

Figure 9 **(a)** Error statistics for different levels of adaptivity and look back. **(b)** Chart of the number of triangles (in multiples of 1.000) in the first 22 time steps of the streak surface shown in Figure 10. Dark blue refers to the adaptive streak surface, light blue to the non-adaptive streak surface, that has an approximately identical number of triangles at step 22.

Table 1 Exemplary measurements of total streak surface generation times in milliseconds for 25 time steps of data set one with 20× parallel particle advection. Final particle number is approx 40.000 for both approaches.

Method	Advection	Delaunay	Adaptivity
non-adaptive	1.768.470	2.865	0
adaptive, 0.002, 2 lb	790.890	1.155	9.081

folds of a high-resolution mesh are correctly refined and represented by an adaptive surface with only half as many triangles as an equivalent non-adaptive surface. Efficient distribution of particle densities can be observed in Figure 15, where the adaptive version of a streak surface clearly has a particle distribution of better quality than a non-adaptive surface with an equal number of points.

Statistical error measurements are shown in Figure 9a. The measured two-sided surface mesh error is computed with respect to (5).

$$E(S_1, S_2) = \frac{1}{|P_1| + |P_2|} \left(\sum_{p \in P_1} d(p, S_2) + \sum_{q \in P_2} d(q, S_1) \right) \tag{5}$$

where S_1 and S_2 are the meshes of two streak surfaces at the same time step with according point sets P_1 and P_2. It is important to note that, as d computes the minimal distance of a point to a surface mesh, error values for the same point sets usually differ if the mesh is changed. While small features in the error curves of Figure 9a may therefore be caused by flipped edges, the overall benefit gained from reducing the minimal surface resolution and increasing the number of look back steps is clearly visible. The plotted exemplary measurements were taken at different time steps of a data set that was scaled to fit into a unit cube and represent absolute per particle error when comparing the according surface to a high resolution ground truth streak surface. The shown measurements represent 15 consecutive time-steps of a surface evolving under extreme stretching and turbulence conditions - a representative scenario that requires reliable surface reconstruction techniques to produce accurate streak surfaces.

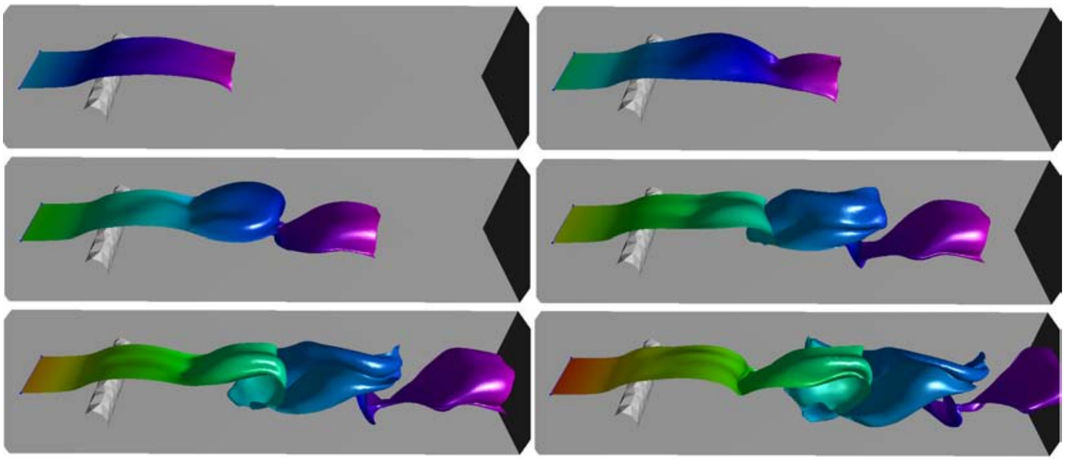


Figure 10 A sequence of six time steps in a three-dimensional Von Kármán vortex street produced by test data set number one. The extracted surface is colored with respect to time, with purple showing the particles generated at time t_0 and red being used for new particles. Details of twisting folding, and stretching are visible.

In addition to the direct impact of our adaptivity approach on particle counts and distributions, the reduced number of particles does influence the computation times of surfaces as well. We show a representative graph of particle number development in a surface generated in the ellipsoidal data set in Figure 9b. The triangle count for the adaptive surface shows exponential growth as long as the surface is completely inside the domain of the data set and is not trimmed. Exponential growth is stopped as soon as parts of the surface leave the data set. Integrals of the depicted curves are directly proportional to the time spent for total particle advection. From the given graph, it is clearly visible that adaptive streak surfaces require the advection of a larger number of particles to reach the same final number. Due to better distribution of particle densities, an adaptive surface with the same number of particles generally represents a better approximation of the real surface. The previously mentioned assumption about particle advection time is verified by measurements during surface integration. The total time spent for adaptivity measures is a combination of computation times for surface and curvature approximation as well as particle insertion. In our tests only a percentage of less than 2% of the total surface generation time was consumed by the adaptivity method, even if particle advection was performed 20 times in parallel. As the adaptive integration scheme discussed in section 5.3 commonly requires multiple evaluations of the vector field for every particle, the amount of time dedicated to particle advection is much higher than the one used up by mesh refinement, thus noticeably speeding up the integration process. Further improvements on this performance can be made by parallelization of the mesh refinement algorithm. While in our tests generation of adaptive surfaces was in average 4 times faster than creation of equivalent higher resolved surfaces, when using 10× parallel particle advection, it is difficult to obtain meaningful absolute time comparisons (cf. Table 1), as speed-up is highly dependent on the method of vector field approximation and flow complexity. In general, the relative speed-up gained by adaptive surface generation is proportional to the complexity of flow and field evaluation methods.

For insightful visualization of generated surfaces, we utilize several known visualization techniques. Four of these techniques are displayed in Figure 12. While the first picture showing a standard particle based visualization of the surface is not capable of conveying the

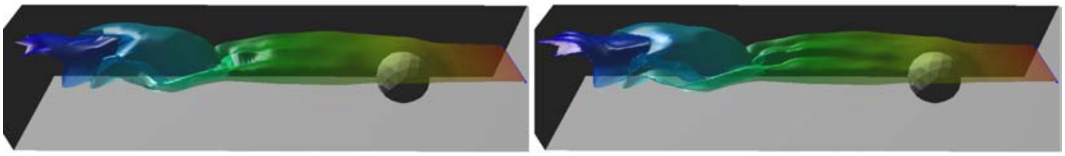


Figure 11 Transparent rendering of a non-adaptive and an adaptive streak surface with the same seeding resolution extracted from test data set number two. The spherical obstacle produces aperiodic disturbances.

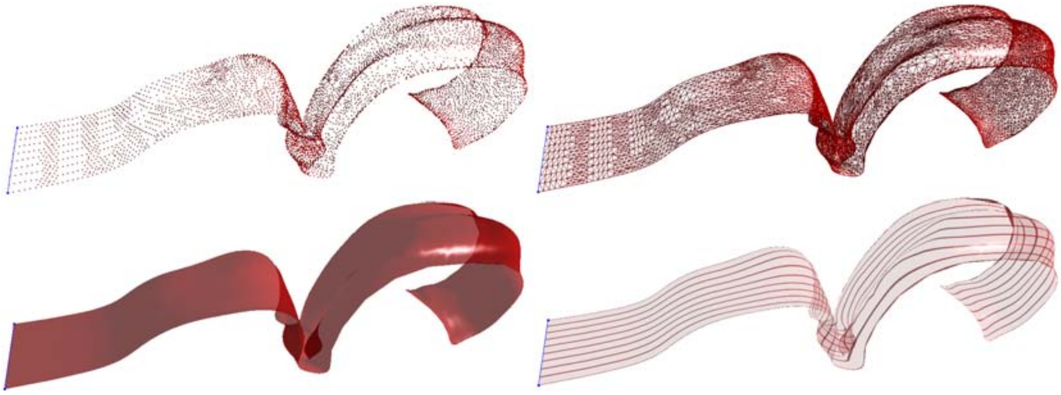


Figure 12 One time step of streak surface integration shown in four different visualization techniques for surface rendering. Techniques are: Surface particles rendered as shaded points, wire-frame triangulation, solid transparent triangulation, and texture mapping showing streak line structures.

impression of a homogeneous surface, it gives an insight into adaptive particle distributions. Particle density and connectivity information are shown by a direct wire-frame representation of the Delaunay construct as used in the second frame. Shading and texturing techniques are applied to the solid Delaunay triangulation based surface visualizations. We use a simple axis-based triangle pre-sorting approach for transparent surface rendering as shown in the last two frames. Texturing allows the rendering of streak- or time-line like parts of the surface, former is shown in frame four.

These renderings of streak surfaces have in common, that the Delaunay construct used for density estimation is used directly for surface visualization, leading to a simple and fast visualization allowing the analysis of adaptivity properties. For high quality surface rendering, one would typically use a smooth interpolation of the surface point set and a much finer resolved triangle grid to get rid of fine discontinuities of the simple linear representation.

To illustrate the use of the methods introduced in this paper in a real application, we present an example of an adaptive evolving mesh being equivalent to a streak surface without a seeding rake in Figure 13. The application is concerned with stirring of a fluid at high temperature and consists of a cylindrical barrel and two rotating wheels with four attached mixing poles. The extracted mesh segments the fluid in a stirring simulation into two separate volumes, giving an impression of how our approach to adaptive streak surfaces can be used to separate regions of different flow as done in vector field topology.

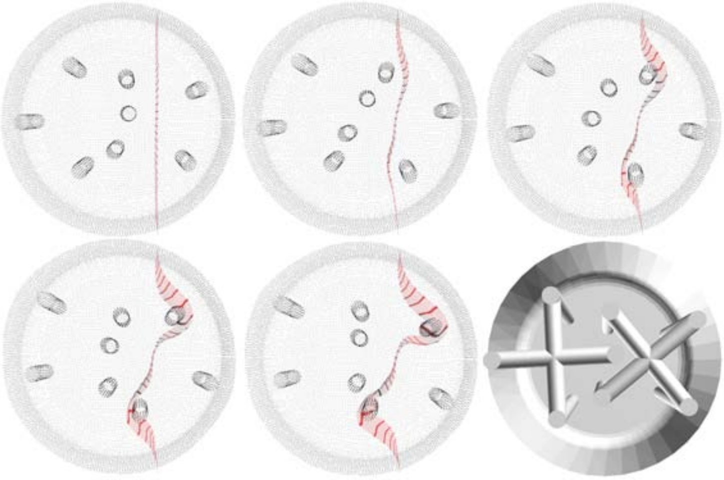


Figure 13 Refinement mesh in a stirring simulation.

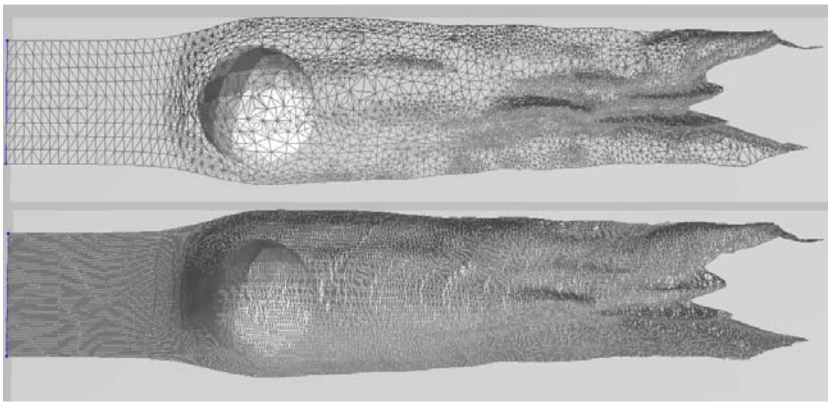


Figure 14 Triangulations of an adaptive (top) and a high-resolution non-adaptive streak surface passing a spherical obstacle. Triangle counts in the time step shown evaluate to approximately 20.000 and 45.000. The adaptive surface yields a highly diverse particle density, while accurately representing fold-like features.

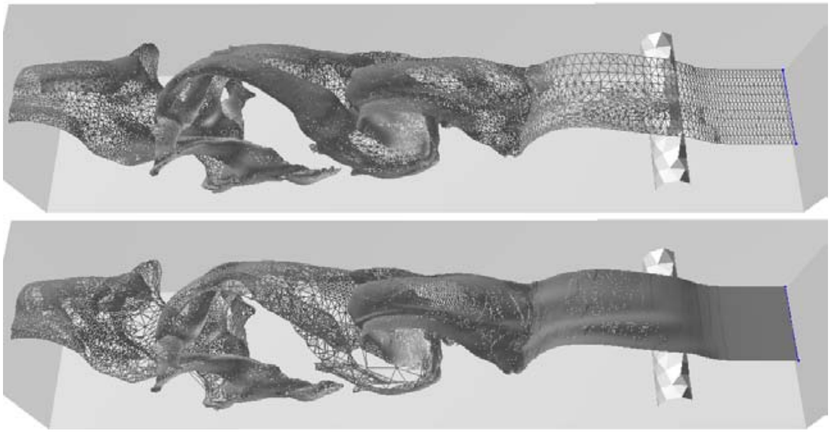


Figure 15 Triangulations of an adaptive (top) and a non-adaptive streak surface with approximately 110.000 triangles each. The non-adaptive version shows less optimal particle distributions and regions with low mesh resolution.

7 Conclusion

In this paper we have introduced an approach to fully adaptive streak surface integration. Our method is based on MLS approximation for the evaluation of the vector field and surface as well as curvature approximation. For particle insertion we utilized the circumradius properties of a Delaunay type evolving mesh, which is restructured after every time step to restore Delaunay properties. The distinction between two types of particle location during insertion avoid the generation of triangles with bad aspect ratios. A look back strategy during particle insertion further reduces approximation errors.

The results shown in section 6 demonstrate the accurate, robust, and fast integration of adaptive streak surfaces generated by our method. Moreover we have presented results in a real world application of a mixing process, illustrating the suitability for practical flow analysis and topology-based visualization.

The methods introduced in this work are portable to other fields of visualization where moving or evolving meshes are concerned, such as the concept of unsteady flow volumes presented by Becker et al. [3]. Common approaches to the extraction of vector field topology of fluid flows [20] based on the use of stream surfaces as three-dimensional separatrices in the stationary case can be generalized to vector field topology methods in non-stationary fields by the application of streak surfaces for segmentation of time-varying volumes.

Future work on the topic of adaptive streak surface construction may include the utilization of improved spatial clustering for further parallelization of the surface generation process as well as the integration of vector field singularities into the streak surface generation process.

Acknowledgments

This work was supported by the German Research Foundation (IRTG 1131) and by the Center for Mathematical and Computational Modeling (CM)^2.

References

1 Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishmann, David Levin, Claudio T. Silva: Point Set Surfaces *Visualization '01, pp. 21–28 (2001)*

2 Bruno Rodrigues de Araujo, Joaquim Armando Pires Jorge: Curvature Dependent Polygonization of Implicit Surfaces *Proceedings of the Computer Graphics and Image Processing 2004, pp. 266–273 (2004)*

3 Barry G. Becker, Nelson L. Max, David A. Lane: Unsteady Flow Volumes *Proceedings of the 6th conference on Visualization '95, p. 329 (1995)*

4 Hao Chen, Jonathan Bishop: Delaunay Triangulation for Curved Surfaces *6th International Meshing Roundtable Proceedings, pp. 115–127 (1997)*

5 L. Paul Chew: *Guaranteed-Quality Mesh Generation for Curved Surfaces 9th Symposium on Computational Geometry, pp. 274–280 (1993)*

6 Nicolas Cuntz, Andreas Kolb, Robert Strzdka, Daniel Weiskopf: Particle Level Set Advection for the Interactive Visualization of Unsteady 3D Flow *Eurographics/IEEE VGTC Symposium on Visualization 2008*

7 Ramsay Dyer, Hao Zhang, Torsten Möller: Delaunay Mesh Construction. *Eurographics Symposium on Geometry Processing, pp. 273–282 (2007)*

8 Wolfram von Funck, Tino Weinkauf, Holger Theisel, Hans-Peter Seidel: Smoke Surfaces: An Interactive Flow Visualization Technique Inspired by Real-World Flow Experiments *IEEE TVCG 2008, pp. 1396–1403 (2008)*

9 Christoph Garth, Hari Krishnan, Xavier Tricoche, Tom Bobach, Kenneth I. Joy: Generation of Accurate Integral Surfaces in Time-Dependent Vector Fields *IEEE TVCG 2008, pp. 1404–1411 (2008)*

10 M. Gopi, S. Krishnan, C. T. Silva: Surface Reconstruction Based on Lower Dimensional Localized Delaunay Triangulation *Computer Graphics Forum 19, (2000)*

11 J. P. M. Hultquist: Constructing stream surfaces in steady 3D vector fields *Visualization '92, pp. 173–175 (1992)*

12 Soo-Kyun Kim, Chang-Hun Kim: Finding ridges and valleys in a discrete surface using a modified MLS approximation *Computer-Aided Design, Vol. 38, No. 2, pp. 173–180 (2006)*

13 Jens Krüger, Peter Kipfer, Polina Kondratieva, Rüdiger Westermann: A Particle System for Interactive Visualization of 3D Flows *IEEE TVCG 2005*

14 Boris Mederos, Luiz Velho, Luiz Henrique de Figueiredo: Moving Least Squares Multiresolution Surface Approximation *Proceedings of the Computer Graphics and Image Processing 2003, pp. 19–26 (2003)*

15 Miriah D. Meyer, Pierre Georgel, Ross T. Whitaker: Robust Particle Systems for Curvature Dependent Sampling of Implicit Surfaces *Proceedings of the International Conference on Shape Modeling and Applications 2005, pp. 124–133 (2005)*

16 A. Nealen: An As-Short-As-Possible Introduction to the Least Squares, Weighted Least Squares and Moving Least Squares Methods for Scattered Data Approximation and Interpolation, *Discrete Geometric Modeling Group, TU Darmstadt (2004)*

17 Tobias Schafhitzel, Eduardo Tejada, Daniel Weiskopf, Thomas Ertl: Point-Based Stream Surfaces and Path Surfaces *Graphics Interface 2007, pp. 289–296 (2007)*

18 S. Tiwari, J. Kuhnert: Finite Pointset method based on the projection method for simulations of the incompressible Navier-Stokes equations. *Springer Lecture Notes in Computational Science and Engineering: Meshfree Methods for Partial Differential Equations I (2001)*

19 Daniel Weiskopf: Dye advection without the blur: A level-set approach for texture-based visualization of unsteady flow *Proceedings of Eurographics 2004, pp. 479–488 (2004)*

20 Alexander Wiebel, Xavier Tricoche, Gerik Scheuermann: Extraction of Separation Manifolds using Topological Structures in Flow Cross Sections. *Topology-Based Methods in Visualization II, pp. 31–44 (2009)*

Interactive Isocontouring of High-Order Surfaces

Christian Pagot[1], Joachim Vollrath[2], Filip Sadlo[2], Daniel Weiskopf[2], Thomas Ertl[2], and João L. D. Comba[1]

1 **Instituto de Informática, UFRGS**
Porto Alegre, RS, Brazil, CP 15064
{capagot,comba}@inf.ufrgs.br

2 **VISUS, Universität Stuttgart**
Allmandring 19, 70569, Stuttgart, Germany
{vollrath,sadlo,weiskopf,ertl}@visus.uni-stuttgart.de

Abstract

Scientists and engineers are making increasingly use of *hp*-adaptive discretization methods to compute simulations. While techniques for isocontouring the high-order data generated by these methods have started to appear, they typically do not facilitate interactive data exploration. This work presents a novel interactive approach for approximate isocontouring of high-order data. The method is based on a two-phase hybrid rendering algorithm. In the first phase, coarsely seeded particles are guided by the gradient of the field for obtaining an initial sampling of the isosurface in object space. The second phase performs ray casting in the image space neighborhood of the initial samples. Since the neighborhood is small, the initial guesses tend to be close to the isosurface, leading to accelerated root finding and thus efficient rendering. The object space phase affects the density of the coarse samples on the isosurface, which can lead to holes in the final rendering and overdraw. Thus, we also propose a heuristic, based on dynamical systems theory, that adapts the neighborhood of the seeds in order to obtain a better coverage of the surface. Results for datasets from computational fluid dynamics are shown and performance measurements for our GPU implementation are given.

1998 ACM Subject Classification I.3.3 [Computer Graphics]: Picture/Image Generation–Display algorithms; I.3.1 [Computer Graphics]: Hardware Architecture–Graphics processors;

Keywords and phrases High-order finite elements, isosurface visualization, GPU.

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.276

1 Introduction

Computational simulations use finite element methods to approximate the solution of partial differential equations. While the use of low-order linear basis functions within the finite elements was common in the past, the increasing need for numerical accuracy and efficiency motivates the development of more sophisticated numerical methods. An example is the *hp*-adaptive finite element method used in computational fluid dynamics, which produces high-order approximations inside each element. This leads to more accurate representations of physical phenomena. However, for visual inspection, the rendering of such data can become costly due to expensive evaluation. Approximate representation is one way to offer a viable visualization approach, allowing a trade-off between rendering speed and accuracy. Low-order resampling and isocontouring algorithms such as Marching Cubes (MC) [8] are among the simplest solutions to this problem. Adaptive sampling variations of MC can further reduce the error and capture more complex structures [14, 17]. However, resampling approaches

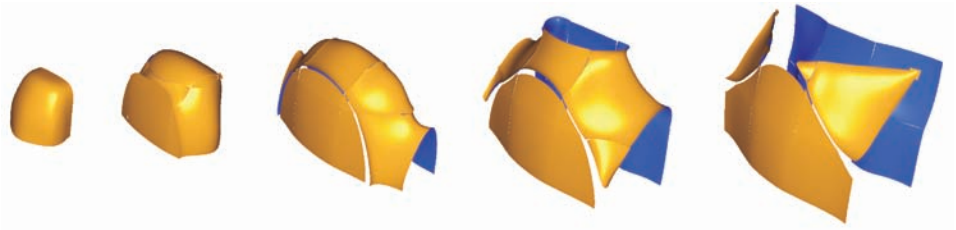


Figure 1 Isocontouring of a sequence of different density levels for a high-order dataset obtained from a Discontinuous Galerkin flow simulation.

introduce error and often lead to increased memory demands due to the large number of lower-order elements needed to represent the original data with sufficient accuracy.

To overcome these limitations, more attention has been paid recently to methods for direct contouring of high-order data. Usually, in these methods, isocontouring is realized as a root finding or gradient descent problem. Due to the high-order nature of the data, there is no closed-form solution for these approaches, and numerical methods must be used instead. Since these methods are typically computationally expensive, they are often carried out in a pre-processing step, which was applied in several mesh-extraction [14, 16, 17] and point-based algorithms [3, 20, 18, 10]. Although interactive rendering is possible with these techniques because the isosurface is computed during pre-processing, changing isovalues requires recomputation of the isosurface and hence the overall visualization is often no longer interactive. The ability to change isovalues interactively is present in some ray casting or ray tracing isocontouring algorithms [19, 11, 6]. However, the evaluation during rendering of such numerical methods typically leads to lower frame rates.

In this work we present a technique for the interactive approximate contouring of high-order data. It allows to choose the trade-off between rendering speed and rendering quality and is called iHOS (isocontouring of High-Order Surfaces) in this paper. It relies on a hybrid algorithm that distributes the isocontouring workload over pre-processing and rendering phases, allowing for interactive exploration by changing isovalues. The main components include:

- A parallel particle-based technique to generate a coarse, view-independent sampling of the isocontour in object space, which is only executed when the isovalue changes.
- An algorithm based on dynamical systems theory that allows for improved isosurface sampling.
- GPU-based ray casting that shoots a collection of rays for each quad around each sample in object space.

The method is designed to be efficiently mapped to parallel architectures and does not rely on inter-element connectivity information. Element reordering is not necessary for correct rendering, although performance improvement is achieved by processing elements in front-to-back order. Results are presented for datasets composed by convex cells. However, since no cross-cell rays are used, the method can be easily adapted to work with concave cells through the use of a more general point-in-polyhedron test. These are desired features that allow easy handling of structured and unstructured meshes as well. Figure 1 presents the results obtained with our method for a sequence of different isovalues for a high-order dataset obtained from a Discontinuous Galerkin flow simulation.

2 Related Work

There is a vast literature on isocontouring; the discussion below focuses on techniques for high-order data. Figueiredo *et al.* [3] proposed physically-based approaches for extracting triangle meshes from implicit surfaces. One method is particle-based while the second is based on a mass-spring system, and represents the first attempt on using particles to sample high-order data. This work inspired Witkin and Heckbert [20] to develop a point-based tool for the modeling and visualization of implicit surfaces. When used as a modeling tool, points represent handles for changing shapes. As a visualization tool, points are projected onto the surface and rendered as discs, with size and distribution adaptively computed according to the curvature of the surface. Meyer *et al.* [10] proposed a technique for isosurfaces generated from high-order finite element simulations that builds upon the approach of Witkin and Heckbert. Potential functions are used for particles repulsion, giving more control over particle distribution. This method handles cells with curved surfaces and allows for interactive visualization of a given isosurface. However, interactive exploration of different isovalues is not viable since every isovalue change typically takes several minutes due to resampling.

Kooten *et al.* [18] presented the first interactive particle-based method for implicit surface visualization that runs entirely on the GPU. The projection method is an adaptation of that by Witkin and Heckbert. Particle repulsion relies on a spatial-hash data structure that requires costly and frequent updates, thus preventing its use for rendering large datasets. Although not directly related to isocontouring, we mention the work by Zhou and Garland [21], which presents a point-based approach for the direct volume rendering of high-order tetrahedral data. Despite their effectiveness, point-based methods typically only provide a coarse representation for the isosurface, and accurate representations require a huge number of points. Haasdonk *et al.* [4] presented a multi-resolution rendering method for *hp*-adaptive data based on polynomial textures targeted to 2D rendering. Schroeder *et al.* [16] presented a mesh extraction method that explores critical points of basis functions to provide topological guarantees on the extracted mesh. Similar to other mesh extraction methods, it employs computationally expensive pre-processing to allow interactive exploration of arbitrary isovalues. Remacle *et al.* [14] proposed a method for resampling high-order data inspired by adaptive mesh refinement (AMR) methods. The original high-order data is down-sampled to a lower-order representation which is suitable for lower-order visualization algorithms like Marching Cubes [8]. The resampling error threshold can be adjusted, but low thresholds can lead to memory consumption explosion.

A competing strategy is to avoid pre-computation and compute isocontours during rendering. Nelson and Kirby [11] presented a ray tracing-based method for the isocontouring of spectral/*hp*-adaptive data. The method works by projecting the high-order functions onto each traced ray and computing the intersection with the isosurface from the resulting univariate function. Visualization error is carefully quantified and reduced. On the other hand, it typically takes several seconds to generate the final image, prohibiting interactive exploration of the data.

Knoll *et al.* [6] presented an interactive interval arithmetic-based method for the ray tracing of implicit surfaces on the GPU. It is currently one of the fastest ray tracers for implicits, presenting also robustness with respect to the root-finding process. However, the equations of the implicits must be converted to an inclusion-computable form, which increases the number of arithmetic operations needed to evaluate the equations. This is not a problem in the case of polynomials with only few coefficients, as can be verified by their results. However, in the case of simulation data we usually have polynomials with hundreds of coefficients and thousands of polynomials to be evaluated for each frame, and the use of interval arithmetic may impact performance.

3 Isocontouring High-Order Surfaces

In this section we describe the details of the isocontouring High-Order Surface (iHOS) algorithm. We start by first giving an overall description, followed by details of each step.

3.1 iHOS design

Interactive rendering rates in iHOS are obtained by dividing the isocontouring workload between object and image space computations, each defined in a separate phase. The first phase, in object-space, generates an initial sampling of the isosurface by projecting particles (here called *seeds*) along the gradient field onto the surface. Only seeds successfully projected onto the surface are considered for further processing. Each projected seed generates a surface-tangent, seed-centered quadrilateral (quad) that covers the neighborhood of the seed. The second step uses the fragments generated by the rasterization of those quads as the initial points for a ray casting that refines the isosurface representation in image space. Figure 2 gives an overview of the iHOS pipeline.

To compute the initial isosurface approximation, we start by placing a set of seeds uniformly distributed inside each cell and projecting them onto the surface. The projection affects the density of the seeds and thus the sampling quality. There are methods described in the literature that handle this problem by using repulsion/attraction or birth/kill of seeds in a pre-processing stage [20, 10, 18]. Despite their effectiveness, these procedures are computationally expensive, requiring the update of complex data structures that are hard to efficiently map to parallel architectures. Although our experimental results show that satisfactory quality images are obtained even without handling the sampling problem, we additionally propose a heuristic that helps in reducing the artifacts of the final image. This heuristic is implemented as a pre-processing stage that analyzes, under the dynamical systems theory perspective, the seeds behavior along the gradient field during projection. Differently from the previous methods that must compute the pre-processing for each isovalue, this pre-computation is executed only once for the entire dataset and does not depend on a specific isovalue. The next sections give an in depth explanation of each step of our algorithm. We start by first reviewing the finite element (FE) high-order data subject to this work.

3.2 *hp*-adaptive FE Data

FE methods imply discretization. The physical domain Ω is discretized in a compatible mesh (meaning that the elements intersect only at the boundary faces, edges, or vertices). A given mesh M with N elements [14] can be defined as

$$M = \bigcup_{i=1}^{N} e_i \tag{1}$$

where e_i is the i-th element. In some FE method simulations, results are stored at the vertices of regularly distributed elements. However, recent numerical methods brought the concepts of h- and p-adaptiveness. h-adaptive methods vary the element size (h-refinement) while p-adaptive methods are those that can vary the order of the basis functions (p-refinement). The hp-adaptive methods combine both. These methods are of special interest because they

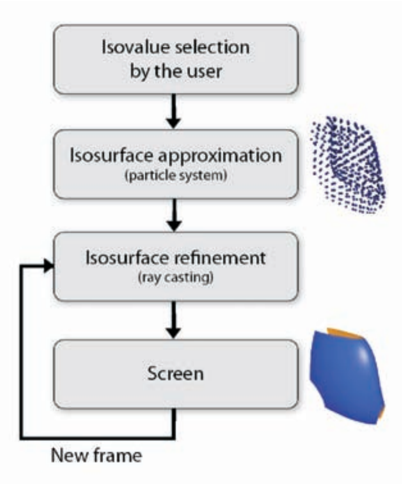

Isovalue selection by the user
Isosurface approximation (particle system)
Isosurface refinement (ray casting)
Screen
New frame

Figure 2 iHOS pipeline: An initial sampling of the isosurface is generated in object space by projecting coarsely seeded particles over the isosurface along the gradient field in object space. Each projected point generates a quadrilateral whose fragments will be used as starting points for the ray casting in the image-based phase.

can, through the balance between *h*- and *p*-adaptiveness, increase the convergence order of computations. In *p* or *hp* methods, results are defined continuously for each element using high-order functions.

In the specific case of Discontinuous Galerkin methods [13] the solution is computed separately for each element, which means that the solution can be discontinuous at the element boundaries. It is also common to have the higher-order basis functions defined in a reference space, and a mapping function that transforms coordinates from reference to world space. Although this mapping can be non-linear, we currently only handle transformations that include rotation, translation, reflection, and scaling.

3.3 Object Space Sampling

The task of finding an isosurface for a given polynomial data can be formulated as a root-finding problem. For high-order polynomials, the lack of closed form solutions lead to the use of iterative numerical methods. Newton-Raphson (NR) is one of the best known methods for numerical root finding. Additionally, it presents good convergence rates when the starting point is already close to the desired solution. However, uniformly distributed seeds may not be close to the solution and NR may fail, leading to poor sampling. One possible solution is to guide the seeds through the gradient field until they get closer to the isosurface and then apply NR to refine the projection. Since we just want to bring the seeds closer to the isosurface, we decided for guiding the points by means of integration. After the integration step it is assumed that the seeds are closer to the desired isosurface and we start with the NR iterations. Directional NR [7] along the gradient is used, since it has the additional advantage of quadratic convergence. Equation 2 shows the formulation used for the NR iterations.

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} - \frac{f(\mathbf{x}^{k})}{\nabla f(\mathbf{x}^{k}) \cdot \nabla f(\mathbf{x}^{k})} \nabla f(\mathbf{x}^{k}) \tag{2}$$

where k is the time step ($k = 0, 1, \ldots$), $\mathbf{x}$ is the point position in reference space, and f is the polynomial for the current cell.

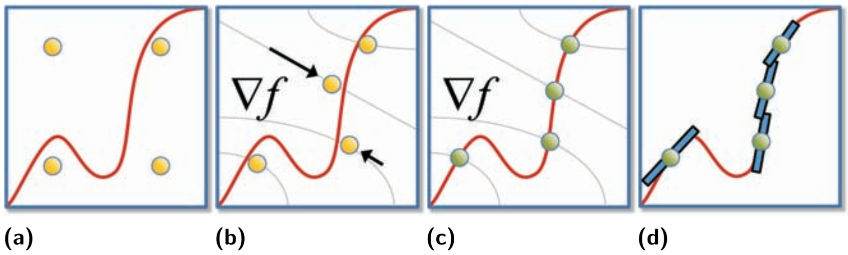


Figure 3 Isosurface approximation: (a) Seeds are initially distributed uniformly into each cell. (b) Seeds are guided through integration along the gradient field towards the desired isosurface. (c) Seeds are projected using directional NR on the isosurface. (d) Surface-tangent quads are generated for each point projected onto the isosurface.

All computations are performed separately for each seed point since no neighborhood information is needed. Figure 3 gives an overview of the first phase.

3.4 Image Space Refinement

The input for the second phase of our method is a set of surface-tangent, seed-centered quads that together cover the isosurface. Points lying on these quads can be used as starting points for the ray casting that will refine the representation of the isosurface. These points are efficiently obtained through the rasterization of the quads in image space. The fact that these points are already close to the isosurface implies that we can use the directional NR method along the ray direction, as show in Figure 4. Equation 3 shows the formulation for the directional NR used for the fragment projection:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \frac{f(\mathbf{x}^k)}{\nabla f(\mathbf{x}^k) \cdot \mathbf{r}}\mathbf{r}\,, \tag{3}$$

where k is the time step ($k = 0, 1, \ldots$), $\mathbf{x}$ is the point position in reference space, f is the polynomial for the current cell, and $\mathbf{r}$ is the ray vector in reference space.

The generated ray and fragment coordinates are transformed into the reference coordinate system, where the ray-isosurface intersection computations are performed. After the maximum number of iterations is reached, if the fragment is not projected onto the isosurface (within an error tolerance bound) or it has been moved outside of the cell, it is discarded. If the fragment is successfully projected, shading is computed and the corresponding pixel color is updated.

4 Sampling Strategies

Both steps, the object space sampling (Section 3.3) and the image-based refinement (Section 3.4), require appropriate sampling for obtaining a complete isosurface representation. In some cases holes can result due to improper sampling. As a limit case of the object space sampling, infinitely dense seeding would guarantee a complete isosurface representation, but in practice, no guarantees about the resulting isosurface sampling can be given.

The reason is that in general no assumptions about the behavior of the gradient field can be made, and hence of the mapping of the seeds to the isosurface. It is, for example, generally not possible to estimate the maximum distance between two seeds when they are mapped to an isosurface of arbitrary isolevel. A limit case of the image space refinement is a single

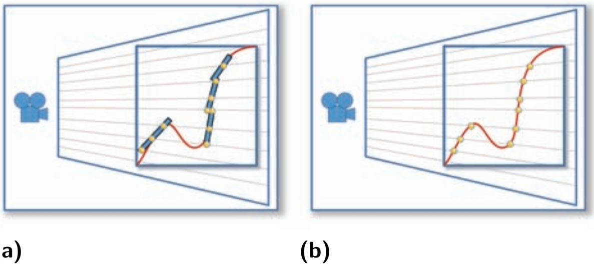


Figure 4 Isosurface refinement: (a) Quads are rasterized in image space. (b) Each fragment is projected against the isosurface through directional Newton-Raphson.

quad that spans the complete screen, which would be equivalent to a ray casting approach. This would lead to the typically low performance of these methods due to computationally expensive root finding along the viewing rays. Our method operates between the two mentioned limit cases. To motivate the parametrization and a sampling technique, we will first discuss the relevant issues and implications that arise in the context of sampling.

4.1 Sampling Issues

The proposed approach suffers from the fact that the fragments are processed independently and therefore it is impossible to detect, and to directly fix holes in the isosurface. Holes can be caused by different reasons:

1. the root finding may fail, e.g. due to insufficient iteration count,
2. the root finding may deliver a point on a nearby isosurface if the quad is not well aligned with the isosurface, i.e. the isosurface exhibits too high curvature with respect to the initial sampling,
3. the quad sizes are insufficient.

The first cause is hard to address explicitly because the efficiency of our method relies on synchronized root finding and hence constant iteration count (see Section 5.2). However, this reason is typically negligible. Because we use quads that are aligned with the tangent plane of the isosurface at each sampling seed, the second case depends only on the quality of the initial sampling and the last one on the quad sizes. The initial sampling has to be performed each time the isolevel is changed and therefore the exploration of isocontours requires high performance for this operation. This makes the use of expensive approaches not feasible for this step. Unfortunately, it is generally not possible to find a distribution of the seeds for the object space sampling that leads to uniformly sampled isosurfaces for all possible isovalues. This makes non-uniform quad sizes for the image space refinement necessary.

Therefore, we propose a method for estimating sufficient quad sizes in Section 4.2 and place the seeds for the object space sampling regularly inside each cell. One problem is that even comparably large quads cannot guarantee hole-free coverage of the isosurface because the sampling points can be arbitrarily distant due to the possible intricacy of the gradient field. Additionally, large quads typically lead to high overdraw and hence to lowered performance. All in all, a balance has to be found between the number of initial seeds for the object space sampling and the sizes of the quads.

For approximate results at high performance that can serve as a preview to more expensive but robust methods such as that by Knoll *et al.* [6], the sampling in object space can be parametrized and the size for the quads can be derived from that sampling as shown in Section 6. As an alternative, we now propose a method to estimate conservative quad sizes for individual quads.

4.2 Estimating Quad Sizes using FTLE

A method to determine quad sizes that guarantees a complete covering of the isosurfaces under the assumption of an appropriate object-space sampling can be derived from dynamical systems theory. The finite-time (or local) Lyapunov exponent (FTLE) was originally defined to measure the predictability of dynamical systems [9, 5]. More precisely, it measures the exponential growth that a perturbation undergoes when it is advected for a finite time interval in a vector field. The flow map $\boldsymbol{\phi}^T(\mathbf{x})$, which maps a sample point $\mathbf{x}$ to its advected position $\boldsymbol{\phi}^T(\mathbf{x})$ after advection time T, is the basis for the Cauchy-Green deformation tensor $\nabla\boldsymbol{\phi}^T(\mathbf{x})$. Using

$$\Delta^T(\mathbf{x}) = (\nabla\boldsymbol{\phi}^T(\mathbf{x}))^\top \cdot \nabla\boldsymbol{\phi}^T(\mathbf{x}) \tag{4}$$

and $\lambda_{\max}$ being the largest eigenvalue, leads to the finite-time Lyapunov exponent [5]:

$$\sigma^T(\mathbf{x}) = \frac{1}{|T|} \ln \sqrt{\lambda_{\max}(\Delta^T(\mathbf{x}))}\,. \tag{5}$$

For our application, we are only interested in the growth of the distance between two seeds as they are guided by the gradient field and therefore we omit the normalization by advection time and the logarithm in Eq. 5. Additionally, since we look for points on the isosurfaces, i.e. the intersections of isosurfaces with gradient field lines starting at the seeds, the integration along the field lines shall not be limited by integration time (or length). Instead, it has to be limited by the prescribed isolevel l. We define $\boldsymbol{\phi}^l(\mathbf{x})$ to be mapping the position $\mathbf{x}$ to the isosurface of isolevel l along the corresponding gradient field line (the field line is stopped if a critical point of the gradient field is reached). Using Eq. 5, this would lead to the separation factor

$$s^l(\mathbf{x}) = \sqrt{\lambda_{\max}(\Delta^l(\mathbf{x}))}\,. \tag{6}$$

We follow a direct approach for evaluating $s^l(\mathbf{x})$. To avoid numerical issues and to be able to exclude certain seeding neighbors $\mathbf{n} \in \mathcal{N}(\mathbf{x})$ of $\mathbf{x}$, as described below, the computation of $s^l(\mathbf{x})$ is not based on the gradient of $\boldsymbol{\phi}^l(\mathbf{x})$, but it is calculated directly:

$$s^l(\mathbf{x}) = \max_{\mathbf{n}\in\mathcal{N}(\mathbf{x})} \frac{||\boldsymbol{\phi}^l(\mathbf{n}) - \boldsymbol{\phi}^l(\mathbf{x})||}{||\mathbf{n}-\mathbf{x}||}\,. \tag{7}$$

The quad size $d(\mathbf{x})$ at $\boldsymbol{\phi}^l(\mathbf{x})$ is $s^l(\mathbf{x})$ times the corresponding seeding distance $||\mathbf{n}-\mathbf{x}||$. If the resolution of the object space sampling is appropriate, using these quad sizes guarantees a complete covering of the isosurface at level l. For obtaining quad sizes that are appropriate for any isolevel, the maximum separation $\hat{s}(\mathbf{x})$ over all isolevels, computed in a preprocessing step, is used to determine the quad sizes:

$$\hat{s}(\mathbf{x}) = \max_{l_{\min}\le l\le l_{\max}} s^l(\mathbf{x})\,. \tag{8}$$

This quantity relates to the FTLE maximum by Sadlo and Peikert [15]. Unfortunately, neighboring seeds can get mapped to different isosurface parts, making the resulting quad sizes too conservative. This leads to unnecessary overdraw during rendering and hence lowers the overall performance. As a remedy, we propose a heuristic that reduces the overdraw caused by these cases. The idea is to exclude neighbors of $\mathbf{x}$ from the computation of $\hat{s}(\mathbf{x})$, that are not located in its vicinity on the same isosurface. There are several possible approaches for detecting if two points are adjacent on the same isosurface. The straightforward approach would be to compute the geodesic distance between the points along the isosurface. However, this is considered impractical and too expensive.

A criterion for testing if a mapped point $\boldsymbol{\phi}^l(\mathbf{x})$ and its mapped neighbors $\boldsymbol{\phi}^l(\mathbf{n}), \mathbf{n} \in \mathcal{N}(\mathbf{x})$ are located close to each other on the same isosurface can be motivated by the fact that the gradient is always aligned with the isosurface normal. To test if a point and its neighbor are located on the same isosurface, the behavior of the gradient field is analyzed on the straight line connecting them. If the isosurface is planar between the two points, the line follows the isosurface and intersects the gradient field everywhere perpendicularly. If, on the other hand, the isosurface is non-planar between the points or if the points even lie on different isosurface regions, there are positions on the segment where the gradient is not perpendicular. We introduce the following measure $a_{\min}$ for the minimum angle between the gradient and the segment $\mathbf{s}_n$ going from $\mathbf{x}$ to $\mathbf{n}$:

$$a_{\min} = \max_{t \in [0,1]} |\mathbf{s}_n \cdot \nabla s(\mathbf{x} + t\mathbf{s}_n)| / (||\mathbf{s}_n|| ||\nabla s(\mathbf{x} + t\mathbf{s}_n)||) \,. \tag{9}$$

Then, the neighbor $\mathbf{n}$ is excluded from the computation if $a_{\min}$ exceeds a user-defined threshold. In practice, this threshold can be typically set to 0.5 for suppressing most of the erroneous neighbors in order to prevent too conservative quad sizes and reduce the performance loss due to overdraw.

5 Implementation

The design of iHOS was to take full advantage of parallel architectures. Although our current implementation runs on a single GPU, it could be easily distributed over a GPU cluster. We start by first describing the data structures used. Afterwards we go into the details of the implementation.

5.1 Data Storage

For the early cell culling we use an interval-tree [2] that stores the extrema of the scalar field for each cell. This tree is stored and processed on the CPU side. The remaining data structures are stored on the GPU through the use of different buffers. Each time the user selects a different isovalue, we need to load the initial set of uniformly distributed seeds and compute the isosurface approximation. Thus we allocate two seed arrays on the GPU: one with the initial unprojected seeds and another to store the projected seeds. For each unprojected seed, its corresponding FTLE value is stored as its w coordinate. Actually, those seed arrays are implemented as vertex buffer objects (VBOs). We refer to VBO_{unproj} for the VBO with unprojected seeds and VBO_{quads} for the VBO that stores the quads. Polynomial, gradient, and cell boundary data are read-only and are made available for GLSL through the use of bindable uniform buffer objects (BUBOs). Since BUBOs are limited in size (usually 64KB) we chose to create a separate BUBO for each cell and to bind them on demand.

5.2 GLSL Shaders

Both phases of the algorithm are implemented through a set of shader programs. The first step of the algorithm is implemented through a GLSL program composed of a vertex shader and a geometry shader. The vertex shader reads the VBO_{unproj} and projects the seeds onto the isosurface (Section 3.3). These seeds are streamed up to the geometry shader and used to generate surface tangent quads. The quads are resized accordingly to the FTLE value of the seed (Section 5.1). In order to record the generated quads into the VBO_{quads} we interrupt the pipeline just after the geometry shader through the use an OpenGL extension called *Transform Feedback* (or stream-out in DirectX terminology). The GLSL program responsible for the second step is composed of a vertex and a fragment shader. The vertex shader just redirects the vertices of the quads to the rasterization stage. The fragment shader implements the core of the second step. It performs a ray casting using the input fragment as the starting point for the root-finding iterations (Section 3.4). As said before, fragments that are not sucessfully projected are discarded. However, it must be observed that fragments successfully projected onto the isosurface are only shaded according to the new position, without having their depth coordinate updated. This procedure greatly increases method's performance by allowing the use of the early-z test. A side effect of this approach is that some noise can appear on some regions of the isosurface. This happens because a fragment closer to the camera can be projected on a far surface, being thus wrongly shaded. This effect is greatly reduced through the comparison between the angles of the polynomial gradient vectors at the fragment positions before and after projection. If the two angles differ above a certain value, it is assumed that the fragment jumped to another surface. In this case one have just to discard the fragment. In both steps, the shaders must compute several evaluations of the polynomials and their gradients. A shader capable of evaluating a general polynomial with an arbitrary degree should contain a loop that could not be unrolled by the compiler. To increase performance, we decided to keep several versions of these shaders, each one targeted to a specific polynomial degree. For the polynomial evaluation itself we use static expressions generated by a multivariate Horner scheme approach [1]. We decided to keep the number of integration and NR iterations static in order to reinforce thread synchronization, increasing performance.

5.3 Computation Flow

When the user chooses a desired isovalue the cells containing the isovalue are selected. These cells are inserted in a separate list and arranged in front-to-back order through fast sorting that does not have to be exact. This ordering is not necessary, but it helps the second step of the algorithm to avoid processing occluded fragments, increasing performance. The shaders corresponding to the polynomial degree of the current cell are activated. The corresponding BUBO (containing the polynomial, gradient, and boundary data) for the current cell is bound and the seeds are projected. After all seeds are projected, we start with the second step. Again, the cell list is traversed, and now the ray casting shaders and corresponding BUBOs are bound. Fragments successfully projected onto the isosurface are recorded in the frame buffer.

6 Results

Results and performance measurements were obtained on a computer with an Intel Core 2 Quad 2.4 GHz processor, with 4 GB of RAM, Linux operating system, and NVidia GeForce

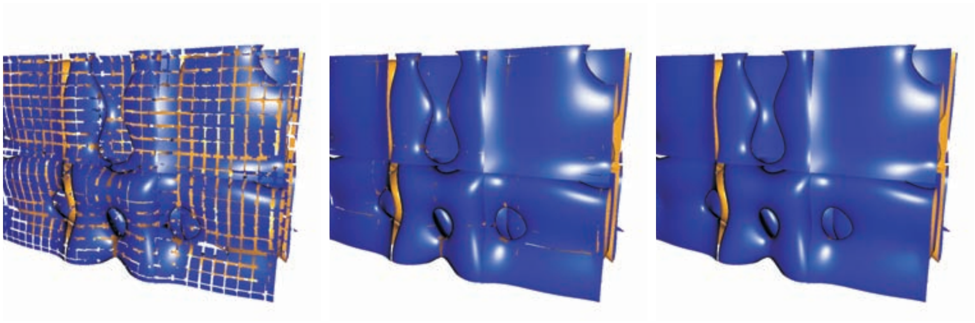


Figure 5 Renderings of isovalue ≈ 2 with varying q_s. From left to right, the s_q value and corresponding frames per second (fps) are: $s_q = 0.75$ and 45 fps, $s_q = 1.0$ and 39 fps, $s_q = 1.25$ and 29 fps.

8800 GTX video card. To demonstrate the method we used synthetic data and two density datasets generated by *hp*-adaptive Discontinuous Galerkin flow simulations. The scalar fields of all datasets are defined analytically by a multivariate polynomial defined in the reference space in a monomial basis of the form:

$$P(\mathbf{x}) = \sum_{i+j+k<=n} c_{ijk} x^i y^j z^k \tag{10}$$

with i, j, $k \in \mathbb{N}_0$, the maximum order n, the coefficients c_{ijk}, and the reference space coordinates x, y, z. The mapping functions for all datasets include only translations. The smaller dataset is composed of 119 hexahedral cells with polynomial degrees varying between 5 and 6. The larger dataset is composed of 34,535 convex polyhedral cells (tetrahedra, prisms, and hexahedra) with a scalar field defined by degree 3 polynomials.

6.1 Performance

For each phase of the algorithm a set of parameters can be adjusted. For the first phase these parameters are the number of initial seeds (n_s), number of integration steps (n_i), number of NR steps (n_{N1}), and the error for the NR projections (ϵ_1). For the second phase the parameters that can be adjusted are the number of NR steps (n_{N2}), the scale factor for the quad size (s_q), and the error for the NR projections (ϵ_2). Although the great number of parameters may be confusing at first, they offer great flexibility to the user, allowing the tuning towards accuracy or interactivity. From our experience we observed also that in general one can explore both accuracy and performance by adjusting just a few of them, keeping the others fixed. For all presented measurements we used screen resolutions of 1024^2 pixels.

All renderings were made with active early cell culling and with front-to-back ordering for the cells. Figure 5 shows performance measurements for the small dataset. The fixed parameters, and their respective values, are: $n_s = 10^3$, $n_i = 50$, $n_{N1} = 10$, $\epsilon_1 = 10^{-3}$, $n_{N2} = 3$ and $\epsilon_2 = 10^{-4}$. These values were chosen through trial-and-error. The only variable parameter is s_q. Due to early cell culling only 16 cells (≈ 7.43% of the total) were processed. The results of Figure 6 show performance measurements for the bigger dataset. The fixed parameters are: $n_s = 8^3$, $n_i = 50$, $n_{N1} = 10$, $\epsilon_1 = 10^{-3}$, $n_{N2} = 3$ and $\epsilon_2 = 10^{-4}$. These values were also chosen through trial-and-error. Again, the only variable parameter is s_q. Due to early cell culling 3,781 cells (≈ 9.13% of the total) were processed.

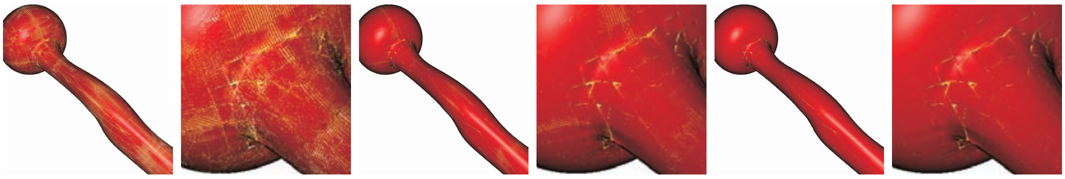


Figure 6 Renderings of isovalue ≈ 0.9983 with varying s_q. From left to right, rendering and zoomed images with the following s_q value and corresponding frames per second (fps): $s_q = 0.5$ and 21 fps, $s_q = 0.75$ and 14 fps, $s_q = 1.0$ and 12.5 fps.

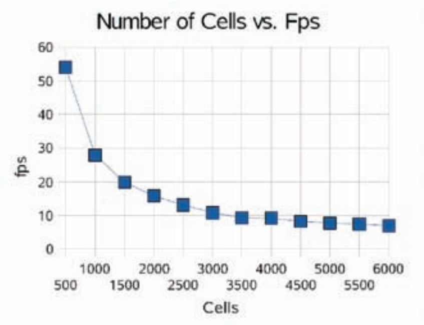


Figure 7 Performance vs. number of cells. The non-linearity can be explained by the use of the early-z test, which avoids the processing of occluded fragments.

Figure 7 shows how the method scales with respect to the number of cells. These measurements were made with degree 3 polynomials extracted from the bigger dataset. It can be seen that the rendering cost does not grow linearly. This can be explained by the use of the early-z test, which avoids the processing of fragments or even cells that are totally occluded. This behavior is reinforced since we use front-to-back ordering for the cells. We may still have some overdraw since we do not order the quads inside the cell. However, from our experiments, this has not affected significantly the performance in the general case.

The method proposed by Knoll *et al.* [6] may look, at first, as a competing approach. Despite the fact that their method is meant for the robust rendering of implicits, it is fast and could be applied separately to each cell of our datasets. Figure 8 shows a comparison between our method and Knoll's for the rendering of a single cell of our small dataset. Visual inspection shows that our method is able to achieve higher framerates at better image quality. One reason for the difference in performance is probably related to the extra cost involved with the use of inclusion arithmetic in Knoll's method. Although the use of inclusion arithmetic guarantees error bounds for the results, it tends to produce inferior results if larger error bounds are used. Since we are not primarily interested in the control of accuracy, we rely on the direct evaluation of the original polynomial representation, which is computationally cheaper and thus can be used in a previewing system.

Regarding the time spent to recompute the object space approximation at each isovalue change, for all our tests, they took less than 2 seconds. In the case of the smaller dataset, it takes less than 1 second to recompute the seed projection. All measurements were made with 50 integration iterations and 10 NR iterations.

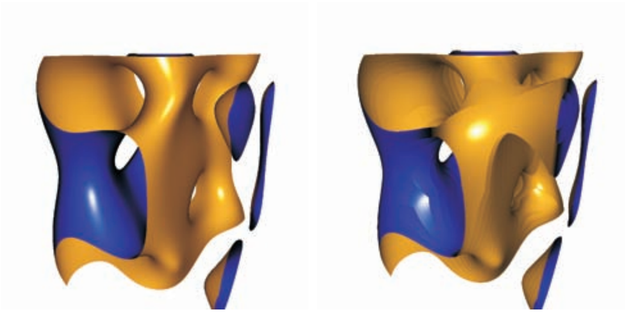


Figure 8 Comparison with Knoll's method. (left) Polynomial of degree 5 rendered by our method at ≈ 30 fps. With the standard settings (depth = 10 and error = 0), Knoll's method rendered this image at ≈ 8 fps without visible artifacts. (right) After optimization of their parameters (depth = 4 and error = 0) to reach the best possible image quality at ≈ 30 fps. Both images rendered in a 1024^2 pixel screen.

6.2 Isocontouring Quality

As a previewing system the proposed method does not focus on high accuracy. Nevertheless, it is capable of delivering results of reasonable quality. Figure 9 shows a comparison between images generated by our method and the same images generated by POV Ray, a well known, tested, and freely available ray tracer [12]. The discontinuities observed in these renderings are due the Discontinuous Galerkin method used to compute the simulations. Our method handles continuous data properly, as can be seen in Figure 10.

Despite the reasonable quality of the images, one may want to have better visual quality for the isocontouring. In such cases the user can switch to a robust isocontouring system, as the one proposed by Knoll *et al.* [6]. However, we have also developed some heuristics in order to allow our method to generate better quality images. The quality of the final sampling depends on the relation between the number of seeds and the size of the quads. The exact relation between these two parameters is difficult to assess since the projection changes seed density. The FTLE-based heuristic tries to estimate "optimal" quad sizes that

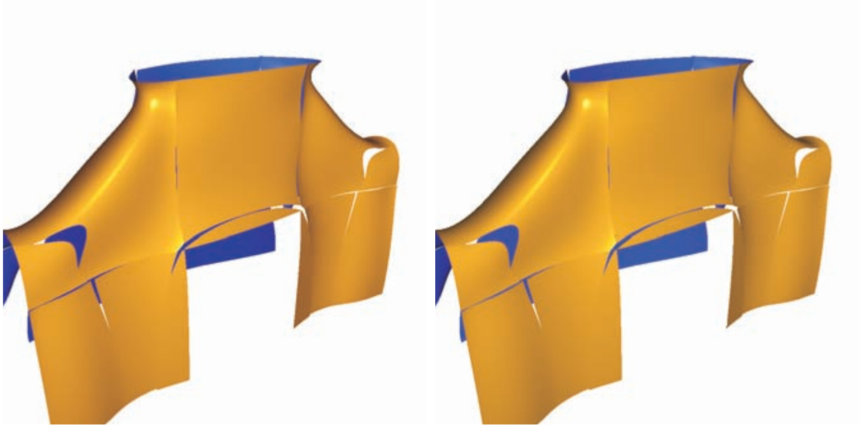


Figure 9 Comparison against POV Ray. (left) Image generated with our method, with the following settings: $n_s = 10^3$, $n_i = 50$, $n_{N1} = 10$, $\epsilon_1 = 10^{-3}$, $n_{N2} = 3$, $\epsilon_2 = 10^{-4}$, $s_q = 1.5$ and isovalue ≈ 5.8437. This scene was rendered at 8.7fps. (right) Reference image generated from the same dataset and same isovalue with POV Ray. Both images rendered at 1024^2 pixels.

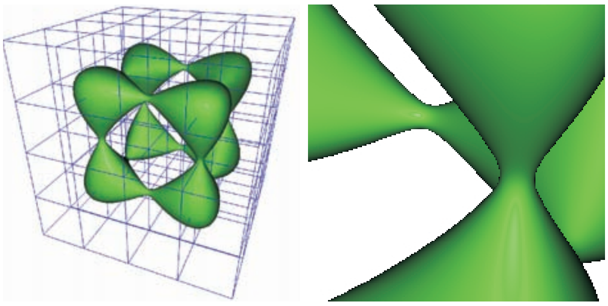


Figure 10 Rendering of Tangle dataset sampled with a grid of $4^3 = 64$ cells (left). The dataset is continuous inside and across cells. Zoomed image (right) shows how our method handles the continuity at the borders correctly. Settings for this rendering are: $n_s = 10^3$, $n_i = 50$, $n_{N1} = 10$, $\epsilon_1 = 10^{-3}$, $n_{N2} = 3$, and $\epsilon_2 = 10^{-4}$. The isovalue is ≈ -1.98. Image was rendered at 1024^2 with ≈ 9.6 fps.

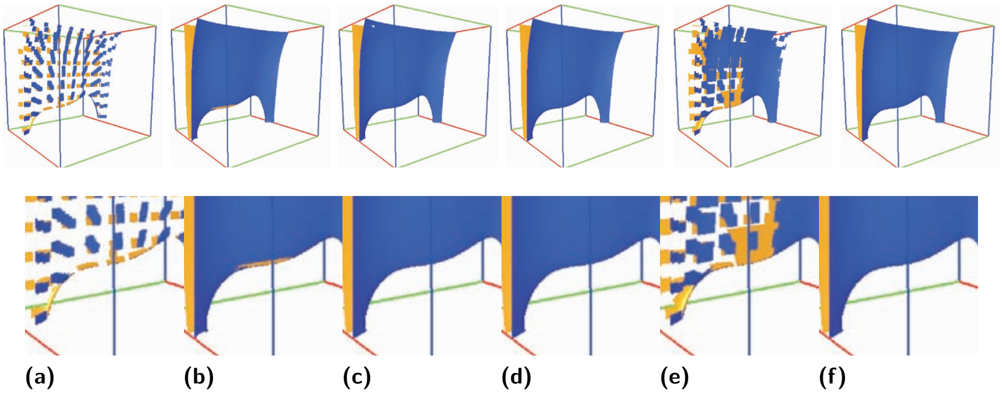


Figure 11 Sampling issues related to quad size: (a) Standard quads reduced to 30% of original size. (b) Sampling using standard quads: some artifacts can be seen due to poor sampling. (c) Quads scaled by the FTLE and reduced to 30% of their size. (d) Sampling obtained with quads resized by the FTLE approach. (e) Quads resized by our FTLE approach in conjunction with a heuristic that reduces its conservativeness, reduce in 30%. (f) Sampling obtained with our FTLE approach in conjunction with the heuristic.

“work” for all isovalues considering an initial seeding given by the user. Since the FTLE does not take into account the topology of the isosurface, it can be too conservative, by considering distances between seeds in disjoint surfaces. For this case we adopt a test that estimates whether points are in disjoint surfaces (Section 4.2). Figure 11 illustrates the sampling problems related to bad seeding/quad size relation and how our heuristic resizes the splats according to that seeding, reducing effectively the artifacts in the final image. It also illustrates the effect of our disjoint-surfaces test on reducing the FTLE conservativeness. Despite the fact that it improves the quality of the final rendering, the FTLE makes the method less attractive for previewing since it reduces its performance.

7 Conclusion and Future Work

This work has presented iHOS, an interactive approach for isocontouring higher-order data generated by *hp*-adaptive discretization methods. The method operates with high-order data whose mapping function includes only rotation, translation, reflection, and scaling. The method does not rely on complex data structures and does not resample the original data into lower-order representations. Results are presented for datasets composed by convex cells. However, since no cross-cell rays are used, the method can be easily adapted to work with concave cells through the use of a more general point-in-polyhedron test. Although not meant to generate accurate images, the presented results show that general quality of the generated images is good. Several parameters can be adjusted in order to control the trade-off between efficiency and accuracy; it has been shown that one can explore both (accuracy and performance) by adjusting just a few parameters, keeping the others at fixed values. Some points should be addressed in future work: further reduction of the overdraw during quad rasterization, development of an improved initial seed distribution heuristic for the first phase, handling of large datasets, optimizations for subpixel cells and handling of non-linear mapping functions between reference and world spaces. For future work we also plan to adapt this method to handle dynamic datasets.

Acknowledgements

We thank our colleagues from the Institut für Aero- und Gasdynamik from Universität Stuttgart, Germany, for their continuous support and for providing datasets. We also thank Aaron Knoll for kindly providing code for performance tests and comparisons, and the anonymous reviewers for their comments and insightful suggestions. C.A. Pagot would like to acknowledge the support of CAPES-Brazil (Probral 3192/08-3) and CNPq-Brazil (140238/2007-7).

References

1 Martine Ceberio and Vladik Kreinovich. Greedy algorithms for optimizing multivariate Horner schemes. *ACM SIGSAM Bulletin*, 38(1):8–15, 2004.

2 P. Cignoni, P. Marino, C. Montani, E. Puppo, and R. Scopigno. Speeding up isosurface extraction using interval trees. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):158–170, 1997.

3 Luiz Henrique de Figueiredo, Jonas de Miranda Gomes, Demetri Terzopoulos, and Luiz Velho. Physically-based methods for polygonization of implicit surfaces. In *Proceedings of Graphics Interface '92*, pages 250–257, 1992.

4 B. Haasdonk, M. Ohlberger, M. Rumpf, A. Schmidt, and K. G. Siebert. Multiresolution visualization of higher order adaptive finite element simulations. *Computing*, 70(3):181–204, 2003.

5 G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D*, 149(4):248–277, 2001.

6 Aaron Knoll, Younis Hijazi, Andrew Kensler, Mathias Schott, Charles D. Hansen, and Hans Hagen. Fast ray tracing of arbitrary implicit surfaces with interval and affine arithmetic. *Computer Graphics Forum*, 28(1):26–40, 2009.

7 Yuri Levin and Adi Ben-Israel. Directional Newton methods in n variables. *Mathematics of Computation*, 71(237):251–262, 2002.

8 William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.

9 E. N. Lorenz. A study of the predictability if a 28-variable atmospheric model. *Tellus*, 17:321–333, 1965.

10 M. Meyer, B. Nelson, R. M. Kirby, and R. Whitaker. Particle systems for efficient and accurate high-order finite element visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1015–1026, 2007.

11 Blake Nelson and Robert M. Kirby. Ray-tracing polymorphic multidomain spectral/hp elements for isosurface rendering. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):114–125, 2006.

12 Povray. Persistence of Vision Raytracer. Website, 2009. `http://www.povray.org`.

13 W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

14 B. Remacle, N. Chevaugeon, É. Marchandise, and C. Geuzaine. Efficient visualization of high order finite elements. *International Journal for Numerical Methods in Engineering*, 69(4):750–771, 2006.

15 Filip Sadlo and Ronald Peikert. Efficient visualization of Lagrangian coherent structures by filtered AMR ridge extraction. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1456–1463, 2007.

16 William J. Schroeder, Francois Bertel, Mathieu Malaterre, David Thompson, Philippe P. Pebay, Robert O'Bara, and Saurabh Tendulkar. Framework for visualizing higher-order basis functions. In *Proceedings of the IEEE Visualization Conference*, pages 43–50, 2005.

17 William J. Schroeder, Francois Bertel, Mathieu Malaterre, David Thompson, Philippe P. Pebay, Robert O'Bara, and Saurabh Tendulkar. Methods and framework for visualizing higher-order finite elements. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):446–460, 2006.

18 Kees van Kooten, Gino van den Bergen, and Alex Telea. *GPU Gems 3*, chapter Point-based visualization of metaballs on a GPU. Addison-Wesley,Nguyen, Hubert, 2007.

19 David F. Wiley, Hank Childs, Bernd Hamann, and Kenneth I. Joy. Ray casting curved-quadratic elements. In *VISSYM'04: Proceedings of the Symposium on Data Visualisation*, pages 201–210, 2004.

20 Andrew P. Witkin and Paul S. Heckbert. Using particles to sample and control implicit surfaces. In *Proceedings of ACM SIGGRAPH 1994*, pages 269–277, 1994.

21 Yuan Zhou and Michael Garland. Interactive point-based rendering of higher-order tetrahedral data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1229–1236, 2006.

HCI in Medical Visualization

Bernhard Preim[1]

1 **University of Magdeburg, Department of Simulation and Graphics**
39106, Magdeburg, Universitätsplatz 2, Germany
bernhard.preim@ovgu.de

Abstract

Research in medical visualization lead to a remarkable collection of algorithms for efficiently exploring medical imaging data, such as CT, MRI and DTI. However, widespread use of such algorithms requires careful parameterization, integration of individual algorithms in *solutions* for real-world problems in diagnosis, treatment planning and intraoperative navigation. In the field of HCI, input devices, interaction techniques as well as a process for achieving usable, useful, and attractive user interfaces are explored. Findings from HCI may serve as a starting point to significantly improve visual computing solutions in medical diagnosis and treatment. We discuss general issues, such as input devices for medical visualization, and selected examples.

1998 ACM Subject Classification I.3 Computer graphics, I.3.3 Picture, Image Generation, I.3.6 Methodology and Techniques, J.3 Life and medical sciences

Keywords and phrases Medical visualization, Human Computer Interaction, Input devices, Scenarios

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.292

1 Introduction

The visualization and exploration of 3D medical image data, such as CT, MRI, ultrasound or PET, is an important application area of scientific visualization. Developments in these areas drove research in volume visualization, mesh processing, tensor visualization and image filtering. With new and improved imaging modalities along with changes in the available graphics hardware there is still a need for improved algorithms with respect to visualization quality, accuracy and performance. However, there is an important trend towards more applied research focussing on specific applications, to provide visual computing solutions integrating image analysis, visualization and appropriate graphical user interfaces. Indeed, these issues are essential to achieve clinical use of medical visualization algorithms. The goal of regular clinical use requires to put the user in focus or – as HCI (Human computer interaction) researchers would call it – to adopt a user-centered design approach [33]. Users of medical visualization systems are primarily medical doctors from a specific discipline, such as radiology, surgery, nuclear medicine or radiation treatment, who use such tools for diagnosis support, treatment planning, intraoperative navigation and follow-up studies to evaluate the success of their treatment. In this article, we focus on these user groups. Thus, we do not address students of medicine who might use advanced 3D visualization for anatomy teaching, or researchers in medicine, biology and chemistry who optimize medical imaging with respect to sequences, protocols and contrast agents. The focus on medical doctors in clinical settings is justified, since this group is by far larger than medical researchers and students, and if these users can be provided with better tools, a direct and significant impact on patient health is possible. It is essential to be clear about the user group: The working place of a medical doctor in a hospital and particularly in the operating room differs strongly

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 292–310
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

from the office of a researcher or the desk of a student. Thus it is naive to assume that computer support developed for the latter is also appropriate for the first user group.

To discuss these issues, we need to dive into human computer interaction, the discipline which deals with the process of generating usable and enjoying software systems directly supporting specific users and their tasks. Since HCI itself is a huge discipline, we focus on a few areas within HCI which are particularly relevant for the clinical use of medical visualization. These include:

1. modern task analysis methods, which carefully incorporate the context of use, usage scenarios, preferences and acceptance criteria,
2. an appropriate and refined use of input devices and input options, such as pen, touch display and, gestured input,
3. strategies and recommendations which improve the user experience [1],
4. an adequate use of the large variety of displays ranging from handheld devices to very large screens, and
5. prototyping and evaluation techniques which support the exploration of a wide variety of techniques, options and combinations.

After an overview of medical visualization (Sect. 2), we discuss task analysis (Sect. 3) and input devices (Sect. 4) in more detail and relate them to medical visualization in clinical medicine. We continue with a discussion of evaluation strategies (Sect. 5).

2 Medical Visualization: From Pure Research to Advanced Applications

Pioneering work in medical visualization was aimed at efficiently displaying a set of medical image data with direct volume rendering or isosurface rendering incorporating shading, efficiently using graphics hardware and improving image quality, for example with pre-integrated shading. Another branch of research focussed on non-photorealistic rendering, such as incorporating hatching to convey surface curvature [42]. NPR techniques have later been adapted and used in volume rendering [6]. With these and other developments a large set of powerful rendering techniques is available which provides – from an application's point of view – a sufficiently good performance and quality. However, basic algorithms alone do not provide adequate support for clinical tasks. Medical doctors often need to

1. follow an inner path of an elongated branching structure (e.g. airway tree, vascular structure, colon) to understand the branching pattern before an endoscopic intervention or to detect abnormalities along the wall,
2. closely examine the local surrounding of pathologic structures,
3. explore possible resection areas and resection planes [52], and
4. integrate the results from different examinations, such as fMRI and DTI.

We discuss general strategies of providing useful and appropriate solutions in the following subsections.

[1] The user experience (UX) extends the older term *usability* and covers also perceived attractiveness. Amongst others, a distinct visual design, typography, the careful use of colors, shapes, and animations contribute to the user experience [7].

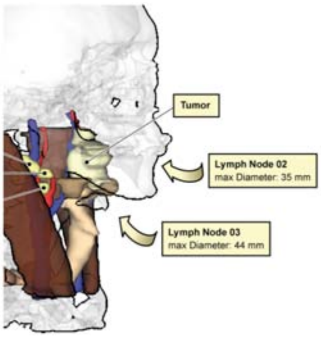


Figure 1 Labelled visualization for neck surgery planning. All labels relate to potentially pathologic structures and are placed automatically. Hidden structures are annotated using a bended arrow. Thus, the user knows where to look for further critical and important structures. Clicking on the annotation starts an animation that leads to a good viewpoint on the structure. (From: [30])

Incorporation of semantic information

A more user-centered approach was first adopted to support anatomy and surgery education [21, 36]. We do not dive into such applications, but a few trends emerging in teaching applications are relevant for the purpose of this paper. In particular, medical image data was connected with a large variety of semantic information which enables the user to explore data with respect to organ systems, nerve supply and other application-specific questions [45]. Based on such semantic information, anatomic structures may be labelled automatically [17]. Labelling is also useful for surgical planning (see Fig. 1 for an example carefully discussed with a surgeon).

The extended use of semantic information is an important trend in other areas within medical visualization. Importance-driven rendering relies on a priori knowledge of the relevance of certain structures [51] and transfer function specification benefits from a user-centered approach and "knowledge" of the actual data distribution as well as frequently used settings [38]. In clinical settings, DICOM data are employed which come along with a variety of semantic information, e.g. with respect to data resolution, specific scanning parameters, which may be employed for annotation and parameterizing algorithms as well.

The essential role of segmentation

To provide advanced support, relevant structures have to be identified and delineated (to be *segmented*). Segmentations are used to label or emphasize them for example with importance-driven rendering. Segmentation is beyond the scope of this paper. However, similar to medical visualization, there is a trend to focus on HCI aspects in segmentation as well instead of purely developing highly sophisticated model-based approaches (the seminal paper of Olabarriaga and Smeulders [34] laid the foundation for a systematic exploration of interaction techniques in segmentation). Recently Heckel et al. [19] introduced a promising method for semi-automatic 3D segmentation, where users may provide input in arbitrary orientations. While many research papers are based on a comprehensive segmentation of many structures, in clinical medicine, segmentation is performed rather rarely due to the effort involved. Exceptions are advanced applications in cardiology and orthopedics as well as radiation treatment planning where segmentation of the target structure (a cancer) and surrounding vital structures is mandatory for optimizing the treatment plan.

2.1 Virtual endoscopy

With respect to specific applications, virtual endoscopy, in particular virtual colonoscopy for colon cancer screening, often serves as a showcase task, since it is highly relevant and convincing solutions have been developed [22, 2]. The success of virtual endoscopy depends strongly on application-specific aspects, such as an emphasis of suspicious structures by advanced image analysis [53], adequate overviews which prevent that crucial information is missed [50], and the presentation of actually hidden anatomic structures relevant for the particular intervention [31]. Virtual endoscopy is also an excellent example, for the successful use of a *metaphor*. This relates to the term "endoscopy", that is very familiar to medical doctors and to the specific visualization, e.g. the implementation of various lens characteristics, that mimic real endoscopy. We will discuss metaphors in Sect. 3.4.

2.2 Integration of image analysis, simulation and visualization

Other applications also require substantial image analysis prior to a visualization and exploration of patient-specific models. As an example, [46] describes the image analysis pipeline necessary for delineating all structures relevant for liver surgery planning. In a similar way, neck surgery planning [27] requires segmentation of many different soft-tissue structures, such as muscles, glands, and lymph nodes. Advanced visual solutions for surgical planning often include simulations as well. As an example, Zachow et al. [52] simulated soft-tissue deformations for different variants of facial surgical interventions. Similarly, Krekel et al. simulated the range of motion of a shoulder implant to guide shoulder replacement [25]. The most advanced visual support is now available in neurosurgery planning applications, where anatomic, functional and DTI data are combined [12, 4] and sophisticated visualization techniques, such as depth-enhancing illustrative techniques, are used [39]. Enders et al. [12] introduce a visualization of DTI fiber bundles which is carefully adapted to the information needs of a neurosurgeon with respect to planning tumor surgery. All systems, mentioned in this subsection, are good examples for carefully incorporating HCI aspects and focussing on real needs and problems documented also with expressive evaluations.

2.3 Visual exploration of 3D models for neck surgery planning

We will explain the example of neck surgery planning in more detail. The in-depth exploration of enlarged lymph nodes – potential lymph nodes metastases – and the primary tumor with respect to its shape, size, location and surrounding structures represents the core of preoperative planning. Specific solutions, such as cutaways for emphasizing lymph nodes, and silhouettes for presenting context structures turned out to be useful in clinical tests. However, our general approach of presenting as many anatomic structures as possible in a convenient way, was eventually considered not appropriate, since it does not fit well to the specific questions of surgeons. In extensive discussions we learned how careful possible infiltrations are assessed. Surgeons want to know whether there is an infiltration of a vascular structure for example, how likely the infiltration is, which portion along the vessel and which portion of the vessel's circumference is involved. The answer to each of these questions may alter the surgical strategy considerably. Therefore, we carefully designed a workflow with a sequence of 3D models to be used to assess the infiltration of different anatomic structures. Figure 2 shows two examples of such specific visualizations. We cannot claim that these are optimal visualizations for this purpose, but at least we have identified the visualization of (potential) infiltrations as an important research topic, probably relevant for a wide spectrum of surgical interventions.

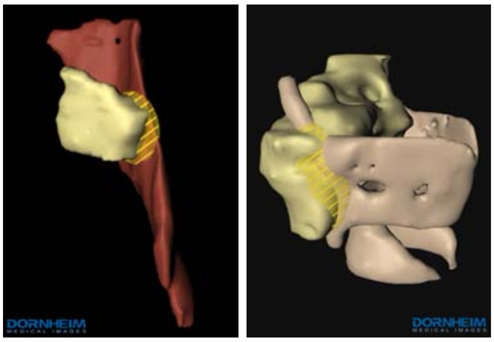


Figure 2 Specific visualizations of the neck anatomy. Both images indicate the possible infiltration of an anatomic structure by a lymph node metastasis (left: a large muscle, right: thyroid cartilage). The possible infiltration area is semitransparently visualized and hatched. (Copyright: Dornheim Medical Images)

2.4 Discussion

Successful applications require to deeply understand the characteristics of the underlying imaging data, the variety within such data, including pathologic situations and the specific diagnostic and treatment planning questions. From the huge variety of options to display such data, appropriate default settings are necessary to support users working under severe time pressure adequately. These default settings relate to colours, rendering styles, transparency and viewing directions. More often than not, different views to the data need to be carefully combined and synchronized, such as internal and external 3D views along with cross-sectional views in virtual endoscopy. Moreover, in diagnosis but also in treatment planning it is often essential to understand existing workflows (without advanced computer support). This enables to envision new workflows which are as as close as possible to the original workflow. In tumor surgery, for example, the location, shape and size of a tumor needs to be explored before possible infiltrations of important risk structures are analyzed in detail. We will discuss workflow analysis in Sect. 3.2.

3 User and Task Analysis

When research and development is indeed targeted at clinical applicability, user and task analysis are key elements. The failure of many attempts to create useful systems for clinicians is often largely based on an incomplete task analysis, where major requirements were not identified or their priority was underestimated. The naive approach to ask the users, what they need, may be a reasonable start, but there are several reasons why the immediate answer to this question is completely insufficient. Typical users have no idea what could be done with adequate technical support, they are accustomed to certain kind of technology and try to cope with it. Thus, user needs have to be very carefully elicited. HCI experts, specialized on this activity, are referred to as *user researchers*[2], a term which illustrates how complex, challenging and creative task analysis actually is.

Modern task analysis combines a variety of methods including observations, interviews and questionnaires. As a result, a hierarchical task analysis (HTA), workflow diagrams, or a set of (informal natural language) scenario descriptions eventually enriched with digital

[2] Very often, these persons have a Phd in psychology.

photos from important artifacts arise. Diagnostic and treatment planning tasks are complex and demanding. It is therefore necessary to carefully prepare the analysis in order to pose the right questions and follow-up questions to reveal the implicit knowledge of medical experts.

3.1 What has to be analyzed?

The details of a particular treatment are, of course, different. However, a few general questions may serve as orientation:

- Which pathologies should be diagnosed or treated?
- Which imaging modalities are used either in isolation or combination for diagnosis or treatment planning?
- How is the pathology described and which alternative pathologies are considered (confirm or exclude a certain diagnosis)?
- How is the severity of the disease described and which criteria are employed for this description?
- What are the therapeutic consequences of the diagnosis?
- Which treatment options exist, e.g. surgery, radiation treatment, interventional treatment? How can they be combined?
- Which criteria drive the decision for these treatment options?
- Which further details have to be determined prior to surgery or intervention, e.g. access path for a cathether or stent, extent of a surgical resection, necessity of vessel reconstructions?
- Who is involved in these decisions?
- What kind of technical support is used during the intervention, e.g. navigation or surgery assistance systems?
- Which decisions have to be performed during an intervention?

It is crucial to understand these questions, to verify the answers by discussing with several medical doctors, and to discuss the results of your analysis with them. More often than not, it turns out that some facts have been confused or the relevance of some aspects is not correctly understood. As a consequence, the computer support should focus on generating visualizations which support diagnostic or treatment decisions directly. Later in a project, evaluations should focus on the influence of computer support on these questions. Our experience indicates that observations at clinical workplaces are a mandatory aspect of task analysis.

3.2 How to represent the results?

Task analysis yields a wealth of data which needs to be structured, prioritized and consolidated before concise results can be extracted. Audio recordings from interviews or "think aloud" sessions, handwritten notes, schematic drawings of workplaces or tasks are typical examples. Recently, two different representations have been used and refined for medical visualization applications: workflows and scenarios.

Workflow analysis and redesign is a core activity in business informatics where business processes should be designed, evaluated and optimized. Workflows are formal graph or network representations which contain actions (nodes in the graph) and their logical sequence (edges in the graph). The design of medical visualization may borrow from these experiences, notations and tools to manage such workflows to characterize treatment planning, interventional procedures and outcome control. Workflows may contain (a few) variants and may emphasized typical sequences of actions. They may also encode how often certain procedures occur, and

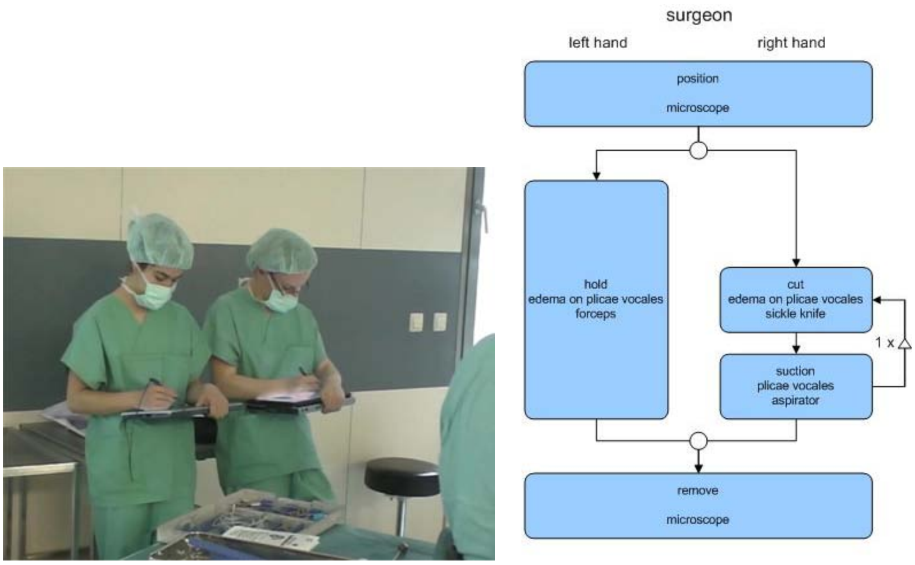


Figure 3 A workflow for a surgical intervention in cardiology resulting from careful observations (left with a tablet PC and dedicated software) in the operating room. (Courtesy of Thomas Neumuth, ICCAS Leipzig)

how long they are [32] – an information which is essential to consider which processes may improve from computer support. Workflows are described after observing several instances of a process (see Fig. 3). They may describe processes at various levels, thus allowing an analyis at different levels of granularity.

The formal character of this representation is a benefit which clearly supports the software development process. However, since this notation is not familiar to medical doctors, workflows are not particularly useful for discussions with them. Also, at different sites or even among different doctors at one side, there might be huge differences in their specific workflows. Unlike in manufacturing and administrative procedures, medical treatment is and must be more individualized with respect to the patient and the medical doctor. Workflow diagrams can hardly represent that variability but are often restricted to a somehow averaged instance.

Scenarios are now widely used in HCI, in particular to characterize and envision radically new software systems [3]. Scenarios are natural language descriptions which include statements about which technology or feature is used for which purpose. They contain different perspectives as well as motivations from users. Scenarios are more open to interpretation, which may be considered as a drawback. However, they are clearly useful as a basis to discuss with medical doctors. For three larger projects in computer-assisted surgery, scenario descriptions have been used, discussed within the development team and with medical doctors resulting in a large corpus of descriptions, annotations and refined descriptions [10]. Figure 4 shows different types of scenarios and their relations as they have been used for liver surgery training, neck surgery planning as well as minimally-invasive spine surgery. For a detailed discussion of these scenario types, see [10].

The following is a short portion from a longer user story and the derived scenarios for a SpineSurgeryTrainer (see Fig. 5 and [11]):

User Story: ?The doctor in training has to place an injection in the area of the cervical spine for the first time. He is insecure and he wants to train this procedure to test his skills

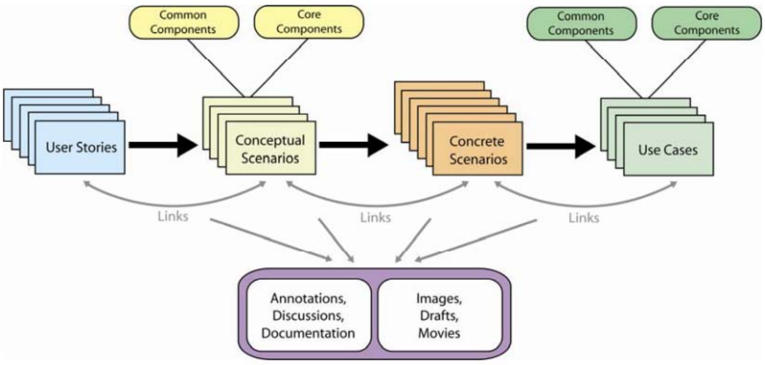


Figure 4 To envision a planned application, high-level user stories are stepwise refined by providing detail on *how* a function should be performed, and by considering constraints from the context of the intended system use. The links between the documents and the related annotations need to be managed. (From: [10])

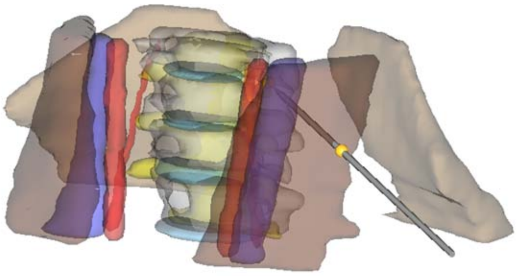


Figure 5 A screenshot from a training system for minimally-invasive spine surgery. A difficult decision relates to the access path and needle placement. (From: [11])

and to do the real injection with self-confidence. But there is no expert and no cadaver available at the moment. Because he is pressed for time and wants to start the training directly, he decides to train the injection virtually. . . . ?

Conceptual Scenario: ?He starts with the survey of the patient data and anamnesis. After that, he decides for an injection as therapy and starts the training of the virtual placement of the needle [Concrete Scenario 1] based on the MRI data and the 3D model of the patient's anatomy. . . . ?

Concrete Senario 1: (Details of injection planning): ?With the mouse (left mouse click) he defines one marker for the penetration point and one for the target point of the needle in the 2D data. The needle takes up its position. In an animation the user can view the injection process of the needle to his defined position. . . . ?

In total, six scenarios related to cases with different levels of difficulty and different viable treatment options have been explored. The discussion of such scenarios with medical doctors revealed a variety of insights and ideas for visualization and exploration of the data, in particular when the decision between two alternative therapies depends on subtle details of the patient anatomy.

3.2.1 Experiences with the Use of Scenarios

The visualization group in Magdeburg employed scenario descriptions consequently for several comprehensive projects. The use of scenarios, in particular the discussions with medical doctors, lead to several unexpected ideas and features. As an example, it turned out that our NeckSurgeryPlanning-system is also relevant for patient consulting, where surgical options are explained to the patient and to family members. For this purpose, a large display device is useful and the set of available features may be strongly reduced. An essential lesson learned in these discussions is that user stories need to be combined with sketches, screenshots of a mock-up or storyboards to further strengthen the imagination of medical doctors. The purely textual character of all scenario types does not sufficiently support the discussion of the strongly visual components of a diagnostics and treatment planning systems. Fig. 5 is an example, which was created to support the reflection on the previously described user stories.

3.2.2 Combination of scenarios and workflows

A development team need not to decide whether (exclusively) workflows or scenarios should be used to guide the development process. Both methods provide useful and complementary information. While scenarios better support the discussion between user researchers and the target users, they do not inform the actual developers in a concise manner. For the developers, a validated workflow description is a valuable support, in particular for implementing wizard-like systems which guide the user in a step-by-step manner. The systems developed in Magdeburg were also based on workflow descriptions at different granularities. Surgical planning, for example, at the highest level follows often the workflow: diagnosis, assessment of the general operability (Can the patient tolerate anesthesia?, . . .), resectability (Is the pathology accessible and may be removed without damage of vital structures?), access planning, in-depth planning including vascular reconstructions.

3.3 Understanding the User

This stage in a user interface lifecycle aims at understanding users' qualifications, preferences, needs and attitudes in order to create solutions which are acceptable and appropriate for them. In medical visualization, users are primarily radiologists, radiology technicians, medical doctors from different operative subjects, such as orthopedics, neurosurgery, or urology.

There are significant differences between radiologists and medical doctors from operative disciplines. While the former use the computer for a large part of their work, the latter consider their cognitive and manual skills to perform surgery as the core of their activity and use the computer only for a small portion of their work, often considering this work as less important. This difference has huge consequences for what is considered as appropriate visualization and interaction technique and user interface. While radiologists prefer a very efficient interaction even at the expense of more complexity and a longer learning period, doctors in operative subjects prefer simple easy-to-use interfaces even at the expense of longer interaction sequences and reduced flexibility. Therefore, radiologists (and medical doctors from related disciplines as nuclear medicine and radiation treatment) efficiently use systems with rather dense user interface panels, invisible interactions, such as short-cuts, popup-menus and other interaction facilities that only appear in a certain context. On the other hand, doctors in operative disciplines favor simplicity and thus prefer strongly reduced interaction with only a few large buttons at the same time. For radiologists it is essential that they can stay focussed on a certain region in a 2D or 3D visualization while performing changes on the visualization parameters, such as brightness, contrast, transfer function, or the currently selected slice. Thus, they prefer in-place interaction with mouse movements, such as scrolling through the slices with mouse wheel and changing brightness/contrast with left/right up/down movements. Interfaces for surgeons perform the same task with a control panel, where (large) sliders enable control of these parameters.

3.4 Metaphors

The identification and use of appropriate metaphors is an essential aspect of a user-centered process. Beyond requirements, scenarios and workflows, the user and task analyis *may* elicit suitable metaphors. The suitability of metaphors depends on the familiarity of users, the structure and richness of the metaphor (what do people associate with a metaphor?) and the degree of correspondence between the source domain (where the metaphor is known) and the target domain (the new application where the metaphor is employed to label and visually illustrate application concepts). Successful applications of metaphors in medicine are virtual endoscopy (recall Sect. 2.1), digital microscopy (a metaphor for designing solutions for pathologists), the digital lightbox (a general metaphor for radiology workstations, particularly for X-ray based image analysis). The further study the use of metaphors in science and in interactive system the following sources are recommended [5, 8, 14, 28].

4 Input Devices

Input and output devices play an essential role for the usability of medical visualization systems. There is a large variety of input and output devices, potentially relevant for medical visualization applications.[3]. We focus here on input devices because there is considerable more experience documented in scientific publications. In the future, however, autostereoscopic displays and mobile devices need to be carefully analyzed with respect to their potential for medical visualization.

[3] The virtual autopsy table with multi-touch input is an inspiring example, see the TED talk: Visualizing the medical data explosion at
http://www.ted.com/talks/anders_ynnerman_visualizing_the_medical_data_explosion.html

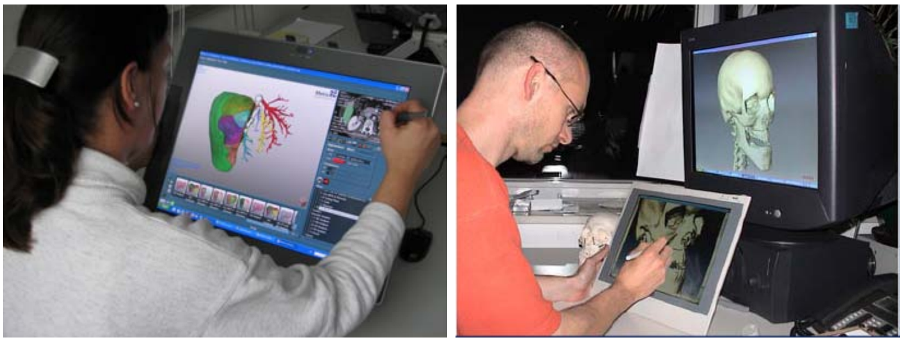


Figure 6 Left: A radiology technician performs segmentation and other analysis tasks on medical imaging data. Pen-based input meets her needs for precise, fast and convenient interaction (Courtesy of MeVis Medical Solutions). Right: Specification of resection lines on a 3D model of the facial bones by means of pen and graphics tablet (From: Zachow et al. [52]).

Software systems for medical diagnosis and treatment planning are almost exclusively operated by means of mouse and keyboard. This was reasonable in the past, since only a few different input devices were available and advanced devices have been very expensive. This situation has radically changed with the advent of a large variety of affordable input devices (see [20] for a recent and comprehensive overview).

For intraoperative use, mouse and keyboard are not appropriate, since all devices have to be sterile. As an alternative, gesture input and the use of the Nintendo Wii have been explored [23, 9, 40]. Before we discuss the special situation inside the operating room (Sect. 4.2), it should be mentioned that also for preoperative diagnosis and planning, alternative input devices should be considered. Pen input is promising for tasks where paths are specified manually, e.g. in case of edge-based image segmentation methods, such as LiveWire [13], where the user sketches the contours of anatomic structures. As an example, radiology technicians frequently use a graphics tablet with pen input (see Fig. 6). Similarly, Zachow et al. [52] used pen input and a graphics tablet to precisely specify resection lines. However, there was neither a systematic comparison of input devices for typical medical visualization tasks nor a solid set of recommendations for the selection of input devices.

Function key pads. Often, a few commands are frequently used in diagnostic and radiation treatment planning systems. These commands may, in principle, be invoked with the function keys of the keyboard or other shortcuts. However, this is neither intuitive nor optimal, since for consistency reasons with other software tools, some function keys cannot be used in a very application-specific way. Also, to invoke the keys, the visual focus has to be put on the keyboard. As an alternative, in an airplane cockpit or a car, a gear can be used without visual attention due to its specific shape which allows to use the tactile sense to grasp it. Thus, joysticks or function pads are a promising alternative. For an application in diagnosis of mammography images, a team around Anke Boedicker, MEVIS Breastcare, developed a special function pad where the size and placement of keys are carefully adapted to the frequency of use. In Figure 7, a general and a specific keypad for diagnosis of mammography are shown. It is likely that radiologists and experts in radiation treatment planning benefit from advanced input devices because the increased learning effort pays off for them.

In the future, other input devices and more variants of the existing devices should be considered for medical visualization. In particular, the popular interaction with gestures and touch screens should be considered to provide a convenient user experience. Moreover, the recent introduction of the Kinect-Controller for the Xbox has potential for intraoperative

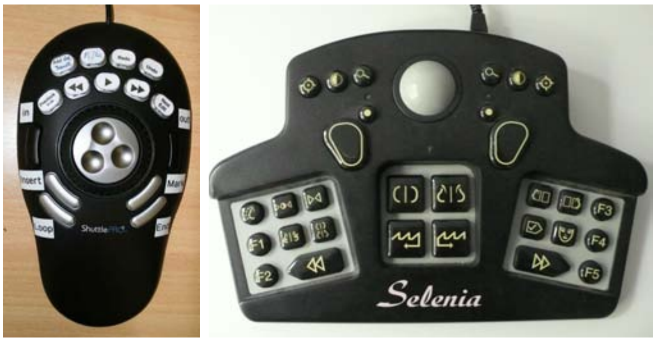


Figure 7 Left: A general function pad may be used to provide fast access for the most important interactions. Right: A dedicated function pad has been developed and refined in various iterations to provide fast access to frequent commands in a diagnostic system for mammography data. (Right image: Courtesy of MeVisBreastCare)

use since the user may control interfaces just with body movements. Empirical evaluations are needed to compare the usability of different input devices for frequent medical visualization tasks.

4.1 3D Input

Many tasks in medical visualization require the interaction with 3D data. Patient-specific 3D models are rotated, 3D measurements are accomplished [37], 3D models of implants, biopsy needles or catheters are inserted (translation, rotation), the virtual camera is moved inside air-filled structures (virtual endoscopy) or virtual resection areas are specified [52]. Also the segmentation of medical volume data, discussed earlier, is a 3D interaction task, where 3D interaction techniques and 3D input devices are essential, in particular to locally refine an initial segmentation (recall [19]).

4.1.0.1 3D widgets.

To optimally support these 3D interaction tasks, 3D widget design and 3D input devices are essential. 3D widgets provide different handles and thus allow to decompose a 3D transformation ([37] discusses 3D widget design in detail for measurement tasks). Often, it is useful to restrict 3D transformation, e.g. by snapping to edges, vertices or faces, or by integrating a priori knowledge. In medical applications, implant placement is probably the most important application. For the sake of brevity, we cannot discuss 3D widget design in detail, but refer to [44].

4.1.0.2 3D input devices.

All interactions mentioned above may, in principle, be accomplished with a 2D mouse where a 3D transformation is somehow artificially decomposed in orthogonal movements. Six degrees-of-freedom devices, such as 3D Mouse, enable a more natural translation, thus reducing the mental effort.

A 3D mouse may also be used in addition to the 2D mouse to support bimanual interaction. Humans are very effective in coordinated movements of both hands, thus bimanual interaction is very promising. Hinckley et al. developed a successful neurosurgery planning system, where bimanual interaction and physical props were employed [16]. Later, Ritter et al. [41]

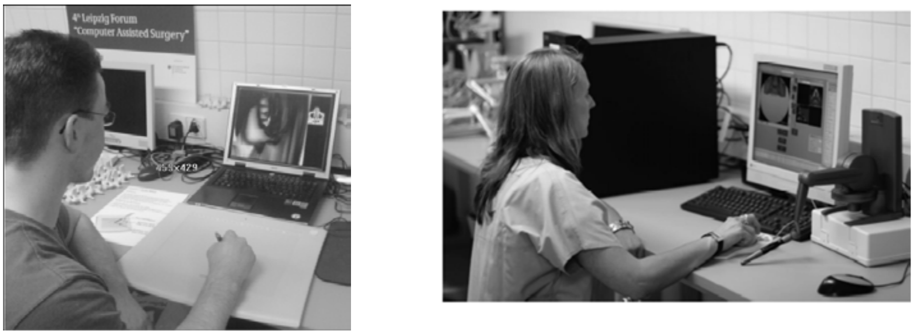


Figure 8 Different input devices such as the graphics tablet with a stylus and the Phantom which provides haptic feedback, graphics tablet with a pen and 3D mouse have been explored for controlling virtual endoscopy in the sinus.

presented an anatomy teaching system, where bimanual interaction was employed successfully as well. As an example, simultaneous rotation and zooming was very effective and satisfying for medical doctors. For virtual endoscopy, a recent investigation of different input devices revealed that after a short learning period, surgeons could track a given path more accurate and faster with a 3D mouse [26]. Haptic input is also essential since it allows the user to better understand complex spatial anatomy such as in the paranasal sinus (Fig. 8).

For the sake of brevity, the area of surgical simulation can only briefly be touched. 3D input devices and haptic feedback are essential in systems to train puncture and needle placement for regional anesthesia or catheter-based interventions. Besides haptic input, usually still with Phantom devices and bimanual control of devices, often complete VR systems are used to provide high degrees of realism (see [49] for a recent example). In particular, if soft tissue structures are involved, elastic deformations play an essential role in surgery simulation. A faithful and efficient realization of this behaviour and its integration in a surgical simulator, is a key aspect.

There is a great need to systematically explore the use of advanced input devices for frequent medical visualization tasks. It might be expected that the use of other input devices give rise to using other interaction techniques. Besides input devices, input techniques also have to be carefully considered. As an example, the transformation of a mouse cursor to a 3D rotation is realized in a strongly different way in popular 3D graphics and visualization toolkits, such as VTK, Open Inventor and 3D Studio. Different usability problems arise in these variants, as a systematic comparison shows [1].

4.2 Interaction Techniques for Intraoperative Use

Time-consuming planning of surgical interventions is primarily accomplished in case of complex and severe surgical interventions, e.g. when rare anatomic variants occur or surgery close to vital risk structures is necessary. In these situations, it is often necessary to compare the intraoperative situation with the preoperative plan, to rehearse preoperative planning or to adapt the plan due to new findings, such as an additional metastasis. Meeting these requirements is challenging in different ways. Accurate navigation systems are needed, intraoperative data has to be precisely registered to the patient, registration has to be updated fast and reliably when the anatomic situation changes, e.g. due to brain shift or soft tissue deformation. Computer-assisted surgery research is focussed on these algorithmic challenges [35]. With respect to the user interface, an important question is the selection

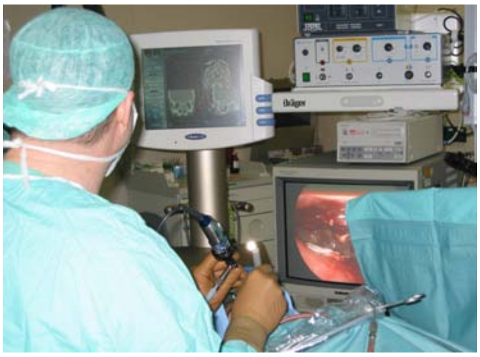


Figure 9 During a navigated intervention of the paranasal sinus the surgeon has to shift his attention from the patient to the endoscopic monitor which presents the images from inside the patient. When he operates close to critical structures he has to focus on the monitor of the navigation system which indicates the instrument position in relation to preoperatively acquired CT data.

and placement of a proper display in an operating room which is already heavily overloaded with various equipment, e.g. from anesthesia. Challenging ergonomic issues arise, e.g. when endoscopic surgery is performed and the monitor has to be placed such that it is not too distracting to look at it during surgery [18]. Even more challenging is the choice of a display solution when a navigation system is used, since an additional monitor has to be carefully placed (see Fig. 9). Visualizations used in these settings have to be carefully adapted to this situation, e.g. by avoiding a too dense display of information. The user has to operate software intraoperatively meeting the requirements of sterility. Voice control has been extensively studied but seems not promising, amongst others, because the environment is noisy. More promising is gesture input, which is a research focus in various groups. Ritter et al. [40] use the Wii interface to operate 3D visualizations (see Fig. 10), whereas [9] employed gestures to operate a touch screen. All of these solutions are based on intensive clinical cooperations, with extensive observations in operating rooms and are now in a state where first trials in realistic settings showed the feasibility.

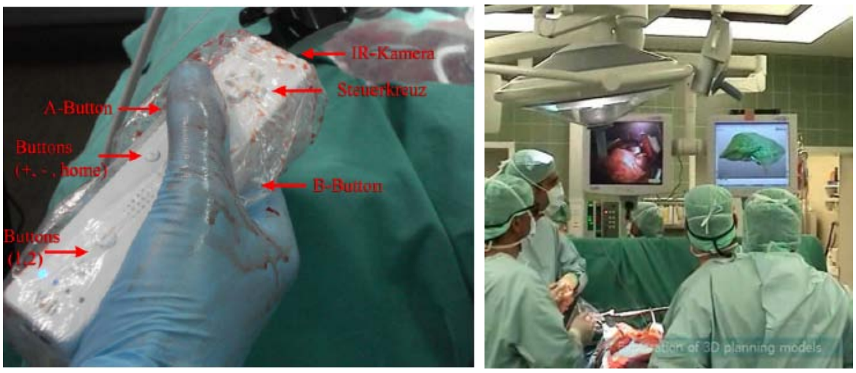


Figure 10 Nintendo’s Wiimote is used under sterile conditions in the operating room to perform simple gesture-based interactions with the 3D model derived in the planning stage. (From: [40])

5 Evaluation

HCI researchers make a basic distinction between *formative* and *summative* evaluations. Formative evaluations are carried out during the development based on prototypes and serve to initiate discussions to receive feedback for guiding the further development. Summative evaluations characterize the final system with respect to ease of use, ease of learning and other usability factors. At most minor problems may be addressed in this late stage. For medical visualizations, both kinds of evaluations are essential. Formative evaluations usually take less effort, they are accomplished with a few users in an informal way. Preparation includes the selection of tasks to focus the evaluation and to carefully think about questions to be answered, including more open questions that stimulate discussions. In early stages, sketches and mockups may be used. Ongoing continuous formative evaluation is essential for research projects where many aspects are not clear and a large design space is explored. Again, [7] gives many convincing examples. One general recommendation is to let users compare alternatives. Users are more critical and discuss much more intensively if they may select between a few alternatives instead of having to comment on the only one solution presented to them.

The "think aloud" technique, like in early task analysis, is helpful. Logging protocols which represent the actions taken by the user are often an invaluable help. Eye-tracking may be a useful ingredient, e.g. to compare visualization techniques with respect to their effect on viewing patterns. However, for most solutions it is not necessary and the interpretation of eye-tracking results is quite challenging.

Summative evaluation often aims at a statistic analysis with a larger number of participants. Many aspects of such an evaluation need to be carefully considered, such as the selection of test persons, the specific questionaire design and the statistical methods used for evaluation. Such summative evaluations have rarely been accomplished in medical visualization. The few such evaluations were web-based questionaires and–as a trade-off-between the number of test persons and their suitability–often not only medical doctors were included.

As a consequence, more insights usually result from formative or informal summative evaluations where a few users are carefully observed and interrogated. In medical visualization, typical tasks include the description of the morphology and spatial surrounding of a pathology, its classification and the assessment of its operability. How long medical doctors need for their decision, how secure they are and whether their assessment is actually correct, are among the aspects which might be explored. Readers interested in evaluation of medical visualization systems should consider the general thoughts on user studies in visualization by Kosara et al. [24] as well as the insight-based evaluation by Saraiya et al. [43]. Ideally, medical visualization systems are evaluated with medical doctors not only as passive sources of information but instead as those who guide the evaluation towards relevant medical problems. Among the few examples of such evaluations are [29] who investigated advanced 3D liver surgery planning and [15], who evaluated virtual endoscopy solutions for surgery planning.

6 Concluding Remarks

The development of visualization systems for clinical medicine requires in-depth analysis of interventions, equipment, usage scenarios and user characteristics. The design of new solutions should comprise a substantial prototyping stage where variants of visualizations, view arrangements and interactions are discussed early and correspondingly refined. The scope of input devices should be carefully considered. This includes a combination of devices

Figure 11 Mobile use of medical image data based on wireless internet connectivity is a huge benefit for the doctor and the patient. (Courtesy of Claus Knapheide, SIEMENS Healthcare)

which may be used bimanually. Graphical user interface design is also an important issue even in research settings. Medical doctors, like many others, expect easy to use and attractive user interfaces, which are perceived as engaging and motivating. Visualization researchers usually do not have an appropriate qualification for all tasks mentioned above. Therefore, cooperations with HCI researchers and practitioners are highly recommended. People with a background in psychology, visual design and user interface programming may be part of interdisciplinary teams to progress medical visualization in a user-centered way. Although this article is focussed on medical applications, it is likely that for advancing other highly specialized professions, such as those in engineering or natural sciences, a similar strategy is needed to improve the impact of medical visualization.

With respect to foreseeable future developments it is very likely that mobile devices in connection with wireless LAN and multitouch input plays an essential role. Leading manufacturers, such as BrainLab, Medtronic, and SIEMENS, already provide systems tailored for use with the Apple iPad (see Fig. 11). These systems enable the selection of cases and image data, zooming in selected data and specifying measurements. In particular, the ability to access medical image and other patient data at the bed of the patient and to enter additional information in digital form is highly welcome by medical doctors. Moreover, the fluent interaction provided by gesture- and touch-based interfaces is considered very attractive by a large majority of them.

There are more HCI-relevant topics to be included in future medical visualization systems (see [47] for an excellent introduction in HCI). An important aspect is whether medical doctors trust the visualizations and analysis results presented to them. In other security-relevant areas, such as aviation, a *level of trust* is determined in order to evaluate this aspect. First attempts to apply these principles to computer-assisted surgery are described in [48]. Finally, treatment decisions in severe cases, such as cancer, are often cooperative decisions where doctors from different disciplines are involved. An open question relates to the optimal support in terms of input devices, displays, visualization and interaction techniques.

Acknowledgements

This paper is based on two joint workshops at the German HCI conference, where Dr. Wolfgang Lauer (RWTH Aachen) and Dr. Werner Korb (HWTK Leipzig) were my co-organizers. I also thank my staff members Steven Birr, Jana Dornheim, Arno Krüger, Jeanette Mönch and Dr. Konrad Mühler as well as Christian Hansen and Dr. Felix Ritter

(Fraunhofer MEVIS) for fruitful discussions on various HCI aspects. I also thank the reviewers for substantially helping to improve this article.

References

1 Ragnar Bade, Felix Ritter, and Bernhard Preim. Usability comparison of mouse-based interaction techniques for predictable 3d rotation. In *Smart Graphics*, pages 138–150, 2005.

2 D. Bartz. Virtual Endoscopy in Research and Clinical Practice. *Computer Graphics Forum*, 24(1):111–126, 2005.

3 David Benyon, Phil Turner, and Susan Turner. *Designing Interactive Systems*. Pearson Education, 2005.

4 Johanna Beyer, Markus Hadwiger, Stefan Wolfsberger, and Katja Bühler. High-quality multimodal volume rendering for preoperative planning of neurosurgical interventions. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1696–1703, 2007.

5 A. F. Blackwell. The reification of metaphor as a design tool. *ACM Trans. on Computer-Human Interaction*, 13(4):490–530, 2006.

6 Stefan Bruckner, Sören Grimm, Armin Kanitsar, and M. Eduard Gröller. Illustrative Context Preserving Volume Rendering. In *Proc. of Eurovis*, pages 69–76, 2005.

7 B. Buxton. *Sketching the User Experience: Getting the Design Right and the Right Design.* Morgan Kaufman, 2007.

8 J. M. Carroll, R. L. Mack, and W. A. Kellogg. *Handbook of Human-Computer Interaction*, chapter Interface Metaphors and User Interface Design, pages 67–85. North-Holland: Elsevier Science Publishers, Helander, M., 1990.

9 Paul Chojecki and Ulrich Leiner. Touchless Gesture-Interaction in the Operating Room. *i-com*, 9(1):13–20, 2009.

10 Jeanette Cordes, Jana Dornheim, and Bernhard Preim. Szenariobasierte Entwicklung von Systemen für Training und Planung in der Chirurgie. *i-com*, 9(1):5–12, 2009.

11 Jeanette Cordes, Katrin Hintz, Jörg Franke, Carsten Bochwitz, and Bernhard Preim. Conceptual design and prototyping implementation of a case-based training system for spine surgery. In Sybille Hambach, Alke Martens, and Bodo Urban, editors, *Proc. of the 1st International eLBa Science Conference (e-Learning Baltics 2008)*, pages 169–178, Rostock, 2008.

12 F. Enders, N. Sauber, D. Merhof, P. Hastreiter, C. Nimsky, and M. Stamminger. Visualization of White Matter Tracts with Wrapped Streamlines. In *Proc. of IEEE Visualization*, pages 51–58, 2005.

13 Alexandre X. Falcao, Jayaram K. Udupa, Supun Samarasekera, Shoba Sharma, Bruce E. Hirsch, and Roberto de Alencar Lofufo. User-steered image segmentation paradigms: Live-wire and live-lane. *Graphical Models and Image Processing*, 60(4):223–260, 1998.

14 G. Fauconnier and M. Turner. *Cambridge Handbook of Metaphor and Thought*, chapter Rethinking Metaphor. New York: Cambridge University Press, Gibbs, R. W., 2008.

15 M. Fischer, G. Strauss, and S. Gahr. Three-dimensional visualization for preoperative planning and evaluation in head and neck surgery. *Laryngorhinootologie*, 88(4):229–233, 2009.

16 John C. Goble, Ken Hinckley, Randy F. Pausch, John W. Snell, and Neal F. Kassell. Two-handed spatial interface tools for neurosurgial planning. *IEEE Computer*, 28(7):20–26, 1995.

17 Timo Götzelmann, Pere-Pau Vázquez, Knut Hartmann, Andreas Nürnberger, and Thomas Strothotte. Correlating text and images: Concept and evaluation. In *Smart Graphics*, pages 97–109, 2007.

18 George B. Hanna, Sami M. Shimi, and Alfred Cuschieri. Task performance in endoscopic surgery is influenced by location of the image display. *Annals of Surgery*, 227(4):481–484, 1998.

19 Frank Heckel, Olaf Konrad, Horst K. Hahn, and Heinz-Otto Peitgen. Interactive 3d medical image segmentation with energy-minimizing implicit functions. *Computers & Graphics*, 35(2):275–287, 2011.

20 K. Hinckley. *Handbook of Human-Computer Interaction*, chapter Input Technologies and Techniques. Lawrence Erlbaum & Associates, Andrew Sears and Julie A. Jacko, Washington, DC, 2007.

21 Karl Heinz Höhne, Andreas Pommert, Martin Riemer, Thomas Schiemann, Rainer Schubert, Ulf Tiede, and W. Lierse. Anatomical atlases based on volume visualization. In *Proc. of IEEE Visualization*, pages 115–123, 1992.

22 L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He. Virtual Voyage: Interactive Navigation in the Human Colon. In *Proc. of ACM SIGGRAPH*, pages 27–34, 1997.

23 Eva Kollorz, Jochen Penne, Joachim Hornegger, and Alexander Barke. Gesture recognition with a Time-Of-Flight camera. *International Journal of Intelligent Systems Technologies and Applications*, 5(3/4):334–343, 2008.

24 Robert Kosara, Christopher G. Healey, Victoria Interrante, David H. Laidlaw, and Colin Ware. User studies: Why, how, and when? *IEEE Computer Graphics and Applications*, 23(4):20–25, 2003.

25 Peter R. Krekel, Charl P. Botha, Edward R. Valstar, Paul W. de Bruin, P. M. Rozing, and Frits H. Post. Interactive simulation and comparative visualisation of the bone-determined range of motion of the human shoulder. In *Proc. of Simulation and Visualization*, pages 275–288, 2006.

26 Arno Krüger, Kristina Stampe, Steffen Irrgang, Ilka Richter, Gero Strauß, and Bernhard Preim. Eingabegeräte und Interaktionstechniken für die virtuelle Endoskopie. In *Mensch & Computer 2008*, pages 237–246. Oldenbourg Verlag, 2008.

27 Arno Krüger, Christian Tietjen, Jana Hintze, Bernhard Preim, Ilka Hertel, and Gero Strauß. Interactive Visualization for Neck-Dissection Planning. In *Proc. of EuroVis*, pages 295–302, 2005.

28 W. Kuhn. 7+/- Questions and Answers about Metaphors for GIS User Interfaces. In W. Kuhn, editor, *Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems*, number 83 in D, pages 113–122, 1995.

29 W. Lamade, G. Glombitza, and L. Fischer. The impact of 3-dimensional reconstructions on operation planning in liver surgery. *Archives of surgery*, 135(11):1256–1261, 2000.

30 Konrad Mühler and Bernhard Preim. Automatic Textual Annotation for Surgical Planning. In *Proc. of Vision, Modelling and Visualization*, pages 277–284, 2009.

31 André Neubauer, Stefan Wolfsberger, Marie-Thérèse Forster, Lukas Mroz, Rainer Wegenkittl, and Katja Bühler. Steps - an application for simulation of transsphenoidal endonasal pituitary surgery. In *Proc. of IEEE Visualization*, pages 513–520, 2004.

32 T. Neumuth, N. Durstewitz, M. Fischer, G. Strauß, A. Dietz, J. Meixensberger, P. Jannin, K. Cleary, H. U. Lemke, and O. Burgert. Structured recording of intraoperative surgical workflows. In *SPIE Medical Imaging 2006 - PACS and Imaging Informatics: Progress in Biomedical Optics and Imaging*. SPIE: Bellingham, 2006.

33 D.A. Norman and S.W. Draper. *User centered system design: New perspectives on human-computer interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

34 Silvia D. Olabarriaga and A.W.M. Smeulders. Interaction in the segmentation of medical images: a survey. *Medical Image Analysis*, 5(2):127–142, 2001.

35 Terry M. Peters and Kevin Cleary. *Image-Guided Interventions: Technology and Applications*. Springer, 2008.

36 Bernhard Pflesser, Ulf Tiede, and Karl Heinz Höhne. Specification, modeling and visualization of arbitrarily shaped cut surfaces in the volume model. In *MICCAI*, pages 853–860, 1998.

37 Bernhard Preim, Christian Tietjen, Wolf Spindler, and Heinz-Otto Peitgen. Integration of Measurement Tools in Medical Visualizations. In *Proc. of IEEE Visualization*, pages 21–28, 2002.

38 Christof Rezk-Salama, Maik Keller, and Peter Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Trans. Vis. Comput. Graph.*, 12(5):1021–1028, 2006.

39 Christian Rieder, Felix Ritter, Matthias Raspe, and Heinz-Otto Peitgen. Interactive visualization of multimodal volume data for neurosurgical tumor treatment. *Comput. Graph. Forum*, 27(3):1055–1062, 2008.

40 Felix Ritter, Christian Hansen, Kjen Wilkens, Alexander Köhn, and Heinz-Otto Peitgen. User interfaces for direct interaction with 3d planning data in the operating room. *i-com*, 9(1):24–31, 2009.

41 Felix Ritter, Bernhard Preim, Oliver Deussen, and Thomas Strothotte. Using a 3D Puzzle as a Metaphor for Learning Spatial Relations. In *Proc. of Graphics Interface 2000*, pages 171–178. Morgan Kaufmann Publishers, 2000.

42 Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-D shapes. In *Proc. of ACM SIGGRAPH*, pages 197–206, 1990.

43 Purvi Saraiya, Chris North, and Karen Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Trans. Vis. Comput. Graph.*, 11(4):443–456, 2005.

44 R. Schmidt, R. Singh, and R. Balakrishnan. Sketching and composing widgets for 3d manipulation. *Computer Graphics Forum*, 27(2):301–310, 2008.

45 Rainer Schubert, Karl Heinz Höhne, Andreas Pommert, Martin Riemer, Thomas Schiemann, and Ulf Tiede. Spatial knowledge representation for visualization of human anatomy and function. In *Information Processing in Medical Imaging*, pages 168–181, 1993.

46 Dirk Selle, Bernhard Preim, Andrea Schenk, and Heinz-Otto-Peitgen. Analysis of Vasculature for Liver Surgery Planning. *IEEE Trans. on Med. Imaging*, 21(11):1344–1357, 2002.

47 B. Shneiderman and C. Plaisant. *Designing the User Interface: Strategies for Effective Human Computer Interaction.* Addison Wesley, 5th edition, 2009.

48 G. Strauß, K. Koulechov, S. Röttger, J. Bahner, C. Trantakis, M. Hofer, W. Korb, O. Burgert, J. Meixensberger, D. Manzey, A. Dietz, and T. Lüth. Ist der Vorteil eines Navigationssystems in der HNO-Chirurgie messbar? *HNO*, 54(12):947–957, 2006.

49 Sebastian Ullrich, Oliver Grottke, Rolf Rossaint, Manfred Staat, Thomas M. Deserno, and Torsten Kuhlen. Virtual needle simulation with haptics for regional anaesthesia. In *Proc. of the IEEE Virtual Reality Workshop on Medical Virtual Envirnoments*, 2010.

50 A. Vilanova, R. Wegenkittl, A. König, and E. Gröller. Nonlinear Virtual Colon Unfolding. In *Proc. of IEEE Visualization*, 2001.

51 Ivan Viola, Armin Kanitsar, and Eduard Gröller. Importance-Driven Volume Rendering. *IEEE Trans. Vis. Comput. Graph.*, 11(4):408–418, 2005.

52 Stefan Zachow, Evgeni Gladilin, R. Sader, and Heinz-Florian Zeilhofer. Draw and cut: Intuitive 3D osteotomy planning on polygonal bone models. In *Proc. of Computer-Assisted Radiology and Surgery (CARS)*, pages 362–369. Springer, 2003.

53 Lingxiao Zhao, Charl P. Botha, Javier Bescos, Roel Truyen, Frans Vos, and Frits H. Post. Lines of curvature for polyp detection in virtual colonoscopy. *IEEE Trans. Vis. Comput. Graph.*, 12(5):885–892, 2006.

Visualizing Spatial Partitions*

Penny Rheingans[1], Blazej Bulka[2], and Marie desJardins[3]

1 University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore MD, 21250, USA
rheingan@cs.umbc.edu

2 Clark & Parsia, LLC
926 N St NW Rear, Studio #1, Washington DC, 2001, USA
blazej@clarkparsia.com

3 University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore MD, 21250, USA
mariedj@cs.umbc.edu

Abstract

We describe an application of geospatial visualization and AI search techniques to the problem of school redistricting, in which students are assigned to home schools within a county or school district. This is a multicriteria optimization problem in which competing objectives must be considered, such as school capacity, busing costs, and socioeconomic distribution. Additionally, school assignments need to be made for three different levels (elementary, middle, and high school) in a way which allows children to move from one school to the next with a cohort of sufficient size. Because of the complexity of the decision-making problem, tools are needed to help end users generate, evaluate, and compare alternative school assignment plans. A key goal of our research is to aid users in finding multiple qualitatively different redistricting plans that represent different tradeoffs in the decision space. We present visualization techniques which can be used to visualize the quality of spatial partititioning plans, compare the alternatives presented by different plans, and understand the interrelationships of plans at different educational levels. We demonstrate these techniques on partitions created through both manual construction and intelligent search processes for the population data of the school district of Howard County, Maryland.

1998 ACM Subject Classification I.3.8 Applications of Computer Graphics, I.2.8 Problem Solving, Control Methods, and Search

Keywords and phrases geospatial visualization, heuristic search, color mapping, multivariate visualization

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.311

1 Motivation and Overview

This research focuses on developing decision support tools for the problem of school redistricting. In this domain, the goal is to assign the students from each geographic region (neighborhood or *planning polygon*) in a school district to a home school at each level (elementary, middle, and high school). We are working with the Howard County (Maryland) school system to develop tools that will aid in generating, evaluating, and comparing alternative school assignment plans. Related applications include emergency response planning, urban planning and zoning, robot exploration planning, and political redistricting.

* This work was supported by the US National Science Foundation ((grant #IIS-0414976).

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 311–321
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany

The school assignment plan should ideally satisfy a number of different goals, such as meeting school capacities, balancing socioeconomic and test score distributions at the schools, minimizing busing costs, allowing students in the “walk area” of a school to attend that home school, and keeping students together in peer groups as they move from school level to level. Since these objectives are often at odds with each other, finding the best plan is a complex multicriteria optimization problem. It is also often desirable to create several alternative plans for consideration; these plans should be *qualitatively different*—that is, they should represent different tradeoffs among the evaluation criteria. Finally, because of the complexity of the problem, it is difficult for users to fully understand these tradeoffs. Therefore, developing effective visualizations is an important challenge.

2 Redistricting Process

The Howard County Public School System (HCPSS) serves a rapidly growing county in suburban Maryland. The pace of development and population growth has necessitated the opening of 25 new schools in the last 16 years, turning the adjustment of school attendance areas into an almost annual event. Under the current process, candidate plans and feasibility studies are generated manually[1] by school system staff. These plans are evaluated and refined by a committee of citizens, then presented at regional meetings for public comment. A small set of candidate plans is forwarded to the Superintendent, who presents two or three recommended alternatives to the Board of Education. The Board has final decision-making authority, and will typically select one of the recommended plans, sometimes making minor modifications in response to concerns raised by parent groups or staff. Note that this process is specific to Howard County; other school districts may have different processes and models.

Candidate plans are evaluated according to eleven measured criteria: (1) the educational benefits for students, (2) the frequency with which students are redistricted, (3) the number and distance of students bused, (4) the total busing cost, (5) the demographic makeup and academic performance of schools, (6) the number of students redistricted, (7) the maintenance of feeder patterns (i.e., the flow of students from elementary to middle to high school), (8) changes in school capacity, (9) the impact on specialized programs, (10) the functional and operational capacity of school infrastructure, and (11) building utilization. Some of these criteria can be clearly quantified (e.g., building utilization and busing costs), while others are harder to quantify (e.g., educational benefits and impact on specialized programs).

In practice, the process is primarily driven by building utilization, but serious consideration is given to feeder patterns, the number of students redistricted, demographic makeup, busing costs, and the frequency with which students are redistricted. Ideally, building utilization should be between 90% and 110% of program capacity and should stay in that range as projected population and capacity changes occur. Desired feeder patterns ensure that there is a critical mass of students who move together from one school level (elementary, middle, and high school) to the next. For instance, the students from a particular middle school should constitute at least 15% of the population of any high school that they feed into. Consideration of the demographic makeup of schools helps to ensure that economically and academically disadvantaged children are not unnecessarily segregated into a few schools.

Figure 1 shows the partitioning of planning regions into school attendance areas at the elementary, middle, and high school levels. A glyph shows the location of each school, and

[1] Map-based tools are used to show the proposed school districts, and a set of spreadsheets is used to generate evaluation data. No other decision support tools are used in the current process.

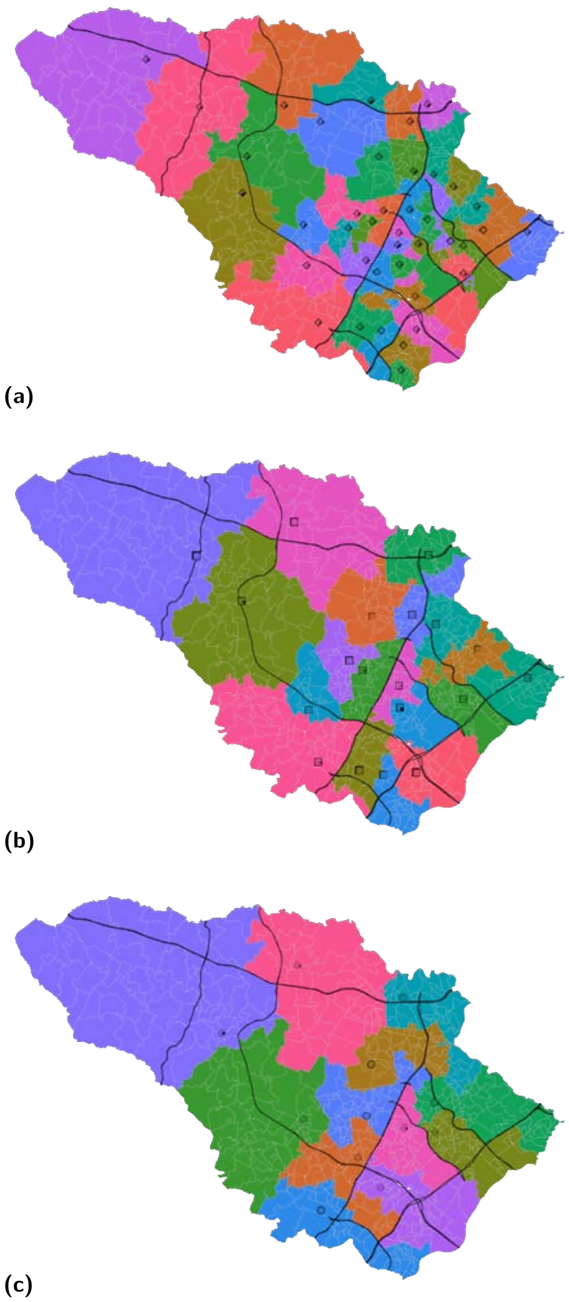


Figure 1 2008-2009 plans for **(a)** elementary, **(b)** middle, and **(c)** high school assignment plans. Each planning polygon is colored according to the school to which it is assigned. Heavy black lines show major roads to give geographic context.

each planning polygon is colored according to the school attended. For instance, in the high school (bottom) image, all students in the northwest region of the county are assigned to Glenelg High School (violet region in northwest) in the current plan.

2.1 Building plans

New plans are typically generated by school system staff in response to changes in capacity, such as the completion of a new school or addition, or a localized problem of overutilization. While one might build a plan from scratch, in practice, new plans are generally derived from base plans. An expert identifies trouble spots in the base plan and reassigns polygons to address the problem. For instance, polygons might be moved out of a school which was over capacity or into another to bolster a small feed. Almost invariably, fixing one problem creates another. The schools adjacent to an overcrowded school may not directly have additional capacity, so adjustments may cascade across multiple schools. Additionally, moving a polygon at one level frequently breaks a feed at another level.

Traditionally, plan developers work at a single level and prefer not to introduce changes at another. For instance, in response to the opening of a new high school, staff prefer to contain changes to the high school level. The effect of a change across level, for instance to feed patterns, is generally evaluated as a second step. The complex nature of the interrelationships between levels makes a clear understanding of the whole pattern both difficult and essential. A clear visual representation of the plan under consideration and its characteristics at this level is a starting point, but does not necessarily indicate which potential changes will have undesired effects across levels. A visual representation which effectively conveys the cross-level interrelationships would allow for more efficient selection of polygons to move.

2.2 Searching for plans

The search space for the redistricting problem is very large, making automated search methods particularly attractive. Automated methods allow for the consideration of a larger number of potential plans than would be practical by manual methods. We have developed methods to search for plans which demonstrate good performance on the measured criteria. The result of the search process is a set of alternative plans. Ideally, these plans are very different from one another, making different compromises between criteria.

For p polygons and s schools, there are

$$s^{(p-s)}$$

possible assignments of schools to polygons (since polygons containing a school are constrained to be assigned to that school). Requiring that school attendance areas be geographically contiguous reduces the number of possible plans, but the number of plans still grows exponentially with the number of schools and polygons. Because of this complexity, we have chosen to use heuristic local search methods, which do not guarantee optimality, but which can be used to find good solutions reasonably quickly.

Our basic approach is a two-stage process: first, we generate an initial “seed” plan using one of several methods described below; second, we use local search to “hill-climb” to a local optimum. Because of the multicriteria nature of the redistricting problem, we have designed several different variations of hill-climbing search that can be used to find qualitatively different alternative plans in the solution space. Variants include basic hill climbing, biased hill climbing with blind bias, and biased hill climbing with diversity bias. Our automatic methods were able to find higher quality plans than manual construction methods, both when optimizing for a single outcome and when balancing among them. These methods are described in more detail in desJardins et al. [4, 5].

3 Scenarios

We present a series of typical exploration and analysis scenarios relevant to school redistricting in order to demonstrate the utility of our system. Different scenarios have different goals, resulting in different visualization requirements. For each, we discuss the visualization techniques developed to address those requirements.

Some visualization requirements and design choices are constant across examples. In order to display multiple values on a single polygon or region, we have chosen to combine the alternatives in the same map. While separate maps are more standard and would be easier to produce, the arbitrary shapes and small sizes of polygons would make correspondences and differences between maps very difficult to perceive. In order to show both possibilities, we have chosen to spilt each polygon or region into rings, displaying different school assignments on different rings. Two alternatives for ring aggregation are shown in the examples below.

All visualizations used the same color assignment for a given school. These colors are selected to be roughly equiluminant, in order to avoid brightness and saturation differences between school colors. In order to show assignment, colors must be distinct from those adjacent. Although it might be nice for colors to be perceptually distinguishable from one another more globally, the number of distinct schools makes that impossible. Consequently, the assignment of colors to schools is arbitrary and two schools at the same level might have similar colors. For example, Figure 1c shows two high schools in very similar shades of blue: Reservoir in the south and Mount Hebron in the north. Intended users of the system would understand the geographic constraints, resulting in no confusion.

3.1 Understanding a plan

Helping viewers understand the nature of a plan under consideration is one core function of a visual plan representation. Situations in which facilitating plan understanding are important include the initial developement of plans, the presentation of potential plans to the public, and the consideration of plans by decision-makers. Two basic types of information should be conveyed by a visual representation of a plan. First, the display should compactly represent the geographic structure of the assignments, specifically which polygons are assigned to which schools at the level(s) under consideration. Geographic properties of interest include the contiguity and compactness of attentence areas, as well as the naturalness of boundaries between schools. Next, a good display should convey the key outcomes resulting from that plan, both quantitative information about such measured criteria as utilitzation and qualitative information about how different values of those criteria are distributed across the space.

Consider a plan which assigned each neighborhood to the closest school at each level. Figure 2 shows two representations of such a plan. On the top, the three rings in each polygon show the assignment at the three school levels; high in the outer ring, middle in the center ring, and elementary in the innermost region. On the bottom, polygons with identical assignments have been grouped into homogeneous regions and each region is ringed to show assignments. The two versions emphasize different types of information. The individual polygon rings emphasize the assignments of each polygon clearly, while the region rings emphasize the makeup of the schools. Polygons with assignments distinct from all neighbors are easier to see with the region rings as small areas with an outlier appearance. One example

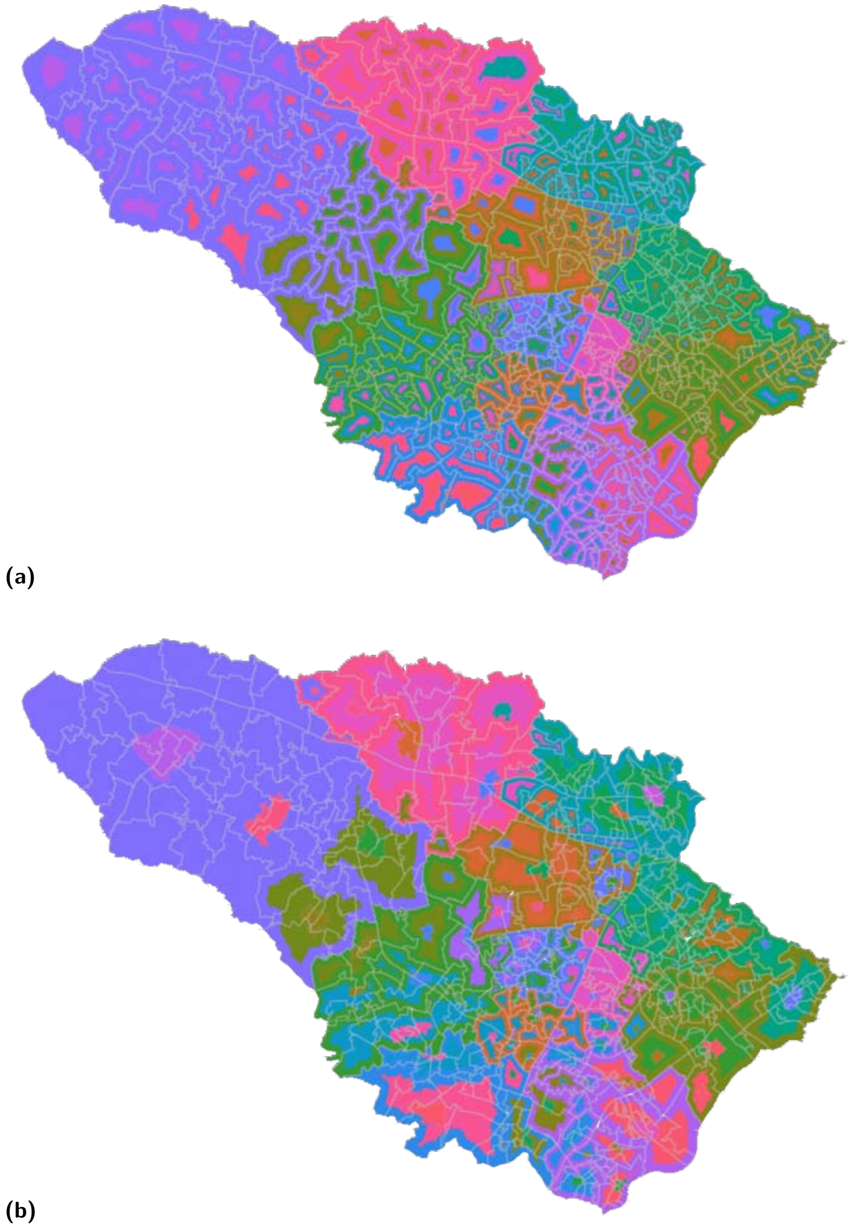
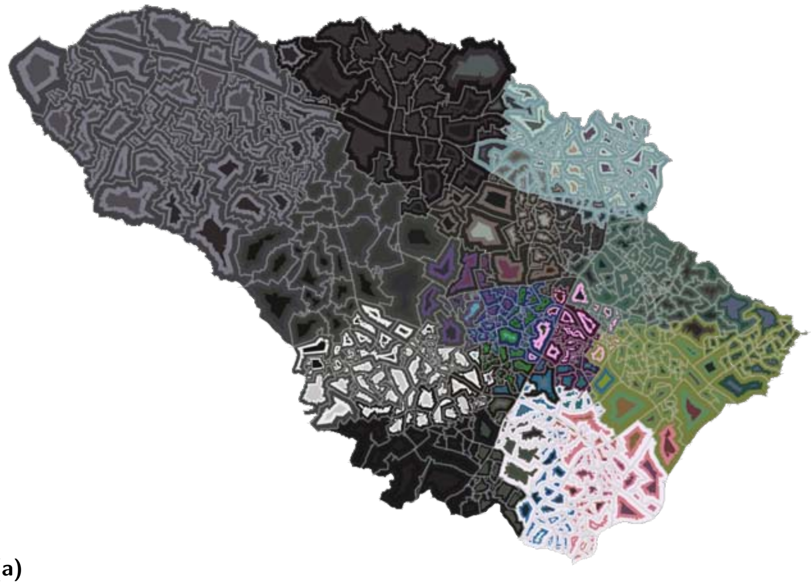


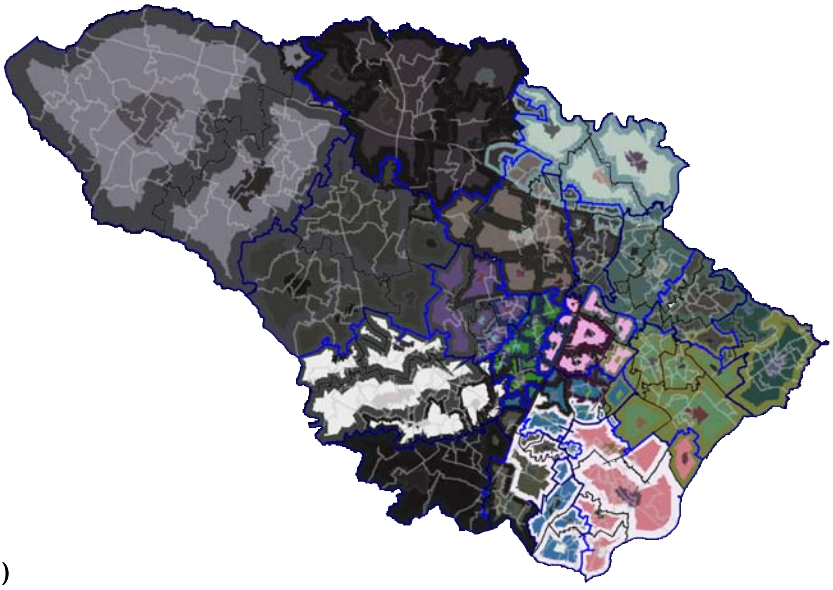
Figure 2 High school plan in which each neighborhood attends the closest school. In the top image **(a)**, each planning polygon is colored according to the schools to which it is assigned (high in the other ring, middle in center ring, and elementary in central area). In the bottom image **(b)**, regions of homogenous assignments are grouped and rings drawn in the resulting region.

can be seen in Figure 2, with a single polygon ringed in orange, green, and olive located toward the eastern side of the county. In this case, those small feeds would not be a problem, since that polygon contains a shopping center and no children (discovered by clicking on the polygon for more detailed information). There is a similar example of an isolated polygon along the northern border of the county; this one does contain children, creating a potential concern. In either view, one can see that the closest school plan makes geographic sense.

Figure 3 shows the pattern of school utilization and free and reduced meal (FARM) percentages created by this plan. Utilization is mapped to brightness (brighter colors show



(a)



(b)

Figure 3 High school plan in which each neighborhood attends the closest school. In the top image **(a)**, each planning polygon is colored according to the schools to which it is assigned (high in the other ring, middle in center ring, and elementary in central area). In each polygon, utilization rate is mapped to brightness, while FARM percentage is mapped to saturation. In the bottom image **(b)**, regions of homogenous assignments are grouped and rings drawn in the resulting region.

fuller schools, while FARM percentage is mapped to saturation (more saturated colors indicate a higher FARM percentage). Since both of these quantities are calculated at the school level, the values are constant across all of the polygons assigned to a school. For such school-level quantities, the choice between individual polygon rings and larger region rings does not make as much difference. Both representations show a great range of brightnesses, indicating that some schools would be dangerously over capacity with such a plan. In particular Hammond

High School in the southeast would be at about 190 percent of capacity, while Marriotts Ridge High School in the northwest would be at about 43 percent of capacity. These images make it clear that a plan which assigns neighborhoods to the closest schools, while intuitively appealing, is not workable given the current distribution of schools and students. FARM percentages, as well, show great disparities among schools.

3.2 Comparing alternative plans

During the redistricting process, two or three alternative plans are prepared and presented to the public and the School Board for comment and consideration. Understanding the differences between these alternative plans, both in terms of geography and outcome, leads to better informed opinions and decisions. The goal of visual support for comparing alternative plans is to effectively convey how plans differ in their assignments, how the resulting school attendance areas correspond, and how these assignments impact the characteristics of the resulting school populations. Since the process of choosing between plans is necessarily one of balancing tradeoffs, an accurate understanding of options and costs is essential.

Figure 4 shows a comparison picture of two alternative plans for county high schools. The school assignments for the closest-school plan are indicated by the inner colored rings in each planning polygon. This plan provides a useful baseline, because it optimizes both walk usage and busing costs, but can be seen to be undesirable in terms of utilization and demographics. The outer ring shows the assignments for a plan which balances capacity utilization across schools. The ring effect allows the user to easily see the planning polygons where the two plans make different recommendations. Polygons with a single color are assigned to the same school in both plans.

For example, in the north center of the county, several polygons are assigned to Marriotts Ridge by the balanced-utilization plan, but to the nearby Mount Hebron, River Hill, and Centennial High Schools by the closest-school plan. The outer ring of these polygons is displayed as pink (Marriots Ridge); the inner ring (center) is either teal (Mount Hebron), green (River Hill), or brown (Centennial). This corresponds to the need to draw from a larger area in order to adequately fill Marriotts Ridge. Along the southeast border of the county, the closest-school plan assigns a number of polygons to Hammond High School (purple innner ring) that are assigned to Reservoir (blue outer ring) by the recommended plan. This difference occurs because assigning them to Hammond would cause that school to be over capacity; also, in this case, those polygons help to balance the socioeconomic distribution at Reservoir. In the bottom image of Figure 4, homogeneous regions are joined. This view emphasizes a higher-level overview of the differences.

Figure 5 shows the comparison of these plans in terms of utilization and FARM outcomes. Since the emphasis is on qualities of the resulting high schools, rather than on assignments of specific polygons, we just show the view with merged rings. From this view, one can clearly see that the outcomes at Reservoir High School (the light blue area in the south) are very different under the two plans, with the balanced-capacity plan increasing both utilization and FARM percentage. With the balanced-capacity plan (shown in the outer ring), there is moderate utilization and FARM percentage (shown by moderate brightness and saturation), while the closest-school plan has very low utilization and FARM percentage (shown by the dark grey inner ring in the area which attends the school under both plans). In the western part of the county, the plans have different outcomes in terms of utilization (utilization is generally slightly lower under the balanced-capacity plan, shown by darker rings), but similar characteristics in terms of FARM percentage (both sets of rings are very unsaturated, indicating low FARM percentage).

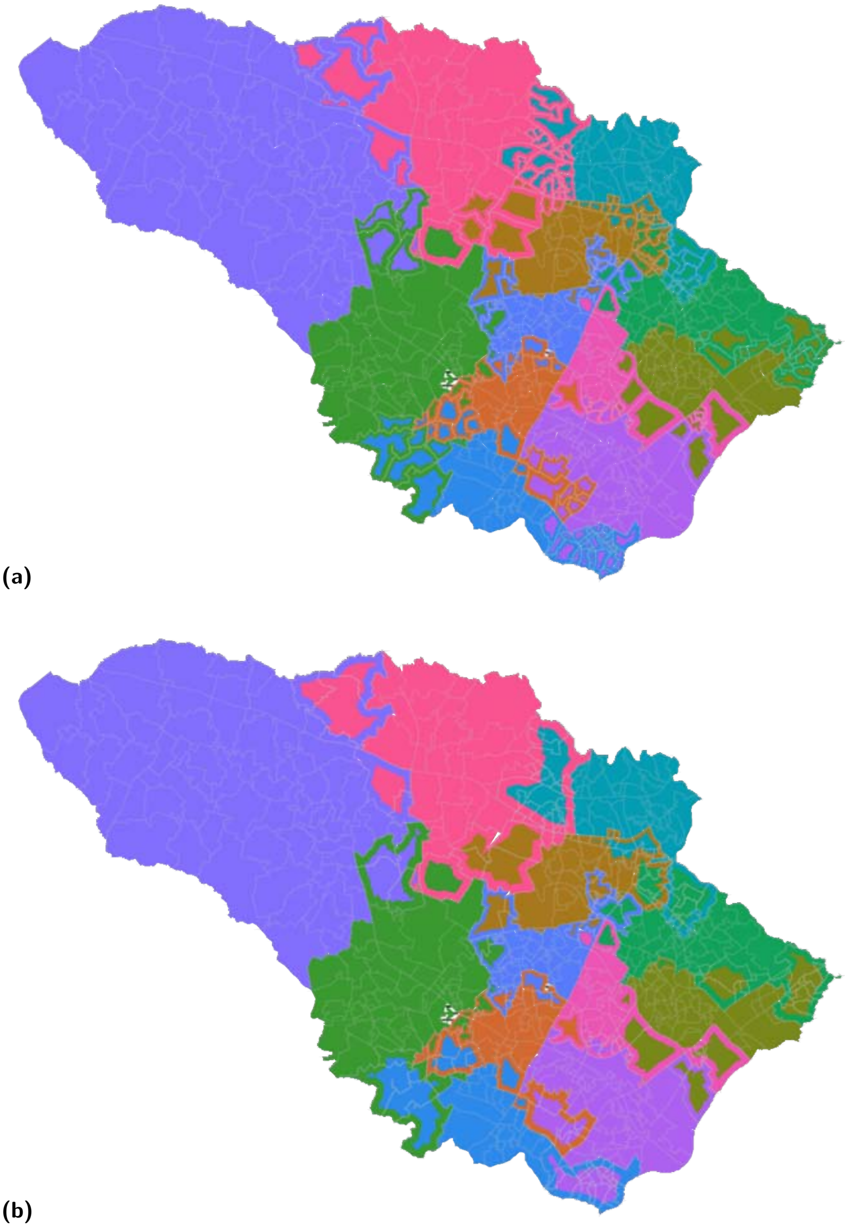


Figure 4 A comparison of the balanced-utilization plan to the closest-school plan. **(a)** The color of the outer and inner rings in each planning polygon indicate the school assignments for the balanced-utilization and closest-school plan, respectively. **(b)** polygons with identical assignments are grouped together into regions.

4 Related Work

The problem of school redistricting is related to that of political redistricting. Several software packages (such as Maptitude [3]) are available for building and analyzing political and school redistricting plans. These packages do not generally provide automated or interactive search methods, do not provide visual comparison techniques such as our ring comparison, and do not facilitate the discovery of qualitatively different plans. Academic research on computational

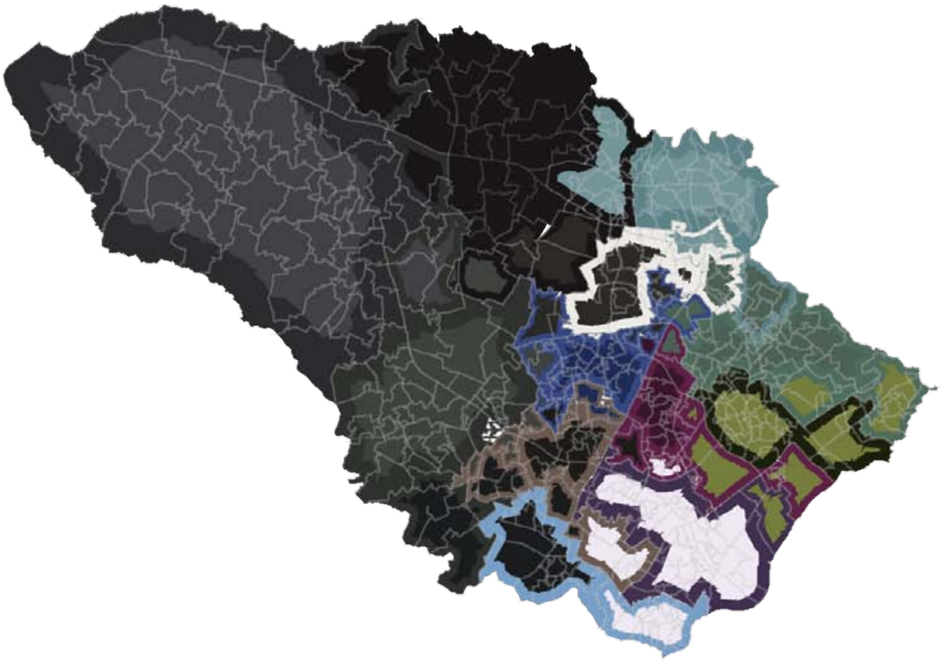


Figure 5 A comparison of the balanced-utilization plan to the closest-school plan. The color of the outer and inner rings in each group of polygons with identical assignments indicate the school assignments for the balanced-utilization and closest-school plan, respectively. Saturation shows FARM percentage, while brightness shows utilization.

approaches to political redistricting [7, 1] has concentrated on methods for constructing plans, rather than visualizing the result.

School redistricting differs from political redistricting in several important ways. First, although compactness is an important factor (both for community building and to minimize busing costs), it is not as important as in political redistricting. Second, the walk usage and feeder issues complicate the scenario for school redistricting. Third, redistricting occurs more frequently (at least in Howard County) than in most political districts, and students are greatly affected by the process. As a result, minimizing the number of students who are redistricted is also an important criterion. Finally, the nature of the decision-making process, in which alternative plans are explicitly compared and contrasted to each other, raises the desirability of generating multiple plans that represent different tradeoffs.

We draw upon previous research on how sets of discrete colors should be used effectively in data visualization. Brewer [2] characterizes different kinds of properties (e.g., numeric and a few types of categorical ones) and their visualization by mixing colors and the use of color components (saturation and brightness). Healey [6] described factors that affect the effectiveness of a particular color selection: the Euclidean distance between the colors in a perceptually uniform color space (L^*, u^*, v^*) and the geometrical positioning of color patterns. Healey also presented a method for manual selection of a fixed number of colors that can be displayed on a monitor by placing them at regular intervals in a carefully positioned circle in L^*, u^*, v^* space. Our method for color selection uses Healey's basic approach.

5 Conclusions

School redistricting is an interesting and challenging problem both computationally and from an application perspective. We have developed a prototype system that uses novel heuristic search and visualization techniques to aid an end user in generating, evaluating, and comparing alternative plans. These tools should provide end users with significant insights into the tradeoffs among alternatives.

The school redistricting problem is closely related to the resource positioning problem of deciding where to build schools, locate fire or police substations, or position emergency response equipment. Our optimization framework, search methods, and visualization tools can all potentially be applied to these other application domains.

Acknowledgements

This work was supported by the National Science Foundation (grant #IIS-0414976). We would like to thank David Drown, Joel Gallihue, and Larry Weaver of the Howard County Public School System's Superintendent's Office for their support, data, and valuable feedback. Thanks also to Poonam Shanbhag, who wrote the original visualization software, and to Andrew Hunt and Priyang Rathod, who also worked on the implementation.

References

1 Micah Altman. *Districting principles and democratic representation.* PhD thesis, California Institute of Technology, 1998.

2 Cynthia A. Brewer. Color use guidelines for data representation. In *Proceedings of the Section on Statistical Graphics*, pages 55–60. American Statistical Association, 1999.

3 Caliper Corporation. Maptitude for redistricting, 2006. `http://www.caliper.com/mtredist.htm`.

4 Marie desJardins, Blazej Bulka, Ryan Carr, Andrew Hunt, Priyang Rathod, and Penny Rheingans. Heuristic search and information visualization methods for school redistricting. In *Proceedings of the Eighteenth Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-06)*, pages 1774–1781. AAAI Press, 2006.

5 Marie desJardins, Blazej Bulka, Ryan Carr, Eric Jordan, Priyang Rathod, and Penny Rheingans. Heuristic search and information visualization methods for school redistricting. *AI Magazine*, 28(3):59–72, 2008.

6 Christopher G. Healey. Choosing effective colours for data visualization. In *Proceedings of IEEE Visualization '96*, pages 263–270, 1996.

7 Robert E. Heldig, Patrick K. Orr, and Robert R. Roediger. Political redistricting by computer. *Communications of the ACM*, 15(8):735–741, August 1972.

Feature Extraction for DW-MRI Visualization: The State of the Art and Beyond*

Thomas Schultz

Computer Science Department and Computation Institute
University of Chicago, USA
t.schultz@uchicago.edu

Abstract

By measuring the anisotropic self-diffusion rates of water, Diffusion Weighted Magnetic Resonance Imaging (DW-MRI) provides a unique noninvasive probe of fibrous tissue. In particular, it has been explored widely for imaging nerve fiber tracts in the human brain. Geometric features provide a quick visual overview of the complex datasets that arise from DW-MRI. At the same time, they build a bridge towards quantitative analysis, by extracting explicit representations of structures in the data that are relevant to specific research questions. Therefore, features in DW-MRI data are an active research topic not only within scientific visualization, but have received considerable interest from the medical image analysis, neuroimaging, and computer vision communities. It is the goal of this paper to survey contributions from all these fields, concentrating on streamline clustering, edge detection and segmentation, topological methods, and extraction of anisotropy creases. We point out interrelations between these topics and make suggestions for future research.

1998 ACM Subject Classification I.3.8 [Computer Graphics]: Applications; I.4.6 [Image Processing and Computer Vision]: Segmentation; I.5.3 [Pattern Recognition]: Clustering; J.3 [Computer Applications]: Life and Medical Sciences

Keywords and phrases Diffusion-Weighted MRI, dMRI, DT-MRI, DTI, HARDI, Streamline Clustering, Edge Detection, DW-MRI Segmentation, Tensor Topology, Crease Surfaces

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.322

1 Introduction

Diffusion-Weighted Magnetic Resonance Imaging (DW-MRI) offers a unique way to investigate the directionally dependent heat motion of water molecules in biological tissue [68]. In nerve fiber bundles or muscles, the diffusion tensor (DT-MRI) model estimates a single principal fiber direction [9]. High angular resolution diffusion imaging (HARDI) allows for even more complex models of apparent diffusion coefficients [90], spin displacement probabilities [122], or fiber densities [118]. Information acquired by DW-MRI is being used, among others, for studying nerve-related disease [48] and normal brain function [104], and for improved planning of brain surgery [134].

Generating visual representations of DW-MRI data has been an active topic in the scientific visualization community in the past decade. Similar to research in flow visualization [97], existing work in this field can be classified as follows:

1. *Direct methods* use simple rules to map data to visual attributes, for example by color mapping [92] or volume rendering [63].

* This work was supported by a fellowship within the Postdoc Program of the German Academic Exchange Service (DAAD).

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 322–345
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

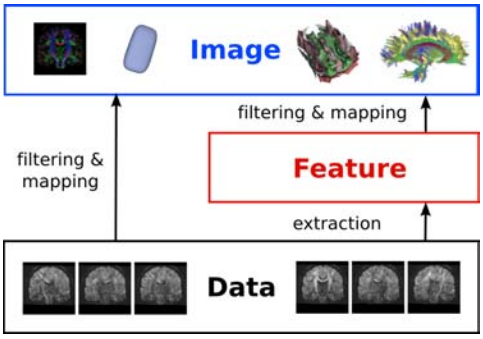


Figure 1 Feature extraction introduces an explicit layer of abstraction that reduces DW-MRI data to the relevant information.

2. *Image-based techniques* generate a texture that conveys certain aspects of the data, for example by variants of line integral convolution [49, 141], reaction-diffusion textures [63], or brush strokes [67].
3. *Geometry-based techniques* generate three-dimensional objects that correspond directly to attributes of the data, like streamlines [82, 139] or glyphs [58].
4. *Feature-based visualization* involves an explicit layer of abstraction between the data itself and its visualization (cf. Figure 1, from [105]), and will be the topic of our paper. Feature definitions meet the requirements of specific research questions, and aim to extract non-local structures from the data.

Several book chapters survey DW-MRI visualization in general [98, 123, 140], but they neither focus on feature-based techniques, nor do they reflect the rapid progress that has been made in the last few years. Therefore, it is the goal of our paper to present a panoramic view of the diverse literature on features in DW-MRI data, to include work that has been published outside the visualization community, and to identify directions of future research. We will assume that the reader is familiar with basic concepts of diffusion-weighted data acquisition, preprocessing, modeling, and derived scalar quantities like fractional anisotropy, which are explained in excellent existing surveys [8, 132, 2].

The material has been organized as follows: Section 2 presents features that are based on clustered streamlines. Section 3 reviews algorithms that segment white or gray matter based on edges or uniform regions in DW-MRI data. Section 4 concentrates on topological methods, while Sections 5 treats crease extraction. Finally, Section 6 concludes the paper.

2 Streamline Clustering and Visualization of Fiber Bundles

Fiber tracking aims at reconstructing the trajectories of major nerve fiber bundles. In the context of visualization, streamline tractography, which integrates lines tangential to inferred fiber directions, represents the most widely used type of fiber tracking [30, 23, 82, 10]. With the diffusion tensor model, it amounts to streamline integration in the principal eigenvector field. Even though a successful streamline visualization requires suitable termination criteria [83] and strategies for seeding, culling, and rendering [139], these problems are outside the scope of our survey, since they do not pertain to feature extraction in the strict sense.

Streamlines visually convey the inferred fiber bundle trajectories in an intuitive manner. However, even though they are frequently called “fibers”, they do not have clear anatomical correlates: Single axons are far below the imaging resolution, and the full fiber bundles

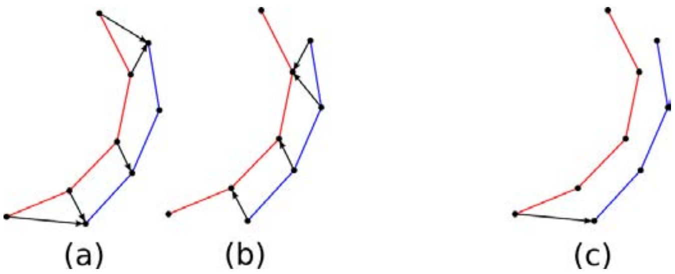


Figure 2 The mean streamline distance d_μ takes an average of the arrow lengths in (a) and (b). The Hausdorff distance d_H only considers the single longest arrow, shown in (c).

typically comprise more than a single streamline. It is the goal of fiber clustering to group individual streamlines into anatomically more meaningful units, which are referred to as fiber bundles. Given a whole brain tractography, clustering is required to make the sheer amount of streamlines more manageable, and it facilitates more abstract and concise visualizations.

Streamline clustering involves at least three major decisions: Defining a measure of similarity between streamlines, chosing a clustering algorithm, and selecting the desired number of clusters. When multiple datasets are considered simultaneously, an additional challenge is to identify correspondences between subjects.

2.1 Distance Measures for Clustering

Some clustering algorithms require a distance measure on pairs of streamlines, while others rely on a similarity measure. In practice, there are various ways to obtain a distance from a similarity and vice versa; a common one is to map high to low values via a Gaussian kernel [16, 86]. For clarity, this section will present all approaches in terms of distance measures.

Most distance measures can be defined in terms of two streamlines F_i and F_j, which are given as uniformly sampled polygonal curves with vertices $\mathbf{p}_k$ and $\mathbf{p}_l$, respectively. The most frequent choice is the mean Euclidean distance d_μ from all points of one streamline to the other curve, averaged over both lines [116, 26, 80] (cf. Figure 2 (a) and (b)):

$$d_\mu(F_i, F_j) = \frac{1}{2}\left(\tilde{d}_\mu(F_i, F_j) + \tilde{d}_\mu(F_j, F_i)\right)$$
$$\text{with } \tilde{d}_\mu(F_i, F_j) = \text{mean}_{\mathbf{p}_k \in F_i} \min_{\mathbf{p}_l \in F_j} \|\mathbf{p}_k - \mathbf{p}_l\| \,. \qquad (1)$$

For matching streamlines from both hemispheres, O'Donnell and Westin [86] use a variant of d_μ, in which they take the minimum of $\tilde{d}_\mu(F_i, F_j)$ and $\tilde{d}_\mu(F_j, F_i)$ instead of the average.

Occasionally, the Hausdorff distance d_H has been considered. It is a "worst case distance", based on the single point from either line which maximizes the distance to the other streamline (cf. Figure 2 (c)) [26, 80]:

$$d_H(F_i, F_j) = \max\left(\tilde{d}_H(F_i, F_j), \tilde{d}_H(F_j, F_i)\right)$$
$$\text{with } \tilde{d}_H(F_i, F_j) = \max_{\mathbf{p}_k \in F_i} \min_{\mathbf{p}_l \in F_j} \|\mathbf{p}_k - \mathbf{p}_l\| \,. \qquad (2)$$

Zhang et al. [138] take the average over points whose distance to the other streamline is larger than some threshold. This measure is in-between d_μ and d_H, in the sense that it emphasizes diverging parts of the curves more than d_μ, but less strongly than d_H. Some authors have also tried the minimum distance between the streamlines as a measure for clustering, but found that it produces inadequate results, since anatomically distinct tracts frequently pass each other at short distances [26, 80].

In a naive implementation, clustering with the distance measures above involves a comparison of all pairs of points for all pairs of streamlines. Given the length and number of streamlines in full-brain tractography, this quickly becomes prohibitively inefficient. The following strategies have been proposed to reduce the computational burden:

- Ding et al. [33] only consider neighboring pairs of streamlines, and match subsequent vertices on both lines, rather than looking for closest points. However, this requires correct alignment of streamlines, and their algorithm to compute it only works if all fibers were seeded along a common plane.
- Brun et al. [17] define a simpler distance measure, which is based on the streamline endpoints. Moberts et al. [80] find that empirically, this measure produces similar results as the Hausdorff distance, which might indicate that for many pairs of streamlines, the maximum distance occurs near their endpoints.
- In a follow-up, Brun et al. [16] generate a descriptor for each streamline by treating its vertices as a point cloud and computing its mean and the square root of its covariance matrix. Then, streamlines are compared by taking the Euclidean distance of the resulting nine-dimensional feature vectors.
- Xia et al. [135] reduce the problem size by initializing the clustering based on a gray matter atlas. Similarly, Maddah et al. [73] only compute distances from streamlines to cluster representatives, which they initialize manually. An additional speedup is obtained by pre-computing a distance transformation of each representative.
- O'Donnell and Westin [86] use an approximative clustering algorithm that only requires them to compute all pairwise distances within a representative subset of all streamlines. Moreover, they evaluate Equation (1) on subsampled streamline representations.

Even though some convergence towards the use of d_μ has been observed [86], it is not entirely settled which measure is the most appropriate, or if different ones should be used depending on the clustering algorithm. Moberts et al. [80] quantify how closely various algorithms and distance measures reproduce a ground truth clustering. In their results, the mean distance d_μ consistently performs better than the Hausdorff distance d_H. However, to our knowledge, there is no systematic comparison available that includes the "in-between" measure by Zhang et al. [138] or the computationally attractive feature-based measure by Brun et al. [16].

Finally, Tsai et al. [121] argue that in order to correctly capture the manifolds on which streamlines lie, the definition of pairwise distances should also take into account the trajectories of all other streamlines. They achieve this by constructing and intersecting minimum spanning trees rooted at each streamline. However, a comparison to traditional distance measures has only been provided on one synthetic example.

2.2 Algorithms for Streamline Clustering

The simplest methods for streamline clustering use the nearest neighbor algorithm or some variant of it [33]. Starting with one cluster per streamline, clusters are merged iteratively if at least one pair of streamlines is closer than some distance threshold θ [26, 80, 138]. Since nearest neighbor clustering produces a hierarchy of refined clusters for decreasing values of θ, it is also called "agglomerative hierarchical clustering" [136].

In nearest neighbor clustering, a single pair of similar streamlines can cause a merge of two otherwise very different clusters. Zhang et al. [138] argue that this helps to cluster sheetlike structures like the corpus callosum. On the other hand, noise in the data can generate spurious streamlines that act as a bridge between anatomically separate clusters and can cause a false merge. Moberts et al. [80] have experimented with some alternative criteria for cluster merging, but none of them improved the results.

Shimony et al. [116] and Maddah et al. [73] have considered fuzzy clustering algorithms, which quantify the certainty to which a streamline belongs to a bundle. These probabilities were both visualized and used in a statistical tract-based analysis [73]. However, to our knowledge, the practical merit over a quantitative analysis that is based on hard clusters [87] has not been studied systematically.

Brun et al. [16] treat the clustering as a graph cut problem. Representing each streamline by a node in a similarity graph, they search for a bipartition that minimizes the summed similarities of the streamlines that are separated, with a normalization that avoids overly small clusters [115]. Like the nearest neighbor methods, this is a hierarchical technique, but it starts with all streamlines in a single cluster, recursively subdividing it until a threshold or a pre-specified number of clusters is reached. The algorithm has also been employed by Enders et al. [36] and Schultz et al. [111]. O'Donnell and Westin [86] use a variant of it, in which they find k-way graph cuts by k-means clustering in a spectral feature space.

The greedy local approaches used in [26, 80, 138] and the global methods in [116, 16, 86, 73] represent two fundamentally different strategies. Generally, local techniques are simpler to understand and to implement, while global ones are more robust against outliers. To our knowledge, no direct experimental comparison is available. However, only one of the global algorithms [86] has been refined to a point at which the reproducibility of its results over subjects and operators has been quantified [124].

2.3 Clustering Parameters

The most important parameter in clustering is the number of desired clusters. In some cases, this parameter is not explicit. Rather, the number of clusters may follow from a desired homogeneity within clusters or a minimum distance between them. In such cases, creating a plot of the number of clusters as a function of the underlying algorithmic parameter can help the user to find the most appropriate value [26].

In theory, such a plot could also be used to try and detect a "natural" clustering implied by the data: Ideally, a clustering that agrees well with the data should remain stable over a considerable range of parameters. To our knowledge, O'Donnell and Westin [86] made the only published attempt to identify the cluster number inherent in the data: Based on a plot of the sum of maximum point-to-centroid distances over the number of clusters, they motivate their choice of 200 clusters for a full-brain tractography.

Moberts et al. [80] use a scoring function to find good parameter values automatically. However, it is defined with respect to a given ground truth clustering, and the authors do not study how well the identified parameters carry over to different datasets for which ground truth may not be known.

At the current state of the art, high-quality streamline clusterings cannot be obtained fully automatically. Therefore, O'Donnell and Westin [86] deliberately generate an oversegmentation, and allow the user to merge clusters manually. Alternatively, Zhang et al. [138] create a user interface for an interactive exploration of the parameter space. As long as streamline clustering requires considerable intervention by an expert, it competes against methods that facilitate a reproducible manual selection of fiber bundles [114, 15, 1]. The frameworks by Chen et al. [21] and Jianu et al. [50] combine both approaches by integrating tools for manual selection in linked two- and three-dimensional views with algorithms for automated clustering.

2.4 Clustering of Populations

A preliminary for cross-subject comparison is spatial alignment of the individual datasets. Multi-subject image registration is a field of its own with a rich body of literature [81]. The specific methods that have been used in the context of streamline clustering include block matching (in [74]), congealing (in [86]), and FLIRT (in [138]). Once alignment has been achieved, there are three general strategies to find corresponding clusters between subjects: Joint clustering, explicit matching, and atlas-based.

O'Donnell and Westin [86] take the union of the streamlines from all coregistered datasets and cluster them jointly. This is a conceptually very simple extension of single-subject clustering, and it automatically yields cross-subject correspondences. However, it increases the problem size dramatically. Therefore, O'Donnell and Westin employ several approximations, the most important of them being the Nystrom method for finding normalized graph cuts.

After combining the streamlines from different subjects, tracts are no longer separated clearly enough to apply the nearest neighbor algorithm. Therefore, Zhang et al. [138] perform clustering on individual subjects and determine correspondences explicitly: Each cluster is described by a nine-dimensional feature vector that captures the mean fiber start-, mid- and endpoints, and correspondences are found based on feature similarity.

The final option is to store descriptors of known fiber bundles in an atlas. Then, streamlines from new subjects are simply assigned to one of the predefined clusters. The resulting labeling is not only consistent over subjects, but also identifies specific tracts automatically. To transfer labels from the atlas to new subjects, Maddah et al. [74] use a B-Spline matching to find the most similar streamline, while O'Donnell and Westin [86] represent fibers as points in a high-dimensional feature space, and assign new streamlines to the closest cluster centroid.

2.5 Representing and Rendering Streamline Clusters

Initially, a clustered fiber bundle is represented as a set of sampled streamlines. Mean or medial lines are a more global representation, and can be constructed in various ways [33, 25, 36, 73, 137, 22]. They have been used to establish correspondences between individual fibers, to act as projection targets for diffusion quantities, to extract geometrical measures like curvature and torsion, or to support visualization. Instead of generating a new line to represent the cluster, O'Donnell et al. [87] select the "most typical" existing one.

For applications in surgical planning, it is important to visualize not just the center, but the full spatial extent of a bundle. Enders et al. [36] use a convex hull algorithm to generate envelopes that wrap a variable percentage of streamlines within the bundle. To give an impression of bundle cores and their full extent, surfaces for different parameter choices are overlaid. To handle more complex bundle geometry, such as branchings, different variants of the alpha shape algorithm have been employed by Merhof et al. [78] and by Chen et al. [22]. Based on a statistical model of the bundle, Maddah et al. [73] generate renderings that depict an interval of three standard deviations around the mean fiber, but they do not provide details on their algorithm for surface construction.

Yushkevich et al. [137] and Schultz et al. [107] have estimated the extent of a bundle by rasterizing the streamlines to a voxel grid and applying a small amount of Gaussian blur on the resulting density volume. Merhof et al. [79] have demonstrated that isosurfaces of such volumes (Figure 3 (a)) provide a simple alternative way to obtain fiber hulls.

In neuroscience, streamline clustering is often used as a preprocess for a tract-based statistical analysis of diffusion measures like mean diffusivity or fractional anisotropy. This

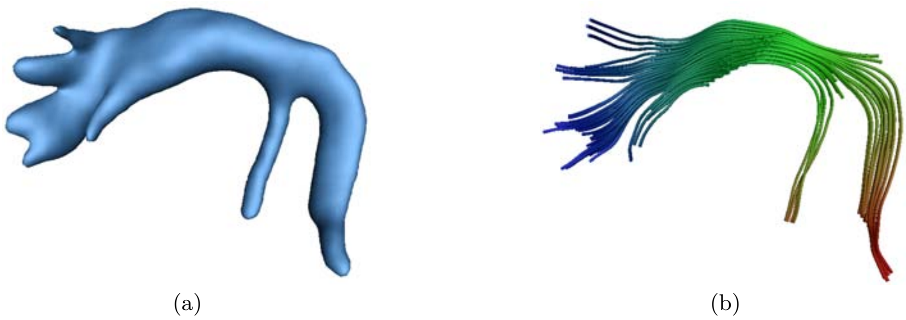


Figure 3 Hulls of fiber bundles (a) have been considered for surgical applications. Correspondences between streamlines (shown by same hue in (b)) are needed for streamline-based spatial statistics.

requires correspondences between points on the individual streamlines or, alternatively, a common shape-based coordinate system. In order to establish such correspondences, Ding et al. [33] match the intersection points of individual streamlines and a plane orthogonal to the medial line. Corouge et al. [24] define a cutting plane through the bundle manually, and match points with the same arc length from the resulting intersection points. This strategy was used to create Figure 3 (b). Maddah et al. [73] create a Voronoi diagram of the mean fiber vertices and match streamline vertices that fall into the same Voronoi cell. After a quantitative comparison to these prior methods, O'Donnell et al. [87] decide to compute an optimal point matching using a variant of the Hungarian algorithm [65].

Intra-cluster variations in streamline shape are modeled by Corouge et al. [25], who perform a Procrustes analysis to achieve curve alignment, followed by a Principal Component Analysis of vertex coordinates. Alternatively, Batchelor et al. [11] explore a variety of quantitative measures of curve shape that do not require point-to-point registration.

3 Edge Detection and Segmentation

White matter segmentation tries to localize fiber bundles without performing streamline tractography, typically by grouping together regions with similar local diffusion properties, or by growing regions until a boundary is reached. It is also possible to segment some gray matter structures from DW-MRI data. This frequently involves probabilistic tractography as a pre-process.

3.1 Distance Measures for Segmentation

In order to identify voxels with similar diffusion properties, the first step is to choose a distance measure $d(\mathbf{T}^{(1)}, \mathbf{T}^{(2)})$ for diffusion tensors $\mathbf{T}$. Since symmetric 3×3 tensors form a six-dimensional vector space, it is possible to simply use the Euclidean metric d_E on that space [133, 38, 100]. In terms of tensor components t_{ij}, it is given as

$$d_E(\mathbf{T}^{(1)}, \mathbf{T}^{(2)}) = \sqrt{\sum_{i=1}^{3}\sum_{j=1}^{3}\left(t_{ij}^{(1)} - t_{ij}^{(2)}\right)^2}. \tag{3}$$

Alternatively, the scalar product $\langle \mathbf{T}^{(1)}, \mathbf{T}^{(2)} \rangle$ that corresponds to the Euclidean space (i.e., $d_E(\mathbf{T}^{(1)}, \mathbf{T}^{(2)}) = \sqrt{\langle \mathbf{T}^{(1)} - \mathbf{T}^{(2)}, \mathbf{T}^{(1)} - \mathbf{T}^{(2)} \rangle}$) can serve as a measure of tensor similarity. It is given in terms of tensor components t_{ij} or, equivalently, in terms of eigenvalues λ_i and

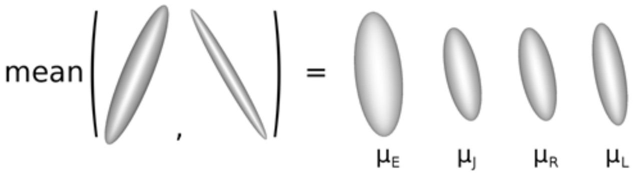


Figure 4 The choice of metric or distance measure changes the definition of the mean tensor.

eigenvectors $\mathbf{e}_i$:

$$\langle \mathbf{T}^{(1)}, \mathbf{T}^{(2)} \rangle = \sum_{i=1}^{3} \sum_{j=1}^{3} t_{ij}^{(1)} t_{ij}^{(2)} = \sum_{i=1}^{3} \sum_{j=1}^{3} \lambda_i^{(1)} \lambda_j^{(2)} \langle \mathbf{e}_i^{(1)}, \mathbf{e}_j^{(2)} \rangle^2 . \tag{4}$$

Occasionally, these two definitions have been presented as distinct measures that happen to behave similarly in practice [3, 95]. Expressing the scalar product as $\langle \mathbf{T}^{(1)}, \mathbf{T}^{(2)} \rangle = \mathrm{tr}(\mathbf{T}^{(1)\mathrm{T}} \mathbf{T}^{(2)})$ (where $\mathbf{T}^{(1)\mathrm{T}}$ is the transpose of $\mathbf{T}^{(1)}$ and $\mathrm{tr}(\cdot)$ denotes matrix trace) and writing $\mathbf{T}$ in terms of its spectral decomposition ($\mathbf{T} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^{\mathrm{T}}$) allows us to verify that they are, in fact, mathematically equivalent [55].

Based on the interpretation of diffusion tensors as the covariance matrix of a multivariate Gaussian distribution, Wang and Vemuri [126] propose to treat diffusion tensor fields as fields of Gaussian probability distribution functions (PDFs), and to consider the square root of the J-divergence, an information theoretical distance measure between PDFs. It is given as

$$d_J(\mathbf{T}^{(1)}, \mathbf{T}^{(2)}) = \frac{1}{2} \sqrt{\mathrm{tr}\left(\mathbf{T}^{(1)-1}\mathbf{T}^{(2)} + \mathbf{T}^{(2)-1}\mathbf{T}^{(1)}\right) - 6} \tag{5}$$

where $\mathbf{T}^{(1)-1}$ denotes the inverse of $\mathbf{T}^{(1)}$. Unlike d_E, d_J is not a metric, since it does not fulfill the triangle inequality. A proper Riemannian metric d_R on the set of positive definite tensors that shares some of the properties of d_J is presented by Batchelor et al. [12]:

$$d_R(\mathbf{T}^{(1)}, \mathbf{T}^{(2)}) = \sqrt{\mathrm{tr}\left(\log^2(\mathbf{T}^{(1)-1/2}\mathbf{T}^{(2)}\mathbf{T}^{(1)-1/2})\right)} \tag{6}$$

where the matrix logarithm $\log(\cdot)$ is evaluated on the eigenvalues. Lenglet et al. adapt the idea of interpreting diffusion tensors via the corresponding Gaussian PDFs and initially employ d_J [69]. In a follow-up work [70], they use d_R and show that it can be derived from the Fisher information matrix.

The choice of metric can have significant impact on the computational efficiency of a segmentation model. In part, this is due to the fact that Equations (5) and (6) involve operations like matrix inverse and logarithm. In addition, many segmentation methods repeatedly compute tensor means, and the definition of a mean μ changes with the chosen metric (cf. Figure 4). When using the Euclidean norm d_E, the mean can be taken componentwise, but its computation becomes more complex for d_J, and when more than two tensors are involved, it does not even have a closed form solution in case of d_R [12, 70]. In order to reduce this computational burden, Arsigny et al. [5] propose the Log-Euclidean distance d_L; it corresponds to applying d_E after taking the matrix logarithm:

$$d_L(\mathbf{T}^{(1)}, \mathbf{T}^{(2)}) = d_E(\log(\mathbf{T}^{(1)}), \log(\mathbf{T}^{(2)})) . \tag{7}$$

For many practical purposes, d_L has similar properties as d_J and d_R: Weldeselassie and Hamarneh [130] find that within their segmentation framework, d_L to d_J produce comparable results. From a theoretical standpoint, d_J and d_R are both affine invariant: For any invertible matrix $\mathbf{A}$, $d_{\{J,R\}}(\mathbf{T}^{(1)}, \mathbf{T}^{(2)}) = d_{\{J,R\}}(\mathbf{A}\mathbf{T}^{(1)}\mathbf{A}^{\mathrm{T}}, \mathbf{A}\mathbf{T}^{(2)}\mathbf{A}^{\mathrm{T}})$. d_L is not affine invariant, but retains, among others, invariance under rotation and scaling [5]. Additionally, d_R and d_L are invariant under inversion, $d_{\{R,L\}}(\mathbf{T}^{(1)}, \mathbf{T}^{(2)}) = d_{\{R,L\}}(\mathbf{T}^{(1)-1}, \mathbf{T}^{(2)-1})$.

Which of these invariances are actually desirable for diffusion tensor processing is a topic of current scientific debate. Affine invariant and Log-Euclidean metrics put positive definite tensors at an infinite distance from tensors with non-positive eigenvalues. Arsigny et al. [5] argue that this is beneficial, since diffusivities should not be negative. When interpolating tensors with the Euclidean metric, the determinant of the interpolant can become larger than the determinants of the original tensors. Arsigny et al. call this effect “tensor swelling” and consider it a major problem, which is avoided by the use of their Log-Euclidean metric.

On the other hand, Pasternak et al. [94] have pointed out that while diffusivity as a physical quantity is indeed non-negative, noise and artifacts can still cause *measurements* of diffusivity to become negative. When performing statistical analysis on such measurements, replacing individual negative values with zero or a small positive value will bias the resulting estimates. More generally, they consider affine invariance to be undesirable, since diffusivity has a meaningful scale, so it makes sense to measure absolute differences (as done by d_E), whereas affine invariant metrics effectively compute ratios. Finally, they argue that unless very specific assumptions can be made on the variability underlying the data, it is more appropriate to preserve tensor trace during interpolation (again done by d_E) than to preserve the determinant (as done by d_L).

Other frameworks for tensor interpolation and distance measurement have monotonically interpolated eigenvalues [76] or invariants such as fractional anisotropy [60]. However, they do not lead to closed-form distance measures, and have so far not been used for segmentation.

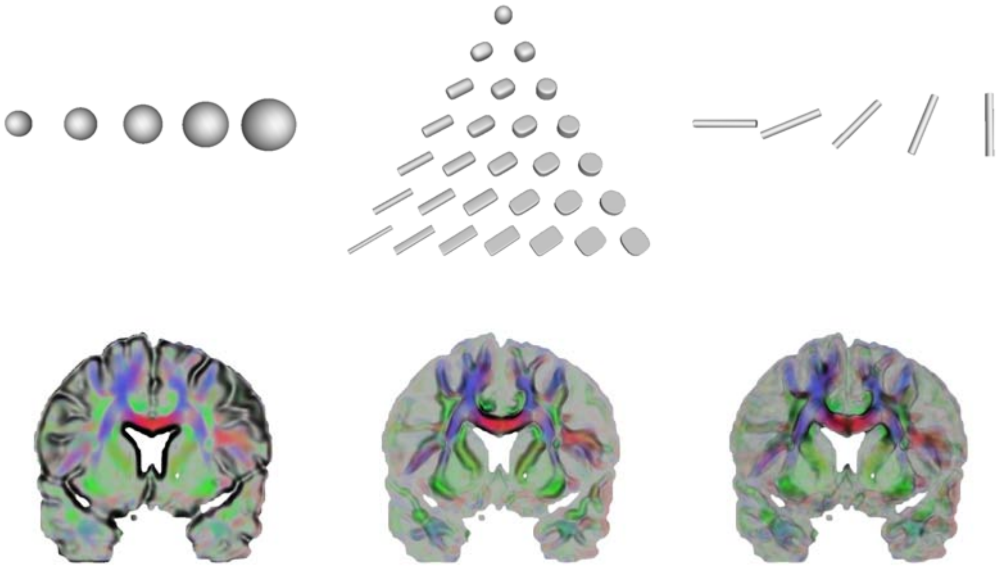
There is evidence that for DT-MRI segmentation, no unique “correct” or “best” distance measure exists. In [53] and [55], Jonasson et al. segment two different anatomical structures and point out that different distance measures gave the best results in these two cases. Similarly, Ziyan et al. [145] find that when segmenting the thalamic nuclei, concentrating on differences in principal eigenvector directions produces better results than using the full tensor information. Schultz et al. [106] demonstrate that, starting from the same initialization, an edge-based level set method will either segment the ventricle or the corpus callosum, depending on the chosen distance measure. Consequently, they propose a flexible framework in which physically meaningful user-defined weights allow the expert to customize the distance measure for specific segmentation problems.

3.2 Edges in Diffusion Tensor Fields

Edges are rapid changes in the data and often indicate meaningful structural boundaries. In grayscale images, edges correspond to variations in intensity, and the gradient vector indicates their direction and magnitude. In DT-MRI fields, they have a more complex structure: Gradients are third-order tensors, and the six degrees of freedom in second-order tensors lead to different types of edges, related to changes in trace, shape, or orientation (cf. Figure 5).

The first edge maps of DT-MRI data were created by Pajevic et al. [91], who distinguish two types of edges by either considering gradients in the full tensor field or only in its deviatoric (trace-free) part. Alternatively, O’Donnell et al. [85] employ normalized convolution to reduce the effect of tensor trace on the overall edge strength.

Kindlmann et al. [59] have presented a framework which decomposes the tensor field gradient into six physically meaningful edge types that cover all degrees of freedom present in the data. This is achieved by considering the gradients of tensor invariants, i.e., changes in tensor value that are associated with changes in scalar measures such as trace or fractional



(a) Changes in trace outline the brain tissue as a whole.

(b) Changes in tensor shape delineate boundaries of white matter tracts.

(c) Changes in orientation indicate interfaces between fiber bundles.

Figure 5 Different types of edges exist in diffusion tensor fields. On the top, variations in specific tensor attributes are illustrated. On the bottom, corresponding edge maps are overlaid on slices in standard XYZ-RGB color coding.

anisotropy. Three scalar measures are used that parameterize the full space of tensor shapes and whose gradients are orthogonal [37]. In order to capture changes in tensor orientation, the framework is supplemented with rotation tangents, describing the effect of infinitesimal rotations around the eigenvectors. Taken together, normalized invariant gradients and rotation tangents form a local basis of the space of symmetric 3 × 3 matrices, and different edge types are distinguished by expressing the tensor field gradient in that basis. In a follow-up work, Schultz and Seidel [109] clarified the relation between invariant gradients and perturbation theory and demonstrate how to find analytical edge maps of anisotropy measures that are defined in terms of sorted eigenvalues, such as c_l, c_p, and c_s [131].

Edges are low-level features; by themselves, they have not found extensive use for visualization or quantitative analysis. However, they provide guidance for some of the segmentation approaches that will be discussed in the remainder of this section [38, 106, 110]. They are also used to define higher-level features which do not imply a partitioning of the field, like interfaces between adjacent tracts [59]. Crease surfaces can be used to obtain explicit geometric representations of such features, and will be covered in Section 5.

3.3 Fiber Tract Segmentation

Contributions on white matter segmentation can be compared along various axes. In this section, we will consider segmentation goals and discuss different segmentation strategies and region models. Most existing approaches are based on the diffusion tensor model, but some [41, 77, 54, 127, 102, 32] have used HARDI data.

Initial works on diffusion MRI segmentation have aimed at delineating the white matter as a whole [144, 38]. However, such segmentations are already afforded by simpler acquisition schemes, such as T1-weighted imaging [27]. Segmentation tasks that explicitly rely on the contrast provided by diffusion MRI include specific structures within the white matter. The corpus callosum is the largest white matter structure in the human brain, connecting the two hemispheres, and has been a very common target of segmentation efforts [100, 70, 106, 54, 29, 130, 127, 102, 32]. Some authors have also aimed at segmenting the cortico-spinal tract [53, 41, 54, 102, 32] and several smaller structures, including (parts of) the cingulum bundles [41, 6, 127], the inferior long association bundles [53] and tracts in the brainstem [41]. Moreover, rat brain and spinal cord [126, 77, 7] as well as a phantom from excised rat spinal cords [18, 70, 32] have been the objects of investigation.

Level set methods [113, 88] have been the most frequent framework for DW-MRI segmentation. They represent the segmented boundary as the zero isocontour of a scalar field and evolve it under some partial differential equation. Zhukov et al. [144] present the earliest level set method for DT-MRI segmentation, but only make use of two derived scalar fields, which limits their method to finding the ventricles and the white matter as a whole. Most later works exploit the full tensor information: Feddern et al. [38, 39] and Schultz et al. [106] use edge-based active contours [19, 57], while most others rely on region models. For example, Wang and Vemuri [125] use a piecewise constant model [20]; Rousson et al. [100] employ multivariate Gaussian modeling of the regions, similar to [99].

As alternatives to level sets, Markov random fields [77, 6, 7] and graph-based methods [130, 127] have been used. Markov random fields [71] are based on a statistical image model. Given the observed image and optional priors, they estimate model parameters that determine a per-region distribution of image values and the per-voxel region membership in a Bayesian framework. Graph-based methods [129, 66] have already been mentioned in Section 2.2: They treat each voxel as a node in a graph, and assign edge weights based on similarity in diffusion properties and spatial proximity. Fiber tract segmentation was also approached with traditional visualization tools by Schultz et al. [110]: They extract isosurfaces that outline the white matter core and use a watershed-type mesh segmentation to identify individual bundles, combining edge- with region-based information.

The simplest models used for segmentation assume a constant diffusion tensor per region, or a Gaussian distribution of tensors. In the presence of bending fibers, such assumptions are problematic, since the mean tensor tends to become isotropic and the model can lose its discriminative power. This problem has been addressed in the following ways: Wang and Vemuri [126] replace their original piecewise constant by a piecewise smooth model [84]. De Luis-García and Alberola-López [29] employ a mixture of Gaussians. Finally, Awate et al. [6] present a fuzzy segmentation framework that is based on nonparametric region models.

Methodically, “fast-marching” [93] or “flow-based” [18] tractography lies in between streamline-based tractography and white matter segmentation. It starts with a level set representation of a small sphere, initialized at some user-specified seed. The initial surface is propagated in normal direction with a speed that depends on the estimated likelihood of a fiber connection in the given direction. Each voxel records the time at which it was first reached by the surface; a gradient descent in the resulting time-of-arrival map connects any point in the domain to the original seed, and assigns a likelihood to those paths, making a connection to probabilistic tractography.

On the other hand, Jonasson et al. [53] define propagation speed based on the local similarity of diffusion properties, stop surface evolution when the speed drops below some threshold, and interpret the final surface as the outline of a fiber bundle, which makes a connection to traditional edge-based segmentation schemes. Conceptually, this strategy corresponds to region growing with curvature-based regularization. Savadjiev et al. [102] use a simpler, unregularized region growing scheme, but derive a similarity measure from the differential geometry of local streamline neighborhoods rather than diffusion properties.

For segmentation of HARDI data, Jonasson et al. [54] transform three-dimensional fields of orientation distribution functions (ODFs) to a five-dimensional scalar field, which they segment with a hypersurface. Hagmann et al. [41] rely on the same five-dimensional position-orientation space, but employ a Markov random field rather than a level set method for the segmentation. Similarly, McGraw et al. [77] use a hidden Markov measure field, but model the ODFs as mixtures of von Mises-Fisher distributions. Finally, Descoteaux and Deriche [32] model the ODFs with spherical harmonics and apply the single-Gaussian region model by Rousson et al. [100] to the resulting coefficients.

3.4 Gray Matter Segmentation

Finding the major nuclei of the thalamus has been the first example of gray matter segmentation based on DW-MRI. The reason for this is that the thalamus contains homogeneously oriented axons (“striations”) that provide contrast in diffusion imaging.

Wiegell et al. [133] combine diffusion tensor dissimilarity and spatial distance to cluster voxels inside of the thalamus via a k-means algorithm. To obtain improved results, other authors employed normalized graph cuts [145], level sets [55], and the mean shift algorithm [34]. To avoid having to determine correspondences explicitly when comparing the segmentation results on different subjects, Ziyan and Westin [146] segment all datasets jointly.

A conceptually different approach to segmenting the thalamus goes back to Behrens et al. [14], who map manually defined cortex areas to the thalamus via a probabilistic tracking of the cortico-thalamic connections. In a follow-up, Johansen-Berg et al. [52] validate this method by comparing the cortex areas that are reached by a tractography from the centers of thalamic activations as determined by functional MRI.

Johansen-Berg et al. [51] and Anwander et al. [4] demonstrate that clustering results from probabilistic tractography allows one to segment certain parts of the cortex into functionally distinct regions. This observation is exploited by Schultz et al. [111] in their definition of topological features in diffusion MRI data, which will be detailed in the next section.

4 Topological Methods

In the feature-based visualization of flow fields, topological methods are a well-established approach. Introduced to visualization by Helman and Hesselink [44], they have been researched widely since then [119, 103, 128]. Topological methods extract qualitative structures from the data by considering the asymptotic behavior of streamlines, effectively partitioning the domain into regions in which all streamlines start in the same source and end in the same sink. The interfaces between these regions constitute the topological skeleton, a concise description of the data which is invariant over all structurally equivalent fields.

The eigenvector fields derived from a symmetric second-order tensor field differ from proper vector fields in the fact that they lack orientation. Delmarcelle and Hesselink [31] have demonstrated how the basic definitions of vector field topology can be transferred to eigenvectors. In a follow-up work, Hesselink et al. [45] also considered three-dimensional tensor fields. Points at which two or more eigenvalues coincide are the fundamental features in tensor topology, and later works [28, 142] have clarified the fact that in generic 3D data, these “degenerate” loci form stable lines.

Even though it had previously been proposed to apply tensor topology to diffusion tensor fields [143], Schultz et al. [111] were the first to publish results on this type of data. Experimenting with the algorithm by Zheng et al. [142], they found that the utility of tensor topology on data from brain diffusion MRI is limited by the facts that the features do not have a clear correlation to anatomical structures, and that they are very sensitive to measurement noise and the choice of interpolation. This observation is partly explained by the fact that standard single fiber models predict diffusion tensors with two equal eigenvalues. On idealized diffusion tensor fields, this imposes a constraint which violates the genericity assumption on which tensor topology is founded.

In order to transfer the basic idea of topological visualization to diffusion MRI, Schultz et al. [111] instead consider the asymptotic behavior of a probabilistic fiber tracking algorithm and identify fuzzy subvolumes which are likely to connect functionally distinct regions of the brain. By rendering semi-transparent confidence hulls around the core structures, the uncertainty in the feature boundaries is conveyed visually.

Inspired by Lagrangian coherent structures [42], Hlawitschka et al. [46] present a framework that visually highlights boundaries along which the streamlines from nearby seeds diverge notably after a fixed and finite integration time. Even though they are not the result of an asymptotic analysis, such boundaries are similar to the topological skeleton in that they separate regions of different qualitative streamline behavior. Finally, it has been pointed out by Salzbrunn and Scheuermann [101] that streamline clustering (cf. Section 2) can be considered as an alternative generalization of vector field topology, in particular when the similarity measure is based on the endpoints, as in [17].

5 Anisotropy Crease Surfaces and Lines

In typical scalar fields, local maxima and minima form isolated points. Creases generalize these extrema to higher-dimensional structures, like extremal lines and surfaces. Ridges generalize local maxima, while valleys correspond to local minima. Even though the most adequate crease definition is not undisputed [64], the so-called "height crease definition" [43] has become a well-researched tool to find medial axes in grayscale images [96] and has been extended towards applications as diverse as medical image analysis, molecular modeling, and analysis of fluid flows [35].

In the context of DW-MRI, ridge surfaces and lines were used in order to obtain reliable spatial statistics in group studies. To localize variations in fractional anisotropy (FA) between two groups (typically, patients vs. healthy controls), many previous studies would register individual datasets to a volumetric template and subsequently perform a local statistical analysis, effectively comparing anisotropy values in individual voxels over all subjects. The reliability of such voxel-based methods is limited by inevitable inaccuracies in registration, varying degrees of partial voluming, and heuristic choices of the smoothing kernel. To ameliorate these problems, Smith et al. [117] extract ridges in a group-averaged FA map. In each individual dataset, locally maximal values of FA are then projected onto the common ridge and statistical tests are done on the ridge manifold. This projection compensates slight misalignments and does not involve any smoothing of the modeled data, leading to more objective and valid results.

The algorithm by Smith et al. [117] employs a thinning technique to produce a binary mask, indicating which voxels are part of the ridge. In contrast, Kindlmann et al. [62] use the height crease definition along with analytical derivatives of FA to extract a high-resolution triangle mesh representation of ridge and valley surfaces, which they name "anisotropy

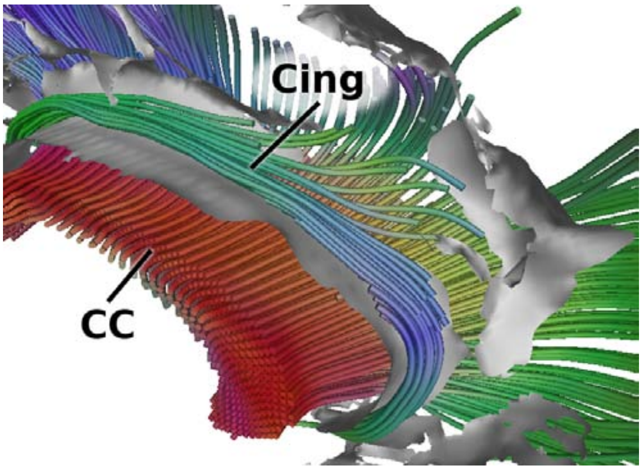


Figure 6 A valley surface in fractional anisotropy (gray) separates the right cingulum bundle (Cing) from the corpus callosum (CC).

creases" (cf. Figure 6). In a follow-up to this work, Schultz et al. [112] present an improved algorithm that captures the surface boundaries more precisely and is approximately one order of magnitude faster on typical DT-MRI data. Some white matter structures, including parts of the cingulum bundles, are tube-like rather than sheet-like, so their cores are better described by lines than by surfaces. In this respect, the algorithm to extract FA ridge lines presented by Tricoche et al. [120] complements the original method from [62].

Intuitively, creases are similar to topological methods as surveyed in Section 4 in that they reduce the dataset to a structural skeleton. More formally, Tricoche et al. [120] point out that degenerate lines in tensor topology are a subset of the crease lines in a tensor invariant called mode, which captures the transition between linear and planar anisotropy. In this sense, anisotropy creases remedy the shortcomings of tensor topology in DT-MRI data by replacing mode with FA, an invariant which is more widely used for anatomical analysis and less affected by noise. A second link between both methods has been found by Schultz et al. [112]. In their theoretical analysis of crease surface topology, they observe that degenerate lines in the Hessians of the considered anisotropy measure (rather than in the diffusion tensor field itself) form one type of crease surface boundary, and they employ the gradient descent proposed by Zheng et al. [142] to localize it.

Since measurement noise and fine-scale structures induce spurious local extrema, extraction of stable and expressive creases typically requires some amount of filtering. While Smith et al. [117] process group mean images which are smoothed implicitly by the involved averaging, Kindlmann et al. [62] work on individual datasets and thus have to perform explicit filtering. However, the adequate smoothing extent is not known a priori and can even vary spatially, because anatomical structures differ in size. In computer vision, this problem has been addressed by treating the amount of smoothing as a free parameter, and analyzing the family of images generated by all possible values along this additional "scale" axis [72]. A particle-based approach to extract creases of variable dimensionality from scale space has been presented by Kindlmann et al. [61].

Extracting anisotropy creases and finding medial structures from fiber bundles (Section 2.5) can be considered as competing approaches to defining a white matter skeleton. This becomes especially clear in a work by Yushkevich et al. [137]: By fitting medial surface models to fiber bundles, they generate surfaces that lend themselves to a similar local statistical analysis as it was done by Smith et al. [117]. Ultimately, the choice between both options might depend on the exact goals of a study: An analysis based on streamline clusters is more specific to individual tracts, but relies on a correct clustering (in [137], this step is performed by an expert). Anisotropy creases are less specific, but require less manual intervention and make it easier to cover the whole white matter.

In the visualization community, crease manifolds were not only treated as a tool for quantitative analysis: Schultz et al. [112] found that DT-MRI streamsurfaces, which had been proposed previously to illustrate regions of planar diffusion [139], are ill-defined in typical data, and demonstrate that planarity ridge surfaces can replace them.

6 Conclusion and Future Directions

One of the challenges in the visualization of DW-MRI data is the high information density: Diffusion tensor MRI produces symmetric second-order tensors with six degrees of freedom per voxel, HARDI models have an even larger number of parameters. Feature extraction is an established strategy to provide a layer of abstraction that allows for visual inspection of such complex data, while preserving information that is relevant to a given application.

In this paper, we have summarized the state of the art in feature-based DW-MRI visualization. Even though different communities have contributed to this field, we were able to highlight a number of close links between the presented approaches. Most prominently, a recurrent theme in streamline clustering (Section 2), white matter segmentation (Section 3), and fuzzy diffusion MRI topology (Section 4) is to identify the spatial extent of anatomically relevant fiber bundles.

DW-MRI visualization is a relatively young field. Consequently, much of the initial work has concentrated on brainstorming new ideas, and many methods have been proposed without a systematic comparison to the state of the art. As the field matures, one important aspect of future research will be to clarify the relation between existing techniques in a more formal manner. Ideally, the community should agree on quantitative benchmarks that help to ensure that we are not only creating more, but actually better ways of analyzing DW-MRI data. This would be greatly facilitated by a public library of datasets, along with atlases that provide ground truth for segmentation and fiber clustering. For grayscale and color images, such libraries exist and are actively used in the computer vision community [75]. Even though some diffusion-based white matter atlases are available on the web,[1] we are not aware of any works that have used them for validation.

When research on DW-MRI visualization was still in its infancy, flow visualization already offered an impressive variety of feature-based techniques [97]. Consequently, many methods for vector fields have been transferred to tensor fields and applied to DT-MRI data. With time, this relation should become more reciprocal: For example, the uncertainty in connections inferred from DW-MRI inspired a fuzzy topological visualization [111] before the topology of uncertain vector fields had been considered [89]. In the relation between flow and diffusion visualization, it is important to remember that despite the many analogies, flow and diffusion are physically different processes. Flow is described by the Navier-Stokes equations, while diffusion is governed by Fick's laws [40]. One important consequence of this fact is that flow has a direction, while the displacement probabilities of free diffusion exhibit antipodal

[1] This includes an atlas available from the Laboratory of Neuro Imaging at UCLA (`http://www.loni.ucla.edu/Atlases/`) and two atlases from John Hopkins University, included in the FMRIB Software Library provided by Oxford University (`http://www.fmrib.ox.ac.uk/fsl/fslview/atlas-descriptions.html`).

symmetry. The hindered diffusion processes typically met in biological tissue are even more complex and still not fully understood [13].

With respect to the development of new methods, future challenges of the field include:

- *Adapt methods to HARDI.* While high-angular resolution imaging has been a very active topic in neuroimaging for several years, the visualization community has started to investigate it only recently [47, 108]. Feature extraction is even more important for this more complex type of data than it is for DT-MRI. For example, Jonasson et al. [56] point out that streamline clustering becomes essential to disentangle intersecting fiber tracts reconstructed via HARDI.
- *Reproducibility and feature matching* over different subjects have to be achieved before methods can be used in neuroscience or medical studies, but are considered only rarely in the context of visualization (e.g., [80, 138]). Moreover, *statistical methods* and the *quantification of uncertainty and variance* still receive too little attention in the visualization literature.
- A closely related task is to *make existing methods more stable.* One example of this is the concept of scale space, which facilitates feature detection under noise and varying scales. For 2D images, scale space is well-established in image processing and computer vision. However, it has started to find its way into the visualization of large 3D datasets only recently [61].
- Even though we believe that there is legitimate fundamental research to be done in visualization, we should keep an eye on the fact that the ultimate goal of our efforts is to *facilitate new insights.* This can be achieved by using visualization to support generation of new hypotheses, as part of a framework that enables quantitative analysis and hypothesis testing, or as a debugging tool for the scientific workflow.

Acknowledgement

I would like to thank Gordon Kindlmann for helpful discussions about this paper, and Alfred Anwander (MPI CBS, Leipzig, Germany) for providing the DW-MRI dataset that was used to create some of the figures.

References

1 D. Akers. CINCH: a cooperatively designed marking interface for 3D pathway selection. In *Proc. ACM UIST*, pages 33–42, 2006.

2 D. C. Alexander. An introduction to computational diffusion MRI: The diffusion tensor and beyond. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, pages 83–106. Springer, 2006.

3 D. C. Alexander, J. C. Gee, and R. Bajcsy. Similarity measures for matching diffusion tensor images. In T. P. Pridmore and D. Elliman, editors, *Proceedings of the British Machine Vision Conference*, pages 93–102, Nottingham, September 1999.

4 A. Anwander, M. Tittgemeyer, D. von Cramon, A. Friederici, and T. Knösche. Connectivity-based parcellation of broca's area. *Cerebral Cortex*, 17(4):816–825, 2007.

5 V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421, 2006.

6 S. P. Awate, H. Zhang, and J. C. Gee. A fuzzy, nonparametric segmentation framework for DTI and MRI analysis: With applications to DTI-tract extraction. *IEEE Transactions on Medical Imaging*, 26(11):1525–1536, 2007.

7 A. Barmpoutis, B. C. Vemuri, T. M. Shepherd, and J. R. Forder. Tensor splines for interpolation and approximation of DT-MRI with applications to segmentation of isolated rat hippocampi. *IEEE Transactions on Medical Imaging*, 26(11):1537–1546, 2007.

8 P. J. Basser and D. K. Jones. Diffusion-tensor MRI: theory, experimental design and data analysis – a technical review. *NMR in Biomedicine*, 15(7–8):456–467, 2002.

9 P. J. Basser, J. Mattiello, and D. L. Bihan. Estimation of the effective self-diffusion tensor from the NMR spin echo. *Journal of Magnetic Resonance*, B(103):247–254, 1994.

10 P. J. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi. In vivo fiber tractography using DT-MRI data. *Magnetic Resonance in Medicine*, 44:625–632, 2000.

11 P. G. Batchelor, F. Calamante, J.-D. Tournier, D. Atkinson, D. L. G. Hill, and A. Connelly. Quantification of the shape of fiber tracts. *Magnetic Resonance in Medicine*, 55:894–903, 2006.

12 P. G. Batchelor, M. Moakher, D. Atkinson, F. Calamante, and A. Connelly. A rigorous framework for diffusion tensor calculus. *Magnetic Resonance in Medicine*, 53:221–225, 2005.

13 C. Beaulieu. The basis of anisotropic water diffusion in the nervous system – a technical review. *NMR in Biomedicine*, 15(7–8):435–455, 2002.

14 T. E. J. Behrens, H. Johansen-Berg, M. W. Woolrich, S. M. Smith, C. A. M. Wheeler-Kingshott, P. A. Boulby, G. J. Barker, E. L. Sillery, K. Sheehan, O. Ciccarelli, A. J. Thompson, J. M. Brady, and P. M. Matthews. Non-invasive mapping of connections between human thalamus and cortex using diffusion imaging. *Nature Neuroscience*, 6(7):750–757, 2003.

15 J. Blaas, C. P. Botha, B. Peters, F. M. Vos, and F. H. Post. Fast and reproducible fiber bundle selection in DTI visualization. In C. Silva, E. Gröller, and H. Rushmeier, editors, *Proc. IEEE Visualization 2005*, pages 59–64, 2005.

16 A. Brun, H. Knutsson, H. J. Park, M. E. Shenton, and C.-F. Westin. Clustering fiber tracts using normalized cuts. In C. Barillot, D. Haynor, and P. Hellier, editors, *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 3216 of *LNCS*, pages 368–375. Springer, 2004.

17 A. Brun, H.-J. Park, H. Knutsson, and C.-F. Westin. Coloring of DT-MRI fiber traces using laplacian eigenmaps. In R. Moreno-Díaz and F. Pichler, editors, *Proc. Computer Aided Systems Theory (EUROCAST)*, volume 2809 of *LNCS*, pages 518–529. Springer, 2003.

18 J. S. W. Campbell, K. Siddiqi, V. V. Rymar, A. F. Sadikot, and G. B. Pike. Flow-based fiber tracking with diffusion tensor and q-ball data: Validation and comparison to principal diffusion direction techniques. *NeuroImage*, 27:725–736, 2005.

19 V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. International Conference on Computer Vision (ICCV)*, pages 694–699, 1995.

20 T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

21 W. Chen, Z. Ding, S. Zhang, A. MacKay-Brandt, S. Correia, H. Qu, J. A. Crow, D. F. Tate, Z. Yan, and Q. Peng. A novel interface for interactive exploration of DTI fibers. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1433–1440, 2009.

22 W. Chen, S. Zhang, S. Correia, and D. S. Ebert. Abstractive representation and exploration of hierarchically clustered diffusion tensor fiber tracts. *Computer Graphics Forum (Proc. EuroVis)*, 27(3):1071–1078, 2008.

23 T. E. Conturo, N. F. Lori, T. S. Cull, E. Akbudak, A. Z. Snyder, J. S. Shimony, R. C. McKinstry, H. Burton, and M. E. Raichle. Tracking neuronal fiber pathways in the living human brain. *Proc. Natl. Acad. Sci. USA*, 96:10422–10427, 1999.

24 I. Corouge, P. T. Fletcher, S. Joshi, S. Gouttard, and G. Gerig. Fiber tract-oriented statistics for quantitative diffusion tensor MRI analysis. *Medical Image Analysis*, 10:786–798, 2006.

25 I. Corouge, S. Gouttard, and G. Gerig. A statistical shape model of individual fiber tracts extracted from diffusion tensor MRI. In C. Barillot, D. R. Haynor, and P. Hellier, editors, *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 3217 of *LNCS*, pages 671–679. Springer, 2004.

26 I. Corouge, S. Gouttard, and G. Gerig. Towards a shape model of white matter fiber bundles using diffusion tensor MRI. In *Proc. IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 344–347, 2004.

27 M. B. Cuadra, L. Cammoun, T. Butz, O. Cuisenaire, and J.-P. Thiran. Comparison and validation of tissue modelization and statistical classification methods in t1-weighted MR brain images. *IEEE Transactions on Medical Imaging*, 24(12):1548–1565, 2005.

28 J. Damon. Generic structure of two-dimensional images under gaussian blurring. *SIAM Journal on Applied Mathematics*, 59(1):97–138, 1998.

29 R. de Luis-García and C. Alberola-López. Mixtures of gaussians on tensor fields for DT-MRI segmentation. In N. Ayache, S. Ourselin, and A. Maeder, editors, *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 4791 of *LNCS*, pages 319–326. Springer, 2007.

30 T. Delmarcelle and L. Hesselink. Visualizing second-order tensor fields with hyperstreamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, 1993.

31 T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. In R. D. Bergeron and A. E. Kaufman, editors, *Proc. IEEE Visualization*, pages 140–147, 1994.

32 M. Descoteaux and R. Deriche. High angular resolution diffusion MRI segmentation using region-based statistical surface evolution. *Journal of Mathematical Imaging and Vision*, 33(2):239–252, 2009.

33 Z. Ding, J. C. Gore, and A. W. Anderson. Classification and quantification of neuronal fiber pathways using diffusion tensor MRI. *Magnetic Resonance in Medicine*, 49:716–721, 2003.

34 Y. Duan, X. Li, and Y. Xi. Thalamus segmentation from diffusion tensor magnetic resonance imaging. *International Journal of Biomedical Imaging*, 2007, 2007.

35 D. Eberly. *Ridges in Image and Data Analysis*, volume 7 of *Computational Imaging and Vision*. Kluwer Academic Publishers, 1996.

36 F. Enders, N. Sauber, D. Merhof, P. Hastreiter, C. Nimsky, and M. Stamminger. Visualization of white matter tracts with wrapped streamlines. In C. Silva, E. Gröller, and H. Rushmeier, editors, *Proc. IEEE Visualization*, pages 51–58, 2005.

37 D. B. Ennis and G. Kindlmann. Orthogonal tensor invariants and the analysis of diffusion tensor magnetic resonance images. *Magnetic Resonance in Medicine*, 55(1):136–146, 2006.

38 C. Feddern, J. Weickert, and B. Burgeth. Level-set methods for tensor-valued images. In O. D. Faugeras and N. Paragios, editors, *Proc. IEEE Workshop on Geometric and Level Set Methods in Computer Vision*, pages 65–72, 2003.

39 C. Feddern, J. Weickert, B. Burgeth, and M. Welk. Curvature-driven PDE methods for matrix-valued images. *International Journal of Computer Vision*, 69(1):93–107, 2006.

40 A. Fick. Über Diffusion. *Annalen der Physik*, 170(1):59–86, 1855.

41 P. Hagmann, L. Jonasson, T. Deffieux, R. Meuli, J.-P. Thiran, and V. J. Wedeen. Fibertract segmentation in position orientation space from high angular resolution diffusion MRI. *NeuroImage*, 32(2):665–675, 2006.

42 G. Haller. Distinguished material surfaces and coherent structures in three-dimensional fluid flows. *Physica D*, 149:248–277, 2001.

43 R. M. Haralick. Ridges and valleys on digital images. *Computer Vision, Graphics, and Image Processing*, 22:28–38, 1983.

44 J. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *Computer*, 22(8):27–36, 1989.

45 L. Hesselink, Y. Levy, and Y. Lavin. The topology of symmetric, second-order 3D tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):1–11, 1997.

46 M. Hlawitschka, C. Garth, X. Tricoche, G. Kindlmann, G. Scheuermann, K. I. Joy, and B. Hamann. Direct visualization of fiber information by coherence. *International Journal of Computer Assisted Radiology and Surgery*, 5(2):125–131, 2010.

47 M. Hlawitschka and G. Scheuermann. HOT-lines: Tracking lines in higher order tensor fields. In C. Silva, E. Gröller, and H. Rushmeier, editors, *Proc. IEEE Visualization*, pages 27–34, 2005.

48 M. A. Horsfield and D. K. Jones. Applications of diffusion-weighted and diffusion tensor MRI to white matter diseases – a review. *NMR in Biomedicine*, 15(7–8):570–577, 2002.

49 E. Hsu. Generalized line integral convolution rendering of diffusion tensor fields. In *Proc. International Society of Magnetic Resonance in Medicine (ISMRM)*, page 790, 2001.

50 R. Jianu, Ç. Demiralp, and D. H. Laidlaw. Exploring 3D DTI fiber tracts with linked 2D representations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1449–1456, 2009.

51 H. Johansen-Berg, T. Behrens, M. Robson, I. Drobnjak, M. Rushworth, J. Brady, S. Smith, D. Higham, and P. Matthews. Changes in connectivity profiles define functionally distinct regions in human medial frontal cortex. *Proc. National Academy of Sciences of the United States of America (PNAS)*, 101(36):13335–13340, 2004.

52 H. Johansen-Berg, T. E. J. Behrens, E. Sillery, O. Ciccarelli, A. J. Thompson, S. M. Smith, and P. M. Matthews. Functional-anatomical validation and individual variation of diffusion tractography-based segmentation of the human thalamus. *Cerebral Cortex*, 15(1):31–39, 2005.

53 L. Jonasson, X. Bresson, P. Hagmann, O. Cuisenaire, R. Meuli, and J.-P. Thiran. White matter fiber tract segmentation in DT-MRI using geometric flows. *Medical Image Analysis*, 9(3):223–236, 2005.

54 L. Jonasson, X. Bresson, J.-P. Thiran, V. J. Wedeen, and P. Hagmann. Representing diffusion MRI in 5-D simplifies regularization and segmentation of white matter tracts. *IEEE Transactions on Medical Imaging*, 26(11):1547–1554, 2007.

55 L. Jonasson, P. Hagmann, C. Pollo, X. Bresson, C. R. Wilson, R. Meuli, and J.-P. Thiran. A level set method for segmentation of the thalamus and its nuclei in DT-MRI. *Signal Processing*, 87(2):309–321, 2007.

56 L. Jonasson, P. Hagmann, J.-P. Thiran, and V. J. Wedeen. Fiber tracts of high angular resolution diffusion MRI are easily segmented with spectral clustering. In *Proc. International Society for Magnetic Resonance in Medicine (ISMRM)*, 2005.

57 S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. International Conference on Computer Vision (ICCV)*, pages 810–815, 1995.

58 G. Kindlmann. Superquadric tensor glyphs. In *Proc. Eurographics/IEEE Symposium on Visualization (SymVis)*, pages 147–154, 2004.

59 G. Kindlmann, D. Ennis, R. Whitaker, and C.-F. Westin. Diffusion tensor analysis with invariant gradients and rotation tangents. *IEEE Transactions on Medical Imaging*, 26(11):1483–1499, 2007.

60 G. Kindlmann, R. San José Estépar, M. Niethammer, S. Haker, and C.-F. Westin. Geodesic-loxodromes for diffusion tensor interpolation and difference measurement. In N. Ayache,

S. Ourselin, and A. Maeder, editors, *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI), Part I*, volume 4791 of *LNCS*, pages 1–9. Springer, 2007.

61 G. Kindlmann, R. San José Estépar, S. M. Smith, and C.-F. Westin. Sampling and visualizing creases with scale-space particles. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Visualization)*, 15(6):1415–1424, 2009.

62 G. Kindlmann, X. Tricoche, and C.-F. Westin. Delineating white matter structure in diffusion tensor MRI with anisotropy creases. *Medical Image Analysis*, 11(5):492–502, 2007.

63 G. Kindlmann, D. Weinstein, and D. Hart. Strategies for direct volume rendering of diffusion tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):124–138, April 2000.

64 J. J. Koenderink and A. J. van Doorn. Local features of smooth shapes: Ridges and courses. In B. C. Vemuri, editor, *Geometric Methods in Computer Vision II*, volume 2031 of *Proc. of SPIE*, pages 2–13, 1993.

65 H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–87, 1955.

66 S. Lafon and A. B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, 2006.

67 D. H. Laidlaw, E. T. Ahrens, D. Kremers, M. J. Avalos, R. E. Jacobs, and C. Readhead. Visualizing diffusion tensor images of the mouse spinal cord. In *Proc. IEEE Visualization*, pages 127–134, 1998.

68 D. Le Bihan, E. Breton, D. Lallemand, P. Grenier, E. Cabanis, and M. Laval-Jeantet. MR imaging of intravoxel incoherent motions: Application to diffusion and perfusion in neurologic disorders. *Radiology*, 161(2):401–407, 1986.

69 C. Lenglet, M. Rousson, and R. Deriche. Segmentation of 3D probability density fields by surface evolution: Application to diffusion MRI. In C. Barillot, D. R. Haynor, and P. Hellier, editors, *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 3216 of *LNCS*, pages 18–25. Springer, 2004.

70 C. Lenglet, M. Rousson, and R. Deriche. DTI segmentation by statistical surface evolution. *IEEE Transactions on Medical Imaging*, 25(6):685–700, 2006.

71 S. Z. Li. *Markov Random Field Modeling in Image Analysis.* Advances in Pattern Recognition. Springer, 3rd edition, 2009.

72 T. Lindeberg. Scale-space. In B. Wah, editor, *Encyclopedia of Computer Science and Engineering*, volume IV, pages 2495–2504. Wiley, 2009.

73 M. Maddah, W. E. L. Grimson, S. K. Warfield, and W. M. Wells. A unified framework for clustering and quantitative analysis of white matter fiber tracts. *Medical Image Analysis*, 12:191–202, 2008.

74 M. Maddah, A. U. J. Mewes, S. Haker, W. E. L. Grimson, and S. K. Warfield. Automated atlas-based clustering of white matter fiber tracts from DTMRI. In J. Duncan and G. Gerig, editors, *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 3749 of *LNCS*, pages 188–195. Springer, 2005.

75 D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. Int'l Conf. on Computer Vision (ICCV)*, volume 2, pages 416–423, 2001.

76 Y. Masutani, S. Aoki, Z. Liu, O. Abe, and K. Ohtomo. A hybrid tensor field interpolation approach for white matter fiber tract modeling. *Int'l J. of Computer Assisted Radiology and Surgery (Proc. CARS)*, 2(Suppl. 1):S22–S24, 2007.

77 T. McGraw, B. Vemuri, R. Yezierski, and T. Mareci. Segmentation of high angular resolution diffusion MRI modeled as a field of von mises-fisher mixtures. In A. Leonardis,

H. Bischof, and A. Pinz, editors, *Proc. European Conference on Computer Vision (ECCV), Part III*, volume 3953 of *LNCS*, pages 463–475. Springer, 2006.

78 D. Merhof, M. Meister, E. Bingöl, P. Hastreiter, C. Nimsky, and G. Greiner. Generation of hulls encompassing neuronal pathways based on tetrahedralization and 3D alpha shapes. In A. Horsch, T. Deserno, H. Handels, H.-P. Meinzer, and T. Tolxdorff, editors, *Bildverarbeitung für die Medizin*, pages 308–312. Springer, 2007.

79 D. Merhof, M. Meister, E. Bingöl, C. Nimsky, and G. Greiner. Isosurface-based generation of hulls encompassing neuronal pathways. *Stereotactic and Functional Neurosurgery*, 87:50–60, 2009.

80 B. Moberts, A. Vilanova, and J. J. van Wijk. Evaluation of fiber clustering methods for diffusion tensor imaging. In *Proc. IEEE Visualization*, pages 65–72, 2005.

81 J. Modersitzki. *Numerical methods for image segmentation.* Oxford University Press, 2004.

82 S. Mori, B. J. Crain, V. P. Chacko, and P. C. M. van Zijl. Three-dimensional tracking of axonal projections in the brain by magnetic resonance imaging. *Annals of Neurology*, 45(2):265–269, 1999.

83 S. Mori and P. C. van Zijl. Fiber tracking: principles and strategies – a technical review. *NMR in Biomedicine*, 15:468–480, 2002.

84 D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational-problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

85 L. O'Donnell, W. E. L. Grimson, and C.-F. Westin. Interface detection in diffusion tensor MRI. In C. Barillot, D. Haynor, and P. Hellier, editors, *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 3216 of *LNCS*, pages 360–367. Springer, 2004.

86 L. O'Donnell and C.-F. Westin. Automatic tractography segmentation using a high-dimensional white matter atlas. *IEEE Transactions on Medical Imaging*, 26(11):1562–1575, 2007.

87 L. J. O'Donnell, C.-F. Westin, and A. J. Golby. Tract-based morphometry for white matter group analysis. *NeuroImage*, 45:832–844, 2009.

88 S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, 2003.

89 M. Otto, T. Germer, H.-C. Hege, and H. Theisel. Uncertain 2D vector field topology. *Computer Graphics Forum (Proc. Eurographics)*, 29(2):347–356, 2010.

90 E. Özarslan and T. Mareci. Generalized diffusion tensor imaging and analytical relationships between diffusion tensor imaging and high angular resolution diffusion imaging. *Magnetic Resonance in Medicine*, 50:955–965, 2003.

91 S. Pajevic, A. Aldroubi, and P. J. Basser. A continuous tensor field approximation of discrete DT-MRI data for extracting microstructural and architectural features of tissue. *Journal of Magnetic Resonance*, 154:85–100, 2002.

92 S. Pajevic and C. Pierpaoli. Color schemes to represent the orientation of anisotropic tissues from diffusion tensor data: application to white matter fiber tract mapping in the human brain. *Magnetic Resonance in Medicine*, 42(3):526–540, 1999.

93 G. J. M. Parker, C. A. M. Wheeler-Kingshott, and G. J. Barker. Estimating distributed anatomical connectivity using fast marching methods and diffusion tensor imaging. *IEEE Transactions on Medical Imaging*, 21(5):505–512, 2002.

94 O. Pasternak, N. Sochen, and P. J. Basser. The effect of metric selection on the analysis of diffusion tensor MRI data. *NeuroImage*, 49:2190–2204, 2010.

95 T. Peeters, P. Rodrigues, A. Vilanova, and B. ter Haar Romeny. Analysis of distance/similarity measures for diffusion tensor imaging. In D. H. Laidlaw and J. Weickert, editors, *Visualization and Processing of Tensor Fields – Advances and Perspectives*, pages 263–280. Springer, 2009.

96 S. M. Pizer, C. A. Burbeck, J. M. Coggins, D. S. Fritsch, and B. S. Morse. Object shape before boundary shape: Scale-space medial axes. *Journal of Mathematical Imaging and Vision*, 4:303–313, 1994.

97 F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Computer Graphics Forum*, 22(4):775–792, 2003.

98 B. Preim and D. Bartz. *Visualization in Medicine. Theory, Algorithms, and Applications.* Morgan Kaufmann, 2007.

99 M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 699–706, 2003.

100 M. Rousson, C. Lenglet, and R. Deriche. Level set and region based surface propagation for diffusion tensor MRI segmentation. In M. Šonka, I. A. Kakadiaris, and J. Kybic, editors, *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, volume 3117 of *LNCS*, pages 123–134. Springer, 2004.

101 T. Salzbrunn and G. Scheuermann. Streamline predicates. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1601–1612, 2006.

102 P. Savadjiev, J. S. Campbell, G. B. Pike, and K. Siddiqi. Streamline flows for white matter fibre pathway segmentation in diffusion MRI. In D. Metaxas et al., editor, *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 5241 of *LNCS*, pages 135–143. Springer, 2008.

103 G. Scheuermann and X. Tricoche. Topological methods in flow visualization. In C. Johnson and C. Hansen, editors, *The Visualization Handbook*, pages 341–356. Academic Press, 2004.

104 V. J. Schmithorst, M. Wilke, B. J. Dardzinski, and S. K. Holland. Cognitive functions correlate with white matter architecture in a normal pediatric population: A diffusion tensor MRI study. *Human Brain Mapping*, 26(2):139–147, 2005.

105 T. Schultz. *Feature Extraction for Visual Analysis of DW-MRI Data.* PhD thesis, Universität des Saarlandes, Germany, 2009.

106 T. Schultz, B. Burgeth, and J. Weickert. Flexible segmentation and smoothing of DT-MRI fields through a customizable structure tensor. In G. Bebis et al., editor, *Advances in Visual Computing (Proc. ISVC)*, volume 4291 of *LNCS*, pages 455–464, 2006.

107 T. Schultz, N. Sauber, A. Anwander, H. Theisel, and H.-P. Seidel. Virtual Klingler dissection: Putting fibers into context. *Computer Graphics Forum (Proc. EuroVis)*, 27(3):1063–1070, 2008.

108 T. Schultz and H.-P. Seidel. Estimating crossing fibers: A tensor decomposition approach. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Visualization)*, 14(6):1635–1642, 2008.

109 T. Schultz and H.-P. Seidel. Using eigenvalue derivatives for edge detection in DT-MRI data. In G. Rigoll, editor, *Pattern Recognition (Proc. DAGM)*, volume 5096 of *LNCS*, pages 193–202. Springer, 2008.

110 T. Schultz, H. Theisel, and H.-P. Seidel. Segmentation of DT-MRI anisotropy isosurfaces. In K. Museth, T. Möller, and A. Ynnerman, editors, *Proc. Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis) 2007*, pages 187–194, 2007.

111 T. Schultz, H. Theisel, and H.-P. Seidel. Topological visualization of brain diffusion MRI data. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Visualization)*, 13(6):1496–1503, 2007.

112 T. Schultz, H. Theisel, and H.-P. Seidel. Crease surfaces: From theory to extraction and application to diffusion tensor MRI. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):109–119, 2010.

113 J. A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, 2nd edition, 2001.

114 A. Sherbondy, D. Akers, R. Mackenzie, R. Dougherty, and B. Wandell. Exploring connectivity of the brain's white matter with dynamic queries. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):419–430, 2005.

115 J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

116 J. S. Shimony, A. Z. Snyder, N. Lori, and T. E. Conturo. Automated fuzzy clustering of neuronal pathways in diffusion tensor tracking. In *Proc. International Society of Magnetic Resonance in Medicine (ISMRM)*, volume 10, 2002.

117 S. M. Smith, M. Jenkinson, H. Johansen-Berg, D. Rueckert, T. E. Nichols, C. E. Mackay, K. E. Watkins, O. Ciccarelli, M. Z. Cader, P. M. Matthews, and T. E. J. Behrens. Tract-based spatial statistics: Voxelwise analysis of multi-subject diffusion data. *NeuroImage*, 31(4):1487–1505, 2006.

118 J.-D. Tournier, F. Calamante, D. G. Gadian, and A. Connelly. Direct estimation of the fiber orientation density function from diffusion-weighted MRI data using spherical deconvolution. *NeuroImage*, 23:1176–1185, 2004.

119 X. Tricoche. *Vector and Tensor Field Topology Simplification, Tracking, and Visualization.* PhD thesis, Fachbereich Informatik, Universität Kaiserslautern, 2002.

120 X. Tricoche, G. Kindlmann, and C.-F. Westin. Invariant crease lines for topological and structural analysis of tensor fields. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Visualization)*, 14(6):1627–1634, 2008.

121 A. Tsai, C.-F. Westin, A. O. Hero III, and A. S. Willsky. Fiber tract clustering on manifolds with dual rooted-graphs. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

122 D. S. Tuch. Q-Ball imaging. *Magnetic Resonance in Medicine*, 52:1358–1372, 2004.

123 A. Vilanova, S. Zhang, G. Kindlmann, and D. H. Laidlaw. An introduction to visualization of diffusion tensor imaging and its applications. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, pages 121–153. Springer, 2006.

124 A. N. Voineskos, L. J. O'Donnell, N. J. Lobaugh, D. Markant, S. H. Ameis, M. Niethammer, B. H. Mulsant, B. G. Pollock, J. L. Kennedy, C.-F. Westin, and M. E. Shenton. Quantitative examination of a novel clustering method using magnetic resonance diffusion tensor tractography. *NeuroImage*, 45(2):370–376, 2009.

125 Z. Wang and B. C. Vemuri. An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 228–233, 2004.

126 Z. Wang and B. C. Vemuri. DTI segmentation using an information theoretic tensor dissimilarity measure. *IEEE Transactions on Medical Imaging*, 24(10):1267–1277, 2005.

127 D. Wassermann, M. Descoteaux, and R. Deriche. Diffusion maps segmentation of magnetic resonance q-ball imaging. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2007.

128 T. Weinkauf. *Extraction of Topological Structures in 2D and 3D Vector Fields.* PhD thesis, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, 2008.

129 Y. Weiss. Segmentation using eigenvectors: A unifying view. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 975–982, 1999.

130 Y. T. Weldeselassie and G. Hamarneh. DT-MRI segmentation using graph cuts. In J. P. W. Pluim and J. M. Reinhardt, editors, *Medical Imaging 2007: Image Processing*, volume 6512 of *Proc. SPIE*, page 65121K, 2007.

131 C.-F. Westin, S. Maier, B. Khidhir, P. Everett, F. Jolesz, and R. Kikinis. Image processing for diffusion tensor magnetic resonance imaging. In *Proc. Medical Image Computing and*

Computer-Assisted Intervention (MICCAI), volume 1679 of *LNCS*, pages 441–452. Springer, 1999.

132 C.-F. Westin, S. Maier, H. Mamata, A. Nabavi, F. Jolesz, and R. Kikinis. Processing and visualization for diffusion tensor MRI. *Medical Image Analysis*, 6:93–108, 2002.

133 M. R. Wiegell, D. Tuch, H. B. Larsson, and V. J. Wedeen. Automatic segmentation of thalamic nuclei from diffusion tensor magnetic resonance imaging. *NeuroImage*, 19:391–401, 2003.

134 B. P. Witwer, R. Moftakhar, K. M. Hasan, P. Deshmukh, V. Haughton, A. Field, K. Arfanakis, J. Noyes, C. H. Moritz, E. Meyerand, H. A. Rowley, A. L. Alexander, and B. Badie. Diffusion-tensor imaging of white matter tracts in patients with cerebral neoplasm. *Journal of Neurosurgery*, 97:568–575, 2002.

135 Y. Xia, A. U. Turken, S. L. Whitfield-Gabrieli, and J. D. Gabrieli. Knowledge-based classification of neuronal fibers in entire brain. In J. Duncan and G. Gerig, editors, *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 3749 of *LNCS*, pages 205–212. Springer, 2005.

136 R. Xu. *Clustering*. Wiley, 2009.

137 P. A. Yushkevich, H. Zhang, T. J. Simon, and J. C. Gee. Structure-specific statistical mapping of white matter tracts. *NeuroImage*, 41:448–461, 2008.

138 S. Zhang, S. Correia, and D. H. Laidlaw. Identifying white-matter fiber bundles in DTI data using an automated proximity-based fiber-clustering method. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1044–1053, 2008.

139 S. Zhang, C. Demiralp, and D. H. Laidlaw. Visualizing diffusion tensor MR images using streamtubes and streamsurfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):454–462, 2003.

140 S. Zhang, D. H. Laidlaw, and G. Kindlmann. Diffusion tensor MRI visualization. In C. D. Hansen and C. R. Johnson, editors, *The Visualization Handbook*, pages 327–340. Elsevier, 2005.

141 X. Zheng and A. Pang. HyperLIC. In *Proc. IEEE Visualization*, pages 249–256, 2003.

142 X. Zheng, B. Parlett, and A. Pang. Topological lines in 3D tensor fields and discriminant hessian factorization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):395–407, 2005.

143 X. Zheng, X. Tricoche, and A. Pang. Degenerate 3D tensors. In J. Weickert and H. Hagen, editors, *Visualization and Processing of Tensor Fields*, pages 241–256. Springer, 2006.

144 L. Zhukov, K. Museth, D. Breen, R. Whitaker, and A. H. Barr. Level set segmentation and modeling of DT-MRI human brain data. *Journal of Electronic Imaging*, 12:125–133, 2003.

145 U. Ziyan, D. Tuch, and C.-F. Westin. Segmentation of thalamic nuclei from DTI using spectral clustering. In R. Larsen, M. Nielsen, and J. Sporring, editors, *Proc. Medical Image Processing and Computer-Assisted Intervention (MICCAI)*, volume 4191 of *LNCS*, pages 807–814. Springer, 2006.

146 U. Ziyan and C.-F. Westin. Joint segmentation of thalamic nuclei from a population of diffusion tensor MR images. In D. Metaxas, L. Axel, G. Fichtinger, and G. Székely, editors, *Proc. Medical Image Processing and Computer-Assisted Intervention (MICCAI)*, volume 5241 of *LNCS*, pages 279–286. Springer, 2008.

Previewing Volume Decomposition Through Optimal Viewpoints*

Shigeo Takahashi[1], Issei Fujishiro[2], Yuriko Takeshima[3], and Chongke Bi[1]

1 Graduate School of Frontier Sciences, The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8561, Japan
takahashis@acm.org, bichongke@visual.k.u-tokyo.ac.jp

2 Department of Information and Computer Science, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan
fuji@ics.keio.ac.jp

3 Institute of Fluid Science, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai 980-8577, Japan
takesima@vis.ifs.tohoku.ac.jp

Abstract

Understanding a volume dataset through a 2D display is a complex task because it usually contains multi-layered inner structures that inevitably cause undesirable overlaps when projected onto the display. This requires us to identify feature subvolumes embedded in the given volume and then visualize them on the display so that we can clarify their relative positions. This article therefore introduces a new feature-driven approach to previewing volumes that respects both the 3D nested structures of the feature subvolumes and their 2D arrangement in the projection by minimizing their occlusions. The associated process begins with tracking the topological transitions of isosurfaces with respect to the scalar field, in order to decompose the given volume dataset into feature components called interval volumes while extracting their nested structures. The volume dataset is then projected from the optimal viewpoint that archives the best balanced visibility of the decomposed components. The position of the optimal viewpoint is updated each time when we peel off an outer component with our interface by calculating the sum of the viewpoint optimality values for the remaining components. Several previewing examples are demonstrated to illustrate that the present approach can offer an effective means of traversing volumetric inner structures both in an interactive and automatic fashion with the interface.

1998 ACM Subject Classification I.3.8 [Computer Graphics]: Applications

Keywords and phrases Interval volumes, viewpoint selection, feature-driven approach, volume peeling, nested structures

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.346

1 Introduction

The most common media for our visual communication have been 2D projections such as photographs, paintings, and computer displays. While the 2D projections are no doubt powerful means of conveying visual information, they still suffer from the inherent limitation on the dimensionality of the space where the information is presented, especially in the

* This work has been partially supported by Japan Society of the Promotion of Science under Grants-in-Aid for Young Scientists (B) No. 17700092, and the Ohkawa Foundation.

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 346–359
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

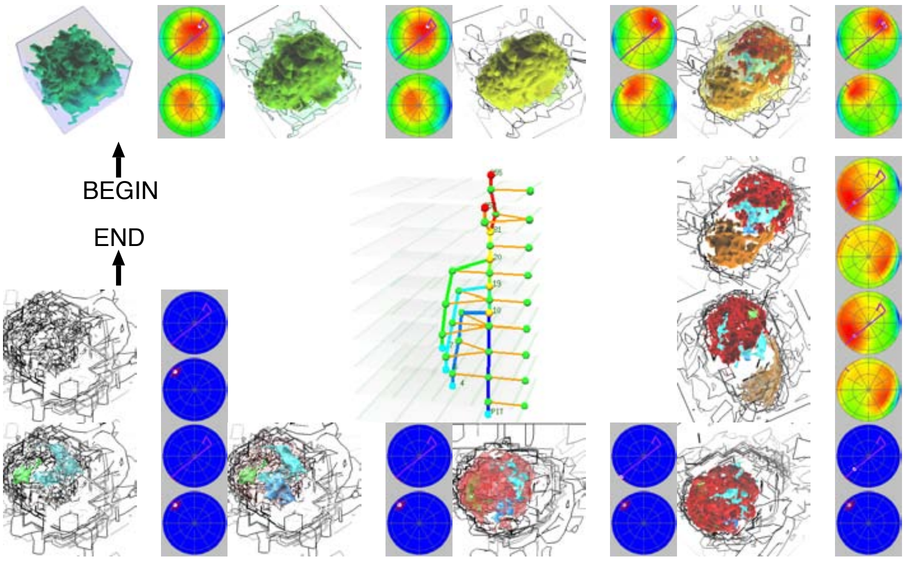


Figure 1 Scenario for decomposing the volume of "sheep heart". Each frame consists of the snapshot of volume decomposition preview and its associated map of viewpoint entropy. The graph at the center shows the corresponding interval volume structure.

case of 3D solid volumes. This is due to the fact that the volume datasets usually contain multi-layered inner structures where specific features cannot be easily identified through simple observations from the outside. Even if the associated features are identified, they cannot be projected clearly onto the 2D image plane because they are usually accompanied with undesirable overlaps while transparent rendering alleviates the problem to a certain extent.

This article therefore presents an approach for previewing volume inner structures by respecting both their 3D nested structures in the volume and their 2D arrangements in the projection. The approach consists of two ingredients: an algorithm for extracting feature components from the given volume so that we can peer into the volume by removing an outer component one by one, and an algorithm for locating an optimal viewpoint that provides the best arrangement of the feature components on the 2D projection. Our contribution lies in the combination of these two feature-driven algorithms because this is, to our knowledge, the first attempt to respect the underlying feature components consistently in both analyzing and previewing the inner structures of the volume.

The first algorithm has been implemented in our previous study as an interface called *interval volume decomposer* [28]. The interface decomposes an entire volume into feature subvolumes by tracking the topological transitions of the corresponding isosurface with respect to the scalar field, and then by constructing a graph structure called *a level-set graph* that represents the topological skeleton of the given volume. Furthermore, the link of the level-set graph corresponds to a feature subvolume called *an interval volume* [10, 13] in this framework. This enables systematic decomposition of the given volume from outside to inside by referring to the constructed level-set graph.

The second algorithm has been implemented in our previous formulation of viewpoint entropy for volumes [29]. Actually, this formation takes account of the combinations of feature interval volumes and evaluates the optimality of their arrangements in 2D projections. In each decomposition step, we can smoothly change the associated viewpoint by updating the map of the entropy values over the viewing sphere.

Fig. 1 shows an example where the inner structure of a sheep heart dataset [23] is effectively presented by incorporating the above two algorithms into our previewing interface. As shown in the figure, the decomposed feature subvolumes are rendered in different colors and the outermost subvolume is faded out one by one while its silhouette gradually appears. Every time one subvolume is taken away, the entropy map over the viewing sphere together with the position of the next optimal viewpoint is updated. In Fig. 1, the upper and lower halves of the viewing sphere are rendered with the color-coded distribution of the viewpoint entropy, where warm colors represent higher entropy regions. Note that the purple line indicates the viewpoint path over the viewing sphere through the entire preview of the decomposition process.

This article is organized as follows: Section 2 surveys previous studies related to the present approach. Section 3 describes an algorithm for decomposing an entire volume into feature subvolumes by referring to its associated level-set graph. Section 4 explains a formulation of the viewpoint entropy for volumes by taking account of the characteristic combinations of the decomposed subvolumes. An interface for systematically decomposing a given volume dataset together with its optimal viewpoint updates is presented in Section 5. Section 6 concludes this article and refers to our future extensions.

2 Related Work

Our approach to volume previewing is related to several areas of research in visualization, computer graphics, and computer vision. Among the research areas, this section gives brief surveys on interactive volume exploration and optimal viewpoint selection.

2.1 Interactive Volume Exploration

Exploring the inside of a volume interactively makes it possible to effectively understand its underlying complex structure. *Volume sculpting* techniques are among such approaches where users can analyze the multi-layered inner structure with interactive carving and sawing operations [11, 36]. Volume peeling can be thought of as one of the principal editing processes used in the volume sculpting systems.

On the other hand, an idea of *haptic rendering* has emerged as the virtual reality technologies have been developed. Actually, the haptic rendering is a process of generating forces and torques in response to user interactions with virtual objects through haptic devices [15], and has also been introduced to volume exploration systems [1]. Anatomical metaphors for medial volumes obtained by CT and MRI scans have also inspired many methods for exploring volume datasets [19, 20]. In addition to these software-based approaches, hardware-assisted volume deformation techniques have also been developed [18, 38].

Another interesting approach is to preform the rigorous analysis of the given volume prior to the volume exploration. Pioneering work was done by Bajaj et al. [2], where they used contour trees to effectively explore the volume inside. Here, the contour tree represents a level-set graph of the given volume, which delineates the topological transitions of an isosurface according to the change of the scalar field value as shown in Fig. 2. Thus the contour trees have been intensively used to as a tool for exploring the isosurface trajectories in subsequent studies [5, 31, 37] while the associated algorithms for computing the level-set graphs have been also developed [34, 22, 6].

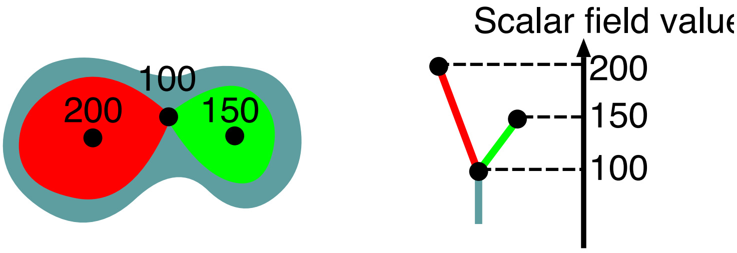


Figure 2 Isosurface transitions (on the left) and the corresponding level-set graph (on the right). The nodes of the level-set graph represent critical points in a volume, and are arranged according to their scalar field values in this article. The same color is assigned to a link of the level-set graph (on the right) and its corresponding subvolume (on the left).

2.2 Viewpoint Selection

The viewpoint selection problem has been studied rather by researchers in computer vision. One of such pioneering studies on this problem was conducted by Koenderink et al. [17], where they defined an aspect graph that partitions the viewing sphere that surrounds the target object into *aspects* by identifying equivalent views of the object edges in a topological sense. The aspect graph representation has been intensively studied as a tool for object recognition [7], while its automatic computation is still the subject of ongoing research [25]. On the other hand, finding a set of optimal camera positions requires some criteria for evaluating the goodness of each classified view position. This problem of planning camera positions is referred to as the *next best view (NBV)* problem in the computer vision literature, and has been investigated by a number of researchers, for example in [12, 24]. The configuration of such optimal viewpoints is also useful for acquiring a minimal set of textures in the image synthesis techniques such as image-based rendering [8, 14].

In the computer graphics applications, several methods have been proposed to locate the single best viewpoint for 3D polyhedral meshes [16, 3]. Among these methods, the most reliable solution especially for the case of general 3D meshes is the *viewpoint entropy*, which is formulated by Vázquez et al. [35] to evaluate the balance of visible faces in 2D projected images.

Locating optimal viewpoints for volumes apparently presents another problem because the volumes usually involve multi-layered inner structures and can be thought of as one-dimension higher objects than surface meshes. Recently, this problem was explored simultaneously but independently by Bordoloi et al. [4] and Takahashi et al. [29]. Bordoloi et al. evaluated the balance between the contributions of voxels to pixels in the resultant image, while Takahashi et al. tried to find the well-balanced arrangement of feature subvolumes on the 2D projected image.

3 Interval Volume Decomposer

This section describes an algorithm that tracks the level-set graph for systematically decomposing a given volume into a set of feature subvolumes, followed by a framework for previewing the volume in our interface called *interval volume decomposer* [28].

3.1 Interval Volumes

For extracting the feature subvolumes from the input dataset, our algorithm first constructs the level-set graph that tracks the topological transitions of an isosurface according to the scalar field, and then identifies each link of the graph with a feature subvolume. Actually, in this case, the subvolume corresponds to an interval volume (IV) bounded by critical isosurfaces. The concept of IV is formulated by Fujishiro et al. [10] and Guo [13] as a generalization of the isosurface, and defined as a subvolume that corresponds to some range of the scalar field value. Our framework employs a set of such IVs obtained by referring to the constructed level-set graph, where the IVs also play an important role in selecting the optimal viewpoint for seeking the best of their arrangements in 2D projection (Section 4.2).

3.2 Level-set Graph Construction

For constructing the level-set graph, we have developed an algorithm called *topological volume skeletonization* [31], which tracks the topological transitions of an isosurface according to the scalar field. Actually, the algorithm is constructed by combining the algorithm of Carr et al. [6] for tracking the number of isosurface connected components, the algorithm of Pascucci et al. [22] for identifying the genus of each isosurface component, and the algorithm of Takahashi et al. [31] for IV clustering. Refer to [30] for the practical implementation of this algorithm.

3.3 Isosurface Inclusion Relationships at Saddles

The constructed level-set graph illuminates a systematic decomposition of the given volume by relating its links to the decomposed feature IVs. This implies that peeling the decomposed IVs from outside to inside requires to locate their inclusion relationships. According to [32], such inclusion relationships can be retrieved by systematically traversing the level-set graph from the end node that corresponds to the exterior surface of the given volume.

A node of the level-set graph represents a critical point at which an isosurface component has topological evolution. Mathematically, critical points in volumes are classified into C_3 (maxima), C_2 (saddles), C_1 (saddles), and C_0 (minima) according to their indices, where the index represents the number of negative eigenvalues of the corresponding Hessian matrix. Taking account of the isosurface embeddings in 3D space allows us to classify the isosurface transitions at C_2 and C_1 into four different isosurface transitions as shown in Fig. 3 [31]. Here, we call an isosurface *solid* if it expands as the scalar field value decreases while *hollow* if it shrinks. The leftmost and rightmost columns in Fig. 3 illustrate subgraphs around the critical points of C_1 and C_2 where a different color is assigned to each link according to whether the corresponding isosurface is solid or hollow. This figure suggests that IV inclusions occur only in the two transition paths of the row (B) for both C_2 and C_1.

3.4 Determining the IV Decomposition Order

In the present algorithm, the IV decomposition order is extracted by tracing the links of the level-set graph from the end node that represents the exterior boundary of the given volume. Fig. 4 shows an example where we determine the decomposition order by tracing the level-set graph. Note that the associated algorithm prepares two FIFO queues Q_{solid} and Q_{hollow} for solid and hollow links, respectively, and adds links traversed in the upward direction to Q_{solid} while those traversed in the downward direction to Q_{hollow}. In addition,

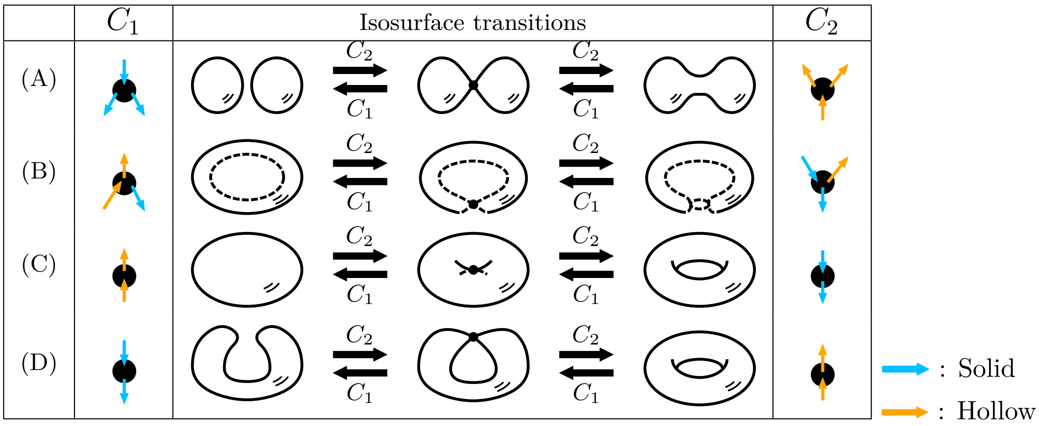


Figure 3 Classification of isosurface transitions at saddle critical points of C_2(join) and C_1(split) according to the spatial configuration in 3D space. The horizontal arrows in the central column indicate isosurface transitions as the scalar field value reduces. Different colors are assigned to solid and hollow links in the level-set graph.

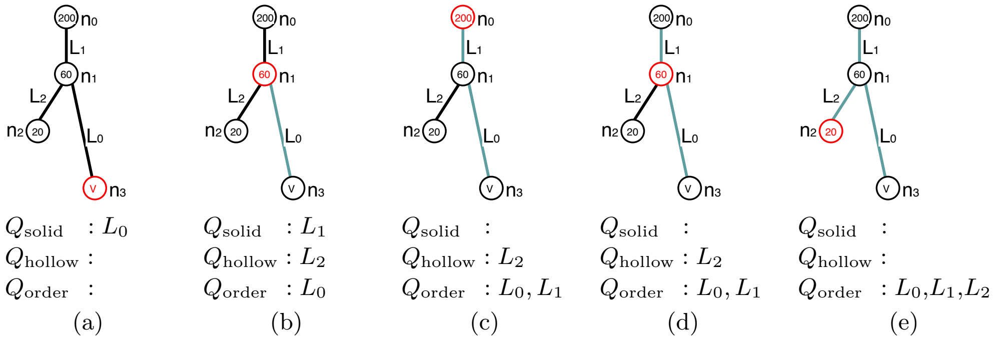

200 n0
L1
60 n1
L2
n2 20
L0
v n3
Q_{solid} : L_0
Q_{hollow} :
Q_{order} :
(a)
Q_{solid} : L_1
Q_{hollow} : L_2
Q_{order} : L_0
(b)
Q_{solid} :
Q_{hollow} : L_2
Q_{order} : L_0, L_1
(c)
Q_{solid} :
Q_{hollow} : L_2
Q_{order} : L_0, L_1
(d)
Q_{solid} :
Q_{hollow} :
Q_{order} : L_0,L_1,L_2
(e)

Figure 4 Steps for extracting the IV decomposition order.

Q_{order} is introduced to retain the links that represent the IV decomposition order. Initially, Q_{hollow} and Q_{order} are empty while Q_{solid} has the link L_0 to be examined first (Fig. 4(a)).

The actual tracing process starts with the traversal from the node n_3 to n_1 through the link L_0 as shown in Fig. 4(a), where L_0 is solid because the it is connected to the exterior surface of the given volume. This lets us delete the link L_0 from Q_{solid} and add it to Q_{order}. Since the associated algorithm identifies the node n_1 as (B)-C_1 type according to the classification in Fig. 3, it inserts L_1 to Q_{solid} and L_2 to Q_{hollow} (Fig. 4(b)). The upward tracing process continues to handle all the links in Q_{solid} until it becomes empty (Fig. 4(c)). The algorithm then begins to handle the links in Q_{hollow}. In this case, it resumes the traversal from n_1 to n_2 through L_2 in the downward direction (Fig. 4(d)). Finally, the tracing process is completed by storing the decomposition order of IVs in Q_{order} while identifying the inclusion relationship between L_0 and L_2 (Fig. 4(e)).

4 Viewpoint Entropy for Volumes

Viewpoint selection for previewing volumes needs criteria for calculating viewpoint optimality for the given volume. The present algorithm calculates such viewpoint goodness by evaluating

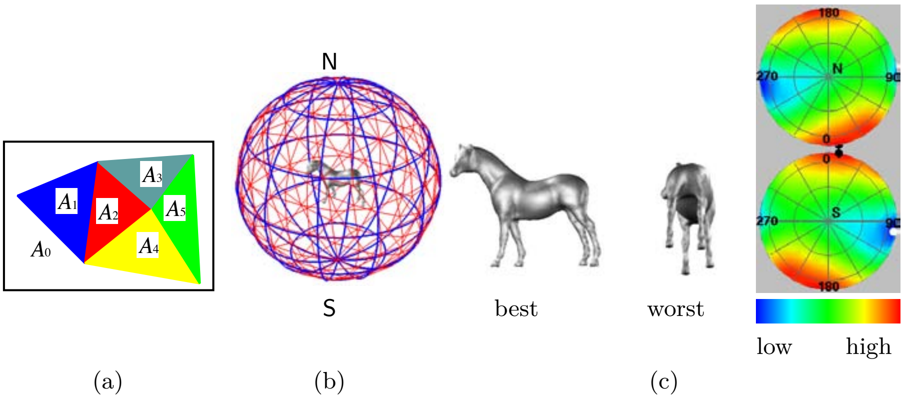


Figure 5 (a) Assignment of different colors to visible faces on the 2D screen. (b) A reference mesh (in red) on the viewing sphere (in blue) for viewpoint samples. (c) The viewpoint entropy distributions and the associated best and worst views for the horse model.

the arrangement of decomposed IVs in the 2D projection using the conventional surface-based technique.

4.1 Formulation of the Viewpoint Entropy for Surfaces

As the criteria for evaluating optimal views of surface meshes, we employ the *viewpoint entropy* formulated by Vázquez et al. [35], which searches for a well-balanced distribution of visible faces using the entropy measure. In our algorithm, we modified the original formulation of the viewpoint entropy so that we can evaluate the optimality of view directions under orthographic projections. Suppose that the j-th face of a given 3D mesh has a visible projected area A_j ($j = 1, \ldots, m$) on the 2D screen, while A_0 denotes the background area as shown in Fig. 5(a). Thus, the total area of the 2D screen S can be calculated as $S = \sum_{j=0}^{m} A_j$. The normalized version of the viewpoint entropy E can be defined as

$$E = -\frac{1}{\log_2(m+1)} \sum_{j=0}^{m} \frac{A_j}{S} \log_2 \frac{A_j}{S}. \tag{1}$$

Note that the quantity E in Equation (1) becomes larger as the corresponding viewpoint achieves more balanced distribution of face visibility, while it ranges from 0.0 (when all the faces are invisible) to 1.0 (when all the faces share the same area).

The actual calculation of the viewpoint entropy in Equation (1) is carried out as follows [3]: Given a viewpoint, the projected area of each face of the target mesh is obtained by counting the number of pixels that belong to the face. Assigning a different color to each face helps us discriminate one face from the others as shown in Fig. 5(a). The associated viewpoint entropy is evaluated at vertices of another reference mesh (in red) that covers the viewing sphere (in blue) as shown in Fig. 5(b). Fig. 5(c) shows the distributions of the viewpoint entropies in Equation (1) over the viewing sphere, together with the associated best and worst views of the 3D horse model, where the best viewpoint captures its silhouettes clearly. The associated top and bottom disks are the projections of the viewpoint entropy distributions on the viewing sphere seen from the top (the North pole) and the bottom (the South pole), respectively. Here, each disk is color-coded by referring to the color legend where the color changes over blue, green, yellow, and red according to the increase in the entropy, and the black and white spots indicate the best and worst locations of the viewpoints, respectively.

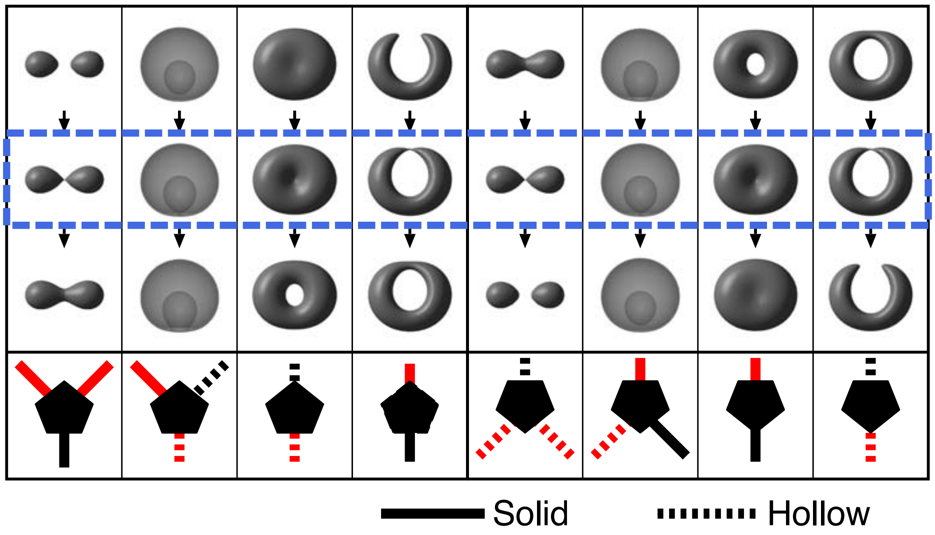


Figure 6 The combinations of IVs for the viewpoint evaluation and their associated links in level-set graphs.

4.2 Formulation of the Viewpoint Entropy for Volumes

For locating the optimal viewpoint for volumes, we introduce our feature-driven approach [29] where we calculate the viewpoint entropies of feature components and then find the global compromise between them, by taking advantage of the aforementioned surface-based entropy formulation. As the feature components, our approach employs combinations of the decomposed feature IVs, each of which constitutes an isosurface join or split of some specific type. Fig. 6 depicts such combinations of IVs where the corresponding links are drawn in red. Note that the figure depicts subgraphs around saddle critical points in the level-set graph when the links are arranged from top to bottom with respect to the scalar field.

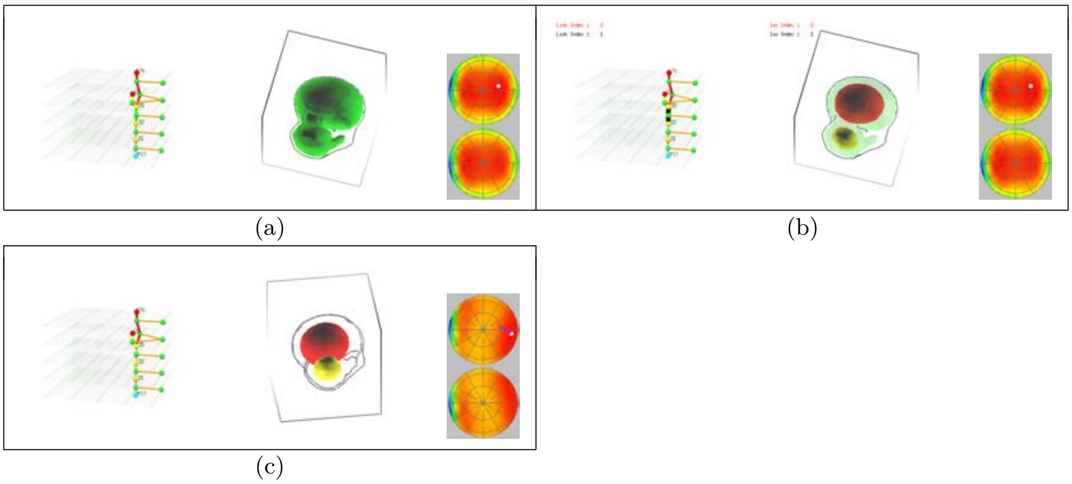
The viewpoint entropy for volumes is formulated as follows: Suppose that we separately evaluate each IV of some specific combination as shown in Fig. 6. The viewpoint entropy for the IV can be calculated using Equation (1) by identifying the visible faces of the IV. However, if the faces of the IV are partially occluded by the other IV in the combination, the occluded regions are assumed be painted in the background color when evaluating the entropy. This handling together with Equation (1) allows us to avoid undesirable occlusions between the IVs in the combination. For evaluating the globally optimal viewpoint, our algorithm calculates the weighted sum of the viewpoint entropies, which is given as

$$E = \sum_{i=1}^{n} \frac{\lambda_i}{L} E_i, \tag{2}$$

by calculating Equation (1) for the i-th IV as E_i. Here, λ_i is the weight value for the i-th IV and $L = \sum_{i=1}^{n} \lambda_i$. In addtition, n is the number of IV. In our framework, λ_i is calculated using an opacity transfer function for direct volume rendering, as follows:

$$\lambda_i = \frac{1}{M_i} \sum_{j=1}^{M_i} o(\boldsymbol{x}_j) \tag{3}$$

where $o(\boldsymbol{x}_j)$ is an opacity value at the j-th voxel sample $\boldsymbol{x}_j$, and M_i is the number of voxel samples $\boldsymbol{x}_j$ contained in the i-th IV. Here, the opacity transfer function is used to assign large


(a)

(b)

(c)

Figure 7 Screen snapshots of the interface where IVs are peeled off interactively: (a) The outermost IV (in green) is specified by a pointing device. (b) The selected IV becomes transparent (in the middle) while its corresponding link is emphasized (on the left). (c) After the selection is confirmed, the IV is removed and its silhouettes are displayed in place (in the middle) while the corresponding link is faded (on the left). The entropy distribution is updated and the viewpoint moves to the next optimal position along the purple path (on the right).

weights to the feature IVs we want to emphasize. Needless to say, more sophisticated transfer functions, such as multi-dimensional transfer functions [33] can be specified in Equation (3), without any modifications to the remainder of the present framework.

5 Interface for Previewing Volume Decomposition

This section describes the implementation of our interface that systematically decomposes the given volume from outside to inside while tracking the movement of its optimal viewpoint over the viewing sphere. The remainder of this section is devoted to describing how to accomplish the IV decomposition using the present interface, either interactively (Section 5.1) or automatically (Section 5.2).

5.1 Interactive IV Decomposition

The interactive IV decomposition proceeds by specifying a feature IV to be excluded using a pointing device such as a mouse. Fig. 7 shows screen snapshots of our interface where it previews the proton atom-hydrogen atom collision dataset [9]. The leftmost window is for displaying the level-set graph together with its IV inclusion relationships. The middle window is for presenting the decomposed feature IVs where a user can peel off IVs from outside to inside one by one. The rightmost window is for exhibiting the color-coded distribution of the viewpoint entropy calculated from the remaining combinations of feature IVs. The two left windows provide an interface for selecting an individual IV using a pointing device while the rightmost window relates the smooth movement of the viewpoint. In addition, the same color is assigned to the link of the level-set graph and its corresponding feature IV in order to visualize the correspondence between them.

As shown in Fig. 7(a), a user specifies the outermost IV to be removed either by clicking the green subvolume in the middle window or by clicking the green link in the left window. Furthermore, the interface indicates the position of the current optimal viewpoint by a black spot on the projection of the upper viewing hemisphere. Note that the distribution of the viewpoint entropy is calculated as the sum of the viewpoint entropies assigned to the remaining combinations of IVs. The interface changes the status of the selected IV by rendering it transparently in the middle window and emphasizing the corresponding link of the level-set graph in the left window, as shown in Fig. 7(b). At this point, the user can look at the structure of the interior IVs through the outer transparent one. When the user confirms this selection, the interface finally removes the selected IV and leaves its silhouettes instead in the middle window, while the corresponding link in the left window is grayed out, as shown in Fig. 7(c). In addition, the interface recalculates the distribution of the viewpoint entropy for the updated combinations of the remaining IVs in the right window, and changes the view of the decomposed IVs in the middle window so that we can smoothly move the corresponding viewpoint from the previous best viewpoint to the next one. The path of the viewpoint movement is obtained by the spherical linear interpolation based on quaternion representations [26]. As demonstrated in Fig. 7, the interface provides an effective means of eliminating decomposed IVs one by one from outside to inside together with the comfortable viewpoint changes, while allowing the user to fully understand the multi-layered inner structures of the target volume.

5.2 Automatic IV Decomposition

Furthermore, the interface automatically generates a scenario for peeling off the IVs from outside to inside using the IV decomposition order as an animation, while smoothly changing its viewpoint positions. Figs. 1, 8, and 9 present scenarios for volume peeling using the present interface. In these scenarios, the decomposed IVs gradually disappear one by one in accordance with the decomposition order by tracking the constructed level-set graph (Section 3). Furthermore, the silhouettes of the IVs will gradually become conspicuous as the corresponding IVs become transparent. Throughout the decomposition scenario, the interface changes the viewpoint position along the path that follows the optimal positions obtained in each step of eliminating the outermost IV.

As described in Section 1, Fig. 1 shows a scenario in which the volume of the sheep heart $(177 \times 177 \times 177)$ [23] is dissected from outside to inside using the interface. This example demonstrates that the present interface works well and provides a systematic process of decomposing nested structures even for such an anatomical volume, together with the smooth change of the viewpoint indicated in purple in the figure. Fig. 8 visualizes a simulated dataset $(41 \times 41 \times 41)$, where the two-body distribution probability of a nucleon in the atomic nucleus ^{16}O is computed [21]. This dataset contains a two-fold nested structure of IVs and reveals its attractive interior while the viewpoint movement along the purple path first provides its side view and then its top view. Fig. 9 represents another dataset $(129 \times 129 \times 129)$, which is obtained by simulating the antiproton-hydrogen atom collision at intermediate collision energy [27]. Note that the interface successfully resolves the complicated structure of this dataset even though it contains a four-fold nested structure of IVs. The viewpoint path (in purple) effectively tracks the best views for the associated decomposition steps while it goes around through the optimal viewpoint positions scattered over the viewing sphere. These results demonstrate the potential and feasibility of the present framework.

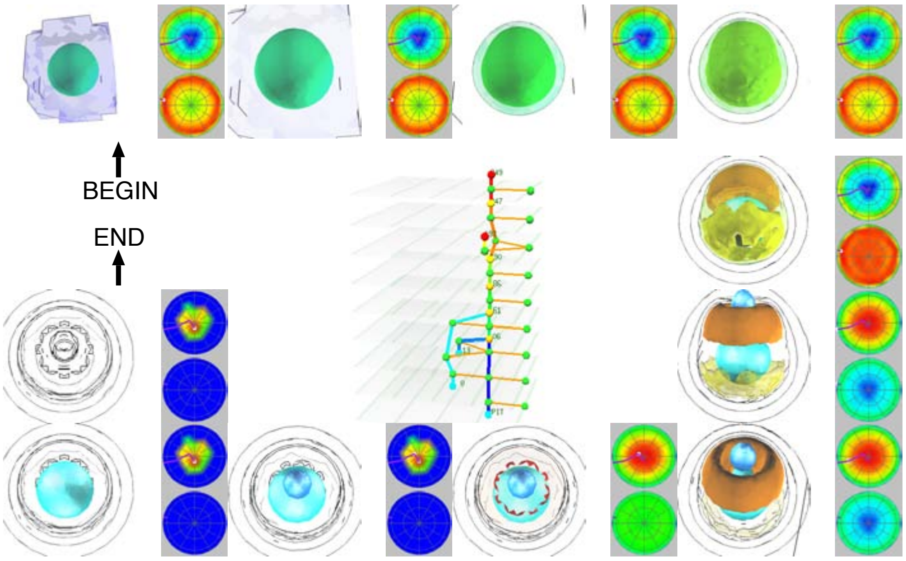


Figure 8 Scenario for decomposing the volume of "nucleon". Each frame consists of the snapshot of volume decomposition preview and its associated map of viewpoint entropy. The graph at the center shows the corresponding interval volume structure.

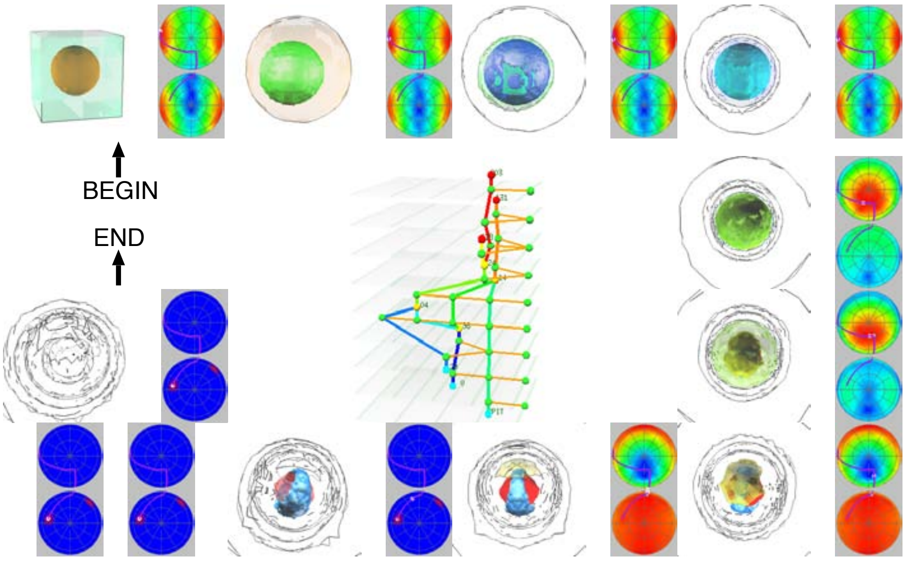


Figure 9 Scenario for decomposing the volume of "antiproton-hydrogen atom collision". Each frame consists of the snapshot of volume decomposition preview and its associated map of viewpoint entropy. The graph at the center shows the corresponding interval volume structure.

6 Conclusion and Future Work

This article has presented an approach to previewing volume datesets respecting both its 3D nested inner structures and their 2D arrangements in the projections. This is accomplished by first constructing the level-set graph that identifies the feature subvolumes by tracking the topological transitions of an isosurface, and then projecting them onto 2D screen so that their arrangements become optimal with less occlusions. Interval volumes (IVs) are introduced as the feature subvolumes where each IV corresponds to a link of the level-set graph, which makes it possible to investigate nested relationships between the decomposed IVs. An algorithm for locating the optimal viewpoint for volumes is also presented that tries to minimize the occlusions between feature interval volumes by taking account of their characteristic combinations inheriting from the level-set graph. Implementation of the associated interface together with several previewing examples is included to demonstrate the feasibility of the present approach.

Future extensions include improving the interface so that it can provide multiple views of the target volumes so that the corresponding view frustums effectively cover the entire 3D domain where the target volume is defined. Currently we use the transfer function to assign a large weight to some specific interval volume. However, we can interactively assign specific weights to the interval volumes with the help of interface and confirm its validity by calculating the corresponding viewpoint position. Furthermore, editing several effects such as coloring, lighting, and rendering styles with the interface is one of the interesting future themes.

References

1 R. S. Avila and L. M. Sobierajski. A haptic interaction method for volume visualization. In *Proc. of IEEE Visualization '96*, pages 197–205,485, 1996.

2 C. L. Bajaj, V. Pascucci, and D. R. Schikore. The contour spectrum. In *Proc. of IEEE Visualzation '97*, pages 167–173, 1997.

3 P. Barral, G. Dorme, and D. Plemenos. Scene understanding techniques using a virtual camera. In *Proc. of Eurographics '00 Short Presentations*, 2000.

4 U. D. Bordoloi and H.-W. Shen. View selection for volume rendering. In *Proc. of IEEE Visualization 2005*, pages 487–494, 2005.

5 H. Carr and J. Snoeyink. Path seeds and flexible isosurfaces using topology for exploratory visualization. In *Proc. of Joint Eurographics-IEEE TCVG Symp. on Visualization*, pages 49–58, 285, 2003.

6 H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.

7 C. M. Cyr and B. B. Kimia. 3D object recognition using shape similarity-based aspect graph. In *Proc. of Int. Conference on Computer Vision*, pages 254–261, 2001.

8 S. Fleishman, D. Cohen-Or, and D. Lischinski. Automatic camera placement for image-based modeling. *Computer Graphics Forum*, 19(2):101–110, 2000.

9 I. Fujishiro, T. Azuma, Y. Takeshima, and S. Takahashi. Volume data mining using 3D field topology analysis. *IEEE CG&A*, 20(5):46–51, 2000.

10 I. Fujishiro, Y. Maeda, H. Sato, and Y. Takeshima. Volumetric data exploration using interval volume. *IEEE Trans. on Visualization and Computer Graphics*, 2(2):144–155, 1996.

11 T. A. Galyean and J. F. Hughes. Sculpting: An interactive volumetric modeling technique. In *Computer Graphics (Proc. of Siggraph '91)*, pages 267–274, 1991.

12 K. D. Gremban and K. Ikeuchi. Planning multiple observations for object recognition. *International Journal of Computer Vision*, 12:137–172, 1994.

13 B. Guo. Interval set: A volume rendering technique generalizing isosurface extraction. In *Proc. of IEEE Visualization '95*, pages 3–10, 1995.

14 Y. Iwakiri and T. Kaneko. PC-based real-time texture painting on real world objects. *Computer Graphics Forum*, 20(3):105–113, 2001.

15 H. Iwata and H. Noma. Volume haptization. In *Proc. of IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, pages 16–23, 1993.

16 T. Kamada and S. Kawai. A simple method for computing general position in displaying three-dimensional objects. *Computer Vision, Graphics, and Image Processing*, 41(1):43–56, 1988.

17 J. J. Koenderink and A. J. van Doorn. The singularities of the visual mapping. *Biological Cybernetics*, 32:211–216, 1979.

18 Y. Kurzion and R. Yagel. Interactive space deformation with hardware-assisted rendering. *IEEE CG&A*, 17(5):66–77, 1997.

19 W. E. Lorensen. Geometric clipping using boolean textures. In *Proc. of IEEE Visualization '93*, pages 268–274, 1993.

20 M .J. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volume data. In *Proc. of IEEE Visualization 2003*, pages 401–408, 2003.

21 M. Meißner. Web Page [http://www.volvis.org/].

22 V. Pascucci and K. Cole-McLaughlin. Efficient computation of the topology of level sets. In *Proc. of IEEE Visualization 2002*, pages 187–194, 2002.

23 H. Pfister, et al. The transfer function bake-off. *IEEE CG&A*, 21(3):16–22, 2001.

24 R. Pito. A solution to the next best view problem for automated surface aquisition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(10):1016–1030, 1999.

25 I. Shimshoni and J. Ponce. Finite-resolution aspect graphs of polyhedral objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):315–327, 1997.

26 K. Shoemake. Animating rotation with quaternion curves. In *Proc. of Siggraph '85*, pages 245–254, 1985.

27 R. Suzuki, H. Sato, and M. Kimura. Antiproton-hydrogen atom collision at intermediate energy. *IEEE Computing in Science and Engineering*, 4(6):24–33, 2002.

28 S. Takahashi, I. Fujishiro, and Y. Takeshima. Interval volume decomposer: A topological approach to volume traversal. In *Proc. of SPIE Conference on Visualization and Data Analysis 2005*, volume 5669, pages 103–114, 2005.

29 S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita. A feature-driven approach to locating optimal viewpoints for volume visualization. In *Proc. of IEEE Visualization 2005*, pages 495–502, 2005.

30 S. Takahashi, G. M. Nielson, Y. Takeshima, and I. Fujishiro. Topological volume skeletonization using adaptive tetrahedralization. In *Proc. of Geometric Modeling and Processing 2004*, pages 227–236, 2004.

31 S. Takahashi, Y. Takeshima, and I. Fujishiro. Topological volume skeletonization and its application to transfer function design. *Graphical Models*, 66(1):22–49, 2004.

32 S. Takahashi, Y. Takeshima, I. Fujishiro, and G. M. Nielson. Emphasizing isosurface embeddings in direct volume rendering. In G.-P. Bonneau, T. Ertl, and G. M. Nielson, editors, *Scientific Visualization: The Visual Extraction of Knowledge from Data*, pages 185–206. Springer-Verlag, 2005.

33 Y. Takeshima, S. Takahashi, I. Fujishiro, and G. M. Nielson. Introducing topological attributes for objective-based visualization of simulated datasets. In *Proc. of Volume Graphics 2005*, pages 137–145, 236, 2005.

34 M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *Proc. of 13th ACM Symp. on Computational Geometry*, pages 212–220, 1997.

35 P.-P. Vázquez, M. Feixas, M.Sbert, and W. Heidrich. Viewpoint selection using view entropy. In *Proc. of Vision Modeling and Visualization Conference (VMV01)*, pages 273–280, 2001.

36 S. Wang and A. E. Kaufman. Volume sculpting. In *Proc. of the 1995 Symp. on Interactive 3D Graphics*, pages 151–156, 1995.

37 G. H. Weber and G. Scheuermann. Automating transfer function design based on topology analysis. In *Geometric Modeling for Scientific Visualization*, pages 293–306. Springer-Verlag, 2004.

38 D. Wiskopf, K. Engel, and T. Ertl. Volume clipping via per-fragment operations in texture-based volume visualization. In *Proc. of IEEE Visualization 2002*, pages 93–100, 2002.

Modeling Multiresolution 3D Scalar Fields through Regular Simplex Bisection

Kenneth Weiss[1] **and Leila De Floriani**[2]

1 **Department of Computer Science**
University of Maryland, College Park, MD
`kweiss@cs.umd.edu`

2 **Dipartimento di Informatica e Scienze dell´Informazione**
Università di Genova, Genova, Italy
`deflo@disi.unige.it`

Abstract

We review modeling techniques for multiresolution three-dimensional scalar fields based on a discretization of the field domain into nested tetrahedral meshes generated through *regular simplex bisection*. Such meshes are described through hierarchical data structures and their representation is characterized by the modeling primitive used. The primary conceptual distinction among the different approaches proposed in the literature is whether they treat tetrahedra or clusters of tetrahedra, called diamonds, as the modeling primitive. We first focus on representations for the modeling primitive and for nested meshes. Next, we survey the applications of these meshes to modeling multiresolution 3D scalar fields, with an emphasis on interactive visualization. We also consider the relationship of such meshes to octrees. Finally, we discuss directions for further research.

1998 ACM Subject Classification I.3.5 Curve, surface, solid, and object representations, I.3.6 Graphics data structures and data types.

Keywords and phrases Tetrahedral bisection, hierarchy of diamonds, mesh-based multiresolution models, regular simplex bisection, scientific visualization.

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.360

1 Introduction

Hierarchical domain decompositions play a fundamental role in scientific analysis and visualization. For example, discrete scalar fields are often sampled at a set of points within a problem domain. Decomposing the domain into a polyhedral mesh enables efficient approximations to the underlying scalar field at new locations. In many contexts, including CT and MRI data, or numerical simulations, the field is sampled at fixed intervals along a cubic grid.

The structure of a scalar field is often analyzed through its *isosurfaces* or by considering subvolumes enclosed between two such surfaces, known as *interval volumes*. However, due to the increasing size of volumetric datasets enabled by advances in computing power and sensing technologies, it is difficult to interactively analyze such meshes at full resolution. This often necessitates analysis and visualization on simplified versions of the underlying field. An important restriction in many applications is the need for a *crack-free* (also referred to as *conforming*) representations of the simplified domain decomposition. This ensures minimal continuity requirements when interpolating the domain across the cells of the mesh.

This has led to the development of *mesh-based multiresolution models* for adaptively controlling and adjusting the *level of detail* in the resolution of the discretized domain. Such

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 360–377
DAGSTUHL FOLLOW-UPS Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

models enable focusing on the more complex regions of the dataset, while retaining access to the original samples if necessary.

When the domain under consideration is a regularly sampled scalar field, exploiting the regularity can yield compact models, which support efficient queries. For example, *octrees* are defined by refining cubes into eight subcubes.[1] When necessary, cracks in the domain decomposition can be patched by modifying the geometry between neighboring cubes. This task can be greatly simplified when applied to *balanced* meshes, in which neighboring nodes differ by at most one level of resolution [54, 40]. In this case, cracks can be fixed by replacing each cell with a triangulated tile [3, 7, 58].

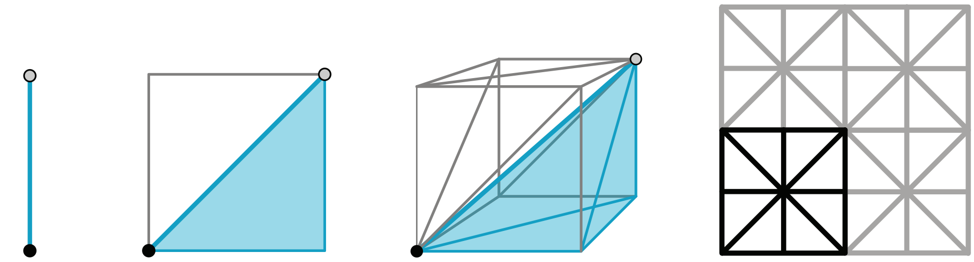
Alternatively, tetrahedral representations can be effective, as they simplify the extraction of crack-free representations. A popular technique is based on tetrahedral bisection along an edge. In the *Regular Simplex Bisection (RSB)* scheme all vertices lie on a regular grid and the generated tetrahedra are well-shaped, which can be important in downstream applications, such as finite element analysis and visualization. The extracted meshes are more adaptive than those obtained from a triangulated octree [11] which can help fine-tune the level of detail in the desired regions of interest.

Here, we provide a taxonomy of approaches that model a three-dimensional scalar field using the regular simplex bisection scheme, and focus on the differences in modeling primitives and extraction techniques that have been introduced in the literature. The primary distinction among the proposed approaches relates to whether individual tetrahedra, or clusters of tetrahedra sharing the same bisection edge, referred to as *diamonds*, are treated as modeling primitives. The latter define the atomic refinements required for conforming refinements. This choice leads to different multiresolution models, querying approaches and possibilities for optimizations in encoding, analyzing and visualizing such datasets.

These hierarchies have primarily been used to model multiresolution scalar fields, where data is associated with the vertices of the mesh. Here, we focus on the developments of interactive isosurface extraction from volumetric datasets. The RSB scheme has also been very popular for terrain visualization (see the recent review by Pajarola and Gobbetti [42]). A more general treatment of the models and applications of the RSB scheme can be found in [60], where the scheme is presented in a dimension–independent manner, and also surveys alternative applications of RSB, including: finite element analysis [47, 35, 21], spatial access structures [8], surface reconstruction [36], and higher-dimensional approaches [52, 26, 32, 2].

The remainder of this paper is organized as follows. After introducing some background notions on simplicial complexes and simplicial decompositions in Section 2, we review the simplex bisection rule and we define diamonds in Section 3. In Section 4, we introduce a classification of the different approaches proposed in the literature. In Section 5, we describe hierarchical representations for RSB meshes. We review tetrahedron-based and diamond-based approaches in Sections 6 and 7. In Section 8, we describe an alternative representation for nested meshes defined by the RSB scheme in the form of balanced octrees whose leaf cubes are tetrahedralized using a bisection-based algorithm. Finally, in Section 9, we present a table summarizing the taxonomy we have developed and discuss directions for further research.

[1] Similar hierarchies can be defined on tetrahedra, where each tetrahedron is refined into eight tetrahedron defined by the edge midpoints [5, 40].



(a) 1D **(b)** 2D **(c)** 3D **(d)** Tiling space (in 2D)

Figure 1 Kuhn-subdivided hypercubes in 1D **(a)**, 2D **(b)** and 3D **(c)**, highlighting one of the $d!$ simplices (blue). All edges are aligned with the diagonal of an axis-aligned hypercube. Translated or reflected **(d)** copies of a Kuhn-subdivided cube can tile space.

2 Background

A k-simplex σ is the convex hull of $k+1$ affinely independent points in a subspace of $\mathbb{R}^d$, where k is the *order* of the simplex. A *simplicial mesh* Σ is a finite collection of simplices such that if σ is a simplex in Σ, then all of the simplices bounding it (called the *faces* of σ) also belong to Σ, and the interiors of all simplices in Σ are disjoint. The *dimension*, or *order*, of a simplicial mesh is the maximum of the orders of the simplices forming it. In a simplicial mesh, simplices that are not on the boundary of any other simplex are called *top simplices*. In 3D, tetrahedra are the top simplices in a *tetrahedral mesh*. A simplicial mesh whose cells are defined by the uniform subdivision of a small set of cells into scaled copies is called a *nested mesh*.

If the intersection of any two simplices σ_1, σ_2 in a simplicial mesh Σ is a lower dimensional simplex on the boundary of σ_1 and σ_2, then Σ is *conforming*. A conforming simplicial mesh is also referred to as a *simplicial complex*.

We are often interested in generating simplicial complexes that cover a hypercubic domain. To this aim, we consider the canonical subdivision of a d-dimensional hypercube h into $d!$ simplices along a diagonal, which we refer to as a *Kuhn-subdivided d-cube* and denote as $\mathcal{K}(h)$ (see Figure 1) [13, 24]. Since the faces of a Kuhn-subdivided cube are also Kuhn-subdivided, a regular grid can be tiled by Kuhn-subdivided cubes [24, 35], that are translated or reflected [52]. The latter is the underlying triangulation for regular simplex bisection meshes at each level of resolution (see Figure 1d for an example in 2D).

3 Regular Simplex Bisection: Tetrahedra and diamonds

In this section, we review the regular simplex bisection scheme, which guides the generation of nested tetrahedral meshes, and the clustering structure for tetrahedra sharing a common bisection edge into *diamonds* which aids in the extraction of conforming tetrahedral meshes.

A well-studied class of nested simplicial meshes is defined by the *simplex bisection* operation. This operation bisects a d-simplex σ along the hyperplane defined by the midpoint $\mathbf{v}_m$ of some edge $\mathbf{e}$ and the $(d-1)$ vertices of σ not incident in $\mathbf{e}$. This generates two d-simplices covering σ, see Figure 2. We refer to $\mathbf{e}$ as the *bisection edge* of σ.

Since the general simplex bisection operation does not indicate which edge to bisect, researchers have proposed schemes to implicitly determine the bisection edge of a tetrahedron based on its geometric properties [47] or on the order of its vertices [4, 23, 33, 1, 35].

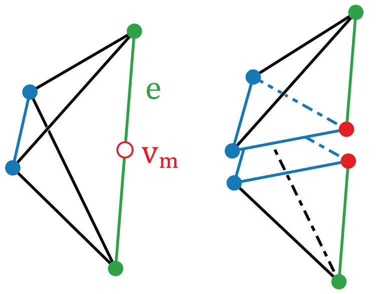


Figure 2 Bisection of a tetrahedron $\mathbf{t}$ along the plane defined by the midpoint $\mathbf{v}_m$ of an edge $\mathbf{e}$, and the two vertices of $\mathbf{t}$ not incident in $\mathbf{e}$.

Approaches in the latter category generalize the 2D *Newest Vertex Bisection* algorithm [38], by choosing the edge opposite the pair of most recently introduced vertices as the bisection edge. This can be achieved through the use of a set of typographic rules for manipulating the order of the vertices. Approaches in the former category, referred to as tetrahedral *Longest Edge Bisection* [47], choose the longest edge of the tetrahedron as its bisection edge.

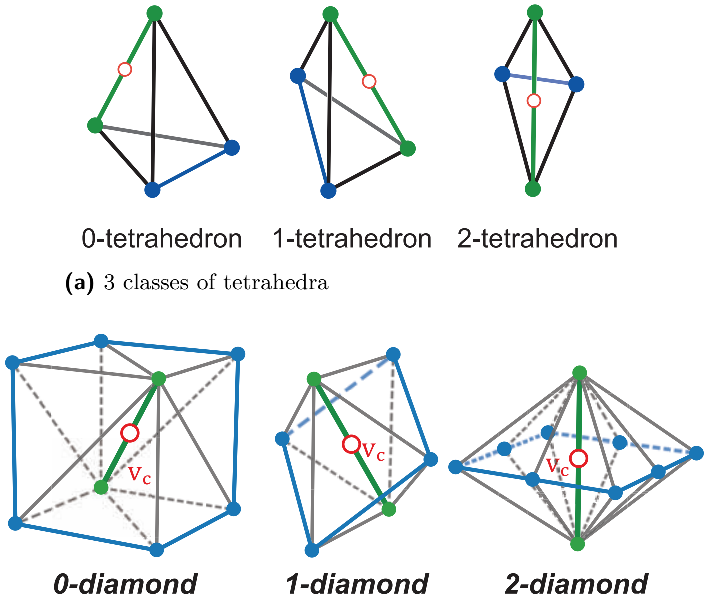
The *Regular Simplex Bisection (RSB)* scheme for tetrahedral meshes is defined by the recursive application of either of the above schemes to the tetrahedra in a Kuhn-subdivided cube $\mathcal{K}(h)$, and cyclically generates tetrahedra whose shapes belong to three similarity classes (see Figure 3a).[2] The six tetrahedra in $\mathcal{K}(h)$, which we refer to as tetrahedra of *class 0*, or as *0-tetrahedra*, have a bisection edge that is aligned with the diagonal of a cube. Tetrahedra in *class 1* are generated by bisecting *0-tetrahedra*, while those in *class 2* are generated by bisecting *1-tetrahedra*. The former have a bisection edge aligned with the diagonal of a square face of a cube, while the latter have a bisection edge aligned with an edge of a cube. Bisection of *2-tetrahedra* generates *0-tetrahedra* with edge lengths half that of their three-fold predecessors. We refer to tetrahedral meshes generated by recursive application of RSB as *tetrahedral regular simplex bisection (tRSB) meshes*, or simply as *RSB meshes*.

Note that individual tetrahedral bisections can introduce cracks into an RSB mesh along tetrahedra incident to the bisected edge, and thus, all such tetrahedra require simultaneous bisection for the mesh to remain conforming. Since the RSB scheme only allows bisection along a tetrahedron's predetermined bisection edge, conforming refinements in the RSB scheme involve tetrahedra belonging to the same similarity class and sharing the same bisection edge.

The cluster of tetrahedra sharing the same bisection edge is referred to as a *diamond* [12, 18, 43, 56], and we refer to the bisection edge as its *spine*. Consequently, the RSB scheme generates three similarity classes of diamonds, in correspondence to its three similarity classes of tetrahedra (see Figure 3b).

A diamond δ is *subdivided* by bisecting all of its tetrahedra using the regular simplex bisection operation. The effect of a diamond subdivision on an RSB mesh Σ is to remove the spine of the diamond, to insert a vertex at the midpoint of its spine, which we call the *central vertex* of the diamond and denote as v_c, and finally to insert edges from v_c to each vertex of δ (see Figure 4). Note that there is a one-to-one correspondence between diamonds and edges (via their spine), and between diamonds and vertices (via their central vertices). Diamond subdivision is an instance of *stellar subdivision* [28] and only affects the interior of

[2] In higher dimensions, the RSB scheme is based on typographical approaches [35, 53].



(a) 3 classes of tetrahedra

(b) 3 classes of diamonds

Figure 3 Regular simplex bisection generates three similarity *classes* of tetrahedra **(a)**. The bisection edge (green) of a class i tetrahedron is aligned with the diagonal of an axis aligned $(3 - i)$-cube. Tetrahedra sharing the same bisection edge can be clustered into *diamonds* **(b)** defined by six tetrahedra of class 0, four tetrahedra of class 1 and eight tetrahedra of class 2, respectively. The bisection edge of a diamond is referred to as its *spine*, and its midpoint v_c as its *central vertex*.

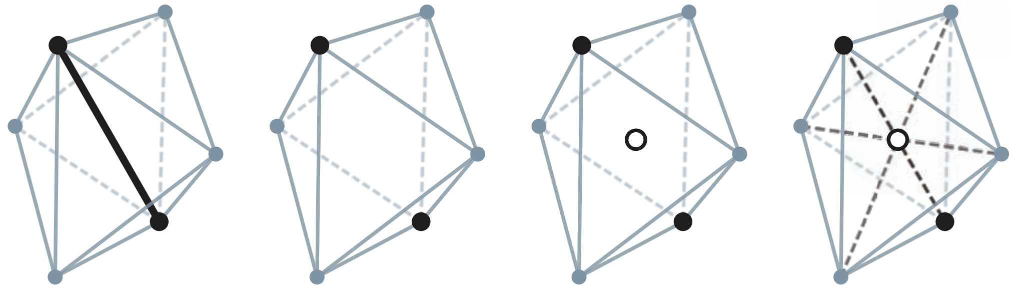
the diamond's domain. The diamond subdivision paradigm, in which $(d - i)$-cells are refined at step i, has been generalized to cell complexes in higher dimensions [43].

Although the regular simplex bisection scheme is defined in arbitrary dimension, it does not provide intuition on the shape or complexity of conforming refinements. The notion of diamonds has recently been generalized to arbitrary dimensions [56] as the combinatorial cross product of two simplicial decompositions of a cube: a Kuhn-subdivided $(d - i)$-cube, and the boundary of a *fully-subdivided* i-cube, a Kuhn subdivided i-cube whose simplices were bisected i times. Figure 5 illustrates how a three dimensional 1-diamond (Figure 5a) can be decomposed into a Kuhn-subdivided 2-cube (Figure 5b) and the boundary of a fully subdivided 1-cube (Figure 5c).

4 Overview

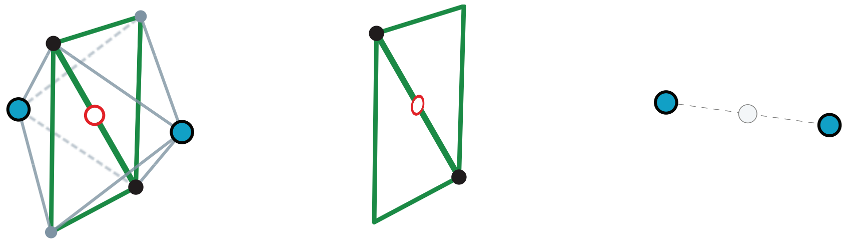
In this section, we review the basic ingredients of approaches that use nested tetrahedral RSB meshes. We classify such approaches on the basis of the choice of the basic primitives of the representation and on the method by which the data structures are queried.

Representations for 3D RSB meshes can be classified into *tetrahedron-based* and *diamond-based* representations. The former consider tetrahedra as the basic modeling primitives, while the latter consider diamonds as the modeling primitives. A simplex-based representation is described by a *hierarchy of tetrahedra* which encodes the containment relation between tetrahedra, while a diamond-based representation is described by a *hierarchy of diamonds* which encodes the parent-child relation between diamonds (i.e. incorporating the containment relations of all tetrahedra within a diamond). Both hierarchies can be encoded through



(a) Original diamond **(b)** Spine removed **(c)** Central vertex added **(d)** Subdivided diamond

Figure 4 A diamond **(a)** is *subdivided* by applying RSB to all of its tetrahedra. The effect of a diamond subdivision is to remove its spine **(b)**, to add its central vertex **(c)** and to add an edge from that vertex to all vertices on the boundary **(d)**.



(a) 1-diamond **(b)** $\mathcal{K}$(2-cube) **(c)** $\mathcal{B}_F$(1-cube)

Figure 5 A 1-diamond **(a)** can be decomposed into a Kuhn-subdivided 2-cube **(b)** and the boundary of a fully subdivided 1-cube **(c)**. In general, an i-diamond of dimension d can be decomposed into a Kuhn-subdivided $(d - i)$-cube and the boundary of a fully subdivided i-cube.

the use of pointers, but the regularity of the domain decomposition admits an implicit formulation of these spatial and hierarchical relationships. Thus, we can distinguish in both cases between explicit and implicit representations.

The most common application of nested RSB meshes is to define a multiresolution model for scalar fields. In these representations, one or more scalar value is associated with the vertices of a cubic grid, and thus the multiresolution model is a precomputed hierarchy of tetrahedra or a hierarchy of diamonds which is queried to extract RSB meshes approximating the underlying scalar field. An interpolant (typically piecewise linear) approximates the scalar field within the mesh. Conforming meshes are important in this context since cracks in the mesh correspond to discontinuities in the field representation. Due to the regularity of the vertex distribution, the scalar values of a dataset of size $(2^N + 1)^3$, are often stored in a linear block of memory. Thus, the data associated with a vertex can be implicitly located in the array using the vertex coordinates and explicit pointers are unnecessary.

Although the minimum required information for each sample is its scalar value, many methods achieve efficiency by encoding additional information, such as the field gradient of each sample, the error associated with each tetrahedron or diamond in the hierarchy, or the range of values within its domain. When data is associated with the simplices, it can be stored in $d!$ binary trees, encoded as linear arrays. Alternatively, data that is associated with the vertices or diamonds of the hierarchy can be uniquely associated with the corresponding central vertices of diamonds. Thus, such data can be encoded as a three-dimensional array.

There has been much research on querying methods to extract nested RSB meshes that approximate the full domain at virtually continuous levels of detail. Such methods can begin with a coarse approximation of the domain which is *selectively refined* by traversing the hierarchy defining the model in a top-down manner [16]. Alternatively, they can be defined by coarsening the full resolution dataset in a bottom-up manner [63], or in an incremental manner by starting with a previously extracted mesh [12, 18, 10].

Queries are implemented by evaluating an application–dependent predicate, known as the *acceptance criterion*, at each node of the hierarchy (i.e. tetrahedron or diamond) to determine whether to refine or coarsen the node. These predicates can be defined by the location of a node in a *region of interest* or by the distance to an object of interest such as the viewpoint. In scalar field representation, the *approximation error* of a node describes the degree to which it locally approximates the scalar field. These approximation errors can be defined *locally* between a node and its children, or *globally* across all descendants. When the error at a node is guaranteed to be greater than those of its descendants (with respect to the diamond dependency relation), it is said to be *saturated* [41, 17].

In applications that require conforming RSB meshes, all simplices within a diamond must be subdivided concurrently. If a tetrahedron-based representation is used, this can be accomplished implicitly, via a top-down traversal using a saturated selection criterion [16, 8], or explicitly through the use of *neighbor-finding* operations [21, 25, 2]. In general, neighbor-finding requires more hierarchical traversals but fewer overall bisections to satisfy a given acceptance criterion [25]. If a diamond-based representation is used, it can be easily shown that the parent-child dependency relation among the diamonds is a partial order relation and any set of diamonds which is *closed* with respect to the partial order defines a conforming mesh. In this case, the model is an instance of the *Multi-Tessellation* framework [11].

5 Representations for nested tetrahedral RSB meshes

As mentioned above, representations for tetrahedral RSB meshes can be classified into *tetrahedron-based* and *diamond-based* representations. We will see that a tetrahedron-based representation implicitly encodes all possible RSB meshes which can be generated from the initial Kuhn-subdivided domain through successive bisections, while a diamond-based representation encodes only the conforming RSB meshes defined on the same domain and set of vertices.

The *containment relation* between the tetrahedra in a nested RSB mesh defines a hierarchical relationship, where the two tetrahedra created during the bisection of a *parent* simplex σ are interpreted as the *children* of σ. This relationship is captured using a binary tree, often referred to as a *bintree* [12, 63], whose root is a tetrahedron of the Kuhn-subdivided domain Ω. Thus, a nested tetrahedral RSB mesh can be modeled as a forest of six binary trees, which we call a *hierarchy of tetrahedra* [25].

The *depth* of a tetrahedron is defined recursively as 0 for the bintree roots, and one greater than the depth of its parent otherwise. All root tetrahedra are *0-tetrahedra* since they belong to $\mathcal{K}(h)$. Since regular simplex bisection is used to generate the tetrahedra at successive depths, tetrahedra at the same bintree depth in the hierarchy belong to the same similarity class, and the classes repeat cyclically. The tetrahedra at three successive depths define a *level* of resolution within the hierarchy. Consequently, the *level* of a tetrahedron at depth m is $\lfloor m/3 \rfloor$ and its class is $(m \mod 3)$.

Since many applications require conforming meshes, diamond-based approaches, which focus on conforming updates to a tetrahedral RSB mesh, have received a lot of attention in the literature. The containment relation among the tetrahedra of a nested RSB mesh induces a hierarchical *dependency relation* on the diamonds within the hierarchy. Specifically, a diamond δ_c is a *child* of a diamond δ_p if δ_c contains at least one tetrahedron generated during the subdivision of δ_p.

In contrast to a hierarchy of tetrahedra, the domain of the children of a diamond is not nested within its own domain, and diamonds can have several parents and children. Since the parents of a diamond at depth $m+1$ belong to depth m, the dependency relation can be modeled as a directed acyclic graph of diamonds, whose single root is the diamond containing the tetrahedra covering Ω.

Figure 6 illustrates the dependency relation for the three classes of diamonds. 0-diamonds (brown) have three parents and six children (upper row), 1-diamonds (red) have two parents and four children (middle row) and 2-diamonds (green) have four parents and eight children (lower row). Thus, each parent diamond contributes a pair of tetrahedra to each of its children, which we refer to as a *parent-child duet* [6, 59]. Generally, a d-dimensional diamond δ has $O(d)$ parents, each of which generates $O(d!)$ simplices belonging to δ upon its subdivision [56].

Due to the correspondence between diamonds and their central vertices, the dependency relation among diamonds is often studied by considering the set of vertices on which it depends. For simplification, a diamond depends on vertices introduced at deeper within the hierarchy, while, for refinements, a diamond depends on vertices introduced at shallower depths.

6 Tetrahedron-based RSB approaches

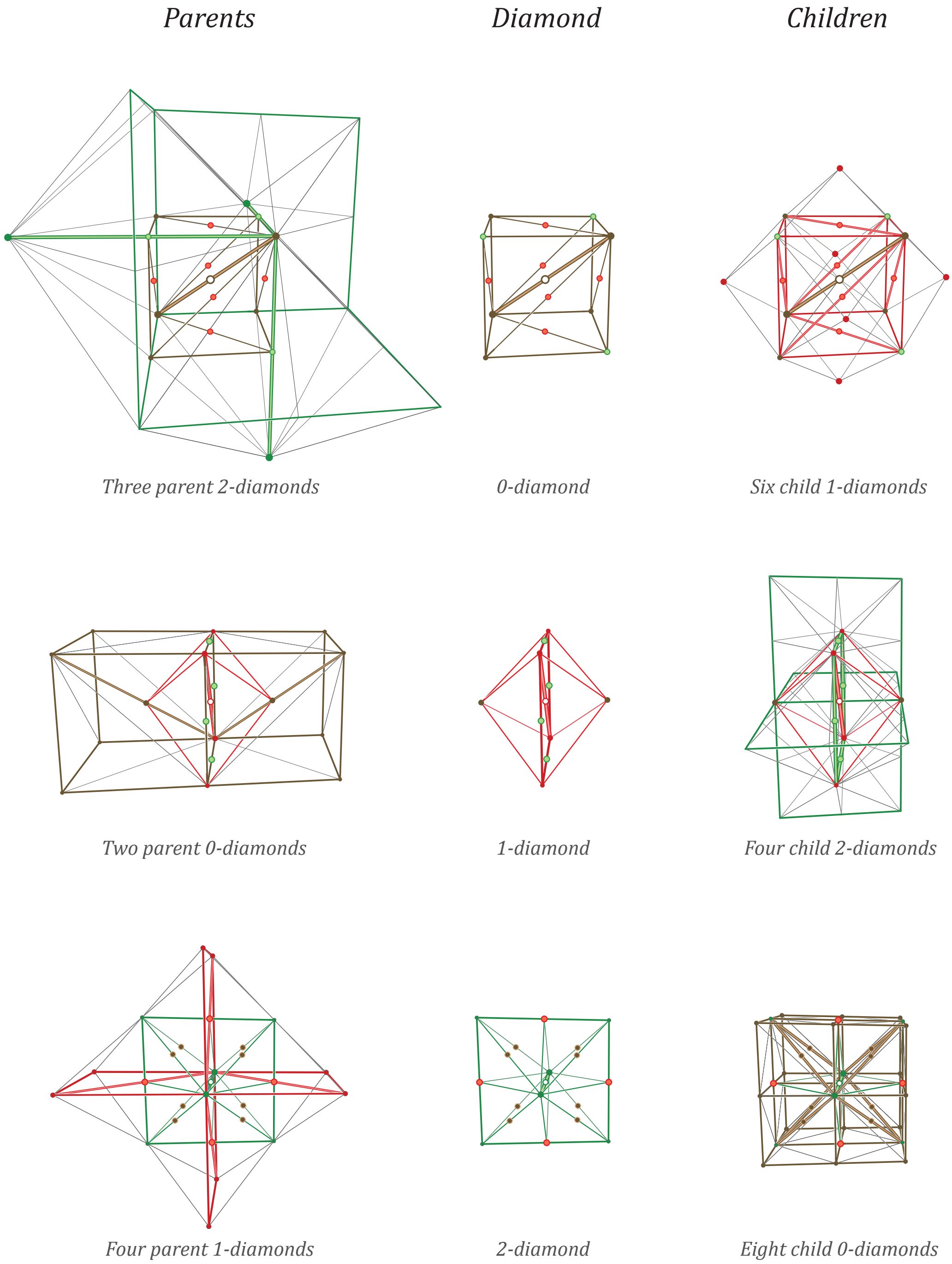
The primary application of 3D RSB has been multiresolution modeling of 3D scalar field for visualization and analysis. This has primarily (but not exclusively) taken the form of isosurface extraction and visualization, in which the goal is to extract an adaptive isosurface from a simplified representation of the underlying field. In this case, a conforming RSB mesh ensures that extracted isosurfaces are conforming (e.g. using the Marching Tetrahedra algorithm [46]).

Zhou et al. [63] extend the 2D simplification approach of Lindstrom et al. [29] with a bottom-up *tetrahedral fusion* operation (i.e. the inverse of an RSB operation). They explicitly encode the vertex simplification dependency relation in a lookup table, and obtain conforming tetrahedral RSB meshes by fusing all pairs of tetrahedra incident in the removed vertex (i.e. the central vertex of a subdivided diamond).

To ensure that the topology of the simplified isosurface matches that of the surface at full resolution, they introduce a *topology-preserving* check into their acceptance criterion that disallows fusion when the bisection edge's endpoints lie on the opposite side of the isosurface as its central vertex.

Gerstner and Pajarola [15] note that, while this topology preserving criterion is sufficient to guarantee accurate topology, it is too conservative. They identify the cases in which the local isosurface topology can change during a diamond's subdivision. They encode with each node the (conservative) range of field values in which the topology of its descendants can change. At runtime, a top-down query is applied to this saturated topology-preserving acceptance criterion to extract a conforming tetrahedral RSB mesh whose embedded isosurface has the same topology as the mesh at full resolution. They also propose *topology control* heuristics to reduce topological noise in the extracted isosurface.

Takahashi et al. [50] extend the above topological criterion to capture topological changes to the entire scalar field rather than those of a specific isosurface. They use this to guide the



Three parent 2-diamonds *0-diamond* *Six child 1-diamonds*

Two parent 0-diamonds *1-diamond* *Four child 2-diamonds*

Four parent 1-diamonds *2-diamond* *Eight child 0-diamonds*

Figure 6 Diamond hierarchy in 3D. A diamond's (middle column) tetrahedra are generated during the subdivision of its *parents* (left column). The central vertex of each parent coincides with a vertex of the diamond, while those of its *children* (right column) coincide with the midpoints of a subset of its edges. 0-diamonds (brown) have three parents and six children (upper row). 1-diamonds (red) have two parents and four children (middle row). 2-diamond (green) have four parents and eight children (lower row).

creation of transfer functions that highlight the topological features of the scalar field during Direct Volume Rendering (DVR).

Marchesin et al. [34] directly use the nested hierarchy for view-dependent DVR and examine the implications of applying non-conforming bisections, which can introduce artifacts into the visualization, but might be sufficient during exploration of the field.

One of the advantages of a saturated acceptance criterion is that it admits parallel extraction of conforming tetrahedral meshes [16]. Gerstner and Rumpf [16] propose a parallel query on the hierarchy to accelerate view-dependent isosurface extraction. Their curvature-based backface culling approach reduces the size of the extracted isosurfaces by a factor of two. In follow-up work [14], Gerstner describes a hierarchical scheme to compute the gradient of a tetrahedron from that of its parent and introduces a back-to-front sorting scheme based on the RSB splitting plane. This enables the extraction of multiple transparent isosurfaces during a single traversal of the hierarchy.

Pascucci [44] introduces a hardware accelerated approach to isosurface extraction, which incorporates a GPU marching tetrahedra algorithm for RSB tetrahedra, as well as a volumetric space-filling curve for generating tetrahedral strips for efficient isosurface rendering.

An alternative approach to extracting an isosurface from an RSB mesh is to define a multiresolution model for isosurfaces extracted from the hierarchy based on the set of conforming refinements. When the modifications follow the diamond subdivisions, they ensure that the extracted surface is manifold and free of self-intersections [45].

Pascucci and Bajaj [45] propose a progressive multiresolution model for extracted isosurfaces based on a small set of local update primitives that correspond to the change in isosurface after each tetrahedral bisection. Borgo et al. [6] describe a top-down progressive isosurface extraction method where the isosurface patches for successive depths of the hierarchy are extracted from those of the previous one. They establish an explicit correspondence between the edges of tetrahedra in successive depths to pass *isovertices* (i.e. isosurface vertices) from one depth to the next. This scheme has an overhead of 70 bytes per encoded diamond and achieves a reported 3 times speedup in isosurface extraction on modestly sized datasets of resolution 65^3 and 129^3. A similar approach is proposed by Lewiner et al. [27] to compress and progressively encode extracted isosurfaces. They encode the relative sign value of each RSB vertex in the desired isosurface's *tubular neighborhood*, that is, the set of tetrahedra intersected by the isosurface, and use a depth-first search to reconstruct the local connectivity of the tetrahedra in the RSB mesh.

Scalar field reconstruction and analysis are other interesting applications of nested RSB hierarchies. In [48], Roxborough and Nielson utilize the hierarchy of tetrahedra to reconstruct volumetric shapes based on freehand ultrasound data. Similarly, Mello et al. [36] and Tanaka et al. [51] utilize a hierarchy of tetrahedra to reconstruct surfaces based on sampled range or volumetric data. In all three cases, the decomposition is used as a spatial access structure on the irregularly sampled field, and the scalar field can be reconstructed on the vertices of an adaptive RSB mesh from these samples. Kimura et al. [22] propose a parallel algorithm to segment a volume dataset represented as a hierarchy of tetrahedra.

7 Diamond-based RSB approaches

Gregorski et al. [18] propose the first diamond-based RSB approach in 3D. They operate on a domain of resolution $(2^N)^3$ and avoid dealing with domain boundaries by treating the domain as a 3-torus. However, since opposite faces of the domain are conceptually glued together in this model, this can increase the size of extracted meshes. For example, refinements near a domain boundary of the scalar field can cause refinements along opposite boundaries that are spatially distant.

Gregorski et al. exploit the regularity of the RSB model to implicitly reconstruct the dependency relation of each diamond in terms of scaled offsets from 26 *archetypal* diamonds, based on the *oriented directions* of a diamond's spine. They also encode the geometry of a diamond, e.g. the locations of its vertices, as scaled offsets from its central vertex, which are accessed from a lookup table. Access to a diamond's entries in this table requires only its central vertex, its level of resolution and its type (i.e. spine orientation), saving 6–12 pointers (i.e. 24–48 bytes) per diamond compared to the explicit encoding of Zhou et al. [63].

Weiss and De Floriani [56, 59] extend this encoding by providing an efficient means of determining a diamond's class, level of resolution and type directly from the binary representation of its central vertex. In their encoding, a diamond's *scale* γ is the minimum of the number of trailing zeros in the binary representation of its three coordinates. The level ℓ and scale γ in a hierarchy of resolution $(2^N + 1)^3$ are related as $\ell = N - \gamma$. The *offset table* for the diamond types can either be generated in a preprocessing step, or can be efficiently calculated at runtime [56].

Gregorski et al.'s diamond-based scheme extends the ROAMing terrain approach for 2D domains [12]. It uses a dual-queue incremental selective refinement algorithm to exploit the frame-to-frame coherence between extracted meshes during view-dependent isosurface extraction. To reduce the size of large datasets, they compress the scalar values, field gradient and ranges of field values within each diamond from 19 bytes to 4 bytes, and rearrange the data using a hierarchical space-filling curve [30] to use the operating system's virtual memory paging for cache-coherent out-of-core memory management. Recently, Gregorski et al. [19] proposed an occlusion culling heuristic to further accelerate view-dependent isosurface visualization.

Linsen et al. [31] use the diamond connectivity [43] as an adaptive subdivision basis for volumetric datasets. They use trilinear B-spine wavelets to downsample the dataset (rather than the more commonly used subsampling) which generate similar approximations to the underlying domain using 10-15% fewer tetrahedra, although this can change the field's topology [50].

Weiss and De Floriani [57] introduce a high-level clustering primitive based on fully subdivided cubes (see Section 3), which they refer to as *supercubes*, to encode information with subsets of an RSB hierarchy. Each supercube uniquely indexes 56 diamonds, of which there are: eight 0-diamonds, twenty-four 1-diamonds and twenty-four 2-diamonds. Due to the one-to-one correspondence between diamonds, edges and vertices of the hierarchy, supercubes can be used to associate information with coherent subsets of the vertices, edges, tetrahedra and diamonds of a nested RSB mesh. For example, a supercube corresponds to the vertices of eight cube centers, 24 face centers of a cube and 24 edge centers within a scaled $(4 \times 4 \times 4)$ grid at a given level of resolution. The advantage of this representation is that it can reduce the geometric overhead associated with the coordinates of the retained elements.

Weiss and De Floriani suggest the use of a supercube-based representation when the number of retained samples is sparse with respect to the original dataset and the average clustering *concentration* of each supercube is high. Empirically, they found many common volume datasets to be oversampled by a factor of three or more with respect to a lossless approximation error. A supercube-based representation can also be used to efficiently encode conforming RSB meshes extracted from the hierarchy. This representation requires less than one byte per tetrahedron in the extracted mesh, approximately half the storage of a corresponding diamond-based representation [18, 56] and one-sixth the storage of a simplex-based representation [25].

The Isodiamond Hierarchy approach [59] introduces two models for decoupling multiresolution isosurfaces or interval volumes extracted from a hierarchy of diamonds from the underlying scalar field. In this framework, irregular modifications to the isosurface or interval volume are encoded in terms of the regular updates to the hierarchy of diamonds, requiring 14 bytes per update, and one byte per encoded isovertex. The *Relevant Isodiamond Hierarchy* encodes a closed set of updates from the corresponding hierarchy, in correspondence to the diamond subdivisions that modify the underlying isosurface (i.e. *active* updates) as well as their hierarchical ancestors (i.e. *relevant* updates). The *Minimal Isodiamond Hierarchy* enables extraction of conforming isosurfaces and interval volumes while only encoding the active updates. As a consequence, the extracted RSB meshes no longer cover the entire cubic domain of the scalar field, but its embedded isosurface or interval volume is still conforming. While both methods efficiently support selective refinement queries on the multiresolution model, the Minimal Isodiamond Hierarchy is more compact and extracts the same mesh in less time than the Relevant Isodiamond Hierarchy, but loses support for spatial selection queries and connectivity on the underlying mesh.

A subject of ongoing research relates to the validity of analyzing properties of the underlying scalar field using adaptive RSB meshes. An example is the computation of *discrete distortion* [37], a discrete curvature estimate on the scalar field, considered as a hypersurface embedded in $\mathbb{R}^4$. Preliminary results indicate that the salient features of the field are already present in approximated meshes, which can lead to accurate curvature estimation [61] and effective morphological segmentation [9] of the field.

8 Octree-based RSB approaches

An alternative representation for nested RSB meshes is in the form of *balanced octrees* whose leaf cubes are tetrahedralized using a bisection-based algorithm [7, 58].

An octree is said to be balanced (or *restricted* [54, 49]) if *neighboring* leaf nodes differ by at most one level of resolution. This leads to several possible balancing conditions depending on the desired neighborhood on which the mesh is balanced. A k-dimensional face in a nested cubic mesh is said to be balanced if its incident cubes belong to at most two consecutive levels of resolution, A nested cubic mesh is k-balanced if all k-cubes are balanced [39]. Specifically, in a 0-balanced octree (i.e. vertex-balanced), all cubes incident to a common vertex are balanced. Similarly, all edges are balanced in a 1-balanced octree (edge-balanced) and all squares are balanced in a 2-balanced octree (facet-balanced). Trivially, all octrees are 3-balanced (unbalanced). The desired property with respect to RSB refinement is edge-balancing, since this guarantees that each edge has only a single interior vertex.

Moore [39] considers the cost of balancing arbitrary octrees, and proves that balancing increases the size of an octree by at most a constant factor which depends only on the dimension of the domain. In 3D, vertex-balanced and edge-balanced octrees are at most 27 and 26 times larger than their unbalanced counterpart, but are typically considerably smaller in practice. Furthermore, every octree has a unique *least-common k-balanced refinement*, which can be attained through a simple greedy refinement strategy.

Weigle and Banks [55] propose a recursive bisection-based triangulation of uniform (hyper)-cubic meshes which treats each d-cube as a Kuhn-subdivided cube whose simplices are bisected $d-1$ times, leading to a mesh with $2^{d-1}d!$ simplices per hypercube (i.e. 24 tetrahedra per cube in 3D).

Castelo et al. [7] propose a recursive triangulation algorithm for the cubes within a balanced octrees (in arbitrary dimension). Specifically, each simplex is defined by connecting the midpoint of a k-cube to the vertices on its $k-1$ facets. When applied to edge-balanced (as well as vertex-balanced) octrees, this generates an RSB mesh. They apply this decomposition to isosurface extraction and surface reconstruction.

Weiss and De Floriani [58] propose a diamond-based triangulation of edge-balanced octrees (in arbitrary dimension) that applies a local selective refinement process to each leaf node based entirely on its *refined edges*, i.e. its edges that are incident to a smaller cube in the mesh. Specifically, they consider the 0-diamond corresponding to each leaf cube h as a base mesh Σ_h. For each refined edge e of h, they add its corresponding 2-diamond δ_e to Σ_h, and locally subdivide δ_e subject to the transitive closure of the diamond dependency relation restricted to the domain of h. In 3D, these triangulations can be precomputed in a lookup table based on the 12 possible edge refinements. An advantage of this approach is that the diamonds from adjacent leaf nodes can be merged into a single diamond-based RSB mesh.

The above triangulation algorithm implies a unique correspondence from every edge-balanced octree (and thus from general octrees) to a single diamond-based RSB meshes. However, since there are many diamond meshes that do not correspond to triangulated octrees, diamond-based approaches have a higher representational power.

In 3D, the triangulation of each cubic leaf node generates between 6 and 48 tetrahedra in the corresponding RSB mesh and has been found to inflate the mesh by a factor of 2-3 with respect to a diamond-based approach using the same acceptance criterion [58]. On the other hand, since octrees are widely implemented, such RSB-based triangulations are an effective means of generating well-shaped tetrahedral meshes.

9 Concluding remarks

We have classified and analyzed approaches for representing nested tetrahedral meshes generated through regular simplex bisection with a focus on how these techniques have been applied to multiresolution modeling of three-dimensional scalar fields.

In Table 1, we present a taxonomy of these approaches. We first distinguish between simplex-based approaches (upper rows) and diamond-based approaches (lower rows). Next, we classify methods based on their support for selective refinement queries. Such queries can be run from a coarse base domain in a top-down manner, from the full resolution mesh in a bottom-up manner or incrementally from a previously extracted mesh. We also distinguish between the class of selection criteria supported by the approach: those based on approximation error, range of field values and view-dependent criteria, which depend on an object's distance to the viewpoint. Similarly, a saturated error metric enables simpler queries but can also increase the size of its resultant meshes and requires preprocessing to saturate the acceptance criterion. The precomputed approximation error can be based on the approximation error between its current value and that of its subdivided children at the next depth, which we refer to as a *local* error metric. Alternatively, it can be based on the maximum interpolation error over all samples within its domain, which we refer to as a *global* error metric.

Our final classification relates to the optimizations implemented in each approach. This includes compressed meshes in the form of tetrahedral strips, view frustum culling and cache-coherent access to subsets of the dataset. Since the underlying data structure in all approaches are simplex-based or diamond-based nested RSB meshes, the optimizations developed for one scheme can usually be applied to the other schemes, but are useful in distinguishing among the various methods and in directing the reader for further details.

Table 1 Taxonomy of RSB-based approaches for 3D scalar fields indicating the modeling primitive (tetrahedron-based approaches in upper rows and diamond-based approaches in lower rows), querying, error metrics and optimizations.

	Query				Error		Optimizations							
Approach	Extraction	Approximation Error	Distance to Viewpoint	Range-based	Saturated	Hierarchical	Frustum Culling	Frame-frame coherence	Simplex Stripping	Out-of-core	Hierarchical Layout	Clustered updates	Incomplete Field	Parallel
Lee et al. [25]	Incremental	✓				Global		✓						
Marchesin et al. [34]	Incremental	✓	✓		✓	Local	✓	✓						
Gerstner et al. [15, 16, 14]	Top-down	✓		✓	✓	Local								[16]
Pascucci [44]	Top-down	✓	✓		✓	Local	✓		✓					✓
Tetrapuzzles [8]	Top-down	✓	✓		✓	Local	✓		✓	✓		✓	✓	✓
Lewiner et al. [27]	Top-down			✓		Local								
Zhou et al. [63]	Bottom-up			✓	✓	Local								
Gregorski et al. [18]	Incremental	✓		✓		Global	✓	✓		✓	✓	[20]		
Linsen [31]	Top-Down	✓				Global					✓			
Borgo [6]	Top-Down	✓	✓		✓	Local								
Weiss [56]	Top-Down	✓		✓		Global							[57]	
Isodiamonds [59]	Top-Down	✓				Global							✓	

The primary focus of this survey has been on the application of RSB meshes to the modeling, analysis and visualization of three dimensional scalar fields, whose samples coincide with the vertices of the RSB decomposition, since conforming meshes are typically required in this context. Another interesting application of these meshes is to use the spatial partitioning induced by the RSB operation as an access structure for spatial data. Cignoni et al. [8] have utilized the hierarchy to generate conforming updates to an irregularly triangulated surface indexed by the RSB tetrahedra. This supports external-memory visualization of conforming approximations to massive triangle meshes. Atalay et al. [2] have utilized this property on a four-dimensional *hierarchy of pentatopes* (i.e. 4-simplices) to accelerate point location queries for ray tracing of atmospheric effects.

As we have seen, many of the 3D approaches have been generalized from earlier two-dimensional versions. Similarly, these techniques have been generalized to problems in four- and higher-dimensions. For example, Lee et al. [26] propose a constant-time neighbor finding algorithm on hierarchies of pentatopes, Linsen et al. [32] generalize their analysis of diamond connectivity to 4D (and higher), and Gregorski et al. [20] propose a multiresolution model for time-varying scalar fields, where they exploit temporal coherence on an extracted tetrahedral RSB mesh to initialize the mesh at the next time step.

We expect the new understandings of the combinatorial structure of diamonds in arbitrary dimension [56] to yield insights into efficient data structures for modeling domains of dimensions greater than or equal to four. This can be used to analyze time-varying volume datasets, where the temporal and spatial dimensions are treated homogeneously as well as to visualize functions on the complex plane [55] and parametric spaces [35, 54].

The recent work on encodings for incomplete scalar fields sub-sampled at the vertices of a regular grid [57] suggests opportunities for efficient representations of adaptive distance fields and for multiresolution modeling of domains that are not strictly defined within a cube.

Finally, there has been recent work on accelerating multiresolution processing of RSB meshes on graphics hardware [62]. The regular structure of simplex and diamond hierarchies offers promising opportunities for such advances on tetrahedral RSB meshes which can further increase the utility of such meshes for interactive analysis and visualization of scientific and medical datasets.

Acknowledgments

This work has been partially supported by the National Science Foundation under grant CCF-0541032.

References

1 D.N. Arnold, A. Mukherjee, and L. Pouly. Locally adapted tetrahedral meshes using bisection. *SIAM Journal on Scientific Computing*, 22(2):431–448, 2000.

2 F.B. Atalay and D.M. Mount. Pointerless implementation of hierarchical simplicial meshes and efficient neighbor finding in arbitrary dimensions. *International Journal of Computational Geometry and Applications*, 17(6):595–631, 2007.

3 R.E. Bank, A. H. Sherman, and A. Weiser. Refinement algorithms and data structures for regular local mesh refinement. In R. Stepleman, M. Carver, R. Peskin, W. F. Ames, and R. Vichnevetsky, editors, *Scientific Computing, IMACS*, volume 1, pages 3–17. North-Holland, Amsterdam, 1983.

4 E. Bänsch. Local mesh refinement in 2 and 3 dimensions. *IMPACT of Computing in Science and Engineering*, 3(3):181–191, 1991.

5 J. Bey. Tetrahedral mesh refinement. *Computing*, 55(4):355–378, 1995.

6 R. Borgo, V. Pascucci, R. Scopigno, and P. Cignoni. A Progressive Subdivision Paradigm (PSP). *Proceedings of SPIE*, 5295:223, 2004.

7 A. Castelo, L.G. Nonato, M.F. Siqueira, R. Minghim, and G. Tavares. The J_1^a triangulation: An adaptive triangulation in any dimension. *Computers & Graphics*, 30(5):737–753, 2006.

8 P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, and R. Scopigno. Adaptive tetrapuzzles: Efficient out-of-core construction and visualization of gigantic multiresolution polygonal models. *ACM Transactions on Graphics*, 23(3):796–803, 2004.

9 L. De Floriani, F. Iuricich, P. Magillo, M.M. Mesmoudi, and K. Weiss. Discrete distortion for 3D data analysis. In L. Linsen, H. Hagen, and B. Hamann, editors, *Visualization in Medicine and Life Sciences (VMLS)*, Mathematics and Visualization. Springer Berlin Heidelberg, 2011.

10 L. De Floriani and M. Lee. Selective refinement on nested tetrahedral meshes. In G. Brunett, B. Hamann, and H. Mueller, editors, *Geometric Modeling for Scientific Visualization*. Springer Verlag, 2004.

11 L. De Floriani and P. Magillo. Multiresolution mesh representation: Models and data structures. In M. Floater, A. Iske, and E. Quak, editors, *Principles of Multi-resolution Geometric Modeling*, Lecture Notes in Mathematics, pages 364–418, Berlin, 2002. Springer Verlag.

12 M. Duchaineau, M. Wolinsky, D. E. Sigeti, M. C. Miller, C. Aldrich, and M. B. Mineev-Weinstein. ROAMing terrain: Real-time Optimally Adapting Meshes. In R. Yagel and H. Hagen, editors, *Proceedings IEEE Visualization*, pages 81–88, Phoenix, AZ, October 1997. IEEE Computer Society.

13 H. Freudenthal. Simplizialzerlegungen von beschrankter flachheit. *Annals of Mathematics*, 43(3):580–582, 1942.

14 T. Gerstner. Multiresolution extraction and rendering of transparent isosurfaces. *Computers & Graphics*, 26(2):219–228, 2002.

15 T. Gerstner and R. Pajarola. Topology-preserving and controlled topology simplifying multi-resolution isosurface extraction. In *Proceedings IEEE Visualization*, pages 259–266, 2000.

16 T. Gerstner and M. Rumpf. Multiresolutional parallel isosurface extraction based on tetrahedral bisection. In *Proceedings Symposium on Volume Visualization*, pages 267–278. ACM Press, 1999.

17 T. Gerstner, M. Rumpf, and U. Weikard. Error indicators for multilevel visualization and computing on nested grids. *Computers & Graphics*, 24(3):363–373, 2000.

18 B. Gregorski, M. Duchaineau, P. Lindstrom, V. Pascucci, and K. Joy. Interactive view-dependent rendering of large isosurfaces. In *Proceedings IEEE Visualization*, pages 475–484. IEEE Computer Society Washington, DC, USA, October 2002.

19 B. Gregorski, J. Senecal, M. Duchaineau, and K. I. Joy. Compression and occlusion culling for fast isosurface extraction from massive datasets. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Mathematics and Visualization, pages 303–323. Springer, 2009.

20 B. Gregorski, J. Senecal, M.A. Duchaineau, and K.I. Joy. Adaptive extraction of time-varying isosurfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):683–694, 2004.

21 D.J. Hebert. Symbolic local refinement of tetrahedral grids. *Journal of Symbolic Computation*, 17(5):457–472, May 1994.

22 A. Kimura, Y. Takama, Y. Yamazoe, S. Tanaka, and H Tanaka. Parallel volume segmentation with tetrahedral adaptive grid. *International Conference on Pattern Recognition*, 2:281–286, 2004.

23 I. Kossaczký. A recursive approach to local mesh refinement in two and three dimensions. *Journal of Computational and Applied Mathematics*, 55(3):275–288, 1994.

24 H.W. Kuhn. Some combinatorial lemmas in topology. *IBM J. Res. Develop*, 4:518–524, 1960.

25 M. Lee, L. De Floriani, and H. Samet. Constant-time neighbor finding in hierarchical tetrahedral meshes. In *Proceedings International Conference on Shape Modeling*, pages 286–295, Genova, Italy, May 2001. IEEE Computer Society.

26 M. Lee, L. De Floriani, and H. Samet. Constant-time navigation in four-dimensional nested simplicial meshes. In *Proceedings Shape Modeling International 2004*, pages 221–230. IEEE Computer Society, June 2004.

27 T. Lewiner, L. Velho, H. Lopes, and V. Mello. Simplicial isosurface compression. In *Vision, Modeling, and Visualization*, pages 299–306, Stanford, CA, November 2004.

28 W.B.R. Lickorish. Simplicial moves on complexes and manifolds. *Geometry and Topology Monographs*, 2(299-320):314, 1999.

29 P. Lindstrom, D. Koller, W. Ribarsky, L. F. Hodges, N. Faust, and G. A. Turner. Real-time continuous level of detail rendering of height fields. In *Proceedings ACM SIGGRAPH*, pages 109–118, August 1996.

30 P. Lindstrom and V. Pascucci. Terrain simplification simplified: A general framework for view-dependent out-of-core visualization. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):239–254, 2002.

31 L. Linsen, J. Gray, V. Pascucci, M. A. Duchaineau, B. Hamann, and K.I. Joy. Hierarchical large-scale volume representation with $\sqrt[3]{2}$ subdivision and trivariate b-spline wavelets. In G. Brunnett, B. Hamann, H. Mueller, and L. Linsen, editors, *Geometric Modeling for Scientific Visualization*, Mathematics + Visualization, pages 359–378. Springer Verlag, Heidelberg, Germany, 2004.

32 L. Linsen, V. Pascucci, MA Duchaineau, B. Hamann, and KI Joy. Wavelet-based multiresolution with $\sqrt[n]{2}$ subdivision. *Journal on Computing, Special Edition: Dagstuhl Seminar on Geometric Modelling*, 72:129–142, 2004.

33 A. Liu and B. Joe. Quality local refinement of tetrahedral meshes based on bisection. *SIAM Journal on Scientific Computing*, 16(6):1269–1291, 1995.

34 S. Marchesin, J.M. Dischler, and C. Mongenet. 3D ROAM for scalable volume visualization. In *IEEE Symposium on Volume Visualization and Graphics*, pages 79–86, 2004.

35 J. M. Maubach. Local bisection refinement for n-simplicial grids generated by reflection. *SIAM Journal on Scientific Computing*, 16(1):210–227, January 1995.

36 V. Mello, L. Velho, and G. Taubin. Estimating the in/out function of a surface represented by points. In *Symposium on Solid Modeling and Applications*, pages 108–114, 2003.

37 M.M. Mesmoudi, L. De Floriani, and U. Port. Discrete distortion in triangulated 3-manifolds. *Computer Graphics Forum*, 27(5):1333–1340, 2008.

38 W.F. Mitchell. Adaptive refinement for arbitrary finite-element spaces with hierarchical bases. *Journal of computational and applied mathematics*, 36(1):65–78, 1991.

39 D. Moore. The cost of balancing generalized quadtrees. In *Proc. ACM Solid Modeling*, pages 305–312. ACM, 1995.

40 D. Moore and J. Warren. Adaptive simplicial mesh quadtrees. *Houston J. Math*, 21(3):525–540, 1995.

41 M. Ohlberger and M. Rumpf. Hierarchical and adaptive visualization on nested grids. *Computing*, 56(4):365–385, 1997.

42 R. Pajarola and E. Gobbetti. Survey of semi-regular multiresolution models for interactive terrain rendering. *The Visual Computer*, 23(8):583–605, 2007.

43 V. Pascucci. Slow Growing Subdivision (SGS) in any dimension: Towards removing the curse of dimensionality. *Computer Graphics Forum*, 21(3):451–460, September 2002.

44 V. Pascucci. Isosurface computation made simple: Hardware acceleration, adaptive refinement and tetrahedral stripping. In *Eurographics/IEEE TVCG Symposium on Visualization (VisSym)*, pages 293–300, 2004.

45 V. Pascucci and C. L. Bajaj. Time-critical isosurface refinement and smoothing. In *Proceedings IEEE Symposium on Volume Visualization*, pages 33–42, Salt Lake City, UT, October 2000. IEEE Computer Society.

46 B.A. Payne and A.W. Toga. Surface mapping brain function on 3D models. *Computer Graphics and Applications, IEEE*, 10(5):33–41, Sept. 1990.

47 M.C. Rivara. Local modification of meshes for adaptive and/or multigrid finite-element methods. *Journal of Computational and Applied Mathematics*, 36(1):79–89, 1991.

48 T. Roxborough and G. Nielson. Tetrahedron-based, least-squares, progressive volume models with application to freehand ultrasound data. In *Proceedings IEEE Visualization*, pages 93–100. IEEE Computer Society, October 2000.

49 R. Sivan and H. Samet. Algorithms for constructing quadtree surface maps. In *Proc. 5th Int. Symposium on Spatial Data Handling*, pages 361–370, 1992.

50 S. Takahashi, Y. Takeshima, GM Nielson, and I. Fujishiro. Topological volume skeletonization using adaptive tetrahedralization. In *Proceedings Geometric Modeling and Processing*, pages 227–236, 2004.

51 H. Tanaka, Y. Takama, and H. Wakabayashi. Accuracy-based sampling and reconstruction with adaptive grid for parallel hierarchical tetrahedrization. In *Proceedings Volume Graphics*, pages 79–86. ACM Press, 2003.

52 M.J. Todd. *The computation of fixed points and applications*. Number 124 in Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 1976.

53 C. T. Traxler. An algorithm for adaptive mesh refinement in n dimensions. *Computing*, 59(2):115–137, 1997.

54 B. Von Herzen and A. H. Barr. Accurate triangulations of deformed, intersecting surfaces. In *Proceedings ACM SIGGRAPH*, pages 103–110, New York, NY, USA, 1987. ACM.

55 C. Weigle and D. Banks. Extracting iso-valued features in 4-dimensional scalar fields. In *Proceedings IEEE Visualization*, pages 103–110. IEEE Computer Society, October 1998.

56 K. Weiss and L. De Floriani. Diamond hierarchies of arbitrary dimension. *Computer Graphics Forum (Proceedings SGP 2009)*, 28(5):1289–1300, 2009.

57 K. Weiss and L. De Floriani. Supercubes: A high-level primitive for diamond hierarchies. *IEEE Transactions on Visualization and Computer Graphics (Proceedings IEEE Visualization 2009)*, 15(6):1603–1610, November-December 2009.

58 K. Weiss and L. De Floriani. Bisection-based triangulations of nested hypercubic meshes. In S. Shontz, editor, *Proceedings 19th International Meshing Roundtable*, pages 315–333, Chattanooga, Tennessee, October 3–6 2010.

59 K. Weiss and L. De Floriani. Isodiamond hierarchies: An efficient multiresolution representation for isosurfaces and interval volumes. *IEEE Transactions on Visualization and Computer Graphics*, 16(4):583 – 598, July-Aug. 2010.

60 K. Weiss and L. De Floriani. Simplex and diamond hierarchies: Models and applications. *Computer Graphics Forum*, 30:–, 2011.

61 K. Weiss, M.M. Mesmoudi, and L. De Floriani. Multiresolution analysis of 3D images based on discrete distortion. In *International Conference on Pattern Recognition (ICPR)*, pages 4093–4096, Istanbul, Turkey, August 2010. IEEE Computer Society.

62 M.A. Yalçın, K. Weiss, and L. De Floriani. GPU algorithms for diamond-based multiresolution terrain processing. In *Eurographics Symposium on Parallel Graphics and Visualization*, Bangor, Wales, April 10–11 2011.

63 Y. Zhou, B. Chen, and A. Kaufman. Multiresolution tetrahedral framework for visualizing regular volume data. In R. Yagel and H. Hagen, editors, *Proceedings IEEE Visualization*, pages 135–142. IEEE Computer Society, October 1997.

ViSSaAn: Visual Support for Safety Analysis

Yi Yang[1], Dirk Zeckzer[2], Peter Liggesmeyer[3], and Hans Hagen[4]

1 **University of Kaiserslautern, Germany**
`yang@informatik.uni-kl.de`
2 **University of Kaiserslautern, Germany**
`zeckzer@informatik.uni-kl.de`
3 **University of Kaiserslautern, Germany**
`liggesmeyer@informatik.uni-kl.de`
4 **University of Kaiserslautern, Germany**
`hagen@informatik.uni-kl.de`

Abstract

Safety of technical systems are becoming more and more important nowadays. Fault trees and minimal cut sets are usually used to attack the problems of assessing safety-critical systems. A visualization system named *ViSSaAn*, consisting of a matrix view, is proposed that supports an efficient safety analysis based on the information from these techniques. Interactions such as zooming and grouping are provided to support the task of finding the safety problems from the analysis information. An example based on real data shows the usefulness of ViSSaAn.

1998 ACM Subject Classification I.3.8 Applications

Keywords and phrases Safety Analysis, Fault Tree Analysis, Minimal Cut Sets, Safety Visualization, Information Visualization

Digital Object Identifier 10.4230/DFU.Vol2.SciViz.2011.378

1 Introduction

Fault Tree Analysis (FTA) [7, 13, 14, 25, 40, 41] is an analysis technique that is based on the graphical model named *fault tree*. It is widely used for analyzing the safety of technical systems. In order to handle large and complicated fault tree models for complex systems, the Component Fault Tree (CFT) was proposed in [22, 23]. The Minimal Cut Sets method (MCSs) [9, 12, 20, 24, 25, 28, 44] is a useful technique for analyzing fault trees. Improving MCS analysis is a good way to improve the safety analysis. For this objective, the following two aspects must be considered: obtaining the MCS analysis information; understanding the obtained information and find safety problems from the information. There are some approaches to obtain the MCS analysis information [9, 12, 20, 24, 28, 44]. Our research is focusing on the second aspect. In this aspect, representation methods of MCSs are more often considered. We try to find a representation to conveniently, quickly, and correctly understand the MCS information and find problems behind the information, in order to improve the safety of the system. Another challenge is the representation of large-scale data sets. For large systems, users have to face thousands of MCSs and all the applicable data associated to them. Finding the relevant information in a large data set is not an easy task.

Information visualization is a suitable technique that can fulfill these requirements. The visualized data is easier and faster to understand. The problems behind the information can be easily found in the large-scale data sets via visualization. It supports users in making correct decisions as soon as possible. Thus, we focus on how to improve the representation of MCS information using information visualization techniques. The commonly used representation

Scientific Visualization: Interactions, Features, Metaphors. *Dagstuhl Follow-Ups, Vol. 2.*
Editor: Hans Hagen; pp. 378–395
Dagstuhl Publishing
Schloss Dagstuhl – Leibniz Zentrum für, Germany

methods for MCS information are textual form, tabular form, and highlighted fault tree paths. They show only the basic MCS information and do not scale to large data sets. A lot of hidden information is not visible. Users have to discover it by themselves. Therefore, a visualization system is needed, that can show hidden information from MCSs and fulfill the complex representation purposes, in order to assist finding the problems conveniently in a large-scale data set.

The visualization system ViSSaAn was implemented according to our research and is presented in this paper. It supports the safety analysis showing MCSs, Basic Events (BEs), and CFTs. A matrix-based representation is used to present the correlation between MCSs and BEs and the correlation between MCSs and CFTs. The safety level method provides a categorization for orders of MCSs, probabilities of MCSs, probabilities of BEs, and probabilities of CFTs. Colors are used to identify safety levels. ViSSaAn provides interactions to support the analysis process for different purposes, such as grouping functionalities that can sort and classify rows and columns of the matrix. Focus&context with semantic zooming is applied to integrate the fault tree structures and their attached data into the matrix view. Users can get detailed information from fault trees inside the matrix view. For the presentation problem of large-scale data sets, ViSSaAn provides DOI zooming and Table zooming, where they can effectively reduce the display space of rows and columns of the matrix view without losing important information.

This paper is structured as follows: In Section 2, we introduce the background of safety analysis, particularly, the Fault Tree Analysis, Component Fault Trees, Minimal Cut Sets, and the safety levels. Related work about representation of MCSs for safety analysis is discussed in Section 3. In Section 4, we introduce the visualization system ViSSaAn, while in Section 5 the interactions in ViSSaAn are presented. An example of ViSSaAn with real data is presented in Section 6. Finally, the discussion of ViSSaAn and conclusions are given in Section 7 and 8.

2 Technical Background

2.1 Basic Concepts

2.1.1 Safety Analysis

Safety is defined as a *State where the danger of a personal or property damage is reduced to an acceptable value* [17]. The safety analysis is a process ensuring that the actual risk is smaller than the acceptable value.

2.1.2 Failure and Fault

Failure and Fault are defined in [15]:

- Failure: *The inability of a system or component to perform its required functions within specified performance requirements.*
- Fault:
  1. A defect in a hardware device or component.
  2. An incorrect step, process, or data definition in a computer program.

2.1.3 Safety-Critical

If the failure of a system could lead to unacceptable consequences and we depend on it for our well-being, then the system is safety-critical [27].

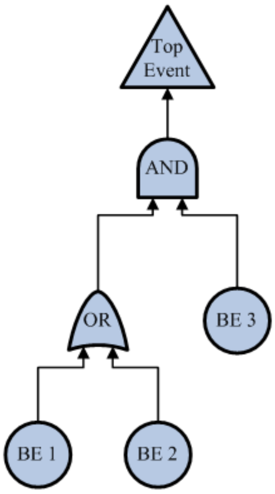


Figure 1 Fault Tree.

2.2 Fault Tree Analysis

Fault Tree Analysis (FTA) [7, 13, 14, 25, 40, 41] is an effective safety analysis technique for technical systems, which is standardized in [14] and [7]. The kernel of FTA is the fault tree model.

A fault tree is a model that graphically and logically represents the various combinations of possible events, both faulty and normal, occurring in a system that lead to the top undesired event [25].

A fault tree is a graphical model that is presented using a tree structure (see Figure 1). It consists of three kinds of elements:

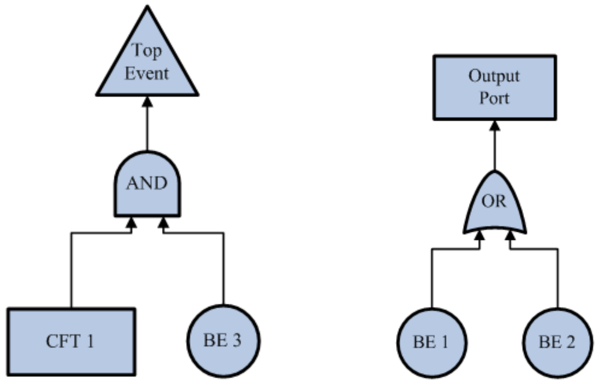
- *Top Event*: root of the tree. It is the top level undesired event.
- *Basic Events (BEs)*: leaves of the tree. They are possible causes of the Top Event. A BE will not be refined any more.
- *Gates*: inner nodes of the tree. They are logical connectives. There are different kinds of gates like the AND gate, the OR gate.

FTA provides some analysis methods in order to analyze fault tree models. Examples are minimal cut sets analysis (see Section 2.4), importance analysis, sensitivity analysis. Using FTA, the safety of systems can be analyzed. Systems can be improved according to the result of FTA.

2.3 Component Fault Trees

For complex technical systems, the fault tree model will be very large and complicated. In order to present the fault tree model more clearly and efficiently, the *Component Fault Tree (CFT)* was proposed in [22, 23]. With the component concept a traditional fault tree of a system can be divided into one or more independent components (see Figure 2). Each component is called a CFT. CFTs are connected amongst each other forming an overall CFT model for a system. The difficulty of FTA is reduced with CFTs.

We give a brief example of CFTs. The fault tree in Figure 1 can be transformed into the CFT depicted in Figure 2. It consists of a main model *CFT0* (see Figure 2 (a)) and a sub component *CFT1* (see Figure 2 (b)). CFT1 consists of the BE1 and the BE2 as well as an



(a) Main component: CFT0 **(b)** Sub component: CFT1

Figure 2 Component Fault Tree.

OR Gate. In CFT0, CFT1 is treated as a black box and used as a part. The CFT concept was implemented in an FTA tool *ESSaREL* (previous name: *UWG3*) [8, 21].

2.4 Minimal Cut Sets

After constructing a fault tree, a method is needed to analyze this model. The *Minimal Cut Sets (MCSs)* method [9, 12, 20, 24, 25, 28, 44] is a useful technique for analyzing fault tree models.

A cut set for a fault tree is a set of basic events whose occurrence causes the Top Event to occur. A cut set is said to be a minimal cut set if, when any basic event is removed from the set, the remaining events collectively are no longer a cut set [25].

The number of different BEs in a minimal cut set is called order of the cut set [32]. Order is also called *size*, e.g., in [25].

For example, in Figure 1 and Figure 2 the MCSs of the fault tree are: MCSs = {MCS1, MCS2}, with MCS1 = {BE1, BE3}, MCS2 = {BE2, BE3}. The order of MCS1 is 2, because it contains two BEs.

With the MCS method users can perform both qualitative analysis and quantitative analysis for fault tree models. Qualitative analysis is used for finding BE combinations that are able to cause the occurrence of the top event. Quantitative analysis is used for calculating the probability of the top event from influencing probabilities. The MCSs method is suitable for CFT analysis as well.

2.5 Safety Levels

The safety level method can be used to estimate the safety states. It provides different quantitative levels to categorize safety. In this paper, we use a simple 3-level criterion for safety that can be extended to more complicated safety level methods. In this criterion, safety can be classified into three levels:

- critical level: It is dangerous and urgent to be solved.
- moderate level: It needs to be solved, but not urgently.
- acceptable level: It is below the acceptable value, i.e., it is not dangerous.

The safety level criterion can be applied to MCSs, BEs, and CFTs. The safety levels of MCSs depend on either the order of the MCS or the failure probability of the MCS. The order of a MCS is inversely proportional to its safety. Basically, a MCS with order 1 is critical for a system. In this situation, the top event will occur by only one BE. The trigger condition is easy to be achieved. The higher the probability of a MCS is, the more dangerous is the MCS. The safety levels of BEs depend on the failure probability of BEs. The higher the probability of a BE is, the more critical is the BE. The same is true for CFTs.

3 Related Work

Currently, most safety analysis tools provide the MCS analysis for FTA. The ordinary representation of MCSs is a textual/tabular form. Some ideas combine a textual/tabular form with highlighted paths on fault tree diagrams. The textual form only lists the MCSs and the contained BEs with simple information. The tabular form is a bit complicated. It provides a table to represent the MCSs and the related information. Some advanced functionalities, such as sort and filter, are possible in the tabular form. Associating to fault tree diagrams helps users to understand the MCS information more clearly.

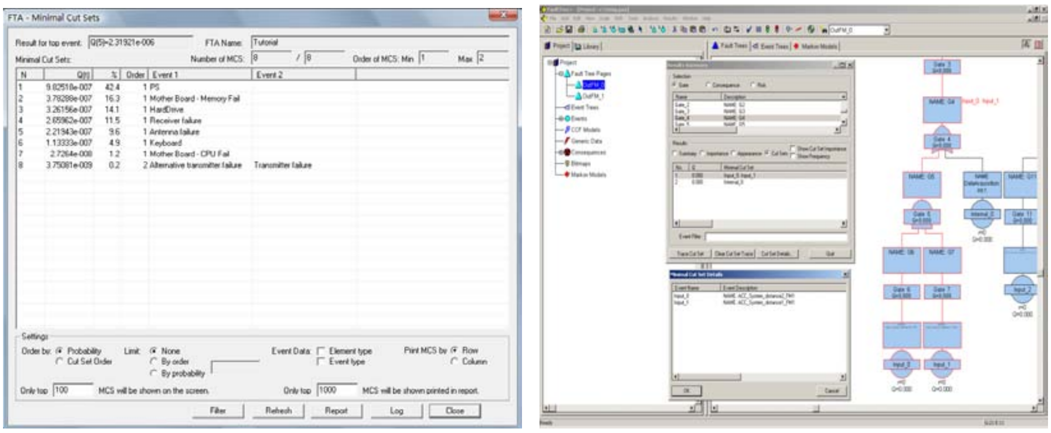
The tool *ESSaREL* [8] gives a textual MCS list. It shows each MCS with its ID and its BEs with BE IDs and BE labels. The MCSs are sorted by order. It also provides some general information, such as the ID of top event, generation date, path of data file, and count of MCSs. The tool *BlockSim* [34] shows the similar information. In additional, it shows probability of the MCSs. It provides filters for the order of MCSs and can sort MCSs by size, reliability, or unreliability. The tool *RAMCommander* [2] (see Figure 3(a)) provides a table for MCS information. Rows are MCSs. Besides the IDs of MCSs, the table also shows probability, contribution, and order of MCSs in the first four columns. The possibly used BEs are in subsequent columns. Sorting MCSs by ID, order, probability, or contribution can be performed in the table. Filtering by order is available to cut out the MCSs having an order outside the filter range. The "Element type" and "Event type" of BEs will be selectively shown in the table. Besides, it shows some common information, e.g., name of the FTA, probability of top event, count of MCSs, statistical information about the order of MCSs. The popular FTA tool *FaultTree+* [18] (see Figure 3(b)) also provides a tabular form for MCS information. In *FaultTree+*, the MCS table and the table for BEs use are separated. When selecting a MCS in the MCS table, the BEs used will be displayed in the BE table that can show a description and more detailed information of BEs. *FaultTree+* provides a filter to BEs. It also provides choices to show the importance and frequency of MCSs. It reflects the MCS information on the fault tree diagram using highlighting the paths that lead from the BEs of the MCS to the top event. It displays how the top event is reached from the BEs. Other tools like *Relex Architect* [33], *DPL faulttrees* [38], *FSAP/NuSMV-SA* [5], and *ITEM ToolKit* [19] provide similar representation methods for MCS information.

4 Visualization

4.1 Visualization Requirements

According to the introduction described in Section 1, the correlation between MCSs and BEs is essential and needs to be visualized. The MCSs can be evaluated using some properties, e.g., probability, and the result is then visualized. The component concept of FTA is also required to be represented, because it makes the system fault tree model easier to understand.

A suitable idea for large-scale data sets is needed. In consideration of the significance of the typical fault tree structure, it will be represented in the visualization system. Users can make sure how the MCSs effect the top event. More visual factors can be considered, if they are helpful for the representation of the MCS analysis information.



FTA - Minimal Cut Sets

Result for top event: Q(5)=2.31921e-006 FTA Name: Tutorial

Minimal Cut Sets: Number of MCS: 8 / 8 Order of MCS: Min 1 Max 2

N	Q(t)	%	Order	Event 1	Event 2
1	9.82518e-007	42.4	1	PS	
2	3.78288e-007	16.3	1	Mother Board - Memory Fail	
3	3.26156e-007	14.1	1	HardDrive	
4	2.65962e-007	11.5	1	Receiver failure	
5	2.21943e-007	9.6	1	Antenna failure	
6	1.13333e-007	4.9	1	Keyboard	
7	2.7264e-008	1.2	1	Mother Board - CPU Fail	
8	3.75081e-009	0.2	2	Alternative transmitter failure	Transmitter failure

Settings

Order by: ● Probability ○ Cut Set Order

Limit: ● None ○ By order ○ By probability

Event Data: □ Element type □ Event type

Print MCS by: ● Row ○ Column

Only top 100 MCS will be shown on the screen.

Only top 1000 MCS will be shown printed in report.

(a) Tabular form for MCSs (RAMCommander [2])

(b) Combination of tabular form and path highlighting (Faulttree+ [18])

Figure 3 MCSs representation methods of current tools.

4.2 ViSSaAn

Based on the requirements we developed a visualization system named *ViSSaAn (Visual Support for Safety Analysis)*. It was developed using Java [37] and the visualization library Prefuse [30]. There are two frames in ViSSan: the configuration frame *Main Control* and the analysis frame *MCS Matrix*.

4.3 Color Encoding

Colors can be used as a nominal code into classify objects to different categories. Color encoding for nominal information was presented, e.g., in [6, 26, 43]. We use colors in ViSSaAn according to the 3-level safety criterion described in 2.5: red (critical level), yellow (moderate level), and green (acceptable level).

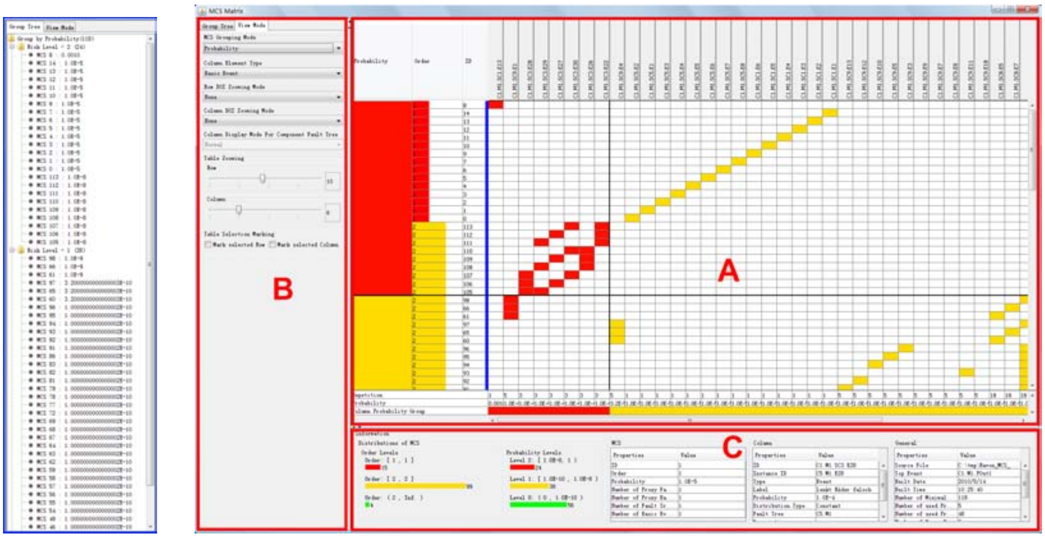
4.4 Main Control

Before starting the MCS analysis, the environment must be configured. *Main Control* is the configuration frame of ViSSaAn. With the help of the *Main Control* the data files can be loaded and users can specify the value ranges for the safety levels.

4.5 MCS Matrix

The tabular form is a useful method for MCS information. But it can not fulfill complex situations for MCS analysis, for example, the component concept of FTA, safety levels.

ViSSaAn provides an interactive matrix based representation to satisfy the advanced purposes. It is the main analysis view of the ViSSaAn. It is used to present correlation between MCSs and BEs/CFTs. The count of MCSs will be more than several thousands for a large fault tree model for a complex system. The matrix visualization is suitable for large-scale data. It is intuitive, clear, and good to represent large-scale data sets (see e.g., in [10] and [39]). After providing the settings in the *Main Control*, the analysis frame *MCS Matrix* will get started. *MCS Matrix* provides three view areas.



(a) Interaction panel (*Group Tree*)

(b) part A: matrix view; part B: interaction panel (*View Mode*); part C: information panel

Figure 4 MCS Matrix.

- The first part is the matrix view (see part A of Figure 4 (b)). It is the central part of the *MCS Matrix*. It shows the correlation between MCSs and BEs or between MCSs and CFTs.
- The second part is an interaction panel (see part B of Figure 4 (b)). It provides interactions for the matrix view that are used for finding the most valuable information for particular purposes.
- The third part is an information panel (see part C of Figure 4 (b)). It provides instant and general information of the elements represented in the matrix view.

4.5.1 Matrix View

The matrix view is the central part of the *MCS Matrix*. Safety levels of MCSs, BEs, and CFTs are coded by colors. There are three areas in the matrix view (see Figure 5). Rows present the MCSs in the first and in the second area. The third area is at the bottom. It shows the probability and the number of repetitions of each BE or CFT (see area 3 of Figure 5). The first area consists of three columns: the MCS ID, the MCS Order, and the MCS Probability (see area 1 of Figure 5). In order to present the order of MCSs more intuitively, a bar graph is used to visualize the order of a MCS. The applications of bar graphs in a table view were introduced in [29, 31]. The larger the order, the longer the bar. Users can read the exact textual value in the bars. The second area starts at the 4th column (see area 2 of Figure 5). There are two types of columns: the *Basic Event* column and the *Component Fault Tree* column. Types can be freely switched for different purposes.

For the type *Basic Event* (see Figure 5), each column of the second area represents a BE. A colored cell indicates that a BE in the current column is used by the MCSs in the current row. The colors represent the safety levels of the BEs. The IDs of the BEs are printed on the column head. For the type *Component Fault Tree* (see Figure 6), each column of the second area represents a CFT. The IDs of CFTs are printed on the column head. Each

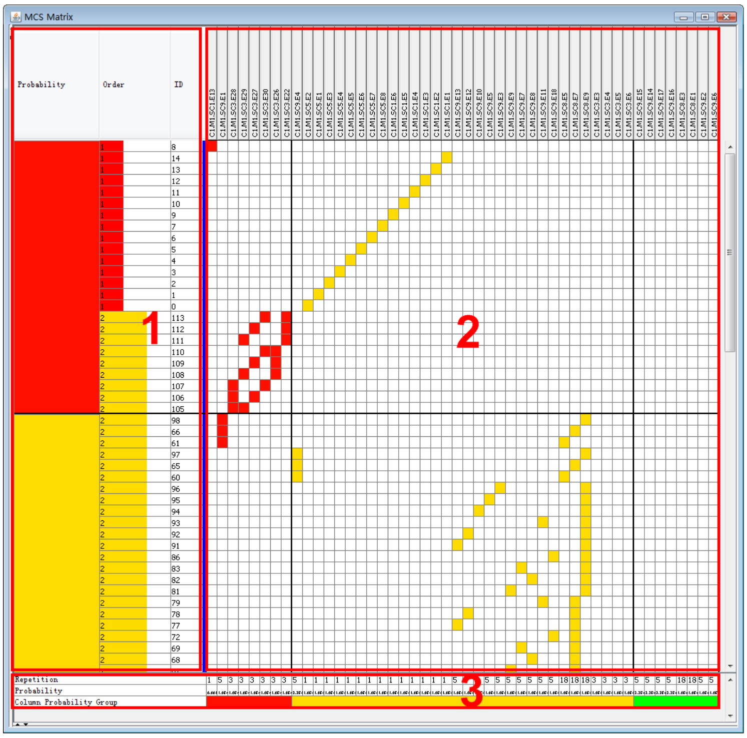


Figure 5 Matrix View – area 1: for MCS; area 2: for CFTs/BEs; area 3: for probability and number of repetitions. (Column Element Type: *Basic Event.*)

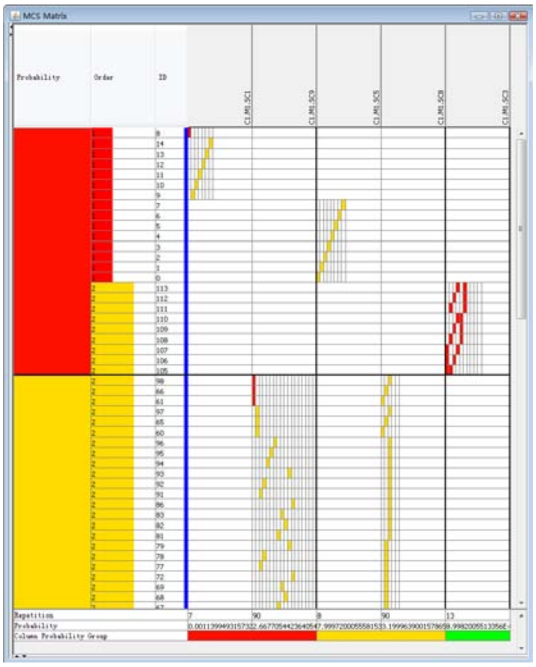


Figure 6 Column Element Type: *Component Fault Tree.*

non-empty cell contains at least one color filled sub-cell. The sub-cells indicate the contained BEs that are sorted by probability in descending order. A color filled sub-cell represents the correlation between inner BEs and the MCS in the current row. An empty cell indicates that no BE of the CFT in the current column is used by the MCSs in the current row. This type can be regarded as another kind of grouping for BEs. In this case, BEs are grouped to different CFTs and they are sorted by probability inside each CFT.

The safety levels are available for different properties, such as probability and order of MCSs. In order to analyze the safety levels of a specified property, ViSSaAn provides the grouping functionalities, where row grouping and column grouping are provided. The row grouping groups and sorts MCSs by the specified property and gathers the elements with the same level. Each group represents an aggregation of elements having the same safety level. There are two grouping modes for MCSs: *Order* and *Probability*. The related interaction will be introduced in 5.1. The column of the grouping property will always be set to the first column. Cells of the first column are merged into blocks according to the groups. The colored blocks can be treated as indicators of different safety levels as well. The column grouping groups and sorts the columns in the second area by probability of BEs or CFTs in descending order. The last row of the third area holds colored blocks as the indicators of the column groups.

Coloring associated to the grouping functionalities identifies the safety levels. Colors are more intuitive than text values, so that the safety levels of the elements are easily to be identified. In cooperation with the grouping functionality, the elements at the same level will be put together and sorted. It speeds up the process of finding the serious problems.

4.5.2 Interaction Panel

The interaction panel gathers the interactions of ViSSaAn that cannot be performed directly in the matrix view. There are two views in the interaction panel: *Group Tree* (see Figure 4 (a)) and *View Mode* (see part B of Figure 4 (b)). The *Group Tree* provides a tree structure for MCSs. The *View Mode* provides some view modes and zooming modes for the matrix view of *MCS Matrix*. The view modes "Column Element Type" have been introduced in Section 4.5.1. The interaction of *Group Tree* and other modes of *View Mode* will be described in Section 5.

4.5.3 Information Panel

The information panel presents the general information about the analyzed data set and the matrix view related data (see part C of Figure 4 (b)). There are four information areas:

- The first information area presents distributions of MCSs. This statistical information provides an overview of the safety state of the system. Users can have a rough idea of the safety of the system.
  - The first diagram visualizes the distribution of order levels. It displays how many MCSs are in each safety level for the order of MCSs.
  - The second diagram visualizes the distribution of probability safety levels.
- The second information area shows information about the current column, either for a CFT or for a BE. It supports to get instant information of BEs/CFTs without having to check fault tree diagrams.
- The third information area gives information about the selected row, i.e., a selected MCS. It gives detailed information about the current MCS.
- The last information area provides statistical information of the analyzed fault tree model. Such as count of BEs, count of MCSs.

5 Interaction with MCS Matrix

Interaction is an important aspect of a visualization system. It helps users to explore information. In this section, the interactions with the *MCS Matrix* is introduced. In the *MCS Matrix* the interaction "Semantic Zooming" is performed in the matrix view directly while others can be performed by using the interaction panel. Some view modes in the interaction panel have been described in Section 4.5.1. The rest of the interaction panel are "MCS Grouping Mode", "Column Display Mode", "Table Zooming", 'Table Selection Marking", and "Group Tree". These are introduced in this section.

5.1 MCS Grouping Mode

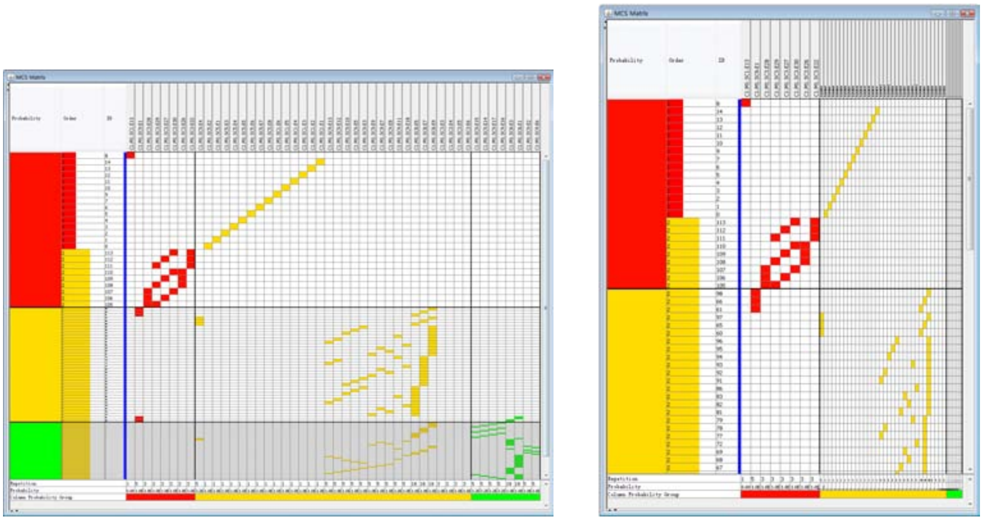
In 4.5.1 the grouping functionality has briefly been mentioned. In this paragraph, the detailed features of this interaction are introduced. "MCS Grouping Mode" is used to sort and group MCSs into different safety levels. There are three modes: *ID*, *Order*, and *Probability*. For *ID*, MCSs are displayed in an ascending sequence sorted by ID. All MCSs are classified in one group. When choosing *Order*, MCSs are grouped by order. The probability information will put on the second column. When choosing *Probability*, MCSs are grouped by probability according to the range defined for each safety level. MCSs are sorted in descending order in each group and the column *probability* will be put in the first place. The order information will be put in the second column. The MCS grouping function is useful to get the most important MCSs for different objectives.

5.2 DOI Zooming

There are two important zooming interactions in ViSSaAn, degree of interest (DOI) zooming described next and semantic zooming being described in Section 5.3.

When the count of MCSs and/or BEs is huge, the representation will be a challenge for MCS analysis. As much information should be shown in a limited amount of space as possible, particularly the interesting information. The DOI zooming is designed to meet this demand. DOI zooming is a zooming technique whose scale depends on the *degree of interest (DOI)*. The DOI distortion was proposed in [11] and used in Table Lens in [29, 31].

In ViSSaAn the DOI depends on the safety levels. The more critical the safety situation, the higher the interest and the larger the zoom scale. Each group has a zoom scale, because groups have one-to-one mappings to safety levels. Therefore, the display size of groups depends on the zoom scale that can be obtained with the help of the safety levels. Basically, the idea of DOI zooming is to compress the display space of uninteresting information, in order to guarantee the display of the overview of a large-scale data set. We assigned following scales to the zooming levels: the group for the critical level has zoom scale "1", the group for the moderate level has zoom scale "0.3", and the group for the acceptable level has the smallest zoom scale "0.15". The display size of a group for critical level remains unchanged. The remaining groups get smaller display space according to their levels. The DOI zooming consists of the "Row DOI Zooming Mode" and the "Column DOI Zooming Mode". By the "Row DOI Zooming Mode", different row heights are used for different MCS groups (see Figure 7 (a)). If a row is in the group for the critical level, it will have the original height. If it is in the group for the moderate level, it will have a smaller height. If it is in the group for the acceptable level, it will have the smallest height. The "Column DOI Zooming Mode" is very similar to the "Row DOI Zooming Mode". By applying the "Column DOI Zooming Mode" the grouped columns have different width because of the different zoom



(a) Row DOI Zooming - rows of different groups have different height

(b) Column DOI Zooming - columns for different safety levels have different width

Figure 7 DOI Zooming.

scales. By combining both DOI zooming methods, the display space is reduced both for rows and for columns. The column elements that are used by the critical MCSs and those that are classified in the critical level have the largest cells (see Figure 8).

In this way, there is enough space to show the important information, and the less important information is shown as context. Therefore, DOI zooming efficiently shows the large MCS information in a limited screen space. At the same time, DOI zooming highlights critical MCSs and critical column elements. The effect is shown in the Figure 8.

5.3 Focus&Context with Semantic Zooming

The MCS analysis information is often associated to fault tree diagrams. The fault tree structure contains a lot of useful information for safety analysis, such as the path from a BE to the top event or the logical connective between BEs. Users can learn how BEs effect the top event. Usually, users have to turn to fault tree diagram from the MCS analysis view to see the detailed information of the fault tree, e.g., the structure of the fault tree or the attached data. This is not convenient and it is easy to miss some context in the MCS information. In order to solve this problem, the fault tree structure should be integrated in the matrix view. The component concept is a powerful tool which needs to be reflected in ViSSaAn. Analyzing the CFTs is more efficient than checking the whole fault tree. Therefore, integrating CFT structure with corresponding data into the matrix view becomes an important task.

Focus&Context with semantic zooming is applied in ViSSaAn to accomplish this task. It was investigated in the projects *Pad, Pad++* [3], and *Jazz* [4], and was evaluated for program visualization in [36]. It was used for the matrix view in [1] as well. With Focus & Context more detailed content of the focused element can be shown dynamically without losing context information. In this paper the technique Focus&Context with semantic zooming is called semantic zooming for convenience.

ViSSaAn provides semantic zooming to access the structure of CFT. If a cell is double

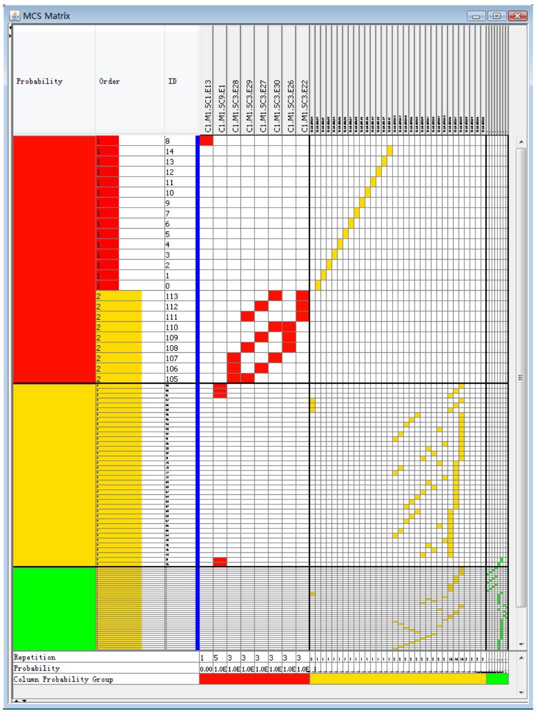
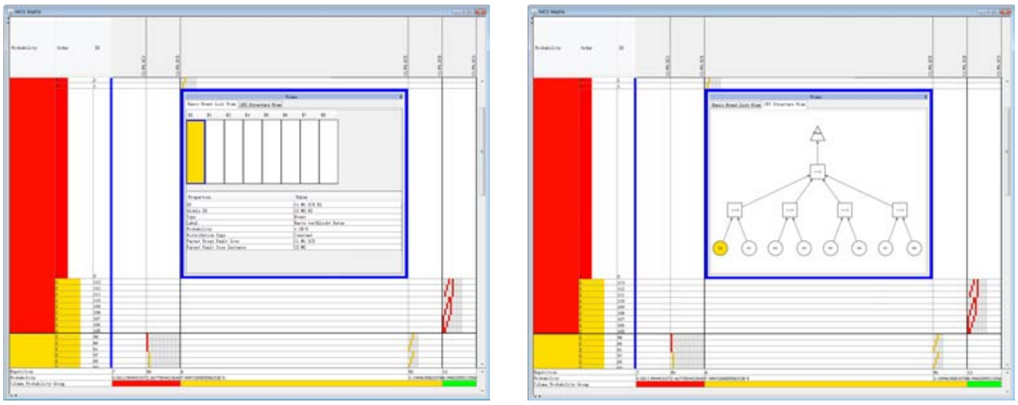


Figure 8 Row DOI Zooming + Column DOI Zooming - The most critical BEs are presented by the columns having the largest cells.

clicked, the cell will zoom out. There will be two views accessible using tabs in the zoomed cell. The first one is the *Basic Event List View.* It consists of a list of blocks and a data table (see Figure 9 (a)). Each block represents a BE in the selected CFT. The BEs represented by color filled blocks are used by the current MCS. The colors are according to the safety levels of the BEs. When a block is selected, the data table will display the data of the corresponding BE. This view gives the detailed data of the current CFT. The second one displays the *CFT Structure View* (see Figure 9 (b)). It shows a traditional structure of current CFT with colored BEs. Like in the *Basic Event List View*, the colors depend on the safety levels. In this view, only the BEs that are used by the MCS in current row are colored. With semantic zooming the detailed information of a CFT can be obtained without losing the context information of MCS analysis. The semantic zooming is similar for the "Column Element Type" *Basic Event*, but there is only the *CFT Structure View.* Pan, Zoom, and ZoomToFit are also provided as common interactions for the node-link diagrams in the *CFT Structure View.*

5.4 Column Display Mode

The "Column Display Mode" is available when using the *Component Fault Tree* mode. It provides an additional *Compact Mode* . By default, columns have the same width allowing the CFTs to be distinguished more clearly. Using *Compact Mode*, the width of a column depends on the number of the BEs contained in the CFT. Columns might have different width. As mentioned before, the column element type "Component Fault Tree" can be treated as the grouping of BEs according to CFTs. During *Compact Mode*, the BEs look more continuous, so that it is more reliable to check BEs grouped by CFTs. It reduces the column display space as well.



(a) Basic Event List View - a list of BEs is in the upper part; a data table is in the lower part

(b) CFT Structure View - *CFT Structure* integrated in the matrix view

Figure 9 Semantic Zooming.

5.5 Table Zooming

Usually, the space of rows and columns will be compressed in order to adapt to screen space. ViSSaAn provides "Row Zooming" and "Column Zooming" for obtaining a suitable overview of MCSs. These functionalities can zoom out / zoom in row height and column width. The zoom scale can be changed flexibly. This is a simple idea, but it is effective for representing the large-scale data set.

5.6 Table Selection Marking

The "Table Selection Marking" points out the selected row or /and column with a light-gray horizontal rectangle or / and vertical rectangle. There are two marking modes for table selection, the "Mark Selected Row" and the "Mark Selected Column". It helps to highlight the selected cell in a large matrix.

5.7 Group Tree

When performing zooming functionalities for representing a large number of MCSs in *MCS Matrix*, it will be hard to locate the desired MCS, because the rows are strongly compressed and the indicators of MCSs. In this case, *Group Tree* will show its usefulness. The *Group Tree* and the *matrix view* are coordinated views. They show the sorted and grouped MCSs in a tree structure. The *Group Tree* is associated with the MCS grouping modes, so that the information of the MCS on the tree is according to the grouping property. The *Group Tree* is helpful for positioning MCSs (see Figure 4 (a)).

6 Example

6.1 Data

We present how CFT models of real data can be analyzed using ViSSaAn. *RAVON (Robust Autonomous Vehicle for Offroad Navigation)* is a mobile robot made by the Robotics Research Lab of the University of Kaiserslautern [35]. It is used as an application example by the project *ViERforES* [42]. *RAVON* is a typical large and complex embedded system, and

Table 1 Settings of the example

	Acceptable	Moderate	Critical
BE Range	$(0, 1e^{-5})$	$[1e^{-5}, 1e^{-4})$	$[1e^{-4}, 1)$
BE Color	green	yellow	red
MCS Range	$(0, 1e^{-10})$	$[1e^{-10}, 1e^{-8})$	$[1e^{-8}, 1)$
MCS Color	green	yellow	red

safety is important for its survival. The safety of *RAVON* is analyzed with the CFT analysis using *ESSaREL*. These CFT models of *RAVON* are then explored using ViSSaAn. For our example the Top Event “C1.M1.POut1” of the CFT “Main, Sensoren, Aktoren” is selected in a CFT Model. There are 118 MCSs and 48 BEs contained in 5 CFTs for this top event.

6.2 Settings

After loading the data, the value ranges of probabilities of BEs, the range of probabilities of MCSs, the range of orders of MCSs, and the colors for the ranges are specified (see Table 1).

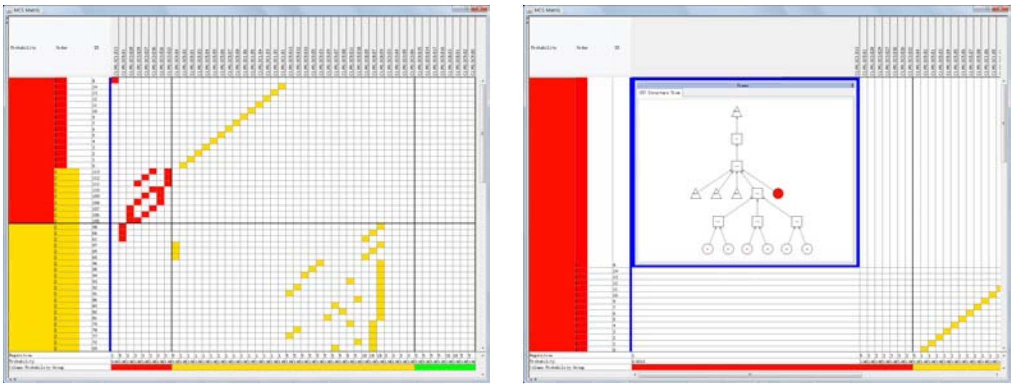
6.3 Scenario

This scenario concerns to explore the important safety problems from the MCS analysis information. The result of the scenario is presented in the Figure 10(a). In the *MCS Matrix*, the “MCS Grouping Mode” is set to *Probability*, in order to group MCSs according to different safety levels by probability. The group with red is the one with a “critical level”. Next, we check the column for order of MCSs. There are 15 MCSs with order 1 and 9 MCSs with order 2 in the “red group”. Because the MCSs with order 1 is more critical than ones with 2, we just focus on the MCSs having one BE in the group. Then, we check the sorted BEs, the BEs in red are critical. Now, we associate the consideration of critical MCSs found before, only the MCS with ID “8” is found to be the most critical MCS, because its BE “C2.M1.SC1.E13” is colored in red, i.e., this BE is critical. The rest critical MCSs found before have BEs in yellow. Thus, The BE “C2.M1.SC1.E13” is the declared important safety problem. Then, we can perform semantic zooming for the cell at the intersection between the row for the MCS with ID “8” and the column for the BE “C2.M1.SC1.E13” in order to check the fault tree diagram to find how this BE effects the Top Event (see Figure 10 (b)). The obtained information helps making a decision to solve the problem. We can also use DOI zooming to show as much information as possible on the screen (see Figure 8). The DOI zooming plays a role with highlighting of the important field as well.

7 Discussion

ViSSaAn is a multi-view visualization system for MCS analysis. It extends the common tabular form to an interactive matrix representation with suitable visual features. The visual features, e.g., coloring, represent the MCS information intuitively.

Generally, the correlation between MCSs and their BEs are the focus for qualitative analysis. In consideration of the contribution of the component concept for the FTA, ViSSaAn also visualizes the involved CFTs in the *MCS Matrix*. ViSSaAn combines the CFT structure and the matrix view by focus&context with semantic zooming interaction. The details of the



(a) The MCS in the first row is most critical. The BE inside is the critical problem

(b) Semantic zooming for the critical BE found

Figure 10 Results of the example.

related CFT can be check, without switching to another fault tree diagram. In this situation, the focus is not interrupted and the context is not missing. This interaction makes checking the CFT structure more convenient and efficient. ViSSaAn uses safety levels for classifying MCSs, BEs, and CFTs. A safety level is assigned to each element according to the specified property. It helps finding the urgent problems efficiently and precisely in a large data set. With the help of coloring, the safety levels are easier to identify. The grouping functionalities aggregate elements with the same safety level, so that users can easily find them. Sorting is also a characteristic of ViSSaAn. Each MCS has a sorted BEs sequence, when the order of the MCS is more than one. Therefore, it is easy to distinguish which is the most important one. Zooming functionalities help analyzing large-scale data sets. Row and Column Zooming can change the size of rows and columns. With DOI zooming, the important information has priority using the screen space. By these zooming techniques, as much information as possible can be shown in the limited screen space.

MCS Matrix is not a simple extension of the tabular form for MCSs information, but an integrated system. The statistical information can be used to understand the safety situation of systems. Combining it with a couple of data tables in the information panel, makes it convenient to get the information of MCSs, BEs, and CFTs in the *MCS Matrix*. ViSSaAn was designed to improve the MCS analysis from the angle of view of information representation. There is no optimization for algorithms and processes. It uses visualization techniques to improve the understandability of the MCS analysis and the readability of large-scale data sets. ViSSaAn provides an intuitive, convenient, and rapid visualization system to support the safety analysis using MCSs. It helps to understand the information analyzed to find the safety areas of a system, and can be used in support for decision making.

8 Conclusions

This paper introduced ViSSaAn, a visualization system for MCSs and CFTs. A matrix-based visualization called *MCS Matrix* is used to present MCS analysis information. Colors are used to encode different safety levels. Grouping functionalities are used to group the elements with the same safety levels. DOI zooming and Table zooming are used for large-scale data sets. Semantic zooming is used in the *MCS Matrix* in order to show details and the inner structure of CFTs. Users can focus on the detailed information while having the MCS information as context. With ViSSaAn users can better understand and analyze fault tree models. Finding safety problems from the MCS analysis information is convenient. Overall, ViSSaAn provides methods to visualize MCSs information improving the MCS analysis for fault trees of complex systems in representation aspect. In the future, ViSSaAn will be extended in several ways, including considering new visual metaphors and interactions. More safety level criteria will be considered.

Acknowledgements

This paper is supported by the International Research Training Group of DFG (IRTG 1131). We thank the Robotics Research Lab of the University of Kaiserslautern [35], the Fraunhofer Institute for Experimental Software Engineering (IESE) [16], and our colleagues for the data, their help, and their comments.

References

1 James Abello and Frank van Ham. Matrixzoom: A visual interface to semi-external graphs. *IEEE Symposium on Information Visualization (INFOVIS04)*, 2004.

2 ALD. RAM Commander. http://www.aldservice.com, Online; accessed 15-July-2010.

3 B.B. Bederson, J.D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas. A Zoomable Graphical Interface for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing*, 1:3–31, 7 1996.

4 B.B. Bederson, J. Meyer, and L. Good. Jazz: an extensible zoomable user interface graphics toolkit in Java. *Proceedings of the 13th annual ACM symposium on User interface software and technology (UIST'2000), San Diego, CA*, pages 171–180, 2000.

5 Marco Bozzano and Adolfo Villafiorita. The FSAP/NuSMV-SA Safety Analysis Platform. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(1):5–24, 2007.

6 Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman, editors. *Readings in information visualization: using vision to think.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

7 DIN 25424. Fehlerbaumanalyse (Fault Tree Analysis). *German Industry Standard (Part 1 & 2)*, 1981/1990. Beuth Verlag, Berlin.

8 ESSaREL. ESSaREL. http://www.essarel.de, Online; accessed 15-July-2010.

9 Nasser S. Fard. Determination of minimal cut sets of a complex fault tree. *Computers & Industrial Engineering*, 33(1-2):59–62, 1997. Proceedings of the 21st International Conference on Computers and Industrial Engineering.

10 J.-D. Fekete, N. Elmqvist, T.-N. Do, H. Goodell, and N. Henry. Navigating with the Zoomable Adjacency Matrix Explorer. *Technical report, INRIA Research Report (Paris)*, RR-6163, 1997.

11 George W. Furnas. Generalized fisheye views. *In Proceedings of the ACMSIGCHI Conference on Human Factors in Computing Systems*, pages 16–23, 04 1986.

12 J.B. Fussel and W.E. Vesely. A New Methodology For Obtaining Cut Sets From Fault Trees. *ANS Trans*, 15, 1972.

13 A.F. Hixenbaugh and The Boeing Company. Fault Tree for Safety. *D6-53604*, 1968.

14 IEC61508. Functional safety of electrical/electronic/programmable electronic safety-related systems. *International Standard IEC 61508*, 2000.

15 IEEE. IEEE Std 610.12-1990. *IEEE Standard Glossary of Software Engineering Terminology*, 1990.

16 IESE. Fraunhofer Institute for Experimental Software Engineering (IESE). http://www.iese.fraunhofer.de, Online; accessed 15-July-2010.

17 DIN EN ISO. DIN EN ISO 8402. *Quality management and quality assurance – Vocabulary*, 1994.

18 ISOGRAPH. FaultTree+. http://www.isograph-software.com, Online; accessed 15-July-2010.

19 ITEM-Software. ITEM ToolKit. http://www.itemtoolkit.com/, Online; accessed 15-July-2010.

20 J.Vatn. Finding minimal cut sets in a fault tree. *Reliability Engineering & System Safety*, 36(1):59–62, 1992.

21 Bernhard Kaiser. Integration von Sicherheits- und Zuverlaessigkeitsmodellen in den Entwicklungsprozess eingebetteter System. *Softwaretechnik-Trends 22(4) Gesellschaft fuer Informatik (Hg.)*, 2002.

22 Bernhard Kaiser. A fault-tree semantics to model software-controlled systems. *Softwaretechnik-Trends 23(3) Gesellschaft fuer Informatik (Hg.)*, 2003.

23 Bernhard Kaiser, Peter Liggesmeyer, and Oliver Maekel. A New Component Concept for Fault Trees. *Proceedings of the 8th Australian workshop on safety critical systems and software (SCS' 03)*, 2003.

24 Chakib Kara-Zaitri. An improved minimal cut set algorithm. *International Journal of Quality & Reliability Management*, 13, 1996.

25 Dimitri Kececioglu. *Reliability Engineering Handbook*, volume 2. DEStech Publications, Inc, 1991.

26 Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, and Chris North, editors. *Information Visualization: Human-Centered Issues and Perspectives.* Springer-Verlag, Berlin, Heidelberg, 2008.

27 John Knight. *Safety critical systems: challenges and directions, ICSE '02: Proceedings of the 24th International Conference on Software Engineering.* ACM, New York, NY, USA, 2002.

28 Kyoichi Nakashima and Yoshio Hattori. An efficient bottom-up algorithm for enumerating minimal cut sets of fault trees. IEEE Trans. Reliab., R-28, (5) 353 (December 1979). *Microelectronics and Reliability*, 20(4):543–543, 1980.

29 Peter Pirolli and Ramana Rao. Table lens as a tool for making sense of data. *Proceedings of the workshop on Advanced visual interfaces*, pages 67–80, 1996.

30 Prefuse. Prefuse. http://prefuse.org/, Online; accessed 15-July-2010.

31 Ramana Rao and Stuart K. Card. The Table Lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI '94)*, 1994.

32 Marvin Rausand and Arnljot Hoyland. *System Reliability Theory: Models, Statistical Methods, and Applications, Second Edition.* Wiley-Interscience, 2003.

33 RELEXSOFTWARE. Relex Architect. http://www.relexsoftware.co.uk, Online; accessed 15-July-2010.

34 RELIASOFT. BlockSim. http://www.reliasoft.com/BlockSim, Online; accessed 15-July-2010.

35 Robotics Research Lab. The Robotics Research Lab of the University of Kaiserslautern. http://agrosy.informatik.uni-kl.de, Online; accessed 15-July-2010.

36 Kenneth L. Summers, Timothy E. Goldsmith, Steve Kubica, and Thomas P. Caudell. An experimental evaluation of continuous semantic zooming in program visualization. *Information Visualization, IEEE Symposium on*, 0:20, 2003.

37 SUN. JAVA. http://www.java.com, Online; accessed 15-July-2010.

38 SYNCOPATIONSOFTWARE. DPL-faulttrees. http://www.syncopationsoftware.com/faulttree.html, Online; accessed 15-July-2010.

39 Frank van Ham. Using multilevel call matrices in large software projects. *IEEE Symposium on Information Visualization*, 2003.

40 W.E. Vesely, F.F. Goldberg, N.H. Roberts, and D.F. Haasl. Fault Tree Handbook. *U.S. Nuclear Regulatory Commission*, 1981. NUREG-0492, Washington.

41 William Vesely, Joanne Dugan, Joseph Fragola, Joseph MinarickIII, Jan Railsback, and Michael Stamatelatos. Fault Tree Handbook with Aerospace Applications. *NASA*, 2002.

42 ViERforES. Virtuelle und Erweiterte Realitaet fuer hoechste Sicherheit und Zuverlaessigkeit von Eingebetteten Systemen (ViERforES). `http://www.vierfores.de`, Online; accessed 15-July-2010.

43 Colin Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann, 2000.

44 G. Zipf. Computation of minimal cut sets of fault trees: Experiences with three different methods. *Reliability Engineering*, 7(3):159–167, 1984.