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 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].

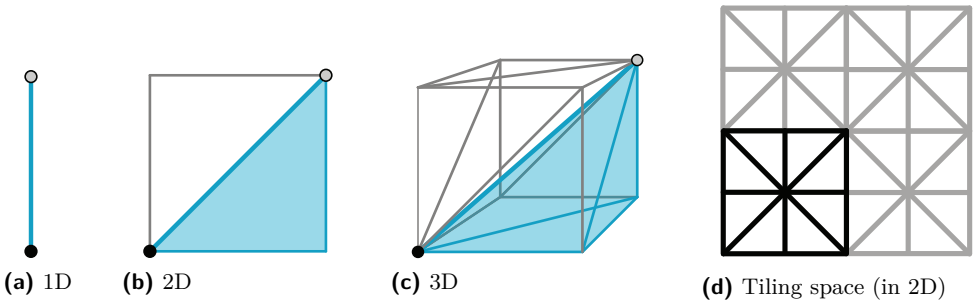


Figure 1 Kuhn-subdivided hypercubes in 1D **(a)**, 2D **(b)** and 3D **(c)**, highlighting one of the $d!$ simplices (blue). All edges are aligned with the diagonal of an axis-aligned hypercube. Translated or reflected **(d)** copies of a Kuhn-subdivided cube can tile space.

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].

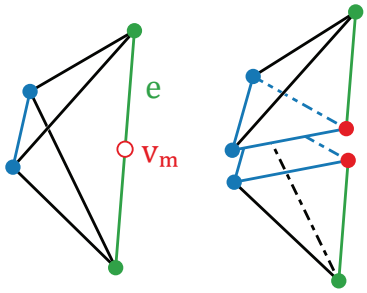


Figure 2 Bisection of a tetrahedron **t** along the plane defined by the midpoint $\mathbf{v}_m$ of an edge **e**, and the two vertices of **t** not incident in **e**.

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.

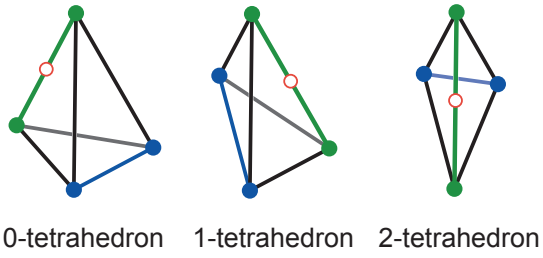
The *Regular Simplex Bisection (RSB)* scheme for tetrahedral meshes is defined by the recursive application of either of the above schemes to the tetrahedra in a Kuhn-subdivided cube $\mathcal{K}(h)$, and cyclically generates tetrahedra whose shapes belong to three similarity classes (see Figure 3a).[2] 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.

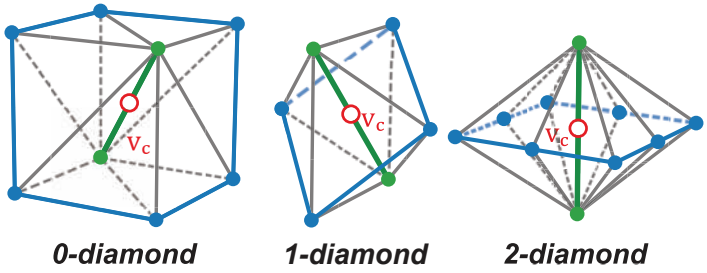
The cluster of tetrahedra sharing the same bisection edge is referred to as a *diamond* [12, 18, 43, 56], and we refer to the bisection edge as its *spine*. Consequently, the RSB scheme generates three similarity classes of diamonds, in correspondence to its three similarity classes of tetrahedra (see 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*.

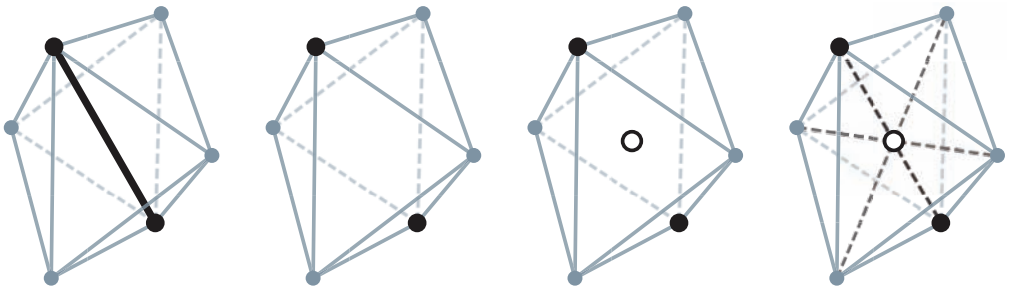
the diamond's domain. The diamond subdivision paradigm, in which $(d-i)$-cells are refined at step i, has been generalized to cell complexes in higher dimensions [43].

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

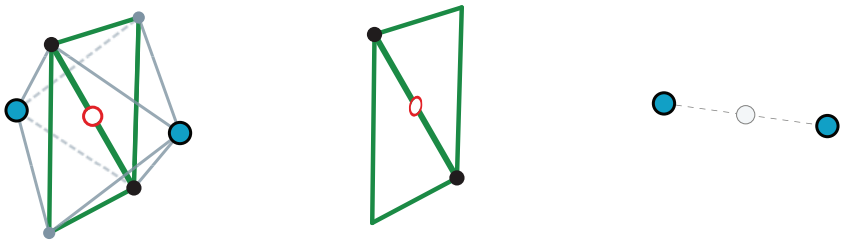
In this section, we review the basic ingredients of approaches that use nested tetrahedral RSB meshes. We classify such approaches on the basis of the choice of the basic primitives of the representation and on the method by which the data structures are queried.

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

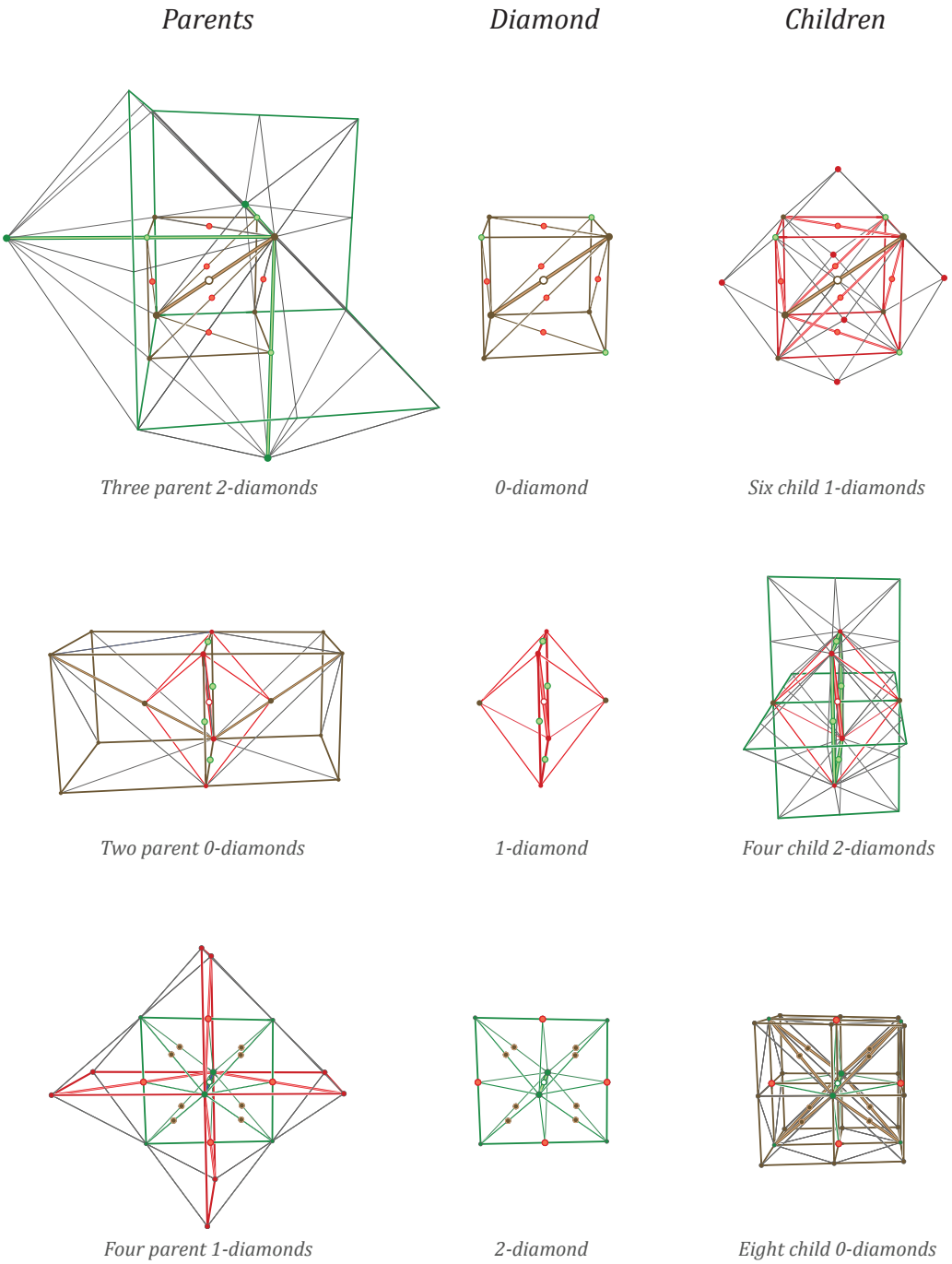


Figure 6 Diamond hierarchy in 3D. A diamond's (middle column) tetrahedra are generated during the subdivision of its *parents* (left column). The central vertex of each parent coincides with a vertex of the diamond, while those of its *children* (right column) coincide with the midpoints of a subset 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.

Approach	Query				Error		Optimizations							
	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.