Open Problems Presented at the 2009 Dagstuhl Workshop on Computational Geometry

Joe Mitchell (Session Chair/Scribe)

March 10, 2009

On Tuesday evening, March 10, 2009, we held an open problem discussion at the Dagstuhl-Seminar on Computational Geometry. The problems span a range of topics, including fundamental algorithms, discrete geometry, algebra, combinatorics, and optimization.

Problem 1 (Jack Snoeyink). *Limited visibility polygons for mobile point agents:* Consider a scenario in which the behavior of each moving agent is affected by the behaviors of nearby agents that they see. Each agent would like to have, not an output-sensitive algorithm for visibility, but an output-bounded algorithm that will compute the visibility polygon until it becomes too cluttered.

Let P be a *floorplan*, which is a polygon with holes made up of n interior-disjoint line segments. Let $A = \{a_1, \ldots a_m\}$ be a set of mobile agents, represented by points in P. We say that the *description complexity of any region* R is the number of line and circle segments needed to represent $R \cap P$, plus the number of agents in $R \cap A$.

Denote the visibility polygon from $p \in P$ by V_p, the ball of radius r by $B_p(r)$. I'll drop most of the p subscripts since all visibility in this paragraph will be from the chosen point or agent p. For a bound K on description complexity, define the *limit radius*

$$r_K = \arg\min_{r \geq 0} \text{description complexity} \left(V_p \cap B(r)\right) \geq K.$$

I'd like an algorithm that computes a region R of description complexity $O(K)$ such that $B(r_K/2) \cap V_p = R \cap V_p$ preferably in $O^*(K)$, where the asterisk denotes that there may be a multiplicative polylog(n)-factor, after preprocessing the floorplan, and possibly the agent positions, into an $O^*(n)$-space data structure.

This problem was suggested by Glenn Elliot at UNC Chapel Hill, who points out that many current agent simulations depend on proximity and do not take visibility into account – thus, in many demos you can see agents interacting across walls.

Problem 2 (Erik Demaine). *Algorithmic 3D Dissection:* Find a finite algorithm to dissect one polyhedron into another with the same volume and Dehn invariant. A *dissection* is a carving of the first polyhedron into finitely many polyhedral shapes and a rigid motion of these pieces so that they union to the second polyhedron. Dehn proved in 1900 that having these two conditions are necessary for there to be a dissection (answering Hilbert's Third Problem), and Sydler proved in 1965 that these conditions are also sufficient. Vladik Kreinovich [Geombinatorics 18(1):26–34, 2008] recently proved that there is a finite algorithm deciding the equality of two Dehn invariants. But what about finding a dissection in that case?

This problem appears in "Hinged Dissections Exist" by Timothy G. Abbott, Zachary Abel, David Charlton, Erik D. Demaine, Martin L. Demaine, and Scott D. Kominers. [SoCG'08; arXiv:0712.2094]

Dagstuhl Seminar Proceedings 09111
Computational Geometry
http://drops.dagstuhl.de/opus/volltexte/2009/2030

Problem 3 (Günter Rote). For 3D convex polytopes, we know the worst-case number of edge sequences for geodesic paths is $\Theta(n^4)$. What can be said about the number of different face sequences for geodesic paths on the surface of a convex polytope in 4D? Is it polynomial? Is it polynomial for a fixed source point? (conjectured by Miller and Pak)

Problem 4 (Günter Rote). Given n red and blue points. Perform least-squares matching, while rotating the blue points. How many different optimal assignments are there? Is there a polynomial bound? (The optimal assignment is unchanged under translations and scalings. With rotation, the problem can be transformed into a parametric assignment problem with costs of the form $c_{ij} = a_{ij} + t \cdot b_{ij}$, $t \in \mathbb{R}$. Carstensen [1] has shown that for a *general* parametric problem of this type, there can be a super-polynomial number of different optimal solutions.)

Problem 5 (Christian Knauer). Klee's measure problem.

Depth of an arrangement of n hyperplane sin $\Re^d$ (d is not fixed): Is it NP-hard to determine? (Sariel mentions that this may be known in the learning theory community.)

Problem 6 (Mohammad Abam). Given a set of disjoint disks in the plane, can one find a t-spanner with $O(n)$ edges and one point per disk that remains a t-spanner for the points no matter where they lie within their respective disks? (It is possible if the disks are all of the same radius and for disks with arbitrary sizes the best known result is a t-spanner with $O(n \log n)$ edges.)

Problem 7 (Mohammad Abam). Perform halfplane range reporting in the kinetic setting (for moving points in the plane) with polylogarithmic event-handling time and few events (perhaps $O(n^3)$ events and $O(n^2)$ events would be perfect) such that at current time a half-plane query can be answered in time $O(\log n + k)$?

Problem 8 (Mohammad Abam). Is it possible to build a BSP of complexity $O(n^2)$ for n disjoint convex objects of arbitrary sizes in 3D? ($O(n^3)$ is known.)

Problem 9 (Mohammad Abam). Given n points in $\Re^2$, can one find a subset of size $\Omega(n)$ that has a bounded degree triangulation? More precisely, for a given constant c, what is the minimum value t such that any set of n points has a subset of t points such that it has a triangulation of maximum degree c?

Problem 10 (Jeff Erickson). Acute triangulations in 2D (can do), 3D (sometimes, but for exactly what values of N?), 4D (open), 5D (known to be impossible).

Problem 11 (Joe Mitchell). (A problem given as homework in my CG class, 10 points extra credit.) Given a planar polygonal domain P (with holes), how complex is it to decide if P has a triangulation whose graph is 3-colorable? Of course, any triangulation of P can be (vertex) 4-colored, by planarity. There are examples (with as few as 7 vertices) for which no triangulation of P can be 3-colored. (Finding a smallest such example was the main exercise in the homework problem; showing hardness was extra credit.)

Follow-up: This problem is indeed NP-hard, from 3-colorability of planar graphs (thanks to Oswin Aichholzer and Günter Rote, who each get 10 points of extra credit!).

Problem 12 (Sariel Har-Peled). Given a set L of n lines in $\Re^2$, and an $\epsilon > 0$. Is there an ϵ-net, $S \subset L$, for L of size $|S| = O(1/\epsilon)$? (so that any vertical segment that intersects ϵn lines of L must intersect at least one line of S)

Problem 13 (Joe Mitchell). Given an n-vertex nonconvex polyhedron P in 3-space, how efficiently can one find a longest line segment within P?

Problem 14 (Günter Rote). Given a self-intersecting closed curve in the plane, how fast can one decide whether it is the vertical projection of the boundary of a disk embedded in space for which the same side is always the "upper side" (facing the $+z$ coordinate direction)? (Shown to be NP-hard when the surface is allowed to have arbitrary topology, see Eppstein and Mumford [2]. note=arxiv:0806.1724,) The restriction to a disk amounts to requiring that the total net turning angle of the curve is 2π.)

Problem 15 (Jean-Daniel Boissonnat). (Contributed by email.) The problem is motivated by the construction of Delaunay triangulations in $\Re^d$. Due to memory storage limitation, we only represent the 1-skeleton of the triangulation (edges and vertices). Two vertices are called neighbors if they belong to a Delaunay edge. The d-simplices are reconstructed on-line when needed (see our paper to be presented at SoCG 2009).

A basic operation is then the following. Given a d-simplex t, and a vertex $p \in t$, we want to find the vertex p' that defines the d-simplex $(p'f)$ adjacent to t through the face $f = t \setminus \{p\}$. Since the edges of the triangulation are stored, the problem reduces to intersecting the lists of neighbors of the vertices of f (and deciding which the true one). The problem is: what is the size W of this intersection (i.e. the vertices that are neighbors of all vertices of f)?

We have observed experimentally that, for uniformly distributed points, W grows quite slowly, much slower than the number M of neighbors of a vertex. For example, for $d = 5$, $M = 73$, $W = 6.5$, $d = 6$, $M = 164.6$, $W = 11.6$ on average.

References

[1] P. J. Carstensen. *The Complexity of Some Problems in Parametric Linear and Combinatorial Programming.* PhD thesis, University of Michigan, 1983.

[2] D. Eppstein and E. Mumford. Self-overlapping curves revisited. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 160–169. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2009.