9707—Computational Geometry

09.02. - 14.02.1997

organized by

Rolf Klein (Hagen), Raimund Seidel (Saarbrücken),
Seth Teller (MIT)

The fifth Dagstuhl seminar on computational geometry was attended by 39 participants from 11 countries.
The aim of this workshop was two-fold: Firstly, to provide a forum for discussions about the interplay between the theory and practice of geometric computing. In particular it gave a chance to hear about (and maybe influence) the nascent efforts of various groups to create libraries of geometric algorithms. Secondly, the workshop provided an opportunity to hear and present some of the latest developments in computer graphics, geometric algorithms, combinatorial geometry, and other related topics.
This report contains the abstracts of all the 29 talks, in the order as they were given at the meeting, as well as abstracts of the problems presented at the open problem session.

Compiled by Rolf Klein (open problem session based on notes by Ricky Pollack).

Contents

1 Computing Roadmaps of Semi-algebraic Sets on a Variety

Ricky Pollack (joint work with S. Basu and M.-F. Roy)

Given a semi-algebraic set S defined by s polynomials of degree d in k variables contained in an algebraic variety V of dimension k' defined as the zero set of a polynomial of degree d and given two points of S defined by polynomials of degree t, we present an algorithm which decides whether or not the two points of S lie in the same semi-algebraically connected component of S and, if they do, computes a semi-algebraic path in S connecting the two points. The complexity of the algorithm is $(s^{k'+1} + k'st^{O(1)})d^{O(k^2)}$.

2 The Shape of a Set of Points

Nina Amenta

Reconstruction of the boundary of an object from an unordered set of points is a fundamental problem in both computer vision and computer graphics, so it would be nice to have a mathematical definition of the "shape" of a set of points which reflects the idea that the points lie on a $d-1$-dimensional surface. We propose a simple definition for points in the plane, called "the crust".

We can prove that if a smooth curve is sampled densely enough, then the crust of the samples approximates the curve, with no extraneous features. The minimum required sampling density varies along the curve according to the "Local Feature Size" (which is also simply defined), so that areas of less detail can be sampled less densely.

The crust is easy to compute using Voronoi diagrams in $O(n \log n)$ time (even in practice). We will demonstrate the computation of crusts on arbitrary point sets.

3 Realistic Input Models for Geometric Algorithms

Mark de Berg

Many algorithms developed in computational geometry are needlessly complicated and slow because they have to be prepared for very complicated, hypothetical inputs. To avoid this, realistic models are needed that describe the properties that realistic inputs have, so that algorithms can de designed that take advantage of these properties. This can lead to algorithms that are provably efficient in realistic situations.
We obtain some fundamental results in this research direction. In particular, we have the following results.

- We show the relations between various models that have been proposed in the literature.
- For several of these models, we give algorithms to compute the model parameter(s) for a given scene; these algorithms can be used to verify whether a model is appropriate for typical scenes in some application area.
- As a case study, we give some experimental results on the appropriateness of some of the models for one particular type of scenes often encountered in GIS, namely certain triangulated irregular networks.

4 Data Structures for Mobile Data

Leo Guibas

Suppose we are simulating a collection of continuously moving bodies, rigid or deformable, whose instantaneous motion follows known laws. As the simulation proceeds, we are interested in maintaining certain quantities of interest (for example, the separation of the closest pair of objects), or detecting certain discrete events (for example, collisions – which may alter the motion laws of the objects). In this talk we will present a general framework for addressing such problems and tools for designing and analyzing relevant algorithms, which we call kinetic data structures. The resulting techniques satisfy three desirable properties: (1) they exploit the continuity of the motion of the objects to gain efficiency, (2) the number of events processed by the algorithms is close to the minimum necessary in the worst case, and (3) any object may change its 'flight plan' at any moment with a low cost update to the simulation data structures.

5 Recent Progress on Computing the Minimum Weight Triangulation

Scot Drysdale

Computing the minimum weight triangulation (MWT) is one of the 12 problems that Garey and Johnson listed as "open" (not known to be polynomial or NP-hard) in their 1979 book. Before a couple of years ago no algorithms were known that could compute the MWT for more than 30 or 40 points in a reasonable amount of time. This talk describes recent progress in this area, including describing beta-skeletons and light edges. It concentrates on the LMT-skeleton, which is a subgraph of any locally minimal triangulation. This was discovered by Dickerson and Montague and independently by Keil. In practice this skeleton is almost always a connected graph, which allows the MWT to be completed in $O(n^3)$ time using a dynamic programming algorithm. In practice the LMT skeletion of uniformly distributed points usually is completely triangulated except for small polygons (a half dozen or a dozen points) which must be triangulated.

The talk describes how to apply techniques that were developed for algorithms to compute greedy triangulations to computing a modified version of the LMT skeleton in near-linear time. This in practice has allowed Jack Snoeyink to compute the exact MWT for tens of thousands of points in under an hour.

6 Non-Canonical Randomized Incremental Construction and General Position

OTFRIED SCHWARZKOPF (joint work with Jiří Matoušek and Jack Snoeyink)

Randomized incremental construction has become very popular in computational geometry in the last years, and standard tools are available to design and analyse such algorithms. However, these standard techniques assume a *canonicality condition*, namely that the structure maintained during the algorithm is independent of the order in which the objects are added. In the presence of degeneracies, however, canonicality can often only be enforced by using a "symbolic perturbation scheme," which makes the implementation unnecessarily complicated. We study this phenomenon on the example of Delaunay triangulations, and mention some other interesting problems.

7 Isomorphic Triangulations and Map Conflation

MARSHALL BERN (joint work with Amit Sahai)

In the isomorphic (sometimes called "compatible") triangulation problem, you are given two polygons P and Q and a correspondence between substrings of P's and Q's vertices. The task is to triangulate P and Q with topologically identical triangulations, that is, in the final pair of triangulations vertices can be numbered such that ijk is a triangle in P if and only $i'j'k'$ is a triangle in Q.

Aronov, Seidel, and Souvaine gave an $O(n^2)$ algorithm for this problem, along with a matching lower bound. Kranakis and Urrutia and Gupta and Wenger have given improvements. In the current work, we add the goal of well-shaped triangles. This goal is important for possible applications such as meshing changing domains, map conflation, and morphing. In "map conflation" P and Q represent different maps of the same region.

We show that it is always possible to give isomorphic triangulations without large angles, but it is not always possible to give triangulations without small angles. The achievable angle depends on the minimum k for which there exists a k-quasiconformal mapping between P and Q, taking corresponding vertices to corresponding vertices.

8 Nearest Neighbor Searching in Metric Spaces

Kenneth L. Clarkson

Given a set S of sites (points) and a distance measure, the nearest neighbor searching problem is: build a data structure so that given a query point q, the site nearest to q can be found quickly. Algorithms are given for this problem when the sites and queries are in a metric space. Some of the algorithms employ an additional set Q, given as input for building the data structure. The resulting data structure is not guaranteed to return the nearest site; however, when the metric space satisfies a sphere packing condition, and when $\{q\}$, Q, and S are random subsets of $\{q\} \cup Q \cup S$, then expected bounds can be placed on the preprocessing time, the space needed, the query time, and the failure probability. These bounds depend on the size of S and Q, and on the distance ratio of S, the ratio of the distance between the farthest pair of sites to the distance between the closest pair of sites.

9 Extending GeomBib

Rolf Klein (joint work with Anne Brüggemann-Klein)

GeomBib is a database on literature in computational geometry that is run by a community effort. Currently, it contains about 8700 references in BibTeX format.
We propose enhancing GeomBib by adding annotated relations among the references stored. Such relations include *A cites B*, *A improves on B*, *A uses technique T of B*. This would enable e. g. queries for all survey papers in a given area, or for all papers dealing with the same topics a given paper does. By visualizing such relationships as a graph it would become possible to capture the development of a whole area.
It should be possible for the user to have personal entries/annotations, or to contribute to the public version of the system. This way, the knowledge of individual geometers could be aggregated. Local and global databases must be integrated; both must be maintained under periodic updates.
More information on this BibRelEx project can be found at
`http://wwwpi6.fernuni-hagen.de/wwwpi6/Forschung/BibRelEx/`

10 A Perturbation Scheme for Spherical Arrangements with Application to Molecular Modeling

Dan Halperin (joint work with Christian Shelton)

We discuss robustness problems that we have encountered while implementing a software package for computing and manipulating the subdivision of a sphere by a collection of (not necessarily great) circles and for computing the boundary surface of the union of spheres. We present a perturbation scheme to overcome these problems while using floating point arithmetic.

The scheme is relatively simple, it balances between the time of the computation and the magnitude of the perturbation, and it performs well in practice. Our package is a major component in a larger package aimed to support geometric queries on molecular models; the spherical subdivisions are used to construct a geometric model of a molecule where each sphere represents an atom. We report and discuss experimental results.

11 Progressive Simplicial Complexes

Hugues Hoppe

We introduce the progressive simplicial complex (PSC) representation, a new format for storing and transmitting triangulated geometric models. Like the earlier progressive mesh (PM) representation, it captures a given model as a coarse base model together with a sequence of refinement transformations that progressively recover detail. The PSC representation makes use of a more general refinement transformation, allowing the given model to be an arbitrary triangulation (e.g. any dimension, non-orientable, non-manifold, non-regular), and the base model to always consist of a single vertex. Indeed, the sequence of refinement transformations encode both the geometry and the topology of the model in a unified multiresolution framework. The PSC representation retains the advantages of PM's. It defines a continuous sequence of approximating models for runtime level-of-detail control, allows smooth transitions between any pair of models in the sequence, supports progressive transmission, and offers a space-efficient representation. Moreover, by allowing changes to topology, the PSC sequence of approximations achieves better fidelity than the corresponding PM sequence.

12 Some Practical Models and Challenges for Computational Geometry

Seth Teller

The theory/practice gap is bridged best when practitioners are aware of powerful techniques developed in the theory community, and when theorists tackle practical barriers to the adoption of such techniques. This talk is intended to address this gap, and spur continued collaboration, in the specific case of algorithms for rendering and global illumination in computer graphics. First presented are several observations about geometric queries as they occur in existing, implemented graphics systems. Among these are that queries are conservative (can admit one-sided error); are spatially and temporally coherent; are often invoked in the innermost loops of numerical algorithms; and are often applied to extremely large data sets. Moreover, exact solutions (such as fully-elaborated arrangements of curved surfaces in high dimension) are difficult to implement robustly. Recommendations for the development of cost models and algorithms are made, corresponding to each of the observations. Finally, a set of specific algorithmic and implementation challenges is posed, including "atomic" and "non-atomic" polygon rendering, meta-data structures which can upper bound their own query time; and the time and space analysis of lazily-elaborated hybrid arrangements of lines and other objects.

13 Checking Geometric Programs or Verification of Geometric Structures

KURT MEHLHORN (joint work with Stefan Näher, Michael Seel, Raimund Seidel, Thomas Schilz, Stefan Schirra, and Christian Uhrig)

A program checker verifies that a particular program execution is correct. We give simple and efficient program checkers for some basic geometric tasks, e.g., convex hulls in arbitrary dimensions, Voronoi diagrams and Delaunay triangulations in two-dimensional space. We report about our experiences with program checking in the context of the LEDA system. We discuss program checking for data structures that have to rely on user-provided functions.

14 CGAL - A C++ Template Library

ANDREAS FABRI

Templates are a relatively new feature of the C++ programming language that allows to write generic code by introducing *parameterized types*. A typical example for a parameterized type is the statement `list<T> L;`, which declares a list of objects of type `T`, where `T` is an arbitrary data type fixed at compile time. Most of the code for the management of the list does not depend on `T`. Template code is generic but type safe, compared to the *C* solution to cast to and away from `void*`. The language provides escape mechanisms from the code generation, that allow to replace parts of the generated code by hand crafted code that is more stable or more efficient (e.g., a `vector<bool>`). The usage of templates is only costly at compile time, and there is no space or time overhead at run time.
Templates are heavily used in the Standard Template Library (STL), which

is a container library that provides basic data structures such as list, set, and map, and algorithms such as sorting. Implementations of STL are shipped with all major C++ compilers (`www.sgi.com/Technology/STL/`), as the library is part of the upcoming language standard (`www.maths.warwick.ac.uk/c++/pub/wp/html/cd2/`). STL is a thoroughly designed conceptual framework, it is coherent and extensible, C++ users will adopt the underlying concepts, and the library is free. These practical reasons led to the decision to adopt and to extend this framework when we worked on the design of the Computational Geometry Algorithm Library (`www.ruu.nl/CGAL`). Data types in the CGAL kernel are parameterized with a type that determines a representation, e.g., Cartesian or homogeneous coordinates. This representation is parameterized with a type determining the type of the coordinates, e.g., built-in `double`, or `real` from the LEDA library (`www.mpi-sb.mpg.dw/LEDA/`). Data structures such as triangulations and planar maps are parameterized with a *traits class*, that encapsulates type information, e.g, the type of the point associated to the vertices of the data structure, and geometric predicates on objects of these types, e.g., orientation tests.
The CGAL library shall be more than a collection of geometric algorithms, but a collection of reusable software components. The mechanisms described in this talk allow to adapt data structures from the CGAL library to a user application, instead of forcing the user to change the application in order to base it on the CGAL library.

15 A Strong and Easily Computable Separation Bound for Arithmetic Expressions Involving Radicals and Geometric Applications

Stefan Schirra

We consider arithmetic expressions over operators $+$, $-$, $*$, $/$, and $\sqrt{\ }$, with integer operands. For an expression E, a separation bound $sep(E)$ is a positive real number with the property that $E \neq 0$ implies $|E| \geq sep(E)$. We propose separation bounds that are easy to compute automatically and stronger than previous bounds. In particular we show that $1/(u(E))^{k-1}$ is a separation bound for a division-free expression E, where k is the product of the indices of the radicals in E and $u(E)$ is the value of the expression obtained from E by replacing all $-$ by $+$ and all integers by their absolute value.
Furthermore we show how the new separation bounds are used to provide exact comparison of numbers in the LEDA number type `real`.
Finally we discuss the application of the separation bound and the number type real to the computation of Voronoi diagrams of line segments and the minimum diameter of a set of moving points.

16 A Simple Polynomial-Type Approximation Scheme for Geometric Network Optimization

Joe Mitchell

We present a simple polynomial-time approximation scheme for geometric instances of the k-MST problem, the traveling salesman problem (TSP), and

other network optimization problems. Our method employs a very simple modification to our SODA'96 paper, which in turn was based on a simplification of a method introduced by Blum, Chalasani, and Vempala (STOC'95). In particular, the method is based on the concept of an "m-guillotine subdivision". Roughly speaking, an "m-guillotine subdivision" is a polygonal subdivision with the property that there exists a line ("cut"), whose intersection with the subdivision edges consists of a small number ($O(m)$) of connected components, and the subdivisions on either side of the line are also m-guillotine. The upper bound on the number of connected components allows one to apply dynamic programming to optimize over m-guillotine subdivisions, as there is a succinct specification of how the subdivision interacts with the cuts that make up the boundary of a rectangle that specifies a subproblem of the dynamic program.

Key to our method is a theorem showing that any polygonal subdivision can be converted into an m-guillotine subdivision by adding a set of edges whose total length is small: at most that of the original subdivision, times $\frac{1}{m}$. This allows us to apply dynamic programming to optimize over the class of m-guillotine subdivisions having some prescribed properties, since the relevant subproblems are specified by only a constant ($O(m)$) number of parameters. This yields a $(1 + \frac{1}{m})$-approximation for various network optimization problems (k-MST, TSP, etc.).

S. Arora has developed an alternative method to achieve similar results – a PTAS for Euclidean TSP and related problems. His remarkable discovery predates the release of this paper by several weeks. Prior to Arora's breakthrough, the best approximation factor for the Euclidean TSP was the Christofides heuristic, which gave a factor of 1.5.

(paper available at http://ams.sunysb.edu/ jsbm/jsbm.html, and has been accepted to SICOMP, pending final revisions)

17 A Probabilistic Analysis of the Power of Arithmetic Filters

Olivier Devillers (joint work with Franco P. Preparata)

The assumption of real-number arithmetic, which is at the basis of conventional geometric algorithms, has been seriously challenged in recent years, since digital computers do not exhibit such capability. A geometric predicate usually consists of evaluating the sign of some algebraic expression. In most cases, rounded computations yield a reliable result, but sometimes rounded arithmetic introduces errors which may invalidate the algorithms. The rounded arithmetic may produce an incorrect result only if the exact absolute value of the algebraic expression is smaller than some (small) ε, which represents the largest error that may arise in the evaluation of the expression. The threshold ε depends on the structure of the expression and on the adopted computer arithmetic, assuming that the input operands are error-free. A pair (arithmetic engine,threshold) is an *arithmetic filter*. In this paper we develop a general technique for assessing the efficacy of an arithmetic filter. The analysis consists of evaluating both the threshold and the probability of failure of the filter. To exemplify the approach, under the assumption that the input points be chosen randomly in a unit ball or unit cube with uniform density, we analyze the two important predicates "which-side" and 'insphere". We show that the probability that the absolute values of the corresponding determinants be no larger than some positive value V, with emphasis on small V, is $\Theta(V)$ for the which-side predicate, while for the insphere predicate it is $\Theta(V^{\frac{2}{3}})$ in dimension 1, $O(V^{\frac{1}{2}})$ in dimension 2, and $O(V^{\frac{1}{2}} \ln \frac{1}{V})$ in higher dimensions. Constants are small, and are given in the paper.
http://www.inria.fr/prisme/biblio/search.html INRIA Research Report, n.2971 (search 2971)

18 The Existence of a Short Sequence of Admissible Pivots to an Optimal Basis in LP and LCP

Komei Fukuda (joint work with Hans-Jakob Lüthi and Makoto Namiki)

We say an LP is fully nondegenerate if both the primal and the dual problems are nondegenerate. In this paper, we prove the existence of a sequence of $|B^* \setminus B|$ admissible pivots from any basis B (not necessarily feasible) to the unique optimal basis B^*, if the given LP has an optimal solution and is fully nondegenerate. Here admissible pivots are those pivots (satisfying certain sign conditions) that exist if the current LP dictionary is not terminal, i.e., neither optimal, inconsistent nor dual inconsistent. A natural extension of the result to LCP's with sufficient matrices is given. The existence itself does not yield a strongly polynomial pivot algorithm for LP's but provides us with a good motivation to study the class of admissible pivot methods for LP's, as opposed to the narrower class of simplex methods for which the shortest sequence of pivots is not known to be polynomially bounded.

19 Direction-Sensitive Voronoi-Diagrams

Franz Aurenhammer

On a tilted plane T in three-space, *skew distances* are defined as the Euclidean distance plus a multiple of the signed difference in height. Skew distances may model realistic environments more closely than the Euclidean distance. Voronoi diagrams and related problems under this kind of distances are investigated. A relationship to convex distance functions and to Euclidean Voronoi diagrams for planar circles is shown, and is exploited for a geometric analysis and a plane-sweep construction of Voronoi diagrams on T.

An output-sensitive algorithm running in time $O(n \log h)$ is developed, where n and h are the number of sites and non-empty Voronoi regions, respectively. The all nearest neighbors problem for skew distances, which has certain features different from its Euclidean counterpart, is solved in $O(n \log n)$ time.

20 On Some Geometric Optimization Problems in Layered Manufacturing

Michiel Smid (joint work with Jayanth Majhi, Ravi Janardan, and Prosenjit Gupta)

Efficient geometric algorithms are given for optimization problems arising in layered manufacturing, where a 3D object is built by slicing its CAD model into layers and manufacturing the layers successively. The problems considered include minimizing the degree of stair-stepping on the surfaces of the manufactured object, minimizing the volume of the so-called support structures used, and minimizing the contact area between the supports and the manufactured object—all of which are factors that affect the speed and accuracy of the process. The stair-step minimization algorithm is valid for any polyhedron, while the support minimization algorithms are applicable to convex polyhedra only. Algorithms are also given for optimizing supports for non-convex, simple polygons. The techniques used to obtain these results include construction and searching of certain arrangements on the sphere, 3D convex hulls, halfplane range searching, ray-shooting, visibility, and constrained optimization.

21 Representing distances by coordinates

Jiří Matoušek

Let M be an n-point metric space with metric ρ, and let X be a d-dimensional normed space with norm $\|.\|$. A mapping $f : M \to X$ is called a D-embedding, $D \geq 1$ a real number, if $\frac{1}{D}\|f(x)-f(x)\| \leq \rho(x,y) \leq \|f(x)-f(y)\|$ holds for any two points $x, y \in M$. We survey some results known about questions of the following type: "What is the minimum D such that all n-point metric spaces can be D-embedded into a given X?" and "What is the minimum dimension d such that all n-point metric spaces admit a D-embedding into a d-dimensional space X, where $D \geq 1$ is a given number?". The known results are mostly pessimistic in the worst case, showing the existence of metric spaces that cannot be embedded very well. The survey is mainly based on work of Bourgain, Lindenstaruss, Johnson, Milman, Schechtman, Enflo, Linial, London, Rabinovich, Arías-de-Reyna, Rodríguez-Piazza, and the speaker.

22 Approximating Weighted Shortest Paths on Polyhedral Surfaces

Jörg-Rüdiger Sack

Shortest path problems are among the fundamental problems studied in computational geometry. In this talk accompanied by a video, we consider the problem of computing a shortest cost path between two points s and t on a (possibly non-convex) polyhedral surface $\mathcal{P}$. The surface is composed of triangular regions (faces) in which each region has an associated positive weight indicating the cost of travel in that region.

The computation of Euclidean shortest paths on non-convex polyhedra has been investigated by a several researchers (for references see [1]); currently, the best known algorithm due to Chen and Han runs in $\mathcal{O}(n^2)$ time. Mitchell and Papadimitriou introduced the Weighted Region Problem and an algorithm that computes a shortest weighted cost path between two points in a planar subdivision; it requires $\mathcal{O}(n^8 \log n)$ time in the worst case. They state that their algorithm applies to non-convex polyhedral terrains with modifications.

Most shortest path applications demand simple and efficient algorithms to compute approximate shortest paths as opposed to a complex algorithm that computes an exact path. Polyhedra arising in these applications approximate real surfaces and thus an approximate path will typically suffice. Our interest is also motivated by our research and development on a parallel system for GIS and spatial modeling.

The talk describes several schemes that allow the computation of approximate shortest paths on polyhedral surfaces in both the weighted and unweighted scenarios (see [1] for details). The schemes are based on adding Steiner points along the polyhedral edges and interconnecting them across each face. The schemes vary in the way in which the Steiner points and edges are added. We represent the Steiner points and edges as a graph in which we compute an approximate path using Dijkstra's graph shortest path algorithm. We show that as more and more Steiner points are added, we obtain increasing path accuracy and with only a few (constant number of) Steiner points per polyhedral edge, the path cost converges to a near-optimal value.

We have implemented these schemes as well as Chen and Han's shortest path algorithm. Each of the schemes is described in the video shown and we also give experimental results that show how well they performed on the test suite of terrain data that we used. In the unweighted case, we compare the path accuracy and algorithm running time with our implementation of Chen and Han's algorithm and we show (with graphs) that our schemes obtain near-optimal results with a running time which is much quicker than that of Chen and Han. Since we do not have an implementation of an algorithm to compute exact weighted shortest path, we are unable to make a comparison of our weighted paths. We do show however, that our weighted paths exhibit similar convergence behavior as in the unweighted case. We also give results showing the effects on the approximated paths when the terrain becomes more “spiky”.

References

[1] M. Lanthier, A. Maheshwari and J.-R. Sack, Approximating Weighted Shortest Paths on Polyhedral Surfaces, Technical report, School of Computer Science, Carleton University, 1996

23 Finding Some Path When You Know At Least One Exists

David Kirkpatrick (joint work with Raimund Seidel)

We address the following open problem mentioned by Marc van Krefeld at the Fourth Dagstuhl Seminar on Computational Geometry in March, 1995: *Given a simple polygon with k disjoint holes and n vertices in total, and also two points s and t in the polygon. Can one compute a path from s to t faster than in $O(n+k\log k)$ time (it need not be the shortest such path)?* We establish an $\Omega(n + k\log k)$ lower bound by showing that the simpler problem of determining an escape path (a path from the origin to the circle at infinity) in the presence of k disjoint obstacles (even elementary obstacles, say horizontal and vertical line segments, or circles) requires $\Omega(k\log k)$ time. Our model is an augmented decision tree, where internal comparisons are arbitrary, and output functions (describing the path as a sequence of points) involve arbitrary analytic functions. Although the bound is the ”expected” one, the question is interesting because disjointness of the holes (obstacles) guarantees that a path exists, but the absence of any optimization criterion means that paths are not constrained to take any particular shape (eg. bending at some restricted subset of points in the plane). The latter makes the formulation of lower bound arguments somewhat more challenging.

24 Raising Roofs, Crashing Cycles, and Playing Pool: Applications of a Data Structure for Finding Pairwise Interactions

Jeff Erickson (joint work with David Eppstein)

We consider the following problems:

- Construct a roof over a building whose floorplan is an arbitrary simple polygon. The *straight skeleton*, introduced by Aurenhammer at the Dagstuhl seminar in 1993, defines a canonical roof for any polygon. The best previously known algorithm for its construction requires time $O(n \log^2 n)$.

- Imagine several motorcycles out in the desert, each specified by an initial position and velocity. Whenever a motorcycle runs over the track left by another motorcycle, it crashes. The pattern of tracks eventually created is called a *motorcycle graph*. Construct the graph.

- Simulate the sequence of collisions between a set of moving billiard balls, each represented by a unit disk in the plane.

These three problems are special cases of the following problem. Given two sets S and T, maintain the pair (s, t) that minimizes some function $f(s, t)$, as elements are inserted and deleted from either set. We describe a simple data strcuture of David Eppstein that maintains the best pair in time $O(n \log^2 n)$ time per update, using only linear space. By exploiting the geometry of the problem, we can recuce the update time even further. For example, we can construct straight skeletons and motorcycle graphs in tiem $O(n^{17/11+\varepsilon})$ and simulate billiards in time $O(n^{2/3+\varepsilon})$ time per collision.

25 How to Move a Mountain

Jack Snoeyink (joint work with Marc van Kreveld and Bernd Jünger (Java applet))

Many of the computational geometers' favorite data structures are planar graphs that are canonically determined by a set of geometric objects and take $\Theta(n \log n)$ time to compute. Using the Delaunay triangulation as an example, we show that, given such a structure, one can determine a permutation of the data in $O(n)$ time such that the diagram can be recomputed from the permuted data in $O(n)$ time by a simple incremental algorithm.
In a Geographic Information Systems (GIS), this allows us to store terrain models based on the Delaunay triangulation in flat files without disrupting other applications, and to read or transmit these models progressively—producing a sequence of better and better approximations as a model is read or received. This was demonstrated by a Java applet that we are developing.

26 Binary Space Partitions: Old and New Results

Pankaj K. Agarwal

Binary space partition (BSP) is a hierarchical decomposition of space, widely used for several problems in computer graphics. Roughly speaking, BSP for a set of objects is a binary tree, each of whose nodes are associated with a convex region. The regions associated with the leaves of the tree form a convex decomposition of the space, and the interior of such a region does not intersect any input object.
The first part of the talk surveys the known results on constructing a BSP of a set of objects in two and three dimensions. The second part of the talk discusses new algorithms for constructing a BSP for a set of orthogonal

rectangles in 3-space, for constructing a BSP of a set of triangles in 3-space, and for maintaining a BSP of a set of moving segments in the plane or of a set of moving triangles in 3-space.

27 Point Set Matching

Remco Veltkamp (joint work with Michiel Hagedoorn)

In applications such as stereo matching, content-based image retrieval, object recognition, and radiotherapy alignment, one of the problems is to determine if there exists a transformation that matches part of one point set A to part of the other point set B. We present a method for finding a set of transformations which map some subset of A arbitrarily close to some subset of B under the Hausdorff distance. The method is based on the volumes traced out by applying a block of transformations. This technique is general in the sense that it works for points of arbitrary dimension, and for various classes of transformations, such as translation, scaling, homothety, rotation, rigid motion, general linear transformation, and affine transformation.

28 A Practial Approximation Algorithm for the LMS Line Estimator

David M. Mount (joint work with N. Netanyahu, K. Romanik, R. Silverman, and A. Y. Wu)

The problem of fitting a straight line to a finite collection of points in the plane is an important problem in statistical estimation. Robust estimators are particularly important because of their lack of sensitivity to outlying data points. The basic measure of the robustness of an estimator is its breakdown point, that is, the fraction (up to 50-percent) of outlying data points that can corrupt the estimator. Rousseeuw's least median-of-squares (LMS) regression (line) estimator is among the best known 50-percent breakdown-point estimators. The best exact algorithms known for this problem run in $O(n^2)$ time, where n is the number of data points. Because of this high running time, many practitioners prefer to use a simple $O(n \log n)$ Monte Carlo algorithm, which is quite efficient but provides no guarantees of accuracy (even probabilistic) unless the data set satisfies certain assumptions. Two algorithms are presented in an attempt to close the gap between theory and practice. The first is a conceptually simple randomized Las Vegas approximation algorithm for LMS, which runs in $O(n \log n)$ time. However, this algorithm relies on somewhat complicated data structures to achieve its efficiency. The second is a practical randomized algorithm for LMS that uses only simple data structures. It can be run as either an exact or an approximation algorithm. This algorithm runs in $O(n^2)$ time in the worst case, but we present empirical evidence that its running time on realistic data sets is much better. This algorithm provides an attractive option for practitioners, combining both the efficiency of a Monte Carlo algorithm and guarantees on the accuracy of the result.

29 On the Complexity of the Union of Fat Convex Objects in the Plane

MICHA SHARIR (joint work with Alon Efrat)

Let $\mathcal{C}$ be a collection of n compact convex sets in the plane, satisfying the following properties:

(i) The objects in $\mathcal{C}$ are α-*fat*, for some fixed $\alpha > 1$; that is, for each $c \in \mathcal{C}$ there exist two disks $D \subseteq c \subseteq D'$ such that the ratio between the radii of D' and D is at most α.

(ii) For any pair of distinct objects $c, c' \in \mathcal{C}$, their boundaries intersect in at most s points, for some fixed constant s.

The *combinatorial complexity* of the union $U = \bigcup \mathcal{C}$ is defined to be the number of intersection points between the boundaries of the sets of $\mathcal{C}$ that lie on ∂U. We show that the combinatorial complexity of U is $O(n^{1+\epsilon})$, for any $\epsilon > 0$, where the constant of proportionality depends on ϵ, α and s.
A key step in the analysis uses a recent result of Pach and Sharir, which analyzes the number R of *regular* vertices on the boundary of the union of n convex sets (no other assumptions are made now) in the plane, where a regular vertex is formed by the intersection of two boundaries which cross exactly twice. Pach and Sharir have shown that

$$R \leq 2I + 6n - 12,$$

where I is the number of irregular vertices on the boundary of the union. The talk has presented the proof in detail.

30 Open Problems

Weavings of fat flexible objects

by Marshall Bern

Intuitively, this problem asks you to count the number of combinatorially distinct arrangements of sheets of paper on a desktop. More formally, assume you are given n (non-axis-aligned) squares in the plane $S_1, S_2, \ldots, S_n$. Over each square S_i there is a continuous real-valued function f_i. Functions are nowhere equal, that is, for all $p \in S_i \cap S_j$, $f_i(p) \neq f_j(p)$, so we can write $f_i < f_j$ if for all $p \in S_i \cap S_j, f_i(p) < f_j(p)$. The question is how many "overlap orders" on the f_i's are possible?
A trivial upper bound is 3^{n^2}, since for each pair of i and j, either $f_i < f_j$, $f_j < f_i$, or the two are incomparable. However, the best lower bound I know is $2^{\Omega(nlogn)}$. This problem seems to be related to the fat-triangle result, as a cycle in the overlap order, $f_i < f_j < ... < f_i$, must surround a hole.
The same problem is open for other fat objects, for example, unit disks instead of unit squares. The problem arose in counting the number of distinct origamis with the same crease pattern.

The number of maxima

by Bernhard Chazelle

Given a finite point set P in the plane and a point q, define $m(P,q)$ to be the number of maxima of $L(q) \cap P$, where $L(q)$ is the south-west quadrant cornered at q; a point is said to be a maximum of S if it is not dominated coordinate-wise by any other point in S. Finally, let $m(P) = \max_q m(P,q)$.
Prove or disprove: Given any P of size n, there exists Q of size $O(n)$ such that $m(P \cap Q) = O(\log n)$.

Polygon unfolding

by Komei Fukuda

An *unfolding* of a convex polytope P in R^3 is a planar embedding of its boundary obtained by cutting the edges of some spanning tree T of the graph of P and flattening the boundary along the remaining edges. Two natural (but naive) questions are

(a) Is every unfolding of a convex polytope non-selfoverlapping?

(b) Is every unfolding of a convex polytope unambiguous?

Here an unfolding is defined to be *unambiguous* if the original polytope is uniquely constructible from it.
Both questions have negative answers. Many constructions are known for the negative answer of (a), but Makoto Namiki (namiki@waka.c.u-tokyo.ac.jp) constructed the smallest example, a skinny tetrahedron, which admits a self-overlapping unfolding, see Figure 1. Note that it has a non-selfoverlapping unfolding as well. For question (b), Tomomi Matsui (tomomi@misojiro.t.u-tokyo.ac.jp) constructed a polytope with 6 facets and 5 vertices which admits an ambiguous unfolding, see Figure 2. Details of these two examples can be found in the UnfoldPolytope package for Mathematica by Namiki and Fukuda [NF92].
Consequently more intelligent questions are

(a') Does every convex polytope admit a non-selfoverlapping unfolding?

(b') Does every convex polytope admit an unambiguous unfolding?

As far as I know, these questions are still open. I conjectured at the meeting that

(1) Any minimum-length spanning tree of a convex polytope induces a non-selfoverlapping unfolding.

The positive answer to this would resolve the question (a') positively as well. Recently Günter Rote has constructed counterexamples to this conjecture, see Figure 3. The smallest among them has 9 vertices and 7 facets [Rot97a]. Rote also constructed a polytope which admits a *combinatorially* ambiguous unfolding [Rot97b]. One can construct two combinatorially different polytopes from such an unfolding, see Figure 4. Matsui's example mentioned above gives rise to two geometrically different polytopes which are combinatorially equivalent.
Note that a question (related to (a')) on the existence of an unfolding without overlaps, where it is allowed to cut any place in the boundary, was answered

positively by Aronov and O'Rouke [AO91]. The key idea was to cut through geodesic paths from a fixed vertex to all other vertices. In fact this result motivates us to pose another open problem.

(1) Does a shortest-path spanning tree of a convex polytope induce a non-selfoverlapping unfolding?

Here a shortest-path spanning tree is a tree composed of shortest paths from a fixed vertex to all other vertices.

Searching a polygon for a mobile intruder

by Leo Guibas

Given a simple polygon, determine the minimum number of searchers required to make sure the polygon does not contain a mobile intruder. In general, $\Theta(\log n)$ many searchers can be necessary and are always sufficient for an n-vertex polygon.

TSP problems

by Joe Mitchell

1. In the *TSP with neighborhoods* problem, we desire a short tour (closed cycle) that visits each of a set of regions (neighborhoods). The problem is clearly NP-hard in general (since TSP for point sites is NP-hard). The question I posed was the case of regions that are given by a set of (infinite) lines. In 2-space, a shortest tour for a set of n lines can be found in polynomial time ($O(n^4)$), based on results for watchman routes in simple polygons (not required to pass through any anchor point). Can this time bound be improved?
Also, what can be said in 3-space for a set of n lines?
2. The *maximum scatter TSP* for a set of point sites asks for a tour that visits all of the sites and maximizes the length of the shortest edge. In contrast, the *bottleneck TSP* problem asks us to minimize the maximum length edge in a tour; this problem is known to be NP-hard for point sites in the plane (the reduction is from Hamilton circuit in grid graphs). However, the complexity of the maximum scatter TSP is open for point sites in the plane. (It is known to be NP-complete for graphs.)

A neighborhood structure between non-crossing simple polygons

by Günter Rote

Let $\mathcal{P}$ be the set of all non-crossing simple polygons with a given set of points as vertices. (It is allowed that a polygon has an angle of 180° at a point.) We call two such polygons *neighbors* if one can be obtained from the other by taking out a point from the cyclic sequence of vertices and inserting it between two other vertices that were previously adjacent in the cyclic order. For example, the polygons $P_1P_2P_3P_4P_5P_6P_7$ and $P_1P_3P_4P_5P_2P_6P_7$ are neighbors, provided that they are both non-self-intersecting.
(a) Is the set $\mathcal{P}$ always connected under this neighborhood structure? (It is not even known that every element of $\mathcal{P}$ has at least one neighbor.)
(b) Can you define another simple neighborhood structure which would make the set $\mathcal{P}$ connected?
NOTE: The problem is due to Jean-François Hêche (Lausanne). Michael Houle (University of Newcastle, Australia) has provided a negative answer to question (a) by exhibiting a simple 19-gon that has no neighbors at all. Regarding question (b), he has come up with a complex proof (not yet written up) that every polygon has at least one neighbor under the *3-flip* operation, where *three* polygon edges can be replaced to obtain a new simple polygon.

Cutting a polygon into congruent polygons

by Günter Rote

Find an algorithm to decide whether a given polygon can be decomposed into three congruent polygons. For the beginning, assume that the given polygon and the three resulting parts are simple polygons, i.e., they are connected and contain no holes.
(For decomposition into two simple polygons, there is a polynomial-time algorithm.)
This problem is due to Kimmo Eriksson (Stockholm).

Counting regular vertices on the boundary of the union of planar sets

by Micha Sharir

We want to analyze the number R of *regular* vertices on the boundary of the union of n convex sets in the plane, where a regular vertex is formed by the intersection of two boundaries which cross exactly twice. We make no other assumptions concerning the given sets, except that they are in general position, meaning that no pair of boundaries are tangent or overlapping, and each pair has a finite (but arbitrary) number of intersections.
Pach and Sharir have shown (the initial proof was obtained during the Dagstuhl seminar) that

$$R \leq 2I + 6n - 12,$$

where I is the number of irregular vertices on the boundary of the union. Moreover, this bound is tight in the worst case.
It is still open (although conjectured to be true) that a similar bound holds for nonconvex sets as well. Joe Mitchell has reported a solution by a student of his, which so far has not been available.
Another open problem is to obtain a subquadratic bound on the number of regular vertices on the boundary of the union, where we also assume that the boundaries of any pair of sets cross in at most some constant number, s, of points. Pach and Sharir give a lower bound construction where the given sets are rectangles and disks, and where the number of regular vertices is $\Omega(n^{4/3})$. A matching upper bound for this special case has been obtained recently by Aronov, Halperin and Sharir. A slightly subquadratic bound for the general case has also been obtained recently by Aronov, Efrat and Sharir.

Storing planar triangulations

by Jack Snoeyink

A planar triangulation of n vertices can be stored in a general winged-edge or quad-edge data structure using $24n$ pointers (4 edge pointers, 2 vertex pointers, and 2 face pointers per segment), or in a triangle-based structure using $12n$ pointers (3 vertex and 3 triangle pointers per triangle). Martin Heller (a GIS researcher in Zurich) observed that edges can be oriented so

that every vertex has outdegree at most 3, and that storing two pointers per edge gives a vertex-based structure with $6n$ edge pointers that supports all standard navigation operations in constant time.
Can the incremental Delaunay triangulation be computed using this structure in $O(n \log n)$ time?
The question is how to efficiently perform diagonal swaps. Let ac be a diagonal from vertex a to vertex c whose adjacent triangles form a quadrilateral $abcd$. Unless ab and ad are also edges out of a, diagonal ac can be swapped for bd, preserving the outdegree-3 property, by adjusting only these four vertices (and pointers to them).
At UBC we have shown that the remaining case can be handled by reversing the edges of a directed cycle through ab or ad and then swapping ac with bd. We use the result that any triangulation can be partitioned into three spanning trees (W. Schnyder, 1st SODA) to find a cycle in time proportional to its length. We do not know, however, how to amortize this additional cost over the algorithm. Other approaches can also be considered.

References

[AO91] B. Aronov and J. O'Rourke. Nonoverlap of the star unfolding. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 105–114, 1991.

[NF92] M. Namiki and K. Fukuda. *UnfoldPolytope: A Mathematica package for unfolding general convex polytopes*, 1992. available from `http://www.ifor.math.ethz.ch/ifor/staff/fukuda/fukuda.html`

[Rot97a] G. Rote. *Counterexample to Fukuda's Conjecture that the unfolding of smallest perimeter is always nonoverlapping*, Mathematica notebook, 1997. figures available from `http://www.ifor.math.ethz.ch/ifor/staff/fukuda/fukuda.html`

[Rot97b] G. Rote. *An unfolding that gives rise to two COMBINATORIALLY different 3-polytopes*, Mathematica notebook, 1997. figures available from `http://www.ifor.math.ethz.ch/ifor/staff/fukuda/fukuda.html`

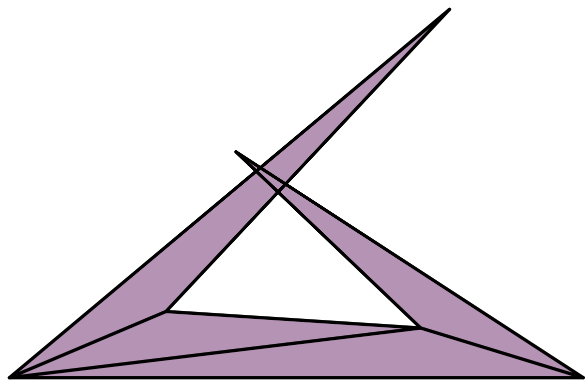

Figure 1: Selfoverlapping unfolding by Namiki

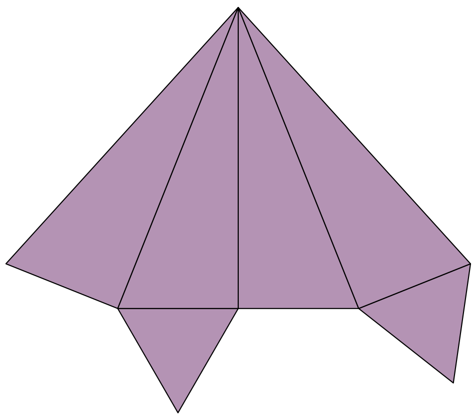

Figure 2: Geometrically ambiguous unfolding by Matsui

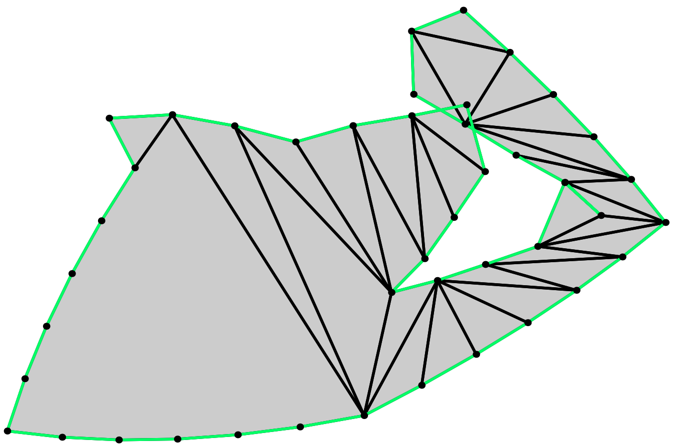

Figure 3: Minimum-perimeter selfoverlapping unfolding by Rote

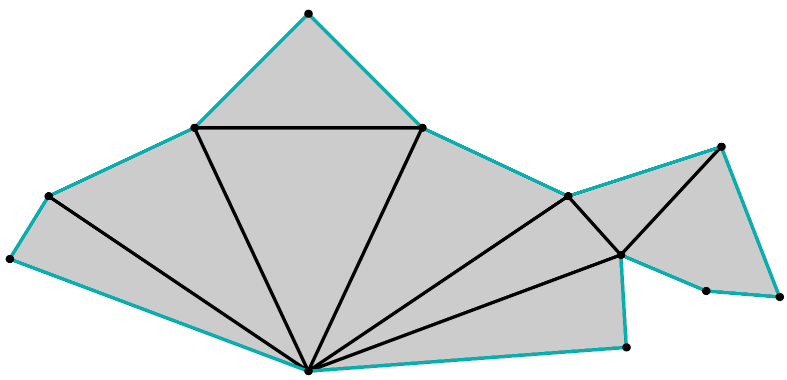

Figure 4: Combinatorially ambiguous unfolding by Rote