Open Problems from CCCG 2005

Erik D. Demaine* Joseph O'Rourke†

The following is a list of the problems presented on August 10, 2005 at the open-problem session of the 17th Canadian Conference on Computational Geometry held in Windsor, Ontario, Canada.

Point Location in an Arrangement
Pat Morin
Carleton University
morin@scs.carleton.ca

Consider the following special case of planar point location: preprocess k sets of lines, where each set consists of parallel lines, to support queries of the form "given a point p, what is the line immediately above or below p?" What is the fastest possible query time as a function of k and the total number n of lines? In other words, the n given lines have k distinct orientations. See Figure 1. In this problem, the data structure must use $O(n \operatorname{polylog} n)$ space, preventing us from simply constructing the arrangement of $\Theta(n^2)$ cells and preprocessing with standard point location. An obvious $O(k \log n)$ algorithm—search for p in each set of parallel lines separately, and then combine the answers—is so far the best solution to this problem.

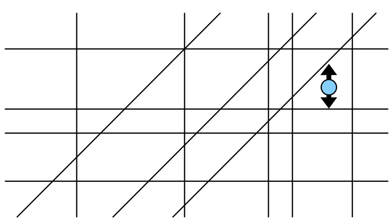

Figure 1: Preprocess these $n = 12$ lines with $k = 3$ orientations to support querying for the line above or below a given point.

The problem remains interesting even for small values of k. For $k = 1$, a simple binary search solves the problem in $\lg n + O(1)$ probes, and this bound matches the information-theoretic lower bound: the arrangement has n cells. For $k = 2$, the obvious algorithm solves the problem in $2 \lg n + O(1)$ probes, and again this bound matches the information-theoretic lower bound because the arrangement has $\Theta(n^2)$ cells. But once $k = 3$, the two bounds diverge: the obvious algorithm solves the problem in $3 \lg n + O(1)$ probes, while the information-theoretic lower bound remains $2 \lg n + O(1)$ because the arrangement has $\Theta(n^2)$ cells. Is the right bound $2 \lg n + O(1)$, $3 \lg n + O(1)$, or something in between?

Update: At the conference, Stefan Langerman obtained a data structure that uses $O(r^2 + (n/r)^2)$ space and with $O(\log r + k \log(n/r))$ query time, for any parameter r, using $(1/r)$-cuttings. In particular, choosing $r = n^{1-s/k}$ for any constant s gives a data structure of size $O(n^{2-2s/k})$ that answers queries in $O(k + s \log n) = O(k + \log n)$ time.

Monotone Chain Visibility
Boaz Ben-Moshe
Ben-Gurion University
benmoshe@cs.bgu.ac.il

Consider an x-monotone chain in the plane, i.e., a planar polygonal chain that is met by any vertical line in at most one point. Imagine the chain as the ground, with the region below the chain the earth (an obstacle), and the region above the air (free space). Call two vertices of the chain *visible* if the line segment connecting them does not go below the chain (what might be called "top-side" visibility). A *guard set* is a set of chain vertices (*guards*) such that every point along the chain vertex is visible to at least one guard. Equivalently, a guard set is a dominating set in the visibility graph. See Figure 2.

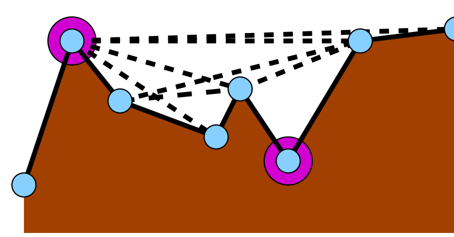

Figure 2: Guarding an x-monotone terrain with two guards.

What is the complexity of computing the guard set of minimum size for a given x-monotone chain in

*MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar Street, Cambridge, MA 02139, USA, edemaine@mit.edu

†Department of Computer Science, Smith College, Northampton, MA 01063, USA. orourke@cs.smith.edu. Supported by NSF Distinguished Teaching Scholars award DUE-0123154.

the plane? According to the poser, "most tenured professors think the problem is NP-hard."

This problem in fact goes back to 1995, when Chen et al. [CEU96] claimed an NP-hardness result, but "the proof, whose details were omitted, was never completed successfully" [Kin06]. The best approximation algorithm so far is a 4-approximation by King [Kin06]; see that paper for references to earlier approximation algorithms.

References

[CEU96] Danny Z. Chen, Vladimir Estivill-Castro, and Jorge Urrutia. Optimal guarding of polygons and monotone chains. In *Proc. 7th Canadian Conference on Computational Geometry*, pages 133–138, Aug. 1995.

[Kin06] James King. A 4-approximation algorithm for guarding 1.5-dimensional terrains. In *Proc. 7th Latin American Symposium on Theoretical Informatics*, volume 3887 of *Lecture Notes Comput. Sci.*, pages 629–640, Apr. 2006.

Unique Ham-Sandwich Cut

Stefan Langerman
Université Libre de Bruxelles
stefan.langerman@ulb.ac.be

How quickly can you determine whether a given set of red and blue points in the plane have a unique ham-sandwich cut? A ham-sandwich cut—a line that divides each color class in half—always exists and one can be found in $O(n)$ time [LMS94]. See Figure 3. A point set has a *unique* ham-sandwich cut if all ham-sandwich cuts are combinatorially equivalent, i.e., induce the same partition of the point set. In particular, if the numbers of red and blue points are both odd, the ham-sandwich cut must pass through a red point and a blue point, so this notion of uniqueness means that there is exactly one ham-sandwich cut.

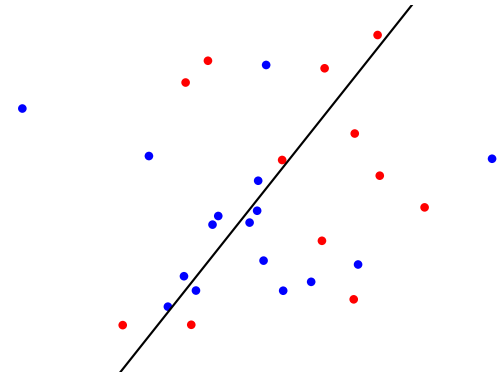

Figure 3: A ham-sandwich cut.

Chien and Steiger [CS95] proved a lower bound of $\Omega(n \lg n)$ on any algorithm in the linear decision tree model that determines whether a point set has a unique ham-sandwich cut, by a reduction from element uniqueness. So testing uniqueness is strictly harder than finding a ham-sandwich cut. But how much harder?

One approach to solving the problem is computing the intersection of the red median level and the blue median level in the dual line arrangement. We can walk those median levels and find intersections in time proportional to the total size of those median levels. The current best upper bound on this size is $O(n^{4/3})$ [Dey98], while Erdős conjectured an upper bound $O(n^{1+\varepsilon})$). Can you do better than this approach? Or are there lower bounds in some model?

The general weighted version of this problem, where points can have positive and negative weights, is essentially solved in [BS04]: determining uniqueness is 3SUM-hard, so likely requires $\Omega(n^2)$ time in any algebraic computation tree, and there is a simple $O(n^2)$-time algorithm. In contrast, a ham-sandwich cut can be found in $O(n \lg n)$ time in this scenario.

References

[BS04] Prosenjit Bose and Stefan Langerman. Weighted ham-sandwich cuts. In *Proc. Japan Conference on Discrete and Computational Geometry*, volume 3742 of *Lecture Notes Comput. Sci.*, pages 48–53, Oct. 2004.

[CS95] Hank Chien and William Steiger. Some geometric lower bounds. In *Proc. 6th International Symposium on Algorithms and Computation*, volume 1004 of *Lecture Notes Comput. Sci.*, pages 72–81, Dec. 1995.

[Dey98] Tamal K. Dey. Improved bounds for planar k-sets and related problems. *Discrete Comput. Geom.*, 19(3):373–382, 1998.

[LMS94] Chi-Yuan Lo, Jiří Matoušek, and William L. Steiger. Algorithms for ham-sandwich cuts. *Discrete Comput. Geom.* 11:433–452, 1994.

Rigid Components of Planar Graph

Ileana Streinu
Smith College
streinu@cs.smith.edu

How quickly can you decompose a planar graph into its generically rigid components? A graph is *generically rigid in the plane* if almost all embeddings into the plane are rigid, i.e., cannot flex while preserving the edge lengths. Equivalently, Laman's characterization says that an n-vertex graph is generically rigid in the plane precisely if it has $2n-3$ edges on which every induced subgraph of k vertices has at most $2k-3$ edges. The problem asks to decompose a graph into maximal subgraphs each of which is generically rigid in the plane. See Figure 4.

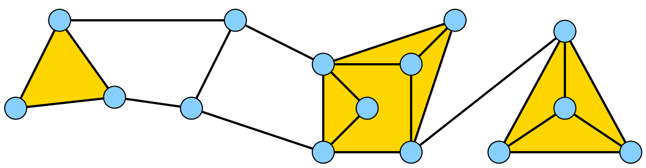

Figure 4: Decomposing a graph into generically rigid components. (Each uncolored edge also belongs to its own component.)

This problem can be solved on general graphs in $O(n^2)$ time [LST05] and on pseudotriangulation mechanisms in $O(n)$ time [SS05]. What about planar graphs? Can we at least test rigidity of planar graphs in subquadratic time? Both problems are posed explicitly in [SS05].

References

[LST05] Audrey Lee, Ileana Streinu, and Louis Theran. Finding and maintaining rigid components. In *Proc. 17th Canadian Conference on Computational Geometry*, pages 219–222, Aug. 2005.

[SS05] Jack Snoeyink and Ileana Streinu. Computing rigid components of pseudo-triangulation mechanisms in linear time. In *Proc. 17th Canadian Conference on Computational Geometry*, pages 223–226, Aug. 2005.

Invertible Turbulence Function for Ray Tracing

Robert Dawson
St. Mary's University
rdawson@husky1.stmarys.ca

This problem arose some years ago in the context of shader functions for ray tracing and similar applications. To make a texture look natural, it is often perturbed with a 3D "vector fractal noise function": the texture at $\vec{x}$ is taken to be $t(\vec{x} + k\vec{f}(\vec{x}))$ where t is the basic shader function and k the amount of perturbation. For instance, an unperturbed wood grain texture consists of perfect concentric cylinders of light and dark brown. A slighty perturbed version will have slight wobbles in the grain; a heavily perturbed version will be "wild" like burl walnut. The exact nature of the vector noise is not terribly important.

However, this method of randomizing has limitations. One can ask "what gets moved to the point I'm looking at?" but not "where does this feature get moved to?" This makes certain desirable types of shader difficult; I found that this was a problems while attempting to create a "random rectangular masonry" shader and, on another occasion, a "snowflake shader" that would give the effect of a particle cloud without having to create and store individual particles (an important consideration on standard-issue hardware). A useful mathematical tool would be an invertible turbulence function such that $\vec{x} \mapsto \vec{x} + \vec{f}(\vec{x})$ is easily inverted. The exact nature of the function is still not too crucual, although it should be smooth and continuous. Piecewise-quadratic functions are probably a good place to start looking.

Omnidirectional Visibility Representations

Therese Biedl
University of Waterloo
biedl@waterloo.ca

This problem is motivated by (one-directional) visibility representations of triangulated planar graphs, in which each vertex is represented by a horizontal line segment, and edges are represented by vertical visibility between line segments. See Figure 5 (left). Any triangulated planar graph can be represented in this way—see, e.g., [LLS03]—and every graph represented in this way must be planar and triangulated (treating distinct visibilities between two segments as multiple edges). (Similar problems have also been considered in 3D [FM99] with one- or two-directional orthogonal visibility.)

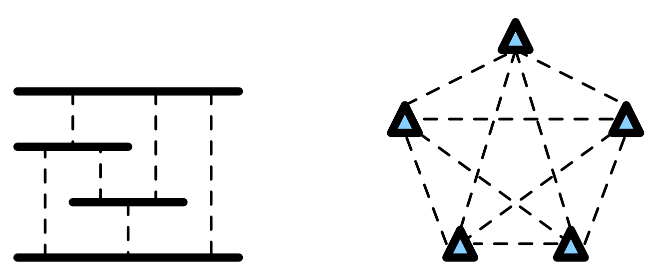

Figure 5: Representing K_4 with vertical visibility among horizontal segments, and representing K_5 with omnidirectional visibility among polygons.

What if the shapes are simple polygons instead of line segments, and the visiblity is omnidirectional instead of just vertical? What graphs can be so represented? Now the graph need not be planar; for example, we can represent any complete graph by n points (or tiny triangles) in general position. See Figure 5 (right). With very general, nonconvex shapes, it is possible to represent an arbitrary triangulated planar graph. The question is to determine the simplest possible shapes for which either triangulated planar graphs or general graphs can be represented. For example, what graphs can be represented by rectangles? By convex shapes? Also, for each of these scenarios, it is interesting to determine the grid of smallest resolution that enables representation.

Update: At the conference, several participants focused on visibility representations of planar graphs by unit disks (instead of polygons). This case seems interesting and nontrivial, but it is not yet resolved.

References

[FM99] S. P. Fekete and H. Meijer. Rectangle and box visibility graphs in 3D. *International Journal of Computational Geometry and its Applications*, 9:1–27, 1999.

[LLS03] Ching-Chi Lin and Hseuh-I Lu and I-Fan Sun. Improved compact visibility representation of planar graph via Schnyder's realizer. In *Proc. 20th International Symposium on Theoretical Aspects of Computer Science*, volume 2607 of *Lecture Notes Comput. Sci.*, pages 14–25, Feb.–Mar. 2003.

Volume-Maximizing Convex Shape

Joseph O'Rourke
Smith College
orourke@cs.smith.edu

TOPP #62

This problem is a rephrasing and generalization of a question posed by Joseph Malkevitch in 2002. Let C be a convex piece of paper; it may be smooth, or a polygon. A *perimeter-halving folding* is a folding of C obtained by identifying two points x and y on the boundary of C that halve the perimeter, and then folding C by "gluing" xy to yx. This gluing always produces a unique convex shape in 3D, a polyhedron if C is a convex polygon [DO06]. What unit-area shape C achieves the maximum volume possible via a perimeter-halving folding? The answer is only known empirically for the single case of C being a square [ADO03], which achieves about 60% of the volume of a sphere of unit surface area. See Figure 6. The restriction to perimeter halving eliminates more complex foldings possible for some convex polygons. Smooth shapes only admit perimeter-halving foldings.

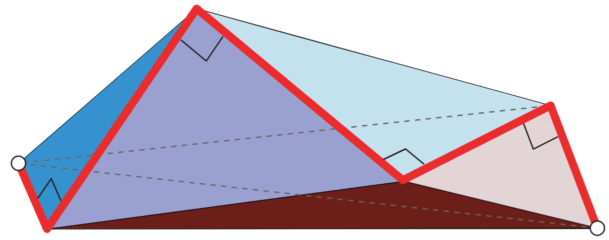

Figure 6: The empirally largest-volume convex polyhedron foldable from a square.

References

[ADO03] Rebecca Alexander, Heather Dyson, and Joseph O'Rourke. The convex polyhedra foldable from a square. In *Revised Papers from the Japan Conference on Discrete and Computational Geometry*, volume 2866 of *Lecture Notes Comput. Sci.*, pages 38–50, Dec. 2002.

[DO06] E. D. Demaine and J. O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, and Polyhedra*. Cambridge University Press. In preparation. http://www.fucg.org.

Ribbon Curves

Joseph O'Rourke
Smith College
orourke@cs.smith.edu

I first posed a version of this problem at the 14th Fall Workshop on Computational Geometry held at MIT, Nov. 2004. Characterize the C^2 space curves that are *ribbon curves*: curves that are the edge of some uncreased paper "ribbon," a rectangle. The motivation here is to understand D-forms, the subject of a problem posed earlier [DO03]. Stephanie Jakus and I showed that it is easy to find C^1 curves that are not ribbon curves. We also constructed a closed C^2 curve that is not a ribbon curve, and believe that this example leads to an open C^2 curve that is not a ribbon curve.

Update: Robert Dawson immediately came up with an open C^∞ curve that is not a ribbon curve. So the answer to the posed question cannot be, "all sufficiently smooth curves."

The poser of the problem conjectures that the following constitute necessary conditions (a key part of the reasoning was in collaboration with Satyan

Devadoss). Parametrize the curve C by t, and let $B(t)$ be the (unit) binormal vector to the curve. $B(t)$ is the cross product of the tangent and the normal vectors at $C(t)$. Then $C(t)$ is a ribbon curve only if (a) $B(t)$ is continuous in t, and (b) for every interval of t over which $|B(t)| = 0$ (e.g., a straight section of $C(t)$), the binormals just before and just after the interval are parallel. It seems unlikely that these conditions are sufficient.

References

[DO03] Erik D. Demaine and Joseph O'Rourke. Open problems from CCCG 2002. In *Proc. 15th Canadian Conference on Computational Geometry*, pages 178–181, 2003. arXiv:cs.CG/0212050.

Illuminating with True Lightbulbs

Joseph O'Rourke
Smith College
orourke@cs.smith.edu

In physics, a point light source illuminates according to an inverse-square law, so that a unit-irradiance light at x illuminates a point at a distance r from x with intensity $1/r^2$. Also, if multiple point light sources illuminate a common point, the intensity sums. For a polygon of n vertices and of diameter d, what is the largest number of lights ever needed to illuminate all points to intensity I? To be specific, set $I = 1$; in this case, points within a radius of 1 around a bulb recieve radiance ≥ 1 from that bulb.

As an example, a disk of diameter $d = 2$ needs just one bulb at it center, but a disk of $d = 2\sqrt{2} \approx 2.83$ needs three bulbs; see Figure 7. As Sándor Fekete observed at the conference, in fact, these three bulbs suffice to illuminate a larger disk as well.

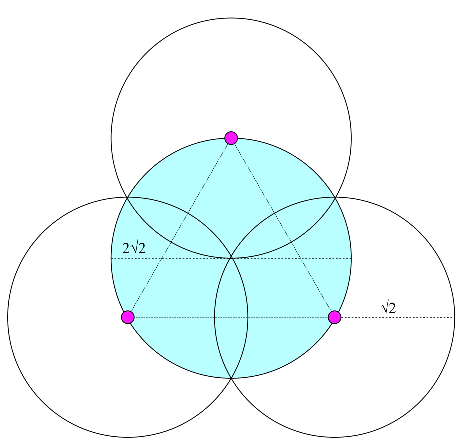

Figure 7: Three bulbs illuminate a disk of diameter $2\sqrt{2}$.

Update: It was pointed out that these types of problems were posed two months earlier in [EFKM05], which addresses the stage-illumination version. Robert Dawson showed that any density of lightbulbs in 3D illuminates all points with divergent intensity (a version of Olber's Paradox), but in 2D this does not hold.

References

[EFKM05] Friedrich Eisenbrand, Stefan Funke, Andreas Karrenbauer, and Domagoj Matijevic. Energy-aware stage illumination. In *Proc. 21st ACM Symposium on Computational Geometry*, pages 336–345, Jun. 2005.

Rolling a Die

Joseph O'Rourke
Smith College
orourke@cs.smith.edu

This problem is inspired by van Deventer's "Rolling block mazes" [vD04]. Label the faces of a unit cube with numbers 1–6 as in a die. (There are actually two standard labelings, with all opposite pairs of face labels summing to 7; pick one.) Place the cube to sit on an integer lattice grid, with one corner at the origin and sides aligned with the axes. Label some finite subset S of n lattice squares with numbers in $\{1, 2, 3, 4, 5, 6\}$. The problem is to roll the cube over its edges so that, for each square $s \in S$ labeled λ, the cube lands on s precisely once, and when it does so, the top face of the cube has label λ. See Figure 8.

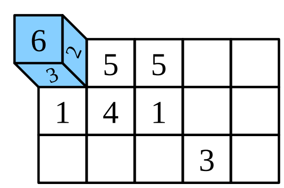

Figure 8: A dice-rolling puzzle with specified start configuration. Although you can roll the die anywhere in the plane, it suffices to stay in the finite board shown. One solution: ESENEESSWWWWN.

What is the computational complexity of solving an instance of this problem? I conjectured that it is NP-complete.

Update: At the conference, Erik Demaine and Sándor Fekete independently developed arguments showing that the problem as posed is NP-complete, by reduction from Hamiltonian paths in grid graphs. Many other variants of the problem were posed, and are under active investigation. For example, Martin Demaine designed a CCCG puzzle with multiple rolling cubes; see Figure 9.

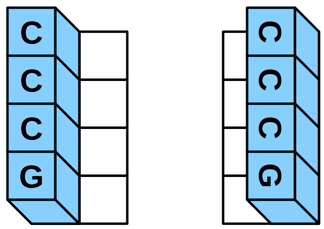

Figure 9: Roll the cubes from the left configuration to the right configuration, at all times remaining in the 2×4 board. Each cube has a label on exactly one face.

References

[vD04] M. Oskar van Deventer. Rolling block mazes. In Barry Cipra, Erik D. Demaine, Martin L. Demaine, and Tom Rodgers, editors, *A Tribute to a Mathemagician*, pages 241–250, A K Peters, Nov. 2004.

Polyhedral 3SUM

Jeff Erickson
University of Illinois Urbana Champaign
jeffe@cs.uiuc.edu

Consider the following *min-convolution* problem: given two arrays $A[0\,.\,.\,n]$ and $B[0\,.\,.\,n]$, compute $\min_i(A[i] + B[k-i])$ for all $k = 0, 1, \ldots, 2n$. Alternatively, we can think of this problem as computing the min of every antidiagonal of the $X+Y$ matrix $C[i,j] = A[i]+B[j]$. It is easy to solve this problem in $O(n^2)$ time. Can it be solved in $o(n^2)$ time? Is there a lower bound in some model? This problem came out of Godfried's Music Information Retrieval Workshop in Jan.–Feb. 2005, but it turns out to be much older; see [BCD+06].

I do not believe that the min-convolution problem is 3SUM-hard. But in the algebraic decision tree model, both this problem and 3SUM are harder than the following *polyhedral 3SUM* problem: given three arrays $A[0\,.\,.\,n]$, $B[0\,.\,.\,n]$, and $C[0\,.\,.\,2n]$, where $A[i] + B[j] \geq C[i+j]$ for all i, j, does $A[i] + B[j] = C[i+j]$ for some i, j pair? This problem has two main differences from 3SUM: the question is about pairs i, j, not triples i, j, k, and there is an extra assumption about the input. A simple iteration solves this problem in $O(n^2)$ time. There is also an $\Omega(n^2)$ lower bound when the algorithm is limited to looking at three items at once. But this lower bound is not useful for the min-convolution problem described above, because any algorithm needs to look at at least four items at once. So min-convolution is just over the boundary of where we know how to prove lower bounds. Is there a stronger lower bound for polyhedral 3SUM?

Polyhedral 3SUM is equivalent to, given a point in the $4n$-dimensional polyhedron defined by the inequalities $A[i] + B[j] \geq C[i+j]$ for all i, j, decide whether the point is on the boundary. This polyhedron has n^2 facets, suggesting an $\Omega(n^2)$ lower bound. If we are allowed to preprocess n (the only parameter defining the polytope), we may be able to solve this problem faster using hyperplane arrangement data structures.

Update: After the conference, Timothy Chan found a subquadratic solution to min-convolution. Details can be found in the forthcoming paper [BCD+06].

References

[BCD+06] David Bremner, Timothy M. Chan, Erik D. Demaine, Jeff Erickson, Ferran Hurtado, John Iacono, Stefan Langerman, and Perouz Taslakian. Necklaces, convolutions, and $X+Y$. In *Proc. 14th Annual European Symposium on Algorithms*, Sept. 2006. To appear.