Robust Proximity Queries: an Illustration of Degree-driven Algorithm Design *
(Short Abstract)

Giuseppe Liotta
Dipartimento Informatica e Sistemistica
Universitá di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
liotta@dis.uniroma1.it

Franco P. Preparata
Center for Geometric Computing
Department of Computer Science
Brown University
115 Waterman Street
Providence, RI 02912–1910, USA
franco@cs.brown.edu

Roberto Tamassia
Center for Geometric Computing
Department of Computer Science
Brown University
115 Waterman Street
Providence, RI 02912–1910, USA
rt@cs.brown.edu

Abstract

In the context of methodologies intended to confer robustness to geometric algorithms, we elaborate on the exact computation paradigm and formalize the notion of degree of a geometric algorithm, as a worst-case quantification of the precision (number of bits) to which arithmetic calculation have to be executed in order to guarantee topological correctness. We also propose a formalism for the expeditious evaluation of algorithmic degree. As an application of this paradigm and an illustration of our general approach where algorithm design is driven also by the degree, we consider the important classical problem of proximity queries in 2 and 3 dimensions, and develop a new technique for the efficient and robust execution of such queries based on an implicit representation of Voronoi diagrams. Our new technique offers both low degree and fast query time, and for 2D queries is optimal with respect to both cost measures of the paradigm, asymptotic number of operations and arithmetic degree.

*Research supported in part by the U.S. Army Research Office under grant DAAH04–96–1–0013, and by the N.A.T.O.-C.N.R. Advanced Fellowships Programme. This work has been done while Giuseppe Liotta was with the Center of Geometric Computing, Department of Computer Science, Brown University. The research of Giuseppe Liotta is currently supported in part by EC ESPRIT Long Term Research Project ALCOM-IT under contract no. 20244, and by Progetto Finalizzato Trasporti 2 (PFT 2) of the Italian National Research Council (CNR).

1 Introduction

The increasing demand for efficient and reliable geometric software libraries in key applications such as computer graphics, geographic information systems, and computer-aided manufacturing is stimulating a major renovation in the field of computational geometry. The inadequacy of the traditional simplified framework has become apparent, and it is being realized that, in order to achieve an effective technology transfer, new frameworks and models are needed to design and analyze geometric algorithms that are efficient in a practical realm.

The real-RAM model with its implicit infinite-precision requirement, has proved unrealistic and needs to be replaced with a realistic finite-precision model where geometric computations can be carried out either exactly or with a guaranteed error bound. This has motivated a great deal of research on the subject of robust computational geometry (see, e.g., [1, 5, 4, 6, 11, 13, 15, 18, 17, 20, 22, 24, 26, 8, 12, 16]). For an early survey of the different approaches to robust computational geometry the reader is referred to [19].

To a first, rough, approximation, robustness approaches are of two main types: perturbing and nonperturbing. Perturbing approaches transform the given problem into one that is intended not to suffer from well-identified shortcomings; nonperturbing approaches are based on the notion of "exact" (rather than "approximate") computations, with the assumption that (bounded-length) input data are error free. In this category falls the exact-geometric computation paradigm independently advocated by Yap [27] and by the Saarbrücken school [3], and so does our approach. Within this paradigm, we introduce the notion of *degree* of an algorithm, which describes, up to a small additive constant, the arithmetic precision (i.e., number of bits) needed by the exact computation paradigm. Namely, if the coordinates of the input points of a degree-d geometric algorithm are b-bit integers, then, as we shall substantiate below, the algorithm may be required on some instances to perform arithmetic computations with bit precision $d(b + O(1))$.

Theoretical analysis and experimental results show that multiprecision numerical computations take up most of the CPU time of exact geometric algorithms (see, e.g., [21, 23]). Thus, we believe that, in defining the efficiency of a geometric algorithm, the degree should be considered as important as the asymptotic time complexity and should correspondingly play a major role in the design stage. In fact, the principal thrust of this paper is to present algorithm degree as a major design criterion for geometric computation. Research along these lines involves re-examining the entire rich body of computational geometry as we know it today.

In this paper, we consider as a test case a problem area, geometric proximity, which plays a major role in several applications and has recently attracted considerable attention because, due to its demands of high precision for exact computation, it is particularly appropriate in assessing effectiveness of robust approaches (see, e.g., [2, 4, 7, 14, 15, 11, 25, 16]). In particular we shall illustrate the role played by the degree criterion if one wishes to comply with the standard exact-computation paradigm.

To illustrate the approach, we recall that Voronoi diagrams are the search structures which permit answering a proximity query without evaluating all query/site distances. Therefore, given the set of sites, their Voronoi diagram is computed and supplied as a planar subdivision to a point-location procedure. Assuming that the coordinates of all input data (also called *primitive* points) are b-bit integers, the coordinates of the points computed by the algorithm (referred to here as *derived* points, e.g., the vertices of a Voronoi diagram of points and segments) must be stored with a representation scheme that supports rational or algebraic numbers as data types (through multiprecision integers). Specifically, the coordinates (x, y) of a Voronoi vertex are rational numbers given by the ratio of two determinants (of respective orders 3 and 2) whose entries are integers of well-defined maximum modulus. The fundamental operation used by any point location algorithm is the point-line discrimination, which consists of determining whether the query point q is to the left or to the right of an edge between vertices v_1 and v_2. For the case of the Voronoi diagram $V(S)$, this is equivalent to evaluating the sign of a 3×3 determinant whose rows are the homogeneous coordinates of q, v_1, and v_2, a computation that needs about $6b$ bits of precision. This should be compared with the $O(n)$-time brute-force method that computes the (squares of the) distances from q to all the sites of S, and executes arithmetic computations with only $2b$ bits of precision (which is optimal).

Guided by the low-degree criterion, in this paper we present a technique – complying with the exact-computation paradigm – which uses a new point-location data structure for Voronoi diagrams, such that the test operations executed in the point location procedure are distance comparisons, and are therefore executed with optimal $2b$ bits of precision. Hence, our approach reconciles efficiency with robustness, and supports an object-oriented programming style where access to the geometry of Voronoi diagrams in point-location queries is encapsulated in a small set of geometric test primitives. It must be pointed out that distance comparisons have already been used nontrivially for proximity search (extremal-search method [10]). However, we shall show that the latter method fails to achieve optimal degree because the search is based on predicates requiring $4b$ bits of precision; moreover, the high overhead of the search technique (which uses the hierarchical polytope representation [9]) casts some doubts on the practicality of the method.

The main results of this work are summarized in Table 1. Considering, for the time being, the degree as a measure of complexity, we show that previous methods exhibit a sharp tradeoff between degree and query time. Namely, low degree is achieved by the slow brute-force search method, while fast algorithms based on point-location in a preprocessed Voronoi diagram or on the extremal-search method have high degree. Our new technique gives instead both low degree and fast query time, and is optimal with respect to both cost measures for queries in sets of 2D point sites.

References

[1] F. Avnaim, J.-D. Boissonnat, O. Devillers, F. Preparata, and M. Yvinec. Evaluating signs of determinants using single-precision arithmetic. Research Report 2306, INRIA, BP93, 06902 Sophia-Antipolis, France, 1994.

Query	Method	Degree	Time
nearest neighbor	brute-force distance comparison	2 *	$O(n)$
	point location in explicit Voronoi diagram	6	$O(\log n)$ *
	extremal-search method	4	$O(\log n)$ *
	point location in implicit Voronoi diagram	2 *	$O(\log n)$ *
k-nearest neighbors and circular range search	brute-force distance comparison	2 *	$O(n)$
	point location in explicit order-k Voronoi diagram	6	$O(\log n + k)$ *
	point location in implicit order-k Voronoi diagram	2 *	$O(\log n + k)$ *
nearest neighbor among points and segments	brute-force distance comparison	6	$O(n)$
	point location in explicit Voronoi diagram	64	$O(\log n)$ *
	point location in implicit Voronoi diagram	6	$O(\log n)$ *
3D nearest neighbor	brute-force distance comparison	2 *	$O(n)$
	point location in explicit 3D Voronoi diagram	8	$O(\log^2 n)$
	point location in implicit 3D Voronoi diagram	3	$O(\log^2 n)$

Table 1: Comparison of the degree and time of algorithms for some fundamental proximity query problems. A * denotes optimality. The new technique introduced in this paper (*point location in an implicit Voronoi diagram*) always outperforms previous methods and is optimal for 2D queries.

[2] C. Burnikel. *Exact Computation of Voronoi Diagrams and Line Segment Intersections.* Ph.D thesis, Universität des Saarlandes, Mar. 1996.

[3] C. Burnikel, J. Könnemann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig. Exact geometric computation in LEDA. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C18–C19, 1995.

[4] C. Burnikel, K. Mehlhorn, and S. Schirra. How to compute the Voronoi diagram of line segments: Theoretical and experimental results. In *Proc. 2nd Annu. European Sympos. Algorithms (ESA '94)*, volume 855 of *Lecture Notes in Computer Science*, pages 227–239. Springer-Verlag, 1994.

[5] C. Burnikel, K. Mehlhorn, and S. Schirra. On degeneracy in geometric computations. In *Proc. 5th ACM-SIAM Sympos. Discrete Algorithms*, pages 16–23, 1994.

[6] K. L. Clarkson. Safe and effective determinant evaluation. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 387–395, 1992.

[7] T. K. Dey, K. Sugihara, and C. L. Bajaj. Delaunay triangulations in three dimensions with finite precision arithmetic. *Comput. Aided Geom. Design*, 9:457–470, 1992.

[8] D. P. Dobkin. Computational geometry and computer graphics. *Proc. IEEE*, 80(9):1400–1411, Sept. 1992.

[9] D. P. Dobkin and D. G. Kirkpatrick. Fast detection of polyhedral intersection. *Theoret. Comput. Sci.*, 27:241–253, 1983.

[10] H. Edelsbrunner and H. A. Maurer. Finding extreme points in three dimensions and solving the post-office problem in the plane. *Inform. Process. Lett.*, 21:39–47, 1985.

[11] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graph.*, 9:66–104, 1990.

[12] S. Fortune. Polyhedral modeling with multiprecision integer arithmetic. *Computer Aided Design*, 1996. to appear.

[13] S. Fortune. Stable maintenance of point set triangulations in two dimensions. In *Proc. 30th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 494–505, 1989.

[14] S. Fortune. Numerical stability of algorithms for 2-d Delaunay triangulations. *Internat. J. Comput. Geom. Appl.*, 5(1):193–213, 1995.

[15] S. Fortune and C. J. Van Wyk. Efficient exact arithmetic for computational geometry. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 163–172, 1993.

[16] S. Fortune and C. V. Wyk. Static analysis yields efficient exact integer arithmetic for computational geometry. *ACM Transactions on Graphics*, 1996. to appear.

[17] D. H. Greene and F. F. Yao. Finite-resolution computational geometry. In *Proc. 27th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 143–152, 1986.

[18] L. J. Guibas, D. Salesin, and J. Stolfi. Epsilon geometry: building robust algorithms from imprecise computations. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 208–217, 1989.

[19] C. M. Hoffmann. The problems of accuracy and robustness in geometric computation. *IEEE Computer*, 22(3):31–41, Mar. 1989.

[20] C. M. Hoffmann, J. E. Hopcroft, and M. T. Karasick. Robust set operations on polyhedral solids. *IEEE Comput. Graph. Appl.*, 9(6):50–59, Nov. 1989.

[21] M. Karasick, D. Lieber, and L. R. Nackman. Efficient Delaunay triangulations using rational arithmetic. *ACM Trans. Graph.*, 10:71–91, 1991.

[22] V. Milenkovic. Verifiable implementations of geometric algorithms using finite precision arithmetic. *Artif. Intell.*, 37:377–401, 1988.

[23] E. Mücke. Detri 2.2: A robust implementation for 3d triangulations. Manuscript, available at URL: http://www.geom.umn.edu:80/software/cglist/lowdvod.html, 1996.

[24] K. Sugihara and M. Iri. Construction of the Voronoi diagram for 'one million' generators in single-precision arithmetic. *Proc. IEEE*, 80(9):1471–1484, Sept. 1992.

[25] K. Sugihara, Y. Ooishi, and T. Imai. Topology-oriented approach to robustness and its applications to several Voronoi-diagram algorithms. In *Proc. 2nd Canad. Conf. Comput. Geom.*, pages 36–39, 1990.

[26] C. K. Yap. Symbolic treatment of geometric degeneracies. *J. Symbolic Comput.*, 10:349–370, 1990.

[27] C. K. Yap. Toward exact geometric computation. *Computational Geometry: Theory and Applications*, 1996. to appear.