Helmut Alt , Bernard Chazelle,
Emo Welzl (editors):

Computational Geometry

Dagstuhl-Seminar-Report; 22
07.10.-11.10.91 (9141)

ISSN 0940-1121

Tel.: +49-6871 - 2458
Fax: +49-6871 - 5942

Das Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloß Dagstuhl
Informatik, Bau 36
Universität des Saarlandes
W - 6600 Saarbrücken
Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de

Report

of the Second Dagstuhl Seminar on

Computational Geometry

October 7th - October 11th, 1991

The second Dagstuhl Seminar on Computational Geometry was organized by Helmut Alt (FU Berlin), Bernard Chazelle (Princeton University) and Emo Welzl (FU Berlin). The 31 participants came from 8 countries, 12 of them came from North America and Israel.

29 lectures were given at the seminar, covering quite a number of topics in computational geometry. Unlike last year, there was no special concentration on any subject. In fact, there were talks on graph algorithms, parallel algorithms, motion planning, application-oriented problems, numerical robustness, similarity and congruence, randomized algorithms, dynamic algorithms, and a talk on implementations.

As last year, an open problem session was held on Monday evening, chaired by Micha Sharir. It was stated that most of the problem discussed in last year's session had been solved (or at least some progress had been made). Let us hope that this year's session (reported here by Micha Sharir) will prove as fruitful.

Berichterstatter: Otfried Schwarzkopf

Participants: Helmut Alt, Freie Universität Berlin
Boris Aronov, Polytechnic University, New York
Franz Aurenhammer, Freie Universität Berlin
Chanderjit Bajaj, Purdue University
Mark de Berg, Utrecht University
Jean-Daniel Boissonnat, INRIA Sophia-Antipolis
Bernard Chazelle, Princeton University
Herbert Edelsbrunner, U. of Illinois at Urbana-Champaign
Steven Fortune, ATT Bell Labs Murray Hill
Danny Halperin, Tel Aviv University
Marc van Kreveld, Utrecht University
Kurt Mehlhorn, MPII Saarbrücken
Stefan Meiser, MPII Saarbrücken
Joe Mitchell, SUNY Stony Brook, NY
Heinrich Müller, Universität Freiburg
Stefan Näher, MPII Saarbrücken
Jürg Nievergelt, ETH Zürich
Mark Overmars, Utrecht University
János Pach, Hungarian Academy of Sciences & Courant Inst.
Ricky Pollack, Courant Institute
Marie-Françoise Roy, Université de Rennes I
Jörg-R. Sack, Carleton University
Stefan Schirra, MPII Saarbrücken
Otfried Schwarzkopf, Freie Universität Berlin
Raimund Seidel, University of California, Berkeley
Micha Sharir, Tel Aviv University and Courant Institute
Michiel Smid, MPII Saarbrücken
Jack Snoeyink, University of British Columbia
Hubert Wagener, Technische Universität Berlin
Emo Welzl, Freie Universität Berlin
Peter Widmayer, Universität Freiburg

A Simple and Correct Proof of the Zone Theorem and its Generalizations

Micha Sharir (Tel Aviv University and Courant Institute, NYU)

Let H be a collection of n hyperplanes in d-space. The arrangement $A(H)$ is the decomposition of d-space into cells (of various dimensions) induced by the hyperplanes of H. Given another "base hyperplane" b, the *zone* of b in $A(H)$ is the collection of cells of $A(H)$ crossed by b, and the complexity of the zone is the total number of faces of all dimensions bounding the zone cells. The *Zone Theorem* states that the complexity of the zone of a hyperplane in an arrangement of n hyperplanes in d-space is $\mathcal{O}(n^{d-1})$ (with the constant of proportionality depending on d). As announced in the first Dagstuhl Seminar on Computational Geometry, a year ago, the original proof of the theorem, by Edelsbrunner, O'Rourke and Seidel, contains an error for $d \geq 3$. The talk presents a new proof, which is much simpler (and hopefully correct), and is based on a simple induction scheme [Edelsbrunner, Seidel, Sharir, to appear in SIAM J. Computing]. The talk also reviews several extensions of the technique, including: (i) A bound of $\mathcal{O}(n^{d-1}\log n)$ on the complexity of the zone of a bounded-degree algebraic surface (or convex surface) in an arrangement of hyperplanes; (ii) A bound of $\mathcal{O}(n^{\lfloor(d+k)/2\rfloor}\log^{d-k \pmod 2} n)$ on the complexity of the zone of a k-dimensional algebraic or convex surface in such an arrangement; and (iii) A bound of $\mathcal{O}(n^d \log^{\lfloor d/2\rfloor -1} n)$ on the sum of squares of cell complexities in a hyperplane arrangement. The talk also mentions several algorithmic applications of the new results, to translational motion planning in 3-space, to point location among hyperplanes, and to ray-shooting among triangles in 3-space. [The results reported are by many authors, including B. Aronov, J. Matoušek, M. Pellegrini, M. Houle, T. Tokuyama, B. Chazelle, J. Friedman, and others]

Two Applications of Power Diagrams

Franz Aurenhammer (FU Berlin)

Let S be a set of n points (called sites) in $\mathbb{R}^d$. The Voronoi diagram of S is the partition of $\mathbb{R}^d$ into regions such that all points within a fixed region have the same closest site. The *power diagram* of S is the generalization of the Voronoi diagram where closeness is specified by the power function $pow(x,s) = d^2(x,s) - w(s)$ of a point x w. r. t. a site s, $w(s)$ denoting the *weight* of s, a real number. Given some finite set X of points, a power diagram of S defines an assignment $X \mapsto S$ in a natural way (according to the partition of X induced by the regions). The following general theorem can be shown to hold:

Let S be any set of sites, let X be any set of points (both finite), and choose any capacities $c(s) \in \mathbf{N}_0$ for the sites $s \in S$ such that $\sum_{s \in S} c(s) = |X|$. There always exists weights $w(s)$ such that

$$|X \cap region(s)| = c(s), \qquad \forall s \in S$$

In other words, a partition of a given point set X into clusters of given size always can be realized by a power diagram of a given set S of sites. The proof is algorithmic and yields a method for finding suitable weights in polynomial time. Among the applications of the theorem (resp. the algorithm) is a method for finding clusters with minimum sum of variances and a method for optimum matching of point sets under the least squares measure.

Another theorem states that, for any simple cell complex C in $\mathbb{R}^d$, $d \geq 3$, there are sites and weights such that the resulting power diagram coincides with C. This allows one to specify any simple cell complex using only $\mathcal{O}(n)$ objects of constant complexity. (C has n convex cells, d is considered a constant.) As one application, point location can be performed in an easy way.

Determining the Convex Hull of a Simple Polygonal Chain in Sublogarithmic Parallel Time

Hubert Wagener (Techn. Univ. Berlin)

Let C be a simple polygonal path in the plane. A new implicit data structure representing the convex hull of C was introduced. Using this datastructure several queries concerning the convex hull can be answered efficiently. For example, given a point p, it is possible to determine whether the point lies within the convex hull or outside in time $\mathcal{O}(\log n / \log k)$ using k processors of a CRCW-PRAM, where n denotes the length of C. In case the point p lies outside, the supporting lines to the convex hull passing through p can be found within this time bound, too.

Using this data structure an algorithm for computing the convex hull of C was given, that runs in time $\mathcal{O}(n/k + \log\log n)$ using k processors of a CRCW-PRAM. The computed convex hull can be represented either by the implicit data structure mentioned above or as a linked list of vertices in the order they appear around the convex hull.

The methods used in this solution of the convex hull problem can be applied to related problems, e.g. to the problem of computing the visibility region from a point with respect to a polygonal chain.

Computing and Verifying Depth Orders

Mark de Berg (Utrecht University)
joint work with Mark Overmars and Otfried Schwarzkopf

A depth order on a set of objects is an order such that object x comes before object x' in the order when x lies behind x', or, in other words, when x is (partially) hidden by x'. We present efficient algorithms for the computation and verification of depth orders on sets of n rods in 3-space. Our algorithms run in time $\mathcal{O}(n^{4/3+\epsilon})$, for any fixed $\epsilon > 0$. If all rods are axis-parallel, or, more generally, have only c different orientiations for some constant c, then the sorting algorithm runs in time $\mathcal{O}(n \log^3 n)$ time, and verification takes $\mathcal{O}(n \log^2 n)$ time. The algorithms can be generalized to handle triangles and other polygons instead of rods. They are based on a general framework for computing and verifying linear extensions of implicitly defined binary relations.

Description of the Connected Components of a Semi-Algebraic Set

Marie-Françoise Roy (Université de Rennes I)
joint work with Joos Heintz and P. Solerno

The connected components of a semi-algebraic set S are semi-algebraic sets.

Given a semi-algebraic set described by s polynomials of degree d in n variables, it is possible to design algorithms with complexity polynomial in d and s, single exponential in n, well parallelizable (with parallel complexity polylog in d and s, polynomial in n), solving the following problems

1. given two points x and x_0, decide if they belong to the same connected component of S
2. if they do, find a connected path connecting them
3. describe (as semi-algebraic sets) the connected components of S

If problem 2 is replaced (in the case when S is open) by problem 2': find a piecewise linear path connecting them, a very easy counterexample (due to Heintz, Kuich, Slissenko, Solerno) shows that the complexity is completely different, and intrinsically exponential in the degree d.

Convex Sets of Lines in Space

Ricky Pollack (Courant Institute, NYU)
joint work with J.E. Goodman, CUNY

Let $\mathcal{L}$ be a set of lines in $\mathbb{R}^3$, we define $\mathcal{L}^* = \{C \mid C \text{ convex}, \ell \cap C \neq \emptyset \, \forall \ell \in \mathcal{L}\}$ and for $\mathcal{K}$ a family of convex sets $\mathcal{K}^* = \{\ell \mid \ell \cap C \neq \emptyset \, \forall C \in \mathcal{K}\}$. The convex hull of $\mathcal{L}$ is defined to be the set $\mathcal{L}^{**}$ and a set of lines is convex if it is its own convex hull. This is a natural generalization of convex sets of points (let $\mathcal{L}$ be a set of points and use the same definition). We observe

1. The Helly number and Caratheodory number of this convexity space is ∞.
2. Any finite set of lines without parallels is convex.
3. If $\mathcal{L}$ is the set of lines tangent to a convex body K, then $\mathcal{L}^{**} = K^*$.
4. Let $\mathcal{L}$ be the set ruling a hyperbolic paraboloid H then $\mathcal{L}^{**} = \{\ell \mid \ell \cap H = \emptyset\} \cup \mathcal{L}$, but if $\ell' \in \mathcal{L}$ then $\mathcal{L} \setminus \ell'$ is convex.

Numerical Stability of Algorithms for 2D Delaunay Triangulations and Voronoi Diagrams

Steven Fortune (ATT Bell Labs)

Consider geometric algorithms implemented using floating point arithmetic. There are versions of the flipping, incremental, and sweepline algorithms for Delaunay triangulations in 2D with the following correctness properties. Let S be a set of n point sites. A circle inscribed in three sites is α-empty if the pseudodisc consisting of three circular arcs connecting the sites, each arc making angle α with the circle, is empty. The floating point Delaunay triangulation algorithms produce triangles with $\mathcal{O}(n\epsilon)$-empty circumcircles. The worst-case running time of the algorithms is unchanged. From the approximate Delaunay triangulation, an approximate Voronoi diagram can be produced in linear time. It can be used to anwer nearest-neighbor queries: the answer to the nearest-neighbor query is not always the true nearest neighbor, but is always within a relative amount of $\mathcal{O}(n\epsilon)$ of being the true nearest neighbor.

C^1-Smoothing of Polyhedra

Chanderjit L. Bajaj (Purdue University)

Polyhedra "Smoothing" is an efficient construction scheme for generating complex boundary models of solid physical objects. This talk presents efficient algorithms for generating families of curved solid objects with boundary topology related to the input polyhedron. Individual facets of a polyhedron are replaced by degree five implicit algebraic surface patches having local support. These degree five patches replace the C^0 contacts of planar facets with C^1 continuity along all interpatch boundaries. Selection of suitable instances of implicit surfaces as well as local control of the individual surface pathes are achieved via simultaneous interpolation and weighted least squares approximation.

The Cartographers' Line Simplification Problem

Jack Snoeyink, University of British Columbia

A portion of the cartographer's art is to distill the crucial information from a large amount of data and preset it on a map. One of their subproblems is a geometric approximation problem that they call "line simplification": given a polygonal chain C on n vertices, find a chain C' with fewer vertices that approximates C well. Of course, this problem crops up in many applications — the distinguishing features in cartography are a large amount of initial data, desire for fast algorithms, and some freedom in defining measures of similarity.

We have begun looking at greedy algorithms based on "fattening" the original chain and finding an approximation within the fattened region. Computing minimum-link paths of agiven homotopy class or computing ordered stabbing chains are two examples. These algorithms avoid minimization and have subquadratic behavior; yet they still give some guarantee on the goodness of the approximation—unlike many of the heuristics in practical use.

We also mentioned that the cartographers' favorite algorithm, a simple recursive procedure by Douglas and Peucker, can be improved from $\Theta(n^2)$ worst case to $\Theta(n \log n)$.

Drawing Graphs

János Pach (Hungarian Academy of Sciences and Courant Institute, NYU)

A *geometric graph* is a graph drawn in the plane by straight-line segments (which may cross each other). Given a class $\mathcal{H}$ of so-called *forbidden* geometric subgraphs let $t(\mathcal{H}, n)$ $(t_c(\mathcal{H}, n))$ denote the maximum number of edges that a (convex) geometric graph with n vertices can have without containing a subgraph from $\mathcal{H}$. Let $\mathcal{C}_k$ denote the class of all geometric graphs consisting of k pairwise crossing edges. Of course, $t(\mathcal{C}_2, n) = 3n - 6$, $t_c(\mathcal{C}_2, n) = 2n - 3$. For any fixed k, $n \to \infty$, we have $t_c(\mathcal{C}_k, n) = \mathcal{O}(n)$. Moreover, we have

Theorem 1 *Every convex geometric graph with n vertices and $m \geq (2k-1)n$ edges contains at least $\lfloor c_k m^{2k-1}/n^{2k-2} \rfloor$ k-tuples of pairwise crossing edges (for some $c_k > 0$).*

Does this statement remain true for not necessarily convex geometric graphs? It should be not too difficult to beat the following bound.

Proposition 2 $t(\mathcal{C}_3, n) = \mathcal{O}(n^{3/2})$

Let $\mathcal{T}_n$ denote the class of all planar drawings of a given tree of n vertices with noncrossing straight-line segments. Then every complete geometric graph with n vertices contains a subgraph from $\mathcal{T}_n$. Moreover, we have

Theorem 3 (P.-Törőcsik) *Given any rooted tree T of n vertices, and any set S of n points in the plane in general position, s_1, $s_2 \in S$, we can draw T by noncrossing straight-line segments on the vertex set S such that the image of the root is either s_1 or s_2. This embedding can be found in time $o(n^{3/2})$.*

Shortest Paths on Convex Polytopes

Boris Aronov (Polytechnic University, New York)

We survey several results concerning shortest paths on the surface of a convex polytope.

The object at the center of our investigation is the so-called 'star-unfolding' of the surface of the polytope, obtained by cutting this surface by shortest paths connecting a given point ('source') to all the vertices of the polytope. We show that this 'unfolding' indeed unfolds in the plane without self-overlap. Moreover, the structure of shortest paths emanating from the source is fully described by a certain Voronoi diagram computed in the plane and restricted to the unfolding. This fully justifies the relatively recent elegant algorithm of Chen and Han for computing shortest paths between the source and all other points on the polytope surface. In addition, the non-overlap of the unfolding and its relation to Voronoi-diagrams imply some improvements in a number of other shortest-path algorithms.

All Pairs Shortest Paths

Raimund Seidel (University of California, Berkeley)

The following algorithm solves the all-pairs-shortest-path problem on undirected graphs with unit edge weights in time $\mathcal{O}(M(n)\log n)$, where $M(n)$ is the time necessary to compute the product of two $n \times n$ matrices. Currently $M(n)$ is known to be $\mathcal{O}(n^{2.37\dots})$.

Input: undirected graph with vertex set $\{1, 2, \dots, n\}$ represented by adjacency matrix.

Output: distance matrix D, where $D[i,j]$ = number of edges on shortest $i - j$ path.

Function APSP (A: adjacency matrix) : distance matrix
  $Z := A \cdot A$
  Let B be $n \times n$ $0-1$ matrix s.t. $B[i,j] := 1$ iff $i \neq j$ and ($A[i,j] = 1$ or $Z[i,j] > 0$)
  If $B[i,j] = 1$ for all $i \neq j$ **then return** matrix $2B - A$
  $T := 2 \cdot \text{APSP}(B)$
  $X := T \cdot A$
  return matrix D **where** $D[i,j] := \begin{cases} T[i,j] & \text{if } X[i,j] \geq T[i,j] \cdot \mathit{degree}(j) \\ T[i,j] - 1 & \text{otherwise} \end{cases}$

Convex Hulls and Range Searching

Bernard Chazelle (Princeton University)

Convex Hulls: We sketch an algorithm for computing the convex hull of n points in $\mathbb{R}^d$ (d fixed) in deterministic $\mathcal{O}(n^{\lfloor n/2 \rfloor} + n \log n)$ time, which is optimal. The method is derived by derandomizing a probabilistic algorithm of Clarkson and Shor, using ϵ-net theory.

Range Searching: (joint work with Hervé Brönniman) We outline the proof of an $\Omega(n^{1-\frac{d-1}{d(d+1)}}/m^{1/d})$ lower bound on the time to answer a halfspace range query in $\mathbb{R}^d$, given n points and m storage cells. This lower bound holds in the Fredman/Yao arithmetic model of computation.

$\mathcal{O}(n^2)$ problems in Computational Geometry

Mark H. Overmars (Utrecht University)

There exist many problems in Computational Geometry for which $\mathcal{O}(n^2)$ are the best known time bounds. Examples are:

- Given a set of points, are three points collinear?
- Given a set of triangles, does their union contain a hole?
- Given a set of line segments, does there exist a non-trivial separating line?
- Given a set of halfplanes, find a point that is covered by most of them
- ...

We show that all these problems can be solved by one general technique in time $\mathcal{O}(n^2)$. Moreover, we establish strict relations between the complexity of the problems. In particular, we show that all these problems are at least as difficult as the problem:

- Given three sets of reals A, B, and C, do $a \in A$, $b \in B$, and $c \in C$ exist with $a + b = c$?

Also for this problem $\mathcal{O}(n^2)$ is the best known bound. Hence, there is no hope of improving the time bound of any of the other problems, without improving this base problem.

Measuring the Distance between Curves

Helmut Alt (FU Berlin)
joint work with Michael Godau

We consider the so-called *Fréchet-Metric* between curves, which is compatible with parametrizations and thus expresses resemblance between curves better than the Hausdorff-metric. For the case of polygonal chains P, Q with p, q edges respectively, we give an $\mathcal{O}(pq)$ algorithm, which given P, Q, and $\epsilon > 0$ decides whether the Fréchet-distance $\delta_F(P,Q) \leq \epsilon$. For actually computing $\delta_F(P,Q)$, given P, Q, we obtain an $\mathcal{O}(pq \log^2 pq)$ algorithm using parametric search directed by an efficient parallel sorting algorithm.

Variants of the problem are the *closed* Fréchet-metric defined for closed polygons P, Q, and the non-monotone Fréchet-metric allowing parametrizations to move on the curve in a nonmonotone way. Runtimes for these variants of the problem are nearly the same as for the original version.

Approximate Decision Algorithms for Approximate Point Set Congruence

Stefan Schirra (Max-Planck-Institut für Informatik, Saarbrücken)
joint work with Paul J. Heffernan

We consider the decision problem whether two sets of points $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$ in the plane are ϵ-congruent by a set $\mathcal{T}$ of allowed transformations, i.e. whether $\tau \in \mathcal{T}$ and a permutation π exist, such that $dist(\tau(a_i), b_{\pi(i)}) \leq \epsilon$. The best known decision algorithms for this problem have running time far from being practical. For $\mathcal{T}$ = set of translations, the best bound is $\mathcal{O}(n^6)$, and for $\mathcal{T}$ = set of isometries, the best bound is $\mathcal{O}(n^8)$. Therefore (α, β)-approximate decision algorithms for these problems become interesting. An algorithm is called (α, β)-approximate if it solves the decision problem of ϵ-congruence of A and B, if $\epsilon \notin [\epsilon^{\mathcal{T}}_{\text{opt}}(A,B) - \alpha, \epsilon^{\mathcal{T}}_{\text{opt}}(A,B) + \beta]$, otherwise the output of the algorithm is **Don't Know**. Here $\epsilon^{\mathcal{T}}_{\text{opt}} = \inf\{\epsilon \mid A \text{ and } B \text{ are } \epsilon\text{-congruent by a transformation in } \mathcal{T}\}$.

We give (γ, γ)-approximate decision algorithms (γ is an additional input parameter of the algorithms) with running time $\mathcal{O}(n^{1.5}(\epsilon/\gamma)^4)$ for translations and $\mathcal{O}(n^{2.5}(\epsilon/\gamma)^5)$ for general isometries. So for $\gamma = \epsilon/c$, c a constant, the running times are $\mathcal{O}(n^{1.5})$ and $\mathcal{O}(n^{2.5})$ compared to $\mathcal{O}(n^6)$ and $\mathcal{O}(n^8)$ resp. The algorithms are much faster than the best known complete decision algorithms, if ϵ is not near $\epsilon^{\mathcal{T}}_{\text{opt}}(A,B)$.

An $\mathcal{O}(n \log n \log\log n)$ Algorithm for the On-line Closest Pair Problem

Michiel Smid (MPI Saarbrücken)
joint work with Christian Schwarz

Let V be a set of n points in k-dimensional space. It is shown how the closest pair in V can be maintained under insertions in $\mathcal{O}(\log n \log\log n)$ amortized time, using $\mathcal{O}(n)$ space. Distances are measured in an arbitrary L_p-metric, where $1 \leq p \leq \infty$.

This gives an $\mathcal{O}(n \log n \log\log n)$ time on-line algorithm for computing the closest pair. The algorithm is based on Bentley's logarithmic method for decomposable searching problems. It uses a non-trivial extension of fractional cascading to k-space.

Shortest Paths of Bounded Curvature in the Plane

Jean-Daniel Boissonnat (INRIA Sophia-Antipolis)
joint work with André Cérézo and Juliette Leblond

Given two oriented points (M_i, θ_i), (M_f, θ_f), in the plane, we want to compute the shortest piecewise regular paths joining them, along which the curvature is everywhere bounded by a given $1/R > 0$. Minimizing the length is meaningful both in the class of paths which are C^1 and piecewise C^2, and in the slightly larger class of paths admitting a finite number of cusps.

Though the problem has been solved recently (Reeds & Shepp, Pacific J. of Math, Dec. 90), we propose an entirely different solution, both much simpler and better adapted to further generalizations. The essential tool we use is the powerful result of optimal control theory known as the "minimum principle of Poutryagin". We insist on local proofs in view of further work dealing with obstacles.

Point Location in Zones of k-Flats in Arrangments

Marc van Kreveld (Utrecht University)
joint work with Mark de Berg, Otfried Schwarzkopf, and Jack Snoeyink

Let $\mathcal{A}(H)$ be an arrangement of a set H of n hyperplanes in d-dimensional space. A k-flat is defined to be a k-dimensional affine subspace of d-space. The *zone* of a k-flat f with respect to H is the closure of all cells of $\mathcal{A}(H)$ that intersect f. It is known that this zone has complexity $\Omega(n^{\lfloor (d+k)/2 \rfloor})$. In this talk we show that we can do point location in the zone with considerably less storage, namely $\mathcal{O}(n^{\lfloor d/2 \rfloor + \epsilon} + n^{k+\epsilon})$, for arbitrarily small constant $\epsilon > 0$. For any point q in d-space, we can determine whether q lies in the zone of the k-flat or not. If it lies in the zone, then we can identify the cell of the zone that contains q. Such a query takes $\mathcal{O}(\log^2 n)$ time.

We also show how to test whether two flats are visible for each other with respect to H. To this end, we apply multidimensional parametric search, and obtain for a k_1-flat and a k_2-flat an $\mathcal{O}(n^{k_1+\epsilon})$ time algorithm if $k_1 \leq k_2 \leq 2k_1 + 1$, and the time bound is $\mathcal{O}(n^{k_1+1-\frac{k_1+1}{\lfloor k_2/2 \rfloor + 1}+\epsilon})$ if $k_2 > 2k_1 + 1$.

Planar Convex Embeddings of Graphs

Jörg-R. Sack (Carleton University, Ottawa)
joint work with F. Dehne and H. Djidjev

We present an algorithm to determine whether a graph can be convexly embedded and if so, for constructing such a convex embedding. An embedding is convex if each face of the graph (except the unbounded) is a convex polygon.

Our parallel algorithm runs in $\mathcal{O}(\log n)$ time and $\mathcal{O}(n)$ space and with the same number of processors as graph connectivity. The algorithm also provides a new linear time sequential method for the problem. The solution is based on two novel concepts, an optimal hierarchical graph decomposition scheme, and the notion of pseudo-complexity.

The Complexity and Construction of a Single Cell in Certain 3D Arrangements

Dan Halperin (Tel-Aviv University)

Given a collection of surfaces in 3-space, we wish to determine the maximum number of features bounding any single cell in the partitioning of space defined by these surfaces, i.e., what is the maximum number of vertices, edges and faces bounding a single three-dimensional component of the partitioning. We present a series of results, using various techniques, that give near-quadratic upper bounds on the complexity of a single cell in certain arrangements of surface patches. Most of our results relate to arrangements of "constraint surfaces" in the free configuration space of a "robot" moving among obstacles, where the single cell of interest is the cell that contains a point representing the initial free placement of the robot. The complexity of the entire arrangement in each case that we consider may be $\Theta(n^3)$ in the worst case and a single cell may have complexity $\Omega(n^2)$ or slightly larger.

We start with a near-tight bound $O(n^2\alpha(n))$ on the complexity of a single cell in the arrangement related to the motion of a so-called telescopic arm. Then we present a theorem that identifies a collection of topological and combinatorial conditions for a set of surface patches in space, which make the complexity of a single cell in an arrangement induced by these surface patches near-quadratic. Applying this result to specific arrangements we get bounds of the form $O(n^2 \log n)$ or $O(n^2\alpha(n) \log n)$. Finally, we mention a bound of $O(n^2 \log^2 n)$ by Aronov and Sharir on the complexity of a single cell in an arrangement of triangles and we show an extension of their result to arrangements of curved surfaces induced by the motion planning problem for certain rigid non-convex bodies moving among obstacles in the plane. For some of the arrangements that we study, we describe near-quadratic time algorithms to compute one cell.

Guard Algorithms Answer Stabbing and Intersection Queries on Fat Spatial Objects

Jürg Nievergelt (ETH Zürich)
joint work with Peter Widmayer

The variety of spatial data structures and retrieval algorithms developed in recent years suggests that it is difficult or impossible to design general-purpose structures that perform well across the entire spectrum of objects to be stored and queries to be processed—generality comes at the cost of performance and increased algorithm complexity. Thus simple algorithms that perform well on a restricted class of problems are clearly of interest.

Guard Algorithms answer stabbing and intersection queries on a dynamic collection of spatial objects embedded in $\mathcal{R}^d$ that satisfy a shape constraint. The objects stored must be **fat** in a technical sense that can be made precise in several ways, but always implies 2 requirements:

- convex, and
- "width"/"length" $\geq f > 0$, where the fatness constant f is characteristic of the class of objects to be stored.

Guard algorithms partition space in a hierarchical grid of cells and guard points. Objects are attached to cells and guard points in such a way that a stabbing query q is answered in a single path from a leaf to the root of the radix treee that represents the hierarchical grid.

Bicriteria Path Problems

Joe Mitchell (SUNY, Stony Brook)
joint work with E. Arkin and C. Piatko

We want to find paths between points s and t in a planar environment (e.g., among a set of polygonal obstacles). Much has been done to devise algorithms that find a shortest path according to a *single* criterion (e.g., Euclidean length, link distance, etc.). We study a broad class of *bicriteria* path problems, in which one asks if there exists a path whose length (according to criterion 1) is at most X, and whose length (according to criterion 2) is at most Y. In graphs, such problems are known to be NP-complete.

We show that many such problems are NP-complete in their geometric versions. In some cases (e.g., with "total turn" and Euclidean length as criteria), we give pseudo-polynomial time algorithms.

We look in more detail at the problem with criteria (number of links, Euclidean length). In this case, we give a fully-polynomial approximation scheme for general polygonal environments, and give a potentially faster algorithm for the case of simple polygons without holes. this algorithm relies on a careful analysis of the local structure of optimal paths (shortest k-link paths), in order to devise a binary search strategy.

A Method for Obtaining Randomized Algorithms with Small Tail Probabilities

Kurt Mehlhorn
joint work with H. Alt, L. Guibas, D. Karp, A. Wigderson

Let $X_1, X_2, \ldots$ be independent, identically distributed nonnegative random variables with common distribution function $f(x)$. For $(Y_1, Y_2, \ldots) \in (\mathcal{R}_{\geq 0})^\infty$, let i_0 be the least i such that $X_i \leq Y_i$. Let $T := Y_1 + Y_2 + \cdots + Y_{i_0-1} + X_{i_0}$. A strategy $\mathcal{S}$ is a probability measure on $(\mathcal{R}_{\geq 0})^\infty$. Let $b_{\mathcal{S},f}(t) = \text{prob}(T \geq t)$ and let $b_\mathcal{S} = \sup\{b_{\mathcal{S},f}(t) \mid f \text{ is such that } \int_0^\infty x f(x)dx = 1\}$. Then

1. $b_\mathcal{S}(t) \geq e^{-t}$ for all $\mathcal{S}$ and t
2. $\exists \mathcal{S} : b_\mathcal{S}(t) \leq e \cdot e^{-t}$ for all t
3. $\exists$ deterministic $\mathcal{S} : b_\mathcal{S}(t) \leq e^{-(t - O(\sqrt{t} \log t)}$

Hierarchical Marching Cube Algorithms

Heinrich Müller (University of Freiburg)
joint work with Michael Stark

Marching Cube Algorithms (Wyvill, McPheeters, Wyvill, The Visual Computer, 1986, Lorensen, Cline, SIGGRAPH 1987) calculate a polygonal approximation of the surface of a volume. The volume has to have the property to decide for every point in space whether it belongs to the volume. Examples are implicitly defined volumes, CSG volumes based on implicitly represented primitives, and rasterized volumes occuring in image processing and computer tomography. Usually, the resulting polygonal surface consists of a huge number of polygons limiting the speed of display on current rendering machines. We present two improvements, called hierarchical marching cube algorithms, which carry out an adaptive approximation of the surface. Experiments with medical data sets performed with a first implementation show a reduction to 50%–60% compared with the original approach even when allowing the new surface not to differ more than the sampling distance from that of the simple marching cube. Further, the new approaches solve the crack problem with only a constant amount of context information, thus carrying over an important property of the simple algorithm which allows an easy parallelization.

On the Post Office Problem

Otfried Schwarzkopf (Freie Universität Berlin)

The post office problem is one of the oldest query type problems in Computational Geometry. While the planar and three-dimensional versions are quite well understood, the only previous solution to the higher dimensional problem gives a data structure with space $\mathcal{O}(n^{\lceil d/2 \rceil + \epsilon})$ and query time $\mathcal{O}(\log n)$ (Clarkson). We show, following ideas by Chazelle & Friedman, that this can be reduced to space $\mathcal{O}(n^{\lceil d/2 \rceil} / \operatorname{polylog} n)$ and time $\mathcal{O}(\log^2 n)$. Using the recent Shallow-Cutting-Lemma by Matoušek, we can do so with only three levels of bootstrapping (instead of five as in Chazelle & Friedman's paper). The problem that combining two solutions requires a more powerful query than in the original problem is resolved by using Agarwal & Matoušek's Ray Shooting by Parametric Search, which, however, increases the query time to $\mathcal{O}(\log^2 n)$. It is left as an open problem to improve this to $\mathcal{O}(\log n)$.

Furthermore, it is argued that Chazelle's "antenna" of a query point can be used for the dynamic version of the post office problem. It is shown that it is sufficient to build a (static) point location structure for the cell of a newly inserted point site. Since this is possible by the above for all odd dimensions, we get a dynamic algorithm in the Mulmuley-Schwarzkopf setting with update time $\mathcal{O}(n^{\lceil d/2 \rceil - 1})$ and query time $\mathcal{O}(\log^4 n)$ for every odd dimension $d \geq 5$.

Using another technique, the problem can also be solved for dimension 6 (dimensions 2 and 4 were known before).

LEDA
A Library of Efficient Data Types & Algorithms

Stefan Näher (MPI Saarbrücken)

LEDA is a software components library that makes the results (data structures and algorithms) from the area of combinatorial computing available to non-expert programmers. The main features of LEDA are

- it gives precise and readable specifications
- it contains efficient implementations
- it offers a very comfortable graph data type
- it helps turning algorithms into efficient and elegant programs

We hope that LEDA will narrow the gap between algorithms research, teaching, and implementation.

Best Enclosures of Rectangles by Two Rectangles

Peter Widmayer (Universität Freiburg)
joint work with B. Becker, P. G. Franciosa, S. Gschwind, T. Ohler, and G. Thiemt

The page split operation in spatial data structures (such as the R-tree) motivates a family of geometric problems, two of which we solve here.

First, for a set of rectangles (parallel to the coordinate axes) we aim at finding two rectangles S, T such that

- each given rectangle is contained in S or in T
- each of S and T contains at least b of the given rectangles
- some measure of S and T is minimized

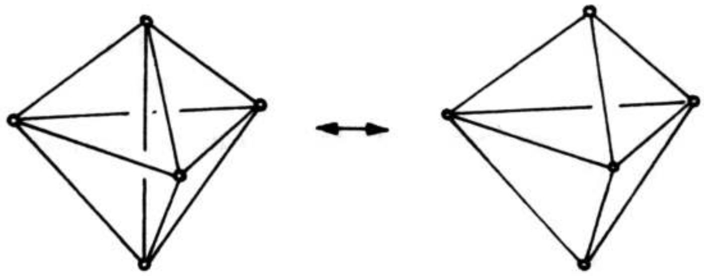
The measure minimization serves to achieve a good data structure performance; typical measures are the sum of the areas of S and T, the area of the intersection of S and T, etc. For the d-dimensional case and any measure function mf for which $mf(S,T) = mf(T,S)$ and $mf(S',T) \geq mf(S,T)$ for $S' \supseteq S$, we propose an algorithm that runs in time $\mathcal{O}(dn \log n + d^2 n^{2d-1})$ for n d-dimensional rectangles in the worst case. In the plane, the $\mathcal{O}(n^3)$ behavior occurs only in one particular case that will not always be of interest.

Second, if we change the problem formulation such that each given rectangle needs to be in $S \cup T$ (but not necessarily in S or T), we set $b = 0$, and we minimize the sum of the areas of S and T, the problem can be solved in optimal time $\mathcal{O}(n \log n)$ for n rectangles in the plane.

Flipping Works for Regular Triangulations

Herbert Edelsbrunner (University of Illinois at Urbana-Champaign)
joint work with Nimish Shah

Let S be a set of n points in $\mathbb{R}^d$, each with a real weight. The *regular triangulation* $\mathcal{R}(S)$ defined by S is the dual of its power diagram, also known as the Dirichlet tessellation and the Voronoi diagram under the Laguerre metric of S. In the assumed non-degenerate case $\mathcal{R}(S)$ is indeed a triangulation, that is, a simplicial cell complex that decomposes the convex hull of S. We prove the correctness of an incremental algorithm that, after adding the i^{th} point, uses an arbitrary sequence of flips to restore the regularity of the triangulation. This generalizes a result of Barry Joe who proved the correctness of the incremental flip paradigm for Delaunay triangulations (regular triangulations for points with equal weights) in $\mathbb{R}^3$. We also show that a non-incremental version of the flip paradigm can get stuck without reaching the regular triangulation, already in $\mathbb{R}^2$.



For 5 points in convex position in $\mathbb{R}^3$, a flip replaces 3 tetrahedra by 2, or vice versa.

The worst-case complexity of the incremental flip algorithm, not counting the time for finding the d-simplex that contains the new point p_i, is $\mathcal{O}(n^{\lceil \frac{d+1}{2} \rceil})$, and if the points are added in a random sequence the expected time/number of flips is $\mathcal{O}(n^{\lceil d/2 \rceil} + n \log n)$. There are ways to do the point location step without increasing the complexity.

Open Problem Session

(reported by Micha Sharir)

Bernard Chazelle: Given a chain-complex induced by a collection of unit-radius balls in d-space, compute its homology efficiently by using "direct" methods (e.g. divide-and-conquer via Mayer-Vietoris sequences).

János Pach: What is the maximum $f(n)$ such that, given n segments in the plane, one can always find at least $f(n)$ of them which are either pairwise disjoint or pairwise crossing.

Ramsey's theory implies $f(n) \geq c \log n$, and a simple construction (of $\sqrt{n}$ families, each consisting of $\sqrt{n}$ pairwise crossing segments) shows that $f(n) \leq \sqrt{n}$. What is the correct bound?

Mark Overmars: (1) Given a planar convex subdivision (of the entire plane), let G denote its dual graph. Does there always exist a spanning tree of G which has bounded degree? The problem is open even for an arrangement of n lines in the plane. The statement is false for nonconvex subdivisions, as can be seen by taking a large convex polygon, attaching triangles to each of its sides and considering the resulting subdivision.

(2) Let V be a set of n non-intersecting axis-parallel cubes in 3-space. Can the complement of V be decomposed into a collection W of $O(n)$ axis-parallel boxes?

(3) (This is a simple case of "curve shooting" in the plane.) Given a set of n arcs in the plane (of 'simple' shape, e.g. algebraic of low degree, etc.), preprocess it so that, given any point on one of the curves, one can quickly find the two neighboring intersections along the curve (the two endpoints of the edge of the arrangement of the curves which contains the query point).

(4) Devise an output-sensitive hidden surface removal algorithm that receives as input m pairwise disjoint convex polyhedra, consisting of n faces altogether, so that the running time of the algorithm is something roughly like $O(m^{2/3}k^{2/3} + \cdots)$, where k is the output size. There are algorithms with running time close to $O(n^{2/3}k^{2/3} + \cdots)$, and the challenge is to replace n by m in the leading term (when approximating curved objects by polyhedra, this can make a lot of difference).

Emo Welzl: (This problem arises in surface reconstruction.) Given a sequence $A = (a_1, a_2, \ldots, a_n)$ of real numbers, define $L(A) = \sum_{i=1}^{n-1} |a_{i+1} - a_i|$. Given two sequences $A = (a_i)_{i=1}^n$, $B = (b_i)_{i=1}^m$, we define a *shuffle* of A and B as a sequence C whose $m + n$ elements are the elements of A and of B, so that the elements of A appear in C in the same order that they appear in A, and similarly for B. The problem is: Given two sequences A and B, find a shuffle C of A and B which minimizes $L(C)$. (As an exercise, one can verify that if A and B are sorted, then the desired shuffle is the merged sorted sequence of A and B.) The problem can be solved in $O(mn)$ time, using dynamic programming. Can this be improved?

Joe Mitchell: (1) Given a set S of axis-parallel rectangles in the plane, and a set P of points, find a smallest subset T of S with the property that $P \subseteq \bigcup T$, and that T is pairwise disjoint. Given that this problem is hard, one is interested in approximating solutions (and one can assume that there always exists a subset T with the desired property, e.g. each point is enclosed by a small rectangle).

(2) What is the maximum complexity of the zone of a line in an arrangement of n rays that have only h endpoints? One can show that the complexity in question is $\Omega(n + h\alpha(h))$, and a trivial upper bound is $O(n\alpha(n))$; a less trivial bound is $O(n + h^2)$.

(3) Given n segments in the plane with h endpoints, how many segments can appear on the boundary of the unbounded cell of their arrangement? Mitchell et al. showed that the answer is $O(h)$, and the question is to calibrate the constant of proportionality. They have shown an upper bound of $16h - 2$ and a lower bound of $2h$, which can probably be improved to $2.5h$.

Pankaj Agarwal (communicated by Micha Sharir): Given n lines in the plane, preprocess them into a data structure, so that, given any query abscissa a, we can compute the number of line intersections that lie to the left of $x = a$. Find any reasonable trade-off between storage of the data structure and the query time. Allowing quadratic storage, a query can be trivially answered in $O(\log n)$ time (store all intersections in a sorted array). If only subquadratic storage is given, can queries be answered in sublinear time? The only other known approach requires linear storage and takes $O(n \log n)$ time per query, by sorting the line intercepts along $x = a$ and counting inversions between this permutation and the similar permutation at $x = -\infty$.

Micha Sharir: Given n surfaces $F_1 = 0, \ldots, F_n = 0$ in 4-space (the F's are algebraic functions of low bounded degree), and m points $p_1, \ldots, p_m$, determine whether there exist a pair i, j such that $F_i(p_j) > 0$.

This can be solved in time roughly $O(n^{5/6}m^{5/6})$, and the goal is to improve it to close to $O(n^{4/5}m^{4/5})$. The technique is based on triangulating the arrangement of a random sample of r of the given surfaces which can be done with roughly $O(r^5)$ cells, as shown by Chazelle et al., and the challenge is to find such a triangulation with only close to $O(r^4)$ cells. However, any solution of the stated problem, regardless of the technique used, is desired. The problem arises, e.g. in computing the closest pair of lines in a collection of n lines in space and other problems involving lines in space.

Participants Seminar 9141:

Helmut **Alt**
Freie Universität Berlin
Institut für Informatik
Arnimallee 2-6
W-1000 Berlin 33, Germany
email: alt@math.fu-berlin.de

Boris **Aronov**
Polytechnic University
Computer Science Dept.
333 Jay Street
Brooklyn, NY 11201, U.S.A.
email: aronov@ziggy.poly.edu

Franz **Aurenhammer**
Freie Universität Berlin
Institut für Informatik
Arnimallee 2-6
W-1000 Berlin 33, Germany
email: aurenham@tcs.fu-berlin.de

Chanderjit **Bajaj**
Purdue University
Computer Science Department
West Lafayette, IN 47907, U.S.A.
email: bajaj@cs.purdue.edu

Mark **de Berg**
Rijksuniversiteit Utrecht
Vakgroep informatica
Padualaan 14, Postbus 80.089
NL-3508 TB Utrecht, The Netherlands
email: markdb@cs.ruu.nl

Jean-Daniel **Boissonnat**
INRIA
Sophia-Antipolis
2004 Route des Lucioles
F-06565 Valbonne CEDEX, France
email: boissonn@mirsa.inria.fr

Bernard **Chazelle**
Princeton University
Department of Computer Science
35 Olden Street
Princeton, NJ 08544-2087, U.S.A.

Herbert **Edelsbrunner**
University of Illinois
Department of Computer Science
1304 W. Springfield Avenue
Urbana, IL 61801, U.S.A
email: edels@cs.uiuc.edu

Steven **Fortune**
AT&T Bell Labs
600 Mountain Avenue
Murray Hill, NJ 07974, U.S.A.
email: fortune@research.att.com

Dan **Halperin**
Tel Aviv University
Department of Computer Science
Tel Aviv 69978, Israel
email: danha@taurus.bitnet

Marc **van Kreveld**
Rijksuniversiteit Utrecht
Vakgroep informatica
Padualaan 14, Postbus 80.089
NL-3508 TB Utrecht, The Netherlands
email: marc@cs.ruu.nl

Kurt **Mehlhorn**
Max-Planck-Institut für Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11, Germany
email: melhorn@cs.uni-sb.de

Stefan **Meiser**
Max-Planck-Institut für Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11, Germany
email: meiser@cs.uni-sb.de

Joseph S.B. **Mitchell**
SUNY Stony Brook
Applied Mathematics
Stony Brook, NY 11794, U.S.A.
email: jsbm@ams.sunysb.edu

Heinrich **Müller**
Albert-Ludwigs-Universität
Institut für Informatik
Rheinstr. 10-12
W-7800 Freiburg i. Breisgau, Germany
email: mueller@informatik.uni-freiburg.de

Stefan **Näher**
Max-Planck-Institut für Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11, Germany

Jürg **Nievergelt**
ETH-Zentrum
Institut für Informatik
CH-8092 Zürich, Switzerland
email: jn@inf.ethz.ch

Mark **Overmars**
Rijksuniversiteit Utrecht
Vakgroep informatica
Padualaan 14, Postbus 80.089
NL-3508 TB Utrecht, The Netherlands
email: markov@cs.ruu.nl

János **Pach**
Mathematical Institute of the
Hungarian Academy of Sciences
Pf. 127
H-1364 Budapest, Hungary
email: h1158pac@ella.hu
or: pach@cims6.nyu.edu

Richard **Pollack**
Courant Institute of Mathematic Sciences
251 Mercer Street
New York, NY 10012, U.S.A.
email: pollack@cims.nyu.edu

Marie-Francoise **Roy**
Institute Mathématique de Rennes
Université de Rennes 1
Campus de Beaulieu
F-35042 Rennes cedex, France
email: costeroy@emmy.cicb.fr

Jörg-Rüdiger **Sack**
Carleton University
School of Computer Science
Herzberg Building
Ottawa K1S 5B6, Canada
email: sack@scs.carleton.ca

Stefan **Schirra**
Max-Planck-Institut für Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11, Germany
email: stschirr@cs.uni-sb.de

Otfried **Schwarzkopf**
Freie Universität Berlin
Fachbereich Mathematik
Institut für Informatik
Arnimallee 2-6
W-1000 Berlin 33, Germany
email: otfried@tcs.fu-berlin.de

Raimund **Seidel**
Univ. of California at Berkeley
Dept. of Electrical Engineering and
Computer Science, 571 Evans
Berkeley, California 94720, U.S.A.
email: seidel@cs.berkeley.edu

Micha **Sharir**
School of Mathematics
Tel Aviv University
Tel Aviv 69978, Israel
email: sharir@math.tau.ac.il

Michiel **Smid**
Max-Planck-Institut für Informatik
Im Stadtwald 15
W-6600 Saarbrücken 11, Germany
email: michiel@cs.uni-sb.de

Jack **Snoeyink**
University of British Columbia
Department of Computer Science
6356 Agricultural Road
Vancouver, BC V6T 1Z2, Canada
email: snoeyink@cs.ubc.cd

Hubert **Wagener**
TU Berlin
Fachbereich Informatik
Franklinstr. 28-29
W-1000 Berlin 10, Germany
email: wagener@cs.tu-berlin.de

Emo **Welzl**
Freie Universität Berlin
Fachbereich Mathematik
Institut für Informatik
Arnimallee 2-6
W-1000 Berlin 33, Germany
email: emo@math.fu-berlin.de

Peter **Widmayer**
Albert-Ludwigs-Universität
Institut für Informatik
Rheinstr. 10-12
W-7800 Freiburg i. Breisgau, Germany
email: widmayer@informatik.uni-freiburg.de

Bisher erschienene und geplante Titel:

W. Gentzsch, W.J. Paul (editors):
Architecture and Performance, Dagstuhl-Seminar-Report; 1, 18.-20.6.1990; (9025)

K. Harbusch, W. Wahlster (editors):
Tree Adjoining Grammars, 1st. International Worshop on TAGs: Formal Theory and Application, Dagstuhl-Seminar-Report; 2, 15.-17.8.1990 (9033)

Ch. Hankin, R. Wilhelm (editors):
Functional Languages: Optimization for Parallelism, Dagstuhl-Seminar-Report; 3, 3.-7.9.1990 (9036)

H. Alt, E. Welzl (editors):
Algorithmic Geometry, Dagstuhl-Seminar-Report; 4, 8.-12.10.1990 (9041)

J. Berstel , J.E. Pin, W. Thomas (editors):
Automata Theory and Applications in Logic and Complexity, Dagstuhl-Seminar-Report; 5, 14.-18.1.1991 (9103)

B. Becker, Ch. Meinel (editors):
Entwerfen, Prüfen, Testen, Dagstuhl-Seminar-Report; 6, 18.-22.2.1991 (9108)

J. P. Finance, S. Jähnichen, J. Loeckx, M. Wirsing (editors):
Logical Theory for Program Construction, Dagstuhl-Seminar-Report; 7, 25.2.-1.3.1991 (9109)

E. W. Mayr, F. Meyer auf der Heide (editors):
Parallel and Distributed Algorithms, Dagstuhl-Seminar-Report; 8, 4.-8.3.1991 (9110)

M. Broy, P. Deussen, E.-R. Olderog, W.P. de Roever (editors):
Concurrent Systems: Semantics, Specification, and Synthesis, Dagstuhl-Seminar-Report; 9, 11.-15.3.1991 (9111)

K. Apt, K. Indermark, M. Rodriguez-Artalejo (editors):
Integration of Functional and Logic Programming, Dagstuhl-Seminar-Report; 10, 18.-22.3.1991 (9112)

E. Novak, J. Traub, H. Wozniakowski (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 11, 15-19.4.1991 (9116)

B. Nebel, C. Peltason, K. v. Luck (editors):
Terminological Logics, Dagstuhl-Seminar-Report; 12, 6.5.-18.5.1991 (9119)

R. Giegerich, S. Graham (editors):
Code Generation - Concepts, Tools, Techniques, Dagstuhl-Seminar-Report; 13, 20.-24.5.1991 (9121)

M. Karpinski, M. Luby, U. Vazirani (editors):
Randomized Algorithms, Dagstuhl-Seminar-Report; 14, 10.-14.6.1991 (9124)

J. Ch. Freytag, D. Maier, G. Vossen (editors):
Query Processing in Object-Oriented, Complex-Object and Nested Relation Databases, Dagstuhl-Seminar-Report; 15, 17.-21.6.1991 (9125)

M. Droste, Y. Gurevich (editors):
Semantics of Programming Languages and Model Theory, Dagstuhl-Seminar-Report; 16, 24.-28.6.1991 (9126)

G. Farin, H. Hagen, H. Noltemeier (editors):
Geometric Modelling, Dagstuhl-Seminar-Report; 17, 1.-5.7.1991 (9127)

A. Karshmer, J. Nehmer (editors):
Operating Systems of the 1990s, Dagstuhl-Seminar-Report; 18, 8.-12.7.1991 (9128)

H. Hagen, H. Müller, G.M. Nielson (editors):
Scientific Visualization, Dagstuhl-Seminar-Report; 19, 26.8.-30.8.91 (9135)

T. Lengauer, R. Möhring, B. Preas (editors):
Theory and Practice of Physical Design of VLSI Systems, Dagstuhl-Seminar-Report; 20, 2.9.-6.9.91 (9136)

F. Bancilhon, P. Lockemann, D. Tsichritzis (editors):
Directions of Future Database Research, Dagstuhl-Seminar-Report; 21, 9.9.-13.9.91 (9137)

H. Alt , B. Chazelle, E. Welzl (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 22, 07.10.-11.10.91 (9141)

F.J. Brandenburg , J. Berstel, D. Wotschke (editors):
Trends and Applications in Formal Language Theory, Dagstuhl-Seminar-Report; 23,14.10.-18.10.91 (9142)

H. Comon , H. Ganzinger, C. Kirchner, H. Kirchner, J.-L. Lassez , G. Smolka (editors):
Theorem Proving and Logic Programming with Constraints, Dagstuhl-Seminar-Report; 24, 21.10.-25.10.91 (9143)

H. Noltemeier, T. Ottmann, D. Wood (editors):
Data Structures, Dagstuhl-Seminar-Report; 25, 4.11.-8.11.91 (9145)

A. Borodin, A. Dress, M. Karpinski (editors):
Efficient Interpolation Algorithms, Dagstuhl-Seminar-Report; 26, 2.-6.12.91 (9149)

B. Buchberger, J. Davenport, F. Schwarz (editors):
Algorithms of Computeralgebra, Dagstuhl-Seminar-Report; 27, 16..-20.12.91 (9151)