

# Computational Geometry: A Retrospective

BERNARD CHAZELLE

Department of Computer Science  
Princeton University  
Princeton, NJ 08544, USA

## Abstract

Computational geometry is at a crossroads. New challenges and opportunities are likely to reshape the field rather drastically in the years ahead. I will survey some of its principal accomplishments, and in light of recent developments, I will discuss the profound transformations the field has begun to undergo. There are reasons to believe that computational geometry will emerge from this transition far richer and stronger but barely recognizable from what it was ten years ago.

Over the last two decades the field has enjoyed tremendous successes. Some of them might be dismissed as the cheap payoffs to be expected from any field lacking maturity. But others are the products of indisputable creativity and should be held as genuine scientific achievements. More important, the field is now able to claim a broad, solid foundation upon which its future can be securely built.

To mature fully as an original subfield of computer science, however, computational geometry must broaden its connections to applied mathematics while at the same time pay more than lip service to the applications areas that it purports to serve. Happily, active efforts to meet these challenges are underway.

Three recent developments are particular encouraging: one is the building of a theory of *geometric sampling* and its revolutionary impact on the design of geometric algorithms. Another is the maturing of *computational real-algebraic geometry* and *computational topology*: both subjects are being revitalized by the introduction of geometric (as opposed to purely algebraic) methods. On the practical end of the spectrum, the emergence of a sub-area concerned specifically with issues of

finite precision and degeneracy in geometric computing is a most welcome development.

## 1 Introduction

Nearly 20 years ago Shamos and Hoey [250] introduced Voronoi diagrams to the world of theoretical computer science. Computational geometry was born. To be sure, Voronoi diagrams had been known long before (surprisingly even Voronoi seemed to have been cognizant of them), but to compute them efficiently was a new concern. In short order, researchers identified a list of top-priority geometric problems to be solved. The oft-cited reason for working on these problems were their relevance to computer graphics, chemistry, statistical analysis, pattern recognition, geographic databases, and other equally vital applications areas.

I forget how convincing these claims might have seemed at the time, but with the benefit of hindsight they appear to have been fairly legitimate. Of course this might have been a happy coincidence. Taking their cue from Sir Hillary who, to the question, "Why did you climb Mount Everest?" replied, "Because it's there!" researchers felt that certain problems were central and compelling enough to merit their attention, with scant regard for practical fallouts. Classical examples are (i) computing the convex hull or the Voronoi diagram of a set of points; (ii) computing the intersection of several objects; (iii) subdividing a complex geometric object into simpler components, e.g., triangulating a simple polygon.

These are a sample of *geometric primitives* which we would expect any geometric software library to have. Two other important classes of problems emerged: *geometric searching* and *fixed-dimensional geometric optimization*. The first category includes problems such as *point location*: store a planar subdivision in some data structure so that given a query point, the face that contains it can be found quickly. It also includes *range searching*, e.g., set up a database to support queries such as: how many unread STOC proceedings are there within  $x$  miles of city  $y$ ? (Of course, we have here the added difficulty that reading the question might change

---

<sup>0</sup>This work was supported in part by NSF Grant CCR-93-01254 and The Geometry Center, University of Minnesota, an STC funded by NSF, DOE, and Minnesota Technology, Inc.

the answer...) *Geometric optimization* includes such problems as: given a set of points, compute its two nearest neighbors, its diameter, its smallest enclosing ellipsoid, etc.<sup>1</sup>

The good news is that most of the theoretical challenges laid down over a decade ago have been met. The field is thus ready to forge ahead and broaden its scope without leaving behind massive areas of darkness. To be sure, trouble spots remain (e.g.,  $k$ -set theory, hidden-surface removal), but most of the classical theoretical questions have been answered. Most satisfying is the fact that a whole body of mathematical tools and data structuring techniques have been created in the process.

The less-than-exhilarating news is that computational geometry has not yet had the practical impact it was expected to have. Issues such as robustness, stability, and degeneracies were long treated as annoying distractions from loftier intellectual pursuits. Computers have trouble with real numbers; theoreticians don't. As a result, the transfer of algorithmic ideas to the "real" world of geometric computing has lagged behind. Fortunately, this is being remedied through several promising research efforts, and one is entitled to a certain degree of optimism.

It is crucial for computational geometry to keep its eyes on the practical aspects of computing. Granted, good research should produce general, not ad hoc, results and this often necessitates a certain distance from specific, narrowly focused problems. Granted, practical problems have a knack for bringing up a hybrid mess of distinct, often contradictory, issues, which no mind of good taste would ever think of putting together. Trying to tackle all these issues at once is likely to result in the algorithmic analog of non-alcoholic diet champagne: pleasing no one by trying to please everyone. And yet one must pause and wonder: What is the point of designing convex hull algorithms if they can never be used or at least influence the way actual code is written? Practical applications are the lifeline of computational geometry. If it ever ceases to be so, chances are computational geometry will cease to be.

One final introductory remark: This little essay does not represent any views besides mine. Feel free to disagree. Of course, if my points don't come across, there won't be much to disagree about, and you might want to look at this piece as a harmless survey that the program committee chairman gently coerced me into writing. Naturally, as a survey it does suffer from its limited scope and my personal biases... All right, having thus

---

<sup>1</sup>It is commonly perceived that computational geometry stops where the dimension of the ambient space ceases to be a constant. For example, linear programming with a fixed number of variables belongs to computational geometry but general linear optimization does not. This view is being challenged by recent developments, however.

pretty much covered all my bases in the disclaiming department, I can now begin.

In the first three sections of this paper I discuss the current status of the "classical problems" (geometric primitives, searching, optimization) and mention some of the most important open problems in these areas. Section 5 is devoted to lower bounds, a particularly sore topic (though no sorer than elsewhere in computer science). Section 6 discusses the mini-revolution of sorts caused by the newly developed theory of geometric sampling. Sections 7 and 8 give a brief account of the emerging areas of computational real-algebraic geometry and computational topology. Section 9 rapidly surveys the current efforts to cope with finite precision and degeneracies. Section 10 mentions some important developments in combinatorial geometry, which have direct bearing on the complexity of certain geometric algorithms. Finally, a few closing words are given in Section 11.

## 2 Geometric Primitives

Historically, the three main categories are: partition, intersection, visibility. The latter is more aptly classified as geometric searching and will be discussed in the next section. Convex hulls and Voronoi diagrams occupy a category of their own, mostly for historical reasons. I will not mention the practical motivations behind these problems, but I will include appropriate references where they can be found.

**Polygon Triangulation.** Triangulating a simple polygon with  $n$  vertices has attracted a great deal of attention over the years. After an early  $O(n \log n)$  solution by Garey et al. [138] dating back to 1978, little tangible progress was reported for many years, besides efficient schemes for restricted classes of polygons. The first major breakthrough came in 1986 with Tarjan and Van Wyk's discovery of an  $O(n \log \log n)$  deterministic algorithm [257]. This was followed in 1989 by a simpler solution with the same complexity due to Kirkpatrick, Klawe, and Tarjan [176]. In the meantime, an  $O(n \log^* n)$  expected-time Las Vegas algorithm was proposed by Clarkson, Tarjan, and Van Wyk [89], later simplified by Seidel [247]. Finally, in 1990, an  $O(n)$  deterministic solution was discovered by the author [55]. (This is a rare case where determinism beats randomness.) The algorithm was parallelized optimally by Goodrich [144].

I suspect that the key to linear complexity is not that a polygon is simple (i.e., free of self-intersections) but that its boundary is a connected curve. Perhaps the most interesting problem left in this general area is to find a "grand unified" solution to polygonal triangula-

tion: Given  $p$  (nonsimple) polygons with a total of  $n$  vertices, compute all  $k$  pairwise intersections and triangulate the induced planar map in optimal  $O(n+k+p \log p)$  time. Such an algorithm would unify, among others, linear-time triangulation ( $p = 1, k = 0$ ) with the optimal line segment intersection algorithm of [60] ( $n = 2p$ ). Another interesting question is to find a truly simple optimal polygon triangulation algorithm, possibly using randomization.

**Partition Problems.** Polygon triangulation is but one example of a large class of problems whose objective is to partition a geometric object into a “few good” pieces. The subject has been heavily researched because of its relevance to the finite-element method, mesh generation, topology of 3-manifolds, interpolation theory, tool design, etc. [19, 21, 20, 24, 34, 35, 51, 59, 72, 76, 104, 219, 222, 227]. For example, the Delaunay triangulation is commonly used in the finite-element method because of its desirable morphology and overall simplicity.

Two goals are usually sought. One is to have partitions with relatively few pieces and the other is to avoid long, skinny ones. For example, we might want to decompose a three-dimensional polytope into a small number of convex pieces with bounded aspect ratios. Avoiding thin angles, large aspect ratios, or other kinds of ill behavior is the main focus of the work in [35, 207].

There is also a tradeoff involved in determining how many Steiner points (i.e., new vertices) can or should be added to the object. Steiner points are a nuisance in practice because they make representations more complicated and cause round-off errors. Unfortunately, they can rarely be avoided. To make matters worse, determining whether they are needed is NP-hard, as was shown by Ruppert and Seidel [238].

To subdivide a polygon into as few convex pieces as possible can be done in polynomial time, but most variants or extensions of the problem (e.g., polygons with holes, covering instead of partitioning) are NP-hard – see [222] for many references on this subject. The author showed that an  $n$ -vertex 3-polytope can always be subdivided into  $O(n^2)$  convex pieces and that this bound is optimal in the worst case [51]. Refinements of this result can be found in [19, 20, 24, 72, 76]. An interesting open problem is to find the complexity of approximating a minimum-size decomposition within, say, a constant factor. Similar questions can be asked involving the other standard parameters (e.g., aspect ratio, angles).

**Intersection Problems.** In two dimensions the archetypical problem is this: given  $n$  line segments, compute all its  $k$  pairwise intersections. In 1979 Bentley

and Ottmann [31] used the by-now classical plane-sweep paradigm to solve the problem in  $O(n \log n + k \log k)$  time.<sup>2</sup> To remove the  $\log k$  factor took some effort, and it was not until 1988 that an optimal  $O(n \log n + k)$  algorithm was discovered (Chazelle and Edelsbrunner [60]). Simpler probabilistic algorithms with optimal expected complexity were given by Clarkson and Shor [88] and Mulmuley [210, 213, 219].

In three dimensions, a natural operation is to intersect two convex polyhedra. Muller and Preparata [209] gave an  $O(n \log n)$  algorithm in 1978, but an optimal  $O(n)$  algorithm was found only much later, in 1989, by the author [56]. The method makes extensive use of a versatile polyhedral hierarchy developed by Dobkin and Kirkpatrick [106].

If all convexity assumptions are removed, only slightly subquadratic overhead can be achieved at this point, i.e.,  $O(n^{8/5+\epsilon} + k \log k)$ , where  $k$  is the number of intersected edges and  $\epsilon > 0$  is any fixed constant (Pellegrini [226]), and there is probably much room for progress. The case where one of the polyhedra is convex is treated by Mehlhorn and Simon [199] and requires  $O((n+k) \log n)$  time. It is likely that further improvement will require an increased understanding of lines in 3-space and ray shooting (see next section). An intriguing open question is whether a convex  $n$ -vertex 3-polytope can be preprocessed in linear space, so that its intersection with a plane can be computed in time  $O(k + \log n)$ , where  $k$  is the size of the cross-section.

**Convex Hulls and Voronoi Diagrams.** I lump these two notions together because, as was shown by Edelsbrunner and Seidel [121], the Voronoi diagram of a set of points in  $E^d$  is isomorphic to (part of) the convex hull of an appropriate lift of the points into  $E^{d+1}$ . In other words, Voronoi diagrams are special cases of convex hulls. Computing the convex hull of  $n$  points in two and three dimensions in  $O(n \log n)$  time can be done by using standard algorithmic tools (Graham [149], Preparata and Hong [229]). For Voronoi diagrams in two dimensions a simple plane-sweep method was developed by Fortune [127]; see also Fortune’s survey [130].

Things get considerably more difficult in higher dimensions, i.e.,  $d > 3$ . The full facial complexity of the hull is  $\Theta(n^{\lfloor d/2 \rfloor})$  in the worst case. Seidel [244] gave an optimal  $O(n^{\lfloor d/2 \rfloor})$ -time algorithm for the case of even dimension  $d$ . An  $O(n^{\lfloor d/2 \rfloor} \log n)$  solution that fell slightly short of optimal was also discovered by Seidel [245]. A remarkable Las Vegas algorithm with optimal expected complexity was found by Clarkson and Shor [88], and a simpler variant was proposed by Sei-

<sup>2</sup>Space sweeping is the algorithmic version of Morse theory; it is one of the most popular techniques in computational geometry.

del in [246]. Finally, an optimal  $O(n^{\lfloor d/2 \rfloor})$  deterministic algorithm was found by the author [58]. Interestingly, the method is based on derandomizing the Clarkson-Shor algorithm. The tools used for the derandomization are fairly involved, however (see Section 6). The technique was recently simplified somewhat and generalized by Brönnimann, Chazelle, and Matoušek [42].

Optimal algorithms now exist, but only in the worst-case sense. An interesting open problem is to find an optimal *output-sensitive* algorithm<sup>3</sup>. The best known general solution is due to Seidel [245]: Its running time is  $O(n^2 + h \log h)$ , where  $h$  is the facial complexity of the hull. As was shown by Matoušek [189], the quadratic overhead can be slightly reduced to  $O(n^{2 - \frac{2}{\lfloor d/2 \rfloor + 1} + \epsilon})$ , for any fixed  $\epsilon > 0$ . Whether the complexity can be reduced to the lower bound of  $\Omega(h + n \log h)$  is an outstanding open problem. Currently, the only optimal output-sensitive algorithms are due to Kirkpatrick and Seidel [177] (deterministic) in two dimensions and Clarkson and Shor [88] (Las Vegas) in three dimensions. The latter algorithm was derandomized optimally by Chazelle and Matoušek [71].

There is a vast amount of literature on Voronoi diagrams. Far-reaching generalizations of the basic notion have been developed over the years: for example, *abstract Voronoi diagrams* and *power diagrams* are treated thoroughly by Klein [178] and Aurenhammer [22], respectively. I highly recommend the comprehensive survey of Voronoi diagrams and their applications by Aurenhammer [23].

### 3 Geometric Searching

Hit detection and ray tracing in computer graphics, range searching in databases, point location in geographical maps are all classical examples of geometric searching for which efficient, preferably dynamic, solutions are sought. We quickly review the state of the art and mention possible directions for future research.

**Point Location.** In its static version, the problem of locating a query point in a planar map is completely solved [92, 119, 175, 180, 240]: It is possible to store a planar subdivision with linear storage, so that given any point, finding the region that encloses it can be done in logarithmic time.

If insertions and deletions of facial features are allowed, however, an optimal  $O(\log n)$  query/update time has been elusive. Many solutions exist, but all suffer from extra logarithmic factors in one or several of the

relevant resource measures (query, update time, storage) [78, 79, 147, 231]. As was shown in [232] an efficient solution in the plane would have impact on the static three-dimensional case. An interesting line of attack is to dynamize randomized incremental methods. Very little work has been done without stochastic assumptions on the sequencing of dynamic operations. A notable exception is Mulmuley's recent work on randomized multidimensional search trees [215, 216, 217], which might contain useful ideas to solve the original problem.

In higher dimensions, searching among  $n$  hyperplanes in  $\mathbf{R}^d$  means locating a query point in the arrangement formed by the hyperplanes. A solution due to Clarkson [81] requires  $O(n^{d+\epsilon})$  space and  $O(\log n)$  query time. The storage was reduced to  $O(n^d)$  by the author [57]. Probabilistic solutions are described in Mulmuley's book [219].

**Visibility and Form Factors.** Ray-shooting in the plane is now well understood [2, 6, 61, 68, 164]: given a simple polygon with  $n$  vertices, it is possible to shoot a ray in any direction and find the first point that it hits in  $O(\log n)$  time, using only  $O(n)$  space. The dynamic case was tackled recently by Goodrich and Tamassia [148]. The three-dimensional case is still wide open, however, despite some progress reported in [63, 95, 154, 192]. Because the problem is akin to ray tracing in computer graphics [139] its practical importance is considerable.

Agarwal and Matoušek [5] have shown how to answer a query in time  $O(n^{1+\epsilon}/m^{1/4})$ , using  $O(m^{1+\epsilon})$  space, for any  $n \leq m \leq n^4$ , which improved on earlier work [6]. Further improvement is likely to require a better computational understanding of the Plücker surface for lines in 3-space. The work in [63, 226] contains preliminary steps in that direction. Generally speaking, however, it seems fair to say that computing with Grassmannian coordinates has been grossly under-researched. Finally, I should mention the interesting use of parametric search for ray shooting discussed by Agarwal and Matoušek [4].

Visibility is a central issue in the rendering problem. Current state-of-the-art rendering techniques use radiosity methods to trace light in order to provide the most realistic rendering of a scene with lighting [91]. At the inner loop of these methods is the computation of form factors. Informally, a form factor is defined for a pair of atomic objects (typically, triangles) as the fraction of one object visible to the other through the obstacles in the scene. Fast, practical methods for computing form factors would have a great impact in rendering.

**Range Searching.** The general setting is this: we are given a collection of (usually parameterized) regions in space, called ranges, and a finite set of points. The

<sup>3</sup>Output-sensitive means that the running time is a function of the input size as well as the actual (as opposed to worst-case) output size.



problem is to store the points in some data structure so that given any range, the points that it contains can be reported or counted efficiently. By and large the classical instances of that problem have been solved satisfactorily. For example, simplex range searching on  $n$  points in  $\mathbf{R}^d$  can be solved in space  $m$  and query time  $O(n/m^{1/d})$  (up to logarithmic factors), where  $m$  is an adjustable parameter. This means that  $n$  points in  $\mathbf{R}^d$  can be stored by using  $m$  memory units, so that given any simplex the number of points that it encloses can be determined within the indicated query time. As it turns out, this result is nearly optimal (see Section 5).

An earlier, quasi-optimal solution by Chazelle, Sharir, and Welzl [75] was improved by Matoušek [187], who introduced the powerful concept of simplicial partitions by extending the earlier notions of partition trees [267] and low-stabbing spanning trees [77, 264]. See also Matoušek [186]. A nice survey by Welzl on the subject of spanning trees with low stabbing numbers appears in [265].

Halfspace range searching (which is the special case of simplex range searching where the query range is a halfspace) arises in many problems, in some cases quite unexpectedly. A general linearization technique of Yao and Yao [272] shows that many range searching problems that are expressed by high-degree polynomials can be reduced to halfspace range searching by lifting them into higher-dimensional space. An alternative treatment of generalized range searching (where the ranges are arbitrary semi-algebraic sets) was provided by Agarwal and Matoušek [5].

Efficient solutions for the reporting case of halfspace range searching (where the points must be explicitly enumerated as opposed to just counted) were proposed by Clarkson [82] and Matoušek [188]: the first solution provides  $O(\log n + k)$  query time for  $O(n^{\lfloor d/2 \rfloor + \epsilon})$  storage, where  $n$  is the number of points and  $k$  is the size of the output; the second method has a query time of roughly  $O(n^{1-1/\lfloor d/2 \rfloor})$  for linear storage. Two general techniques, *filtering search* and *fractional cascading*, have proven quite useful in speeding up range reporting algorithms. The first one, developed by the author [52], exploits the simple idea that if the output is large then the algorithm can slow down proportionately, since the output size can be used to amortize the search cost; this in turn leads to savings in storage. The second technique, developed jointly with Guibas [69], deals with the problem of searching for the same key in several dictionaries at once: auxiliary dictionaries are created that allow instant transitions from one dictionary to the next.

In closing this section, I will mention the comprehensive survey on range searching by Matoušek [190], which I highly recommend.

## 4 Geometric Optimization

I now briefly discuss recent developments that have had an important impact in geometric optimization. One is parametric search; the other concerns fixed-dimensional linear programming.

**Parametric Search.** Almost a decade ago, Megiddo devised a beautiful algorithmic paradigm known as *parametric search* [195, 196]. A clever improvement which under some conditions speeds up the method was proposed by Cole [93]. The main feature of parametric search is to turn a *checking* algorithm for an optimization problem into one that actually *searches* for a solution. Certain conditions must be satisfied, of course, but the setting is general enough to have wide applicability. For example, it can be used to solve the 2-center problem in the plane, selecting slopes or distances, computing the  $k$ th leftmost vertex in a line arrangement, perform ray-shooting in 3-space, etc. See, e.g., [4, 8, 65, 94].

One of the main conditions of Megiddo's technique is that the checking algorithm be parallelizable. This condition can be somewhat weakened, but actually it would appear that in a geometric setting the requirement should be reducible to nothing but a mild form of multidimensional searching.

An interesting open problem is computing the diameter of  $n$  points in  $E^3$  (i.e., the distance between its two furthest points). Clarkson and Shor [88] gave a randomized algorithm with an optimal expected complexity of  $O(n \log n)$ . Attempts to derandomize it have been unsuccessful so far. Building on previous work [58, 65, 191, 235], Brönnimann, Chazelle, and Matoušek [42] gave an  $O(n \log^3 n)$ -time deterministic solution. It is currently the best known but it is unlikely to be optimal. The algorithm uses parametric search as well as the entire derandomization machinery of [58]: one wonders whether the diameter problem really requires all these hybrid complicated techniques.

We close our discussion of parametric searching by mentioning an important generalization of the method, where the parameter to optimize is not a single value but a point in higher-dimensional space [90, 96, 189, 221].

**Linear Programming.** In 1984 Megiddo [197] gave the first linear-time algorithm for linear programming with a fixed number of variables. The running time of his algorithm is  $O(c_d n)$ , with  $c_d = 2^{2^d}$ , where  $n$  is the number of constraints and  $d$  is the number of variables (the dimension of the ambient space). The constant  $c_d$  was improved to  $3^{d^2}$  by Clarkson [80] and Dyer [111]. Still lower dependency on  $d$  was achieved by randomized methods (Dyer and Frieze [113], Clarkson [83], Seidel [246]), the current best being Clarkson's algorithm, with

an expected complexity of  $O(d^2n + d^{d/2+O(1)} \log n)$ .

Kalai [171] and then independently Matoušek, Sharir, and Welzl [193] developed algorithms with a subexponential dependency on both  $n$  and  $d$ . These (dual) simplex-like methods rekindle the hope that a strongly polynomial-time algorithm for general linear programming might exist. Combining these methods with Clarkson's yields a randomized algorithm for linear programming with expected running time  $O(d^2n + e^{O(\sqrt{d} \ln d)})$ . To derandomize these algorithms appears far from obvious. The best current deterministic solutions for linear programming in fixed dimension have complexity  $d^{O(d)}n$  [9, 70].

An abstract framework for optimization problems similar to linear programming but much more general was introduced by Sharir and Welzl [253] and Matoušek, Sharir, and Welzl [193]. Similar techniques to those used in the solutions we mentioned above can be made to bear on these *LP-type* problems. Gärtner [137] was able to derive a subexponential method for a subclass of these problems. Similarly, the method of Chazelle and Matoušek [70] can be used to solve in linear deterministic time problems such as computing the minimum-volume ellipsoid enclosing a set of points or finding the maximum volume ellipsoid within the feasible set of linear constraints. Solutions with higher dependency on the dimension were found earlier by Dyer [112].

## 5 Lower Bounds

Sadly enough, surveying the status of lower bounds in computational geometry is a fairly easy task. As long as the information-theoretic bound is good enough, there are nice tools at one's disposal. A proof technique by Yao [269] for proving that any algebraic decision tree for computing the convex hull of  $n$  points in the plane must have depth  $\Omega(n \log n)$  was generalized by Ben-Or [29] and can be applied to many problems. The case of integer inputs was treated by Yao [270].

If the information-theoretic bound is too weak, however, we are seriously out of luck. Range searching is one of the few exceptions to this rule, because it has a nice combinatorial characterization. In a range searching problem, think of each point as a variable over a semigroup. The answer to a query is the semigroup sum of the variables associated with the points falling in the query range.

In that model, originally introduced by Fredman and Yao, remarkably strong lower bounds can be proven. Fredman established optimal lower bounds for dynamic orthogonal range searching [134] and quasi-optimal ones for dynamic halfplane range searching [135]. Yao [268] established a lower bound for orthogonal range searching in two dimensions, while for arbitrary di-

mensions Vaidya [259] provided nontrivial but suboptimal lower bounds. The author obtained quasi-optimal lower bounds for orthogonal range reporting, orthogonal range searching, and simplex range searching [54, 53]. Other lower bounds were established by the author jointly with Rosenberg [73] (simplex range reporting) and Brönnimann and Pach [43] (halfspace range searching).

Unfortunately, if we allow subtractions, i.e., if we operate over a group as opposed to a semigroup, all these lower bounds collapse miserably. This is very similar to the difficulty of proving lower bounds for general circuits as opposed to monotone circuits. New ideas seem to be required. At this point, *any* nontrivial lower bound for *any* nontrivial (constructive) instance of range searching in the group model would be a major result.

The second challenge is to handle *computational* as opposed to combinatorial models. For example, none of the current methods can be used to derive nontrivial lower bounds for Hopcroft's problem: this is the problem of deciding whether, among  $n$  lines and  $n$  points, all the points are disjoint from all the lines. Several solutions exist that require roughly  $n^{4/3}$  time, the best of which is due to Matoušek [186], but  $\Omega(n \log n)$  is the only existing lower bound.

Problems of that sort can be formalized as follows: Given a  $k$ -variate polynomial  $P$  of fixed degree with rational coefficients, and given  $n$  numbers  $x_1, \dots, x_n$ , is there a  $k$ -tuple among the  $x_i$ 's that is in the zeroset of  $P$ ? There exists a trivial  $O(n^k)$  solution, to which slight improvements can be made by using fairly complicated batching techniques. But again the best known lower bound is only on the order of  $n \log n$ .

Björner, Lovász, and Yao [38] have derived lower bounds (for nongeometric problems) that make novel use of topological invariants and might provide useful tools. However, these tools in themselves are still inherently information-theoretic and therefore unlikely to break the  $n \log n$  barrier. Erickson and Seidel [125] have proved elegant lower bounds for problems such as testing if a set of points is in general position, but their model of computation is far too restrictive to give much hope that a satisfactory solution should be anywhere in sight.

## 6 Randomization and Derandomization

Randomization has had a tremendously positive effect in computational geometry. Not only has it led to a host of fairly simple, efficient probabilistic algorithms, but it has led to a complete rethinking in the design of deterministic algorithms through the process of derandomization. Perhaps this phenomenon is best illustrated by

the convex hull problem discussed in Section 2.

After several years without noticeable progress on the problem, Clarkson and Shor [88] produced an optimal randomized algorithm. Around the same time through the efforts of several researchers, in particular Matoušek [183, 184, 185, 186], a beautiful theory of deterministic sampling in range spaces of finite VC-dimension was being developed. Also, general derandomization techniques began to emerge, such as the method of conditional probabilities of Raghavan and Spencer [234, 254]. It was only with the help of all these powerful tools that the author was able to derive an optimal convex hull algorithm [58]. It is one of a growing number of deterministic algorithms that literally cannot be understood without referring to their probabilistic counterparts. Unfortunately, in this case, derandomization exacts a hefty price, as the algorithm (and especially, its analysis) are quite complicated. If history is a guide, however, one should expect much simpler versions to emerge in the near future.

One of the main ingredients of the algorithm is a method of *deterministic Monte Carlo integration*, using  $\varepsilon$ -nets and  $\varepsilon$ -approximations (see definitions below) as tools for estimating functions that are too hard to evaluate exactly. It is a fascinating open question whether this can be extended to other problems, such as higher-order Voronoi diagrams or even line segment intersection.

The papers of Clarkson [81], Clarkson and Shor [88], and Haussler and Welzl [160], all concerned with probabilistic geometric algorithms, proved to be enormously influential. Two main classes of algorithms emerged: some use divide-and-conquer via random sampling; others, called *randomized incremental constructions*, use insertions in random order. For a small random sample of papers relevant to the first class, see [81, 82, 87, 88, 89, 116, 117, 160, 236]. I also recommend the surveys by Clarkson [85] and Seidel [248], as well as Mulmuley's textbook [219] for excellent introductions to the use of random sampling in computational geometry.

A key notion in the analysis of probabilistic geometric algorithms is that of range spaces of finite VC-dimension. Let  $\Sigma = (X, \mathcal{R})$  be a set system. Given  $Y \subseteq X$ , let  $\mathcal{R}|_Y$  denote the set system  $\{R \cap Y; R \in \mathcal{R}\}$ . We say that  $Y$  is *shattered* (by  $\mathcal{R}$ ) if  $\mathcal{R}|_Y = 2^Y$ . The maximum size of a shattered subset of  $X$  is called the *Vapnik-Chervonenkis dimension*, or VC-dimension, for short, of the set system  $\Sigma$ . Given  $\varepsilon \in (0, 1)$ , a subset  $A \subseteq X$  is an  $\varepsilon$ -approximation for  $\Sigma$ , if

$$\left| \frac{|A \cap R|}{|A|} - \frac{|R|}{|X|} \right| \leq \varepsilon$$

for every set  $R \in \mathcal{R}$ . A subset  $S \subseteq X$  is called an  $\varepsilon$ -net for  $(X, \mathcal{R})$  provided that  $S \cap R \neq \emptyset$  for every set  $R \in \mathcal{R}$  with  $|R|/|X| > \varepsilon$ . Remarkably, as long as it has

finite VC-dimension,  $\Sigma$  admits  $\varepsilon$ -approximations and  $\varepsilon$ -nets whose sizes are independent of  $|X|$  (Vapnik and Chervonenkis [260], Haussler and Welzl [160]).

The question of constructing  $\varepsilon$ -nets deterministically was investigated by Chazelle and Friedman [67], who gave polynomial-time algorithms for computing an  $\varepsilon$ -net of size  $O(\varepsilon^{-1} \log \varepsilon^{-1})$ . A major advance was subsequently made by Matoušek [185], who showed that under some natural computational assumptions about the set system  $\Sigma$ , it is possible to compute  $\varepsilon$ -approximations and  $\varepsilon$ -nets of size  $O(\varepsilon^{-2} \log \varepsilon^{-1})$  and  $O(\varepsilon^{-1} \log \varepsilon^{-1})$ , respectively, in time  $O(\varepsilon^{-2d} \log^d \varepsilon^{-1})|X|$ . A simpler proof was given by Chazelle and Matoušek [70] and slightly better running times were achieved by Brönnimann, Chazelle, and Matoušek [42] by introducing the notion of *sensitive  $\varepsilon$ -approximations*.

With the use of these tools (among others) derandomizing divide-and-conquer type algorithms has been very successful. But lately, the attention has shifted towards randomized incremental constructions [40, 41, 66, 87, 88, 89, 102, 103, 153, 210, 211, 212, 213, 214, 246, 247]. These algorithms examine the input elements in random order and at each step maintain a partial solution based on the portion of input examined so far. This approach has produced surprisingly simple Las Vegas algorithms of low expected complexity. Unfortunately, their derandomization has proven elusive so far. Actually, the author's convex hull algorithm [58] is currently the only successful derandomization of a randomized incremental construction.

To understand the computational power of randomness has been a central preoccupation in theoretical computer science. While only few general derandomization techniques have been discovered [12, 15, 182, 234, 254], several clever schemes have been found to reduce the amount of randomness needed by probabilistic algorithms [14, 28, 32, 33, 170, 208, 220].

As was shown by Mulmuley [218], such techniques can be used to prove that most randomized geometric algorithms require few (truly) random bits; but still too many, however, to allow efficient deterministic simulations. Interesting geometric use of expanders (which act as quasi-random graphs) has been made by Ajtai and Megiddo [11] and Katz and Sharir [174].

Matoušek, Welzl, and Wernisch [194] established that the effectiveness of  $\varepsilon$ -nets and  $\varepsilon$ -approximations for divide-and-conquer in computational geometry is closely tied to the discrepancy of certain set systems. A better understanding of the relationship between discrepancy and sampling might hold the key to understanding what makes derandomization feasible or not.

## 7 Computational Real-Algebraic Geometry

Planning the motion of a robot amidst obstacles usually boils down to asking connectivity questions about semi-algebraic sets. Although reasonably efficient methods are already known for answering such questions, the larger picture is still blurred. For example, one would often like to navigate in configuration space in real-time, perform point location and ray-shooting, modify the constraints on-line, etc. Also, it might be desirable to obtain more information about the cycle structure besides path-connectivity: for example, one might want to compute the full homology of a cell decomposition (see next section). Although much progress has been done in that general area since Collins' pioneering work [97], e.g., Schwartz and Sharir [242], Canny [44], two central questions remain open: (1) triangulating real-algebraic varieties in singly-exponential time, and (2) keeping polynomial coefficients below doubly exponential. Here is a brief explanation:

Given  $n$  polynomials in  $\mathbf{Q}[x_1, \dots, x_d]$ , we wish to partition  $\mathbf{R}^d$  into simple cells over which each polynomial remains sign-invariant. A simple cell could be, for example, a smooth manifold (i) topologically equivalent to a  $k$ -ball ( $k \leq d$ ) and (ii) specified by a constant-size sentence in the theory of the reals. This is the motivation behind a particular type of decomposition known as a *stratification* [44, 64, 74, 97, 233, 242, 266]. Property (ii) is used to ensure that the cells of the decomposition are computationally "tractable."

Informally a triangulation is a special case of a stratification where adjacent faces glue "nicely" to each other, i.e., in a lattice-like fashion. This added property is crucial for, say, navigating in configuration space, e.g., moving a robot amidst obstacles: in configuration space the robot is a point in a space of dimension equal to its degree of freedom and the obstacles are modeled by (usually) algebraic constraints.

As shown by Chazelle and Sharir [74], an economical sign-invariant stratification yields more efficient algorithms for problems as diverse as computing the longest segment fitting inside a polygon, finding the minimum vertical separation between two sets of segments in  $\mathbf{R}^3$ , computing the first time the convex hull of a set of moving objects reaches its stationary configuration, etc. Another motivation for studying this problem is that its language is powerful enough to express *any* multidimensional searching problem expressed as a first-order predicate in the theory of reals.

An effective algorithm for constructing a sign-invariant stratification of size roughly  $O(n^{2d-3})$  was given by Chazelle et al. [64]. Although the algorithm represents an attractive alternative to Collins' decom-

position (whose size is doubly exponential in  $d$ ), it raises three major open questions: One is to reduce the size to  $O(n^d)$ , which by the Thom-Milnor bound, would then be optimal. (We omit to mention the dependency on the size and degree of the polynomials.) Another is to reduce the complexity of the algebraic numbers used to describe the cells from doubly to singly exponential (in the dimension). Although it is known that eliminating quantifiers is inherently doubly exponential (Davenport and Heintz [100], Weispfenning [263]), many restricted problems related to the theory of the reals can be solved in singly-exponential time.

The mathematical landscape in real-algebraic geometry has changed dramatically in the last few years. An eminently readable survey by Heintz, Recio and Roy is given in [163]. A good introduction to the (non-computational part of the) field is the text (in French) by Bochnak, Coste, and Roy [39]. Let us mention a few examples of the recent progress. A very useful lemma by Grigor'ev and Vorobjov [152] gives an upper bound on the size of a ball about the origin that misses any component of a semi-algebraic set (as a function of the number of polynomials, the maximum degree, the coefficient sizes, and the number of variables). This was an important tool used by Goodman, Pollack, and Sturmfels [142] to give sharp estimates on how large a grid must be if it is to accommodate all point configurations of a given size (up to order type).

Pollack and Roy [228] showed that the number of connected components of all non-empty realizations of sign conditions of  $n$  polynomials in  $d$  variables, each of degree at most  $b$ , is bounded by  $O(nb/d)^d$ , thus improving on an earlier bound based on a result of Heintz [161]. (Note that the coefficients need not be in  $\mathbf{Q}$  but in any real-closed field). See also Basu, Pollack, and Roy [27] for a more general result concerning the cross-section of a semi-algebraic set with a given variety.

In [26] Basu, Pollack, and Roy give an algorithm that returns a small set containing an algebraic point in each connected component of each non-empty sign condition in time  $n(n/d)^d b^{O(d)}$  – see also Canny [47, 48, 49]: this improves on the  $(nb)^{O(d)}$  algorithms of Canny [45] and Renegar [237].

Since Collins' work [97], important results in the general (i.e., fully quantified) first-order theory of the reals have been obtained. The works of Ben-Or, Kozen, and Reif [30] and Fitchas, Galligo, and Morgenstern [126] led to an interesting arithmetic-operation counterpart to Collins' bit-operation construction. Grigor'ev [151] discovered a method that is doubly exponential only in the number of quantifier alternations. The best complexity in both the arithmetic and bit models was achieved by Renegar [237]. Another quantifier elimination method was also given by Heintz, Roy, and Solernó [162]. Further improvements over both methods are reported by



Basu, Pollack, and Roy [25].

Somewhat outside the scope of this discussion, the theory of *Groebner bases* ought to be at least mentioned, because it is a very active area in computational algebraic geometry with many applications in geometric modeling. A good introduction can be found in the book by Cox, Little, and O'Shea [98]. A more in-depth treatment is provided by Mishra [206].

## 8 Computational Topology

The purpose of the subject is to study computational questions regarding the classical topological invariants, e.g., fundamental group, homology groups. Some of these problems, like the word problem for automatic groups, are very interesting from an algorithmic and data structuring viewpoint, but they are not germane to computational geometry. It is a different matter with homology theory, however. Computing homology groups can be done by algebraic methods, but it is often highly inefficient. Exploiting the specific geometry of the manifolds can lead to much faster algorithms. Similarly, checking homotopy equivalence between curves or homeomorphisms between surfaces can gain a lot from a computational-geometric approach.

**Homology Groups.** The connectivity properties of a geometric structure can reveal a lot of information about it, but they are usually much more subtle and difficult to compute than for combinatorial structures. The homology groups describe the cycle structure of a manifold. Their main advantage over the fundamental group or higher-dimensional homotopy groups is that they lend themselves to effective computations. It would be extremely useful to be able to compute homology groups efficiently for low-dimensional geometric cell complexes. There are many important applications in pattern recognition and classification in biology and chemistry as well as in robotics and scene analysis [99]. Comparing homologies is a good (if not foolproof) test to rule out topological equivalence: two manifolds with different homologies cannot be homeomorphic; unfortunately, the converse is not true.

There are also fascinating applications to the classification of time series and the analysis of dynamical systems (Freedman [136]). Spectral analysis has long been the standard tool for classifying time series. In the absence of sharp harmonics, the signal is dismissed as noise. Analysis for nonlinear dynamics can be done, however, by computing topological invariants of the attractor set for the system being measured. This can be done by mapping the time series into  $\mathbf{R}^d$  by scanning a "comb" of length  $d$  across the series and thinking of each  $d$ -tuple of entries at the teeth as a point in  $\mathbf{R}^d$ .

(This is the time-lag imbedding of the series; in practice the imbedding dimension  $d$  is often less than 10.) By grouping all  $(k + 1)$ -tuples of points at a distance less than  $\varepsilon$  from one another, we can define  $k$ -simplices. By introducing the obvious boundary operators, this yields a complex whose (Čech) homology is called the  $\varepsilon$ -homology of the set. Because Čech homology commutes with inverse limits, this is an appropriate methods for computing the homology of the attractor. Moreover the use of  $\varepsilon$ -homology (where simplices are defined only in terms of pairwise distances) should have computational advantages. A similar notion, called *alpha-shape*, has been developed by Edelsbrunner. This is a very interesting area to explore where computational-geometric methods can be brought to bear.

Schwartz and Sharir [242] gave a general method for computing the homology of a real-algebraic variety, but unfortunately it is doubly exponential in the dimension of the ambient space. More recently, Donald and Chang [109] proposed a randomized algorithm for computing the homology groups of a cell complex, which takes advantage of the sparsity of its incidence structure. By and large, however, the problem has been treated in full generality, with little focus on the specifically geometric nature of the complex. A notable exception is the work of Delfinado and Edelsbrunner [101], where a nearly optimal method for computing the betti numbers of a subcomplex of a triangulation of  $S^3$  is given. (Recall that these particular homology groups are free, i.e., without torsion, so they are specified entirely by the betti numbers). This result has interesting applications to the so-called alpha-shapes mentioned earlier, which are generalizations of convex hulls used in geometric modeling.

Mayer-Vietoris sequences should be useful in designing efficient divide-and-conquer schemes for computing the homology of higher-dimensional complexes [101, 168, 239]. This is an exciting open area of research. The goal is, in a somewhat narrow sense, to do computational linear algebra by geometric means.

Edelsbrunner [115] has given efficient algorithms for computing topological, combinatorial, and metric properties of the union of finitely many balls in  $\mathbf{R}^d$ . These algorithms are based on short inclusion-exclusion formulas derived from a complex dual to a decomposition of the balls. These methods have applications to biology and chemistry, where unions of balls are commonly used as models of molecules. Further work on triangulating topological spaces has been reported by Edelsbrunner and Shah [123].

**Topology of Curves.** Interesting questions in two-dimensional computational topology have been investigated recently. Mehlhorn and Yap [200] and Vegter [261] considered an algorithmic discrete version of

the Whitney-Graustein theorem, which states that up to kink-free deformations a regular closed curve in the plane is completely determined by its winding number. In particular, Vegter gives a linear algorithm for deforming a regular polygon into any other regular polygon with the same winding number.

Motivated by the desire to be able to decide quickly whether two surfaces are homeomorphic or whether curves on them are homotopic, Vegter and Yap [262] looked at the classical question of putting in canonical form the polygonal schema of a closed triangulated surface. They showed how to implement the Brahana procedure (of the Dehn-Heegaard classification theorem) in  $O(n \log n)$  time, where  $n$  is the size of the triangulation.

In [241] Schipper presented an algorithm for deciding whether a curve on a triangulated surface is null-homotopic, i.e., contractible to a point. A similar question regarding the detection of null-homologous cycles on surfaces was investigated by Dey [105]. See also [258].

## 9 Robustness and Degeneracies

There are two kinds of numerical difficulties in implementing geometric algorithms: finite precision causes round-off errors or inconsistent results, while often more brutal, degeneracies or near-degeneracies cause programs to crash. A program is usually termed *robust* if it is immune to these ills (as much as possible). A growing research effort is underway to provide the programmer and algorithm designer with tools, guidelines, and options to deal with these difficulties. Fortune has an excellent survey [131] on the subject to which the reader is referred. Hoffmann's book [165] is also a good source. I will briefly sketch some of the issues and lines of research in that area.

**Finite Precision.** A typical atomic geometric operation is to determine whether three points make a left turn or if a fourth one lies inside the circle passing through the first three. In general, there is a certain determinant whose sign must be determined. An interesting approach for doing that is given by Clarkson [84].

If we must use exact representation, one option is to use modular arithmetic. As Fortune notes in [131], fairly little seems to be known in a specifically geometric context. This is an interesting subject to explore.

Rational arithmetic is another option. Again, the absence of error is the major drawing card. An obvious problem, however, is that the representation length grows rather quickly. Karasick, Lieber, and Nackman [172] report some cases where appropriate optimization and careful use of interval arithmetic result in running times that are competitive with floating-point computation.

How general these techniques are, however, remains unclear.

Yap [275] has argued that exact geometric computation need not be hopelessly costly, and that the answer might be to enhance traditional BigNumber packages and customize them to geometric situations. See also Fortune and Van Wyk [133].

Away from exact computation, another route, followed by Greene and Yao [150], is to round newly computed points to the nearest point on a fine enough lattice. The usual incidence laws no longer apply, however, and to preserve some measure of consistency is no easy task.

A variant on the Greene-Yao rounding idea that increases its flexibility involves "fattening" the lines. Building on this idea, Milenkovic [204, 205] was able to develop low bit-complexity algorithms for fundamental geometric problems in higher dimensions. See also Milenkovic [203].

Sugihara and Iri [256] considered the problem of maintaining low-bit complexity in the face of solid-modeling operations: these are often nested computations which tend to blow up representation lengths.

In practice the user sometimes has no choice but to use a floating point representation. A standard approach is to add a third truth value, "don't-know", to the logic. Variants of interval analysis techniques (so-called epsilon-geometry) were used by Guibas, Salesin, and Stolfi [155]. Dobkin and Silver [107] suggested adding the use of randomization and sampling.

Backward analysis is another standard error analysis technique which has been adapted with success to some geometric situations. Fortune [128] redesigned the Graham scan method for computing planar convex hulls, so that the output is the true convex hull of an appropriate perturbation of the input points. A variant is described by Li and Milenkovic [179]. The complexity of computing a line arrangement robustly (by stretching lines into pseudolines when necessary) was analyzed by Milenkovic [201, 202] and Fortune and Milenkovic [132].

Other interesting work on maintaining consistency and small error using floating-point arithmetic appears in (Sugihara and Iri [255], Hoffmann, Hopcroft, and Karasick [166, 167], Karasick [173], Fortune [129]). The problem of performing rotations was addressed by Canny, Donald, and Ressler in [50].

**Degeneracies.** Perturbing the input data ever so slightly can sometimes ensure that it is in general position. As has long been known in linear programming, the perturbation can be made symbolic. This approach was adapted to a geometric context by Edelsbrunner and Mücke [120] and Yap [273, 274].

The overhead in using these techniques can be quite high. Emiris and Canny have proposed schemes to alle-

viate the cost of computing long sequences of determinants [124]. Two problems inherent to these methods are the lack of control over the perturbation and the lack of discrimination between coincidental and mathematically implied degeneracies; e.g., three input points might or might not be collinear, but they always are if one is the midpoint of the other two. We close this brief overview by mentioning an interesting investigation of perturbations techniques by Seidel [249].

## 10 Combinatorial Geometry

I will briefly mention a few developments in combinatorial geometry that have had a considerable impact in the analysis of geometric algorithms.

Thanks to the effort of several researchers, (notably Sharir), the theory of Davenport-Schinzel sequences is now fairly advanced [7, 159]. We omit the discussion of this subject and instead refer the reader to the forthcoming book by Sharir and Agarwal [252]. There have been numerous applications to motion planning, triangulation, optimization, etc.

Zone theorems have also played a prominent role lately. Given an arrangement of  $n$  hyperplanes in  $\mathbf{R}^d$ , the zone complexity of any hyperplane  $h$  is the total size of all the cells that  $h$  intersects. Edelsbrunner, Seidel, and Sharir [122] showed that the maximum zone complexity is  $O(n^{d-1})$ . Aronov, Pellegrini, and Sharir [18] generalized the proof technique to the case of the zone of an algebraic surface, while Aronov, Matoušek, and Sharir [17] bounded the mean square cell complexity.

A related question is to bound the sum of the complexities of  $m$  cells in an arrangement of  $n$  hyperplanes in  $\mathbf{R}^d$ . In the two-dimensional case, Clarkson et al. [86] prove that this complexity is at most  $O(n + m + m^{2/3}n^{2/3})$ , which is tight. An efficient algorithm for constructing the corresponding cells is given by Edelsbrunner, Guibas, and Sharir [117]. The higher-dimensional case is treated by Edelsbrunner, Guibas, and Sharir [118], Aronov, Matoušek, and Sharir [17], and Agarwal and Aronov [3].

In problems of placement or Voronoi diagrams-related questions, it is important to have estimates on the combinatorial complexity of the lower envelope of an arrangement of hyperplanes or surfaces: this is the locus of points visible from a point at infinity. This problem is easier than the zone problem, but unfortunately the cases of interest are those for which no zone theorems are available. However, Sharir [251] recently provided a quasi-optimal upper bound of  $O(n^{d-1+\epsilon})$ , for any fixed  $\epsilon > 0$ , on the size of the lower envelope of  $n$  surface patches in  $\mathbf{R}^d$  bounded by algebraic surfaces of constant maximum degree. Specializations to three dimensions were investigated earlier by Halperin and Sharir

[157, 158]. Many applications are discussed in [252].

We conclude by returning to geometric sampling and mentioning an intriguing extension of  $\epsilon$ -nets that was proposed by Alon et al. [13]. Given a finite point set  $P$ , they define a weak  $\epsilon$ -net (for convex sets) to be a collection  $S$  of points such that any convex set that contains a fraction  $\epsilon$  of the points in  $P$  must intersect  $S$ . Alon et al. proved that although the underlying range space does not have finite VC-dimension, such a set  $S$  exists (often outside of  $P$ ) such that the size  $|S|$  is polynomial in  $1/\epsilon$  and, most important, independent of  $|P|$ . The bound was slightly improved by Chazelle et al. [62] for the general case, and optimal bounds were found for special cases by using tilings of the hyperbolic plane. Nevertheless, the problem is still by and large wide open.

We close this section by mentioning the nice survey by Guibas and Sharir [156] and the forthcoming text on combinatorial geometry by Pach and Agarwal [224].

## 11 Concluding Remarks

Computational geometry is more active and diverse than ever. I have not touched on vast areas of research, e.g., motion planning [243], assembly, kinodynamics [110, 108], Groebner bases [98], transversal theory [143], parallel computational geometry [10, 145], hidden surface removal [146, 271], art-gallery theorems [222], Borsuk-Ulam type search (ham-sandwich cuts, centerpoints) [181], arrangements and configurations [140, 141],  $k$ -set theory [225], mesh generation [36, 37], geometric problems in computer vision [16, 169].

There exist several introductory textbooks in computational geometry: Edelsbrunner [114], Mehlhorn [198], Mulmuley [219], O'Rourke [223], and Shamos and Preparata [230].

Today, the field covers a pretty wide front and is still rapidly expanding. There is no way to tell what it will look like in the year 2000. Whatever the new millenium has in store for computational geometry, however, I am sure that it will continue to be vibrant and fun!

## Acknowledgments

I wish to thank Pankaj Agarwal, Hervé Brönnimann, Herbert Edelsbrunner, Jirka Matoušek, Ricky Pollack, Marie-Françoise Roy, Micha Sharir, and Emo Welzl for reading a draft of this survey and giving me many useful comments.

## References

- [1] Agarwal, P.K. *Partitioning arrangements of lines II: Applications*, Disc. Comput. Geom., 5 (1990), 533–573.
- [2] Agarwal, P.K. *Ray shooting and other applications of spanning trees with low stabbing number*, SIAM J. Comput., 22 (1992), 540–570.
- [3] Agarwal, P.K., Aronov, B. *Counting facets and incidences*, Disc. Comput. Geom., 7 (1992), 359–369.
- [4] Agarwal, P.K., Matoušek, J. *Ray shooting and parametric search*, SIAM J. Comput., 22 (1993), 794–806.
- [5] Agarwal, P.K., Matoušek, J. *Range searching with semialgebraic sets*, Disc. Comput. Geom. (1994), in press.
- [6] Agarwal, P.K., Sharir, M. *Application of a new space partitioning technique*, Disc. Comput. Geom., 9 (1993), 11–38.
- [7] Agarwal, P.K., Sharir, M., Shor, P. *Sharp upper and lower bounds on the length of general Davenport-Schinzel sequences*, J. Combin. Theory Ser. A, 52 (1989), 228–274.
- [8] Agarwal, P.K., Sharir, M., Toledo, S. *Applications of parametric searching in geometric optimization*, Proc. 3rd ACM-SIAM Symp. Disc. Alg. (1992), 72–82.
- [9] Agarwal, P.K., Sharir, M., Toledo, S. *An efficient multi-dimensional searching technique and its applications*, Tech. Rep. CS-1993-20, Duke University, 1993.
- [10] Aggarwal, A., Chazelle, B., Guibas, L.J., Ó'Dúnlaing, C., Yap, C.K. *Parallel computational geometry*, Algorithmica, 3 (1988), 293–327.
- [11] Ajtai, M., Megiddo, N. *A deterministic  $Poly(\log \log N)$ -time  $N$ -processor algorithm for linear programming in fixed dimension*, Proc. 24th Ann. ACM Symp. Theory Comput. (1992), 327–338.
- [12] Alon, N., Babai, L., Itai, A. *A fast and simple randomized algorithm for the maximal independent set problem*, J. Algorithms, 7 (1986), 567–583.
- [13] Alon, N., Bárány, I., Füredi, Z., Kleitman, D. *Point selections and weak  $\epsilon$ -nets for convex hulls*, Combinatorics, Probability and Computing, 3 (1992), 189–200.
- [14] Alon, N., Goldreich, O., Hastad, J., Peralta, R. *Simple constructions of almost  $k$ -wise independent random variables*, Random Structures & Algorithms, 3 (1992), 289–304.
- [15] Alon, N., Spencer, J.H. *The Probabilistic Method*, John Wiley & Sons, 1992.
- [16] Arkin, E.M., Chew, L. P., Huttenlocher, D. P., Kedem, K., Mitchell, J.S.B. *An efficiently computable metric for comparing polygonal shapes* Proc. 1st ACM-SIAM Symp. Disc. Alg. (1990), 129–137.
- [17] Aronov, B., Matoušek, J., Sharir, M. *On the sum of squares of cell complexities in hyperplane arrangements*, Proc. 7th Ann. Symp. Comput. Geom. (1991), 307–313. To appear in J. Combin. Theory Ser. A.
- [18] Aronov, B., Pellegrini, M., Sharir, M. *On the zone of a surface in a hyperplane arrangement*, Disc. Comput. Geom., 9 (1993), 177–186.
- [19] Aronov, B., Sharir, M. *Triangles in space or building (and analyzing) castles in the air*, Combinatorica, 10 (1990), 137–173.
- [20] Aronov, B., Sharir, M. *Castles in the air revisited*, Proc. 8th Ann. ACM Symp. Comput. Geom. (1992), 146–156. To appear in Disc. Comput. Geom.
- [21] Aronov, B., Sharir, M. *The union of convex polyhedra in three dimensions*, Proc. 34th Ann. IEEE Symp. Foundat. Comput. Sci. (1993), 518–527.
- [22] Aurenhammer, F. *Power diagrams: properties, algorithms and applications*, SIAM J. Comput., 16 (1987), 78–96.
- [23] Aurenhammer, F. *Voronoi diagrams: a survey of a fundamental geometric data structure*, ACM Comput. Surv., 23 (1991), 345–405.
- [24] Bajaj, C.L., Dey, T.K. *Convex decompositions of polyhedra and robustness*, SIAM J. Comput., 21 (1992), 339–364.
- [25] Basu, S., Pollack, R., Roy, M.-F. *On the combinatorial and algebraic complexity of quantifier elimination*, manuscript, 1994.
- [26] Basu, S., Pollack, R., Roy, M.-F. *A new algorithm to find a point in every cell defined by a family of polynomials*, in “Quantifier Elimination and Cylindrical Algebraic Decomposition”, ed. B. Caviness and J. Johnson, Springer-Verlag, to appear.
- [27] Basu, S., Pollack, R., Roy, M.-F. *On the number of cells defined by a family of polynomials on a variety*, Algorithmic foundations of robotics, AK Peters, ed. K.Y. Goldberg, D. Halperin, J.-C. Latombe, R.H. Wilson. To appear.
- [28] Beck, J. *An algorithmic approach to the Lovász local lemma. I*, Random Structures & Algorithms, 2 (1991), 343–365.
- [29] Ben-Or, M. *Lower bounds for algebraic computation trees*, Proc. 15th Ann. ACM Symp. Theory Comput. (1983), 80–86.
- [30] Ben-Or, M., Kozen, D., Reif, J. *The complexity of elementary algebra and geometry*, J. Comput. Syst. Sci., 32 (1986), 251–264.
- [31] Bentley, J.L., Ottmann, T.A. *Algorithms for reporting and counting geometric intersections*, IEEE Trans. Comput., C-28 (1979), 643–647.



- [32] Berger, B., Rompel, J. *Simulating  $(\log n)^c$ -wise independence in NC*, J. ACM, 38 (1991), 1028–1046.
- [33] Berger, B., Rompel, J., Shor, P. *Efficient NC algorithms for set cover with applications to learning and geometry*, Proc. 30th Ann. IEEE Symp. Foundat. Comput. Sci., (1989), 54–59.
- [34] Bern, M. *Compatible tetrahedralizations*, Proc. 9th Ann. ACM Symp. Comput. Geom. (1993), 281–288.
- [35] Bern, M., Dobkin, D., Eppstein, D. *Triangulating polygons without large angles*, Proc. 8th Ann. ACM Symp. Comput. Geom. (1992), 222–231.
- [36] Bern, M., Eppstein, D. *Mesh generation and optimal triangulation*, in: Computing in Euclidean Geometry, 1, World Scientific, ed. D. Z. Du and F. K. Hwang (1992), 23–90.
- [37] Bern, M., Eppstein, D., Gilbert, J. *Provably good mesh generation*, Proc. 31st Ann. IEEE Symp. Foundat. Comput. Sci. (1990), 231–241.
- [38] Björner, A., Lovász, L., Yao, A.C. *Linear decision trees: volume estimates and topological bounds*, Proc. 24th Ann. ACM Symp. Theory Comput. (1992), 170–177.
- [39] Bochnak, J., Coste, M., Roy, M.-F. *Géométrie algébrique réelle*, Springer Verlag, Heidelberg, 1987.
- [40] Boissonnat, J.D., Devillers, O., Schott, R., Teillaud, M., Yvinec, M. *Applications of random sampling to online algorithms in computational geometry*, Disc. Comput. Geom., 8 (1992), 51–71.
- [41] Boissonnat, J.D., Teillaud, M. *On the randomized construction of the Delaunay tree*, Theoret. Comput. Sci., 112 (1993), 339–354.
- [42] Brönnimann, H., Chazelle, B., Matoušek, J. *Product range spaces, sensitive sampling, and derandomization*, Proc. 34th Ann. IEEE Symp. Foundat. Comput. Sci. (1993), 400–409.
- [43] Brönnimann, H., Chazelle, B., Pach, J. *How hard is halfspace range searching?*, Disc. Comput. Geom., 10 (1993), 143–155.
- [44] Canny, J. *The Complexity of Robot Motion Planning*, MIT Press, 1987.
- [45] Canny, J. *Some algebraic and geometric computations in PSPACE*, Proc. 20th Ann. ACM Symp. Theory Comput. (1988), 460–467.
- [46] Canny, J. *Generalized characteristic polynomials*, J. Symbolic Comput., 9 (1990), 241–250.
- [47] Canny, J. *Some Practical Tools for Algebraic Geometry*, Tech. Rep., Spring school on robot motion planning, Promotion Esprit, 1993.
- [48] Canny, J. *Computing road maps in general semi-algebraic sets*, The Computer Journal, 36 (1993), 504–514.
- [49] Canny, J. *Improved algorithms for sign determination and existential quantifier elimination*, The Computer Journal, 36 (1993), 409–418.
- [50] Canny, J., Donald, B.R., Ressler, G. *A rational rotation method for robust geometric algorithms*, Proc. 8th Ann. ACM Symp. Comput. Geom. (1992), 251–260.
- [51] Chazelle, B. *Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm*, SIAM J. Comput., 13 (1984), 488–507.
- [52] Chazelle, B. *Filtering search: A new approach to query-answering*, SIAM J. Comput., 15 (1986), 703–724.
- [53] Chazelle, B. *Lower bounds on the complexity of polytope range searching*, J. Amer. Math. Soc., 2 (1989), 637–666.
- [54] Chazelle, B. *Lower bounds for orthogonal range searching: I. the reporting case, II. the arithmetic model*, J. ACM, 37 (1990), 200–212 and 439–463.
- [55] Chazelle, B. *Triangulating a simple polygon in linear time*, Disc. Comput. Geom., 6 (1991), 485–524.
- [56] Chazelle, B. *An optimal algorithm for intersecting three-dimensional convex polyhedra*, SIAM J. Comput., 21 (1992), 671–696.
- [57] Chazelle, B. *Cutting hyperplanes for divide-and-conquer*, Disc. Comput. Geom., 9 (1993), 145–158.
- [58] Chazelle, B. *An optimal convex hull algorithm in any fixed dimension*, Disc. Comput. Geom., 10 (1993), 377–409.
- [59] Chazelle, B., Dobkin, D.P. *Optimal convex decompositions*, Computational Geometry, G.T. Toussaint, ed., North-Holland (1985), 63–133.
- [60] Chazelle, B., Edelsbrunner, H. *An optimal algorithm for intersecting line segments in the plane*, J. ACM 39 (1992), 1–54.
- [61] Chazelle, B., Edelsbrunner, H., Grigni, M., Guibas, L.J., Hershberger, J., Sharir, M., Snoeyink, J. *Ray shooting in polygons using geodesic triangulations*, Proc. 18th ICALP, LNCS, Springer-Verlag (1991), 661–673.
- [62] Chazelle, B., Edelsbrunner, H., Grigni, M., Guibas, L.J., Sharir, M., Welzl, E. *Improved bounds on weak  $\epsilon$ -nets for convex sets*, Proc. 25th Ann. ACM Symp. Theory of Comput. (1993).
- [63] Chazelle, B., Edelsbrunner, H., Guibas, L.J., Sharir, M. *Lines in space — combinatorics, algorithms and applications*, Proc. 21st Ann. ACM Symp. Theory of Comput. (1989), 382–393.

- [64] Chazelle, B., Edelsbrunner, H., Guibas, L.J., Sharir, M. *A singly-exponential stratification scheme for real semi-algebraic varieties and its applications*, Theoret. Comput. Sci., 84 (1991), 77–105.
- [65] Chazelle, B., Edelsbrunner, H., Guibas, L.J., Sharir, M. *Diameter, width, closest line pair, and parametric searching*, Disc. Comput. Geom., 10 (1993), 183–196.
- [66] Chazelle, B., Edelsbrunner, H., Guibas, L.J., Sharir, M., Snoeyink, J. *Computing a face in an arrangement of line segments and related problems*, SIAM J. Comput., 22 (1993), 1286–1302.
- [67] Chazelle, B., Friedman, J. *A deterministic view of random sampling and its use in geometry*, Combinatorica, 10 (1990), 229–249.
- [68] Chazelle, B., Guibas, L.J. *Visibility and intersection problems in plane geometry*, Disc. Comput. Geom., 4 (1989), 551–581.
- [69] Chazelle, B., Guibas, L.J. *Fractional cascading: I. A data structuring technique, II. Applications*, Algorithmica, 1 (1986), 133–162 and 163–191.
- [70] Chazelle, B., Matoušek, J. *On linear-time deterministic algorithms for optimization problems in fixed dimension*, Proc. 4th Ann. ACM-SIAM Symp. Disc. Alg. (1993), 281–290.
- [71] Chazelle, B., Matoušek, J. *Derandomizing an output-sensitive convex hull algorithm in three dimensions*, Comput. Geom.: Theory and Appl. (1994), to appear.
- [72] Chazelle, B., Palios, L. *Triangulating a nonconvex polytope*, Disc. Comput. Geom., 5 (1990), 505–526.
- [73] Chazelle, B., Rosenberg, B. *Lower bounds on the complexity of simplex range reporting on a pointer machine*, Proc. 19th ICALP, LNCS 623, Springer-Verlag (1992), 439–449.
- [74] Chazelle, B., Sharir, M. *An algorithm for generalized point location and its applications*, J. Symbolic Comput., 10 (1990), 281–309.
- [75] Chazelle, B., Sharir, M., Welzl, E. *Quasi-optimal upper bounds for simplex range searching and new zone theorems*, Algorithmica, 8 (1992), 407–429.
- [76] Chazelle, B., Shouraboura, N. *Bounds on the size of tetrahedralizations*, Proc. 10th Ann. ACM Symp. Comput. Geom. (1994), to appear.
- [77] Chazelle, B., Welzl, E. *Quasi-optimal range searching in spaces of finite VC-dimension*, Disc. Comput. Geom., 4 (1989), 467–489.
- [78] Cheng, S.W., Janardan, R. *New results on dynamic planar point location*, SIAM J. Comput., 21 (1992), 972–999.
- [79] Chiang, Y.J., Preparata, F.P., Tamassia, R. *A unified approach to dynamic point location, ray shooting, and shortest paths in planar maps*, Proc. 4th ACM-SIAM Symp. Disc. Alg. (1993), 44–53.
- [80] Clarkson, K.L. *Linear programming in  $O(n \times 3^{d^2})$  time*, Inf. Process. Lett., 22 (1986), 21–24.
- [81] Clarkson, K.L. *New applications of random sampling in computational geometry*, Disc. Comput. Geom., 2 (1987), 195–222.
- [82] Clarkson, K.L. *A randomized algorithm for closest-point queries*, SIAM J. Comput., 17 (1988), 830–847.
- [83] Clarkson, K.L. *Las Vegas algorithm for linear programming when the dimension is small*, Proc. 29th Ann. IEEE Symp. Foundat. Comput. Sci. (1988), 452–457.
- [84] Clarkson, K.L. *Safe and effective determinant evaluation*, Proc. 33rd Ann. IEEE Symp. Foundat. Comput. Sci. (1992), 387–395.
- [85] Clarkson, K.L. *Randomized geometric algorithms, in Computing in Euclidean Geometry*, D.-Z. Du and F.K. Kwang ed., Lecture Notes Series on Comput. 1 (1992), World Scientific, 117–162.
- [86] Clarkson, K.L., Edelsbrunner, H., Guibas, L. J., Sharir, M., Welzl, E. *Combinatorial complexity bounds for arrangements of curves and spheres*, Disc. Comput. Geom., 5 (1990), 99–160.
- [87] Clarkson, K.L., Mehlhorn, K., Seidel, R. *Four results on randomized incremental constructions*, Proc. 9th Symp. Theoret. Aspects Comput. Sci., LNCS 577, Springer-Verlag (1992), 463–474.
- [88] Clarkson, K.L., Shor, P.W. *Applications of random sampling in computational geometry, II*, Disc. Comput. Geom., 4 (1989), 387–421.
- [89] Clarkson, K.L., Tarjan, R.E., Van Wyk, C.J. *A fast Las Vegas algorithm for triangulating a simple polygon*, Disc. Comput. Geom., 4 (1989), 423–432.
- [90] Cohen, E., Megiddo, N. *Strongly polynomial-time and NC algorithms for detecting cycles in dynamic graphs*, Proc. 21st ACM Symp. Theory Comput. (1989), 523–534.
- [91] Cohen, M., Wallace, J., *Radiosity and Realistic Image Synthesis*, Academic Press, 1993.
- [92] Cole, R. *Searching and storing similar lists*, J. Algorithms, 7 (1986), 202–220.
- [93] Cole, R. *Slowing down sorting networks to obtain faster sorting algorithms*, J. ACM, 34 (1987), 200–208.
- [94] Cole, R., Salowe, J., Steiger, W., Szemerédi, E. *An optimal-time algorithm for slope selection*, SIAM J. Comput., 18 (1989), 792–810.

- [95] Cole, R., Sharir, M. *Visibility problems for polyhedral terrains*, J. Symbolic Comput., 7 (1989), 11–30.
- [96] Cole, R., Sharir, M., Yap, C.K. *On  $k$ -hulls and related problems*, SIAM J. Comput., 16 (1987), 61–77.
- [97] Collins, G.E. *Quantifier elimination for real closed fields by cylindric algebraic decomposition*, Proc. 2nd GI Conf. on Automata Theory and Formal Languages, Springer-Verlag, LNCS 35, Berlin (1975), 134–183.
- [98] Cox, D., Little, J., O’Shea, D. *Ideals, Varieties, and Algorithms*, Springer-Verlag, 1992.
- [99] Crapo, H., Ryan, J. *Scene analysis and geometric homology*, Proc. 2nd Ann. ACM Symp. Comput. Geom. (1986), 125–132.
- [100] Davenport, J. and Heintz, J. *Real quantifier elimination is doubly exponential*, J. Symbolic Comput., 5 (1988), 29–35.
- [101] Delfinado, C.J.A., Edelsbrunner, H. *An incremental algorithm for betti numbers of simplicial complexes*, Proc. 9th Ann. ACM Symp. Comput. Geom. (1993), 232–239.
- [102] Devillers, O. *Randomization yields simple  $O(n \log^* n)$  algorithms for difficult  $\omega(n)$  problems*, Int. J. Comput. Geom. Appl., 2 (1992), 97–111.
- [103] Devillers, O., Meiser, S., Teillaud, M. *Fully dynamic Delaunay triangulation in logarithmic expected time per operation*, Comput. Geom. Theory Appl., 2 (1992), 55–80.
- [104] Dey, T.K. *Triangulation and CSG representation of polyhedra with arbitrary genus*, Proc. 7th Ann. ACM Symp. Comput. Geom. (1991), 364–372.
- [105] Dey, T. *Optimal algorithms to detect null-homologous cycles on 2-manifolds*, Proc. 5th Canad. Conf. Comput. Geom. (1993), 273–278.
- [106] Dobkin, D.P., Kirkpatrick, D.G. *Fast detection of polyhedral intersection*, Theoret. Comput. Sci., 27 (1983), 241–253.
- [107] Dobkin, D.P., Silver, D., *Recipes for geometry and numerical analysis—part I: an empirical study*, Proc. 4th Ann. Symp. Comput. Geom. (1988), 93–105.
- [108] Donald, B.R. *A geometric approach to error detection and recovery for robot motion planning with uncertainty*, Artif. Intell., 37 (1988), 223–271.
- [109] Donald, B.R., Chang, D.R. *On the complexity of computing the homology type of a triangulation*, Proc. 32nd Ann. IEEE Symp. Foundat. Comput. Sci. (1991), 650–662.
- [110] Donald, B.R., Xavier, P., Canny, J., Reif, J. *On the complexity of kinodynamic planning*, J. ACM, 40 (1993), 1048–1066.
- [111] Dyer, M.E. *On a multidimensional search technique and its application to the Euclidean 1-centre problem*, SIAM J. Comput., 15 (1986), 725–738.
- [112] Dyer, M.E. *A class of convex programs with applications to computational geometry*, Proc. 8th Ann. ACM Symp. Comput. Geom. (1992), 9–15.
- [113] Dyer, M.E., Frieze, A.M. *A randomized algorithm for fixed-dimensional linear programming*, Mathematical Programming, 44 (1989), 203–212.
- [114] Edelsbrunner, H. *Algorithms in Combinatorial Geometry*, Springer-Verlag, 1987.
- [115] Edelsbrunner, H. *The union of balls and its dual shape*, Proc. 9th Ann. ACM Symp. Comput. Geom. (1993), 218–231.
- [116] Edelsbrunner, H., Guibas, L.J., Hershberger, J., Seidel, R., Sharir, M., Snoeyink, J., Welzl, E. *Implicitly representing arrangements of lines or segments*, Disc. Comput. Geom., 4 (1989), 433–466.
- [117] Edelsbrunner, H., Guibas, L.J., Sharir, M. *The complexity and construction of many faces in arrangements of lines and segments*, Disc. Comput. Geom., 5 (1990), 161–196.
- [118] Edelsbrunner, H., Guibas, L.J., Sharir, M. *The complexity of many cells in arrangements of planes and related problems*, Disc. Comput. Geom., 5 (1990), 197–216.
- [119] Edelsbrunner, H., Guibas, L.J., Stolfi, J. *Optimal point location in a monotone subdivision*, SIAM J. Comput., 15 (1986), 317–340.
- [120] Edelsbrunner, H., Mücke, E. *Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms*, ACM Trans. Graphics, 9 (1990), 66–104.
- [121] Edelsbrunner, H., Seidel, R. *Voronoi diagrams and arrangements*, Disc. Comput. Geom., 1 (1986), 25–44.
- [122] Edelsbrunner, H., Seidel, R., Sharir, M. *On the zone theorem for hyperplane arrangements*, SIAM J. Comput., 22 (1993), 418–429.
- [123] Edelsbrunner, H., Shah, N. R. *Triangulating topological spaces*, Proc. 10th Ann. ACM Symp. Comput. Geom. (1994), to appear.
- [124] Emiris, I., Canny, J. *A general approach to removing degeneracies*, Proc. 32nd Ann. Symp. Foundat. Comput. Sci. (1991), 405–413.
- [125] Erickson, J., Seidel, R. *Better lower bounds on detecting affine and spherical degeneracies*, Proc. 34th Ann. IEEE Symp. Foundat. Comput. Sci. (1993), 528–536.
- [126] Fitchas, N., Galligo, A., Morgenstern, J. *Algorithmes rapides en séquentiel et en parallèle pour l’élimination de quantificateurs en géométrie élémentaire*, Séminaire Structures Ordonnées, U.E.R. Math. Univ. Paris VII, 1987.

- [127] Fortune, S. *A sweepline algorithm for Voronoi diagrams*, Algorithmica, 2 (1987), 153–174.
- [128] Fortune, S. *Stable maintenance of point-set triangulation in two dimensions*, manuscript, AT&T Bell Laboratories. Abbreviated version appeared in: Proc. 30th Ann. Symp. Foundat. Comput. Sci. (1989), 494–499.
- [129] Fortune, S. *Numerical stability of algorithms for 2-d Delaunay triangulations and Voronoi diagrams*, Proc. 8th Ann. ACM Symp. Comput. Geom. (1992), 83–92.
- [130] Fortune, S. *Voronoi diagrams and Delaunay triangulations*, in: Computing in Euclidean Geometry, eds: D.-Z. Du, F. Hwang, 1, World Scientific (1992), 193–233.
- [131] Fortune, S. *Computational Geometry*, ed. R. Martin, Directions in Computational Geometry, Information Geometers, to appear.
- [132] Fortune, S., Milenkovic, V. *Numerical stability of algorithms for line arrangements*, Proc. 7th Ann. Symp. Comput. Geom. (1991), 334–341.
- [133] Fortune, S., Van Wyk, C. J. *Efficient exact arithmetic for computational geometry*, Proc. 9th Ann. ACM Symp. Comput. Geom. (1993), 163–172.
- [134] Fredman, M.L. *A lower bound on the complexity of orthogonal range queries*, J. ACM, 28 (1981), 696–705.
- [135] Fredman, M.L. *Lower bounds on the complexity of some optimal data structures*, SIAM J. Comput., 10 (1981), 1–10.
- [136] Freedman, M.H. *Identifying attractors via homology: a technique for time series classification and recognition*, manuscript, 1991.
- [137] Gärtner, B. *A subexponential algorithm for abstract optimization problems*, Proc. 33rd Ann. IEEE Symp. Foundat. Comput. Sci. (1992), 464–472.
- [138] Garey, M.R., Johnson, D.S., Preparata, F.P., Tarjan, R.E. *Triangulating a simple polygon*, Inform. Process. Lett., 7 (1978), 175–180.
- [139] Glassner, A.S. *Ray Tracing*, Academic Press, 1989.
- [140] Goodman, J.E., Pollack, R. *Multidimensional sorting*, SIAM J. Comput., 12 (1983), 484–507.
- [141] Goodman, J.E., Pollack, R. *Upper bounds for configurations and polytopes in  $\mathbb{R}^d$* , Disc. Comput. Geom., 1 (1986), 219–227.
- [142] Goodman, J.E., Pollack, R., Sturmfels, B. *The intrinsic spread of a configuration is  $\mathbb{R}^d$* , J. Amer. Math. Soc., 3 (1990), 639–651.
- [143] Goodman, J.E., Pollack, R., Wenger, R. *Geometric transversal theory*, in: New Trends in Discrete and Computational Geometry, ed. J. Pach, Algorithms and Combinatorics, 10, Springer-Verlag (1993), 163–198.
- [144] Goodrich, M.T. *Planar separators and parallel polygon triangulation*, Proc. 24th Ann. ACM Symp. Theory Comput. (1992), 507–516.
- [145] Goodrich, M. T. *Constructing arrangements optimally in parallel*, Disc. Comput. Geom., 9 (1993), 371–385.
- [146] Goodrich, M. T., Atallah, M. J., Overmars, M. H. *Output-sensitive methods for rectilinear hidden surface removal*, Inform. Comput., 107 (1993), 1–24.
- [147] Goodrich, M.T., Tamassia, R. *Dynamic trees and dynamic point location*, Proc. 23rd Ann. ACM Symp. Theory Comput. (1991), 523–533.
- [148] Goodrich, M.T., Tamassia, R. *Dynamic ray shooting and shortest paths via balanced geodesic triangulations*, Proc. 9th Ann. ACM Symp. Comput. Geom. (1993), 318–327.
- [149] Graham, R.L. *An efficient algorithm for determining the convex hull of a planar point set*, Inform. Process. Lett., 1 (1972), 132–133.
- [150] Greene, D., Yao, F. *Finite-resolution computational geometry*, Proc. 27th Ann. Symp. Foundat. Comput. Sci. (1986), 143–152.
- [151] Grigor'ev, D. *Complexity of deciding Tarski algebra*, J. Symbolic Comput., 5 (1988), 37–64.
- [152] Grigor'ev, D. and Vorobjov, N. *Solving systems of polynomial inequalities in subexponential time*, J. Symbolic Comput., 5 (1988), 37–64.
- [153] Guibas, L.J., Knuth, D., Sharir, M. *Randomized incremental construction of Delaunay and Voronoi diagrams*, Algorithmica, 7 (1992), 381–413.
- [154] Guibas, L.J., Overmars, M., Sharir, M. *Ray shooting, implicit point location, and related queries in arrangements of segments*, Tech. Rep. 433. New York Univ., March 1989.
- [155] Guibas, L.J., Salesin, D., Stolfi, J., *Epsilon geometry: building robust algorithms from imprecise computations*, Proc. 5th Ann. Symp. Comput. Geom. (1989), 208–217.
- [156] Guibas, L.J., Sharir, M. *Combinatorics and algorithms of arrangements*, New Trends in Discrete and Computational Geometry, J. Pach, ed., 1993, Springer-Verlag, 9–36.
- [157] Halperin, D., Sharir, M. *New bounds for lower envelopes in three dimensions, with applications to visibility in terrains*, Proc. 9th Ann. ACM Symp. Comput. Geom. (1993), 11–18.
- [158] Halperin, D., Sharir, M. *Almost tight upper bounds for the single cell and zone problems in three dimensions*, Proc. 10th Ann. ACM Symp. Comput. Geom. (1994), to appear.



- [159] Hart, S., Sharir, M. *Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes*, *Combinatorica*, 6 (1986), 151–177.
- [160] Haussler, D., Welzl, E.  *$\epsilon$ -nets and simplex range queries*, *Disc. Comput. Geom.*, 2 (1987), 127–151.
- [161] Heintz, J., Roy, M.-F., Solernó, P. *On the complexity of semi-algebraic sets*, *Proc. IFIP San Francisco, North-Holland* (1989), 293–298.
- [162] Heintz, J., Roy, M.-F., Solernó, P. *Sur la complexité du principe de Tarski-Seidenberg*, *Bull. Soc. Math. France*, 118 (1990), 101–126.
- [163] Heintz, J., Recio, T., Roy, M.-F. *Algorithms in real algebraic geometry and applications to computational geometry*, *Discrete and Computational Geometry*, DIMACS Series 6, AMS-ACM, ed. J.E. Goodman, R. Pollack, W. Steiger (1991), 137–163.
- [164] Hershberger, J., Suri, S. *A pedestrian approach to ray shooting: shoot a ray, take a walk*, *Proc. 4th ACM-SIAM Symp. Disc. Alg.* (1993), 54–63.
- [165] Hoffmann, C. *Geometric and Solid Modeling*, Morgan Kaufmann, 1989.
- [166] Hoffmann, C., Hopcroft, J., Karasick, M. *Towards implementing robust geometric computations*, *Proc. 4th Ann. Symp. Comput. Geom.* (1988), 106–117.
- [167] Hoffmann, C., Hopcroft, J., Karasick, M. *Robust set operations on polyhedral solids*, *IEEE Comput. Graph. Appl.*, 9 (1989), 50–59.
- [168] Hopcroft, J., Wilfong, G. *Reducing multiple object motion planning to graph searching*, *SIAM J. Comput.*, 15 (1986), 768–785.
- [169] Huttenlocher, D. P. *Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image*, Ph.D. Thesis, MIT, Report TR-1045, 1988.
- [170] Impagliazzo, R., Zuckerman, D. *How to recycle random bits*, *Proc. 30th Ann. IEEE Symp. Foundat. Comput. Sci.* (1989), 248–253.
- [171] Kalai, G. *A subexponential randomized simplex algorithm*, *Proc. 24th Ann. ACM Symp. Theory Comput.* (1992), 475–482.
- [172] Karasick, M., Lieber, D., Nackman, L. *Efficient Delaunay triangulation using rational arithmetic*, *ACM Trans. Graphics*, 10 (1990), 71–91.
- [173] Karasick, M. *On the representation and manipulation of rigid solids*, Ph.D. thesis, McGill U., 1988.
- [174] Katz, M., Sharir, M. *Optimal slope selection via expanders*, *Proc. 5th Canad. Conf. Comput. Geom.* (1993), 139–144.
- [175] Kirkpatrick, D.G. *Optimal search in planar subdivisions*, *SIAM J. Comput.*, 12 (1983), 28–35.
- [176] Kirkpatrick, D.G., Klawe, M.M., Tarjan, R.E. *Polygon triangulation in  $O(n \log \log n)$  time with simple data structures*, *Disc. Comput. Geom.*, 7 (1992), 329–346.
- [177] Kirkpatrick, D.G., Seidel R. *The ultimate planar convex hull algorithm?* *SIAM J. Comput.*, 15 (1986), 287–299.
- [178] Klein, R. *Abstract Voronoi diagrams and their applications*, *Computational Geometry and its Applications*, LNCS 333, Springer-Verlag (1988), 148–157.
- [179] Li, Z., Milenkovic, V. *Constructing strongly convex hulls using exact or rounded arithmetic*, *Proc. 6th Ann. Symp. Comput. Geom.* (1990), 235–243.
- [180] Lipton, R.J., Tarjan, R.E. *Applications of a planar separator theorem*, *SIAM J. Comput.*, 9 (1980), 615–627.
- [181] Lo, C.-Y., Matoušek, J., Steiger, W. *Ham-sandwich cuts in  $\mathbb{R}^d$* , *Proc. 24th Ann. ACM Symp. Theory Comput.* (1992), 539–545.
- [182] Luby, M. *A simple parallel algorithm for the maximal independent set problem*, *Proc. 17th Ann. ACM Symp. Theory Comput.* (1985), 1–10.
- [183] Matoušek, J. *Construction of  $\epsilon$ -nets*, *Disc. Comput. Geom.*, 5 (1990), 427–448.
- [184] Matoušek, J. *Cutting hyperplane arrangements*, *Disc. Comput. Geom.*, 6 (1991), 385–406.
- [185] Matoušek, J. *Approximations and optimal geometric divide-and-conquer*, *Proc. 23rd ACM Symp. Theory Comput.* (1991), 506–511.
- [186] Matoušek, J. *Range searching with efficient hierarchical cuttings*, *Disc. Comput. Geom.*, 10 (1993), 157–182.
- [187] Matoušek, J. *Efficient partition trees*, *Disc. Comput. Geom.*, 8 (1992), 315–334.
- [188] Matoušek, J. *Reporting points in halfspaces*, *Comput. Geom. Theory Appl.*, 2 (1992), 169–186.
- [189] Matoušek, J. *Linear optimization queries*, *J. Algorithms*, 14 (1993), 432–448.
- [190] Matoušek, J. *Geometric range searching*, Tech. Report B-93-09, Free Univ. Berlin, 1993.
- [191] Matoušek, J., Schwarzkopf, O. *A deterministic algorithm for the three-dimensional diameter problem*, *Proc. 25th Ann. ACM Symp. Theory Comput.* (1993), 478–484.
- [192] Matoušek, J., Schwarzkopf, O. *On ray shooting in convex polytopes*, *Disc. Comput. Geom.*, 10 (1993), 215–232.
- [193] Matoušek, J., Sharir, M., Welzl, E. *A subexponential bound for linear programming*, *Proc. 8th ACM Symp. Comput. Geom.* (1992), 1–8. To appear in *Algorithmica*.

- [194] Matoušek, J., Welzl, E., Wernisch, L. *Discrepancy and  $\epsilon$ -approximations for bounded VC-dimension*, *Combinatorica*, 13 (1993), 455–466.
- [195] Megiddo, N. *Combinatorial optimization with rational objective functions*, *Mathematics of Operations Research*, 4 (1979), 414–424.
- [196] Megiddo, N. *Applying parallel computation algorithms in the design of serial algorithms*, *J. ACM*, 30 (1983), 852–865.
- [197] Megiddo, N. *Linear programming in linear time when the dimension is fixed*, *J. ACM*, 31 (1984), 114–127.
- [198] Mehlhorn, K. *Data Structures and Algorithms 3: Multidimensional Searching and Computational Geometry*, Springer-Verlag, Heidelberg, Germany, 1984.
- [199] Mehlhorn, K., Simon, K. *Intersecting two polyhedra one of which is convex*, *Proc. Foundat. Comput. Theory, LNCS 199*, Springer-Verlag (1985), 534–542.
- [200] Mehlhorn, K., Yap, C.K. *Constructive Whitney-Graustein theorem, or how to untangle closed planar curves*, *SIAM J. Comput.*, 20 (1991), 603–621.
- [201] Milenkovic, V. *Verifiable implementations of geometric algorithms using finite precision arithmetic*, *Artificial Intelligence*, 37 (1988), 377–401.
- [202] Milenkovic, V. *Verifiable Implementations of Geometric Algorithms using Finite Precision Arithmetic*, Ph.D. Thesis, Carnegie-Mellon, 1988. Technical Report CMU-CS-88-168, Carnegie Mellon University, 1988.
- [203] Milenkovic, V. *Double precision geometry: a general technique for calculating line and segment intersections using rounded arithmetic*, *Proc. 30th Ann. IEEE Symp. Foundat. Comput. Sci.* (1989), 500–505.
- [204] Milenkovic, V. *Rounding face lattices in the plane*, *Abstracts 1st Canad. Conf. Comput. Geom.* (1989), 12.
- [205] Milenkovic, V. *Rounding face lattices in  $d$  dimensions*, *Proc. 2nd Canad. Conf. Comput. Geom.* (1990), 40–45.
- [206] Mishra, B. *Algorithmic Algebra*, Springer-Verlag New York, Inc., 1993.
- [207] Mitchell, S., Vavasis, S. *Quality mesh generation in three dimensions*, *Proc. 8th Ann. ACM Symp. Comput. Geom.* (1992), 212–221.
- [208] Motwani, R., Naor, J., Naor, M. *The probabilistic method yields deterministic parallel algorithms*, *Proc. 30th Ann. IEEE Symp. Foundat. Comput. Sci.* (1989), 8–13.
- [209] Muller, D.E., Preparata, F.P. *Finding the intersection of two convex polyhedra*, *Theoret. Comput. Sci.*, 7 (1978), 217–236.
- [210] Mulmuley, K. *A fast planar partition algorithm I*, *Proc. 29th Ann. IEEE Symp. Foundat. Comput. Sci.* (1988), 580–589.
- [211] Mulmuley, K. *On obstructions in relation to a fixed viewpoint*, *Proc. 30th Ann. IEEE Symp. Foundat. Comput. Sci.* (1989), 592–597.
- [212] Mulmuley, K. *On levels in arrangements and Voronoi diagrams*, *Disc. Comput. Geom.*, 6 (1990), 307–338.
- [213] Mulmuley, K. *A fast planar partition algorithm II*, *J. ACM*, 38 (1991), 74–103.
- [214] Mulmuley, K. *Hidden surface removal with respect to a moving point*, *Proc. 23rd Ann. ACM Symp. Theory Comput.* (1991), 512–522.
- [215] Mulmuley, K. *Randomized multidimensional search trees: dynamic sampling*, *Proc. 7th Ann. ACM Symp. Comput. Geom.* (1991), 121–131.
- [216] Mulmuley, K. *Randomized multidimensional search trees: lazy balancing and dynamic shuffling*, *Proc. 32nd IEEE Ann. Symp. Foundat. Comput. Sci.* (1991), 180–194.
- [217] Mulmuley, K. *Randomized multidimensional search trees: further results in dynamic sampling*, *Proc. 32nd IEEE Ann. Symp. Foundat. Comput. Sci.* (1991), 216–227.
- [218] Mulmuley, K. *Randomized geometric algorithms and pseudo-random generators*, *Proc. 33rd Ann. IEEE Symp. Foundat. Comput. Sci.* (1992), 90–100.
- [219] Mulmuley, K. *Computational Geometry: An Introduction Through Randomized Algorithms*, Prentice-Hall, 1994.
- [220] Naor, J., Naor, M. *Small-bias probability spaces: efficient constructions and applications*, *Proc. 22nd Ann. ACM Symp. Theory of Comput.* (1990), 213–223.
- [221] Norton, C.H., Plotkin, S.A., Tardos, E. *Using separation algorithms in fixed dimensions*, *J. Algorithms*, 13 (1992), 79–98.
- [222] O'Rourke, J. *Art Gallery Theorems and Algorithms*, Oxford Univ. Press, New York, NY (1987).
- [223] O'Rourke, J. *Computational Geometry in  $C$* , Cambridge Univ. Press, 1994.
- [224] Pach, J., Agarwal, P.K. *Combinatorial Geometry*, John Wiley & Sons, in press.
- [225] Pach, J., Steiger, W., Szemerédi, E. *An upper bound on the number of planar  $k$ -sets*, *Disc. Comput. Geom.*, 7 (1992), 109–123.
- [226] Pellegrini, M. *Ray shooting on triangles in 3-dimensional space*, *Algorithmica*, 9 (1993), 471–494.

- [227] Pellegrini, M. *On point location and motion planning among simplices*, these proceedings.
- [228] Pollack, R., Roy, M.-F. *On the number of cells defined by a set of polynomials*, *Compte-Rendus*, 316 (1993), 573-577.
- [229] Preparata, F.P., Hong, S.J. *Convex hulls of finite sets of points in two and three dimensions*, *Comm. ACM*, 20 (1977), 87-93.
- [230] Preparata, F.P., Shamos, M.I. *Computational Geometry: an Introduction*, Springer-Verlag, New York, 1988.
- [231] Preparata, F.P., Tamassia, R. *Fully dynamic techniques for point location and transitive closure in planar structures*, *Proc. 29th Ann. IEEE Symp. Foundat. Comput. Sci.* (1988), 558-567.
- [232] Preparata, F.P., Tamassia, R. *Efficient point location in a convex spatial cell-complex*, *SIAM J. Comput.*, 21 (1992), 267-280.
- [233] Prill, D. *On approximations and incidence in cylindrical algebraic decompositions*, *SIAM J. Comput.*, 15 (1986), 972-993.
- [234] Raghavan, P. *Probabilistic construction of deterministic algorithms: Approximating packing integer programs*, *J. Comput. System Sci.*, 37 (1988), 130-143.
- [235] Ramos, E., *Intersection of unit-balls and diameter of a set of points in  $\mathbf{R}^3$* , manuscript, 1994.
- [236] Reif, J.H., Sen, S. *Optimal randomized parallel algorithms for computational geometry*, *Algorithmica*, 7 (1992), 91-117.
- [237] Renegar, J. *On the computational complexity and geometry of the first order theory of the reals, I, II, III*, *J. Symbolic Comput.*, 13 (1992), 255-352.
- [238] Ruppert, J., Seidel, R. *On the difficulty of triangulating three-dimensional non-convex polyhedra*, *Disc. Comput. Geom.*, 7 (1992), 227-253.
- [239] Saalfeld, A. *Divide-and-conquer in early algebraic topology: the Mayer-Vietoris exact homology sequence revisited*, *Abstracts 1st Canad. Conf. Comput. Geom.* (1989), 11.
- [240] Sarnak, N., Tarjan, R.E. *Planar point location using persistent search trees*, *Comm. ACM*, 29 (1986), 669-679.
- [241] Schipper, H. *Determining contractibility of curves*, *Proc. 8th Ann. ACM Symp. Comput. Geom.* (1992), 358-367.
- [242] Schwartz, J.T., Sharir, M. *On the "piano movers" problem. II: General techniques for computing topological properties of real algebraic manifolds*, *Adv. in Appl. Math.*, 4 (1983), 298-351.
- [243] Schwartz, J.T., Sharir, M. *Algorithmic motion planning in robotics*, in: *Algorithms and Complexity, Handbook of Theoretical Computer Science*, ed. J. van Leeuwen, Vol. A, Elsevier (1990), 391-430.
- [244] Seidel, R. *A convex hull algorithm optimal for point sets in even dimensions*, *Univ. British Columbia, tech. Rep.* 81-14, 1981.
- [245] Seidel, R. *Constructing higher-dimensional convex hulls at logarithmic cost per face*, *Proc. 18th Ann. ACM Symp. Theory Comput.* (1986), 404-413.
- [246] Seidel, R. *Small-dimensional linear programming and convex hulls made easy*, *Disc. Comput. Geom.*, 6 (1991), 423-434.
- [247] Seidel, R. *A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons*, *Comput. Geom. Theory Appl.* 1 (1991), 51-64.
- [248] Seidel, R. *Backward analysis of randomized geometric algorithms*, *New Trends in Discrete and Computational Geometry*, J. Pach, ed., 1993, Springer-Verlag, 37-67.
- [249] Seidel, R. *The nature and meaning of perturbations in geometric computing*, manuscript, 1994.
- [250] Shamos, M.I., Hoey, D. *Closest-point problems*, *Proc. 16th Ann. IEEE Symp. Foundat. Comput. Sci.* (1975), 151-162.
- [251] Sharir, M. *Almost tight upper bounds for lower envelopes in higher dimensions*, *Proc. 34th Ann. IEEE Symp. Foundat. Comput. Sci.* (1993), 498-507.
- [252] Sharir, M., Agarwal, P.K. *Davenport-Schinzel Sequences and Their Geometric Applications*, Cambridge Univ. Press, to appear.
- [253] Sharir, M., Welzl, E. *A combinatorial bound for linear programming and related problems*, *Proc. 9th Symp. Theoret. Aspects of Comput. Sci. LNCS 577* (1992), 569-579.
- [254] Spencer, J.H. *Ten Lectures on the Probabilistic Method*, CBMS-NSF, SIAM, 1987.
- [255] Sugihara, K., Iri, M. *Construction of the Voronoi diagram for one million generators in single precision arithmetic*, *First Canad. Conf. Comput. Geom.*, 1989.
- [256] Sugihara, K., Iri, M. *A solid modeling system free from topological inconsistency*, *J. Information Processing, Information Processing Society of Japan*, 12 (1989), 380-393.
- [257] Tarjan, R.E., Van Wyk, C.J. *An  $O(n \log \log n)$ -time algorithm for triangulating a simple polygon*, *SIAM J. Comput.*, 17 (1988), 143-178.
- [258] Tourlakis, G., Mylopoulos, J. *Some results in computational topology*, *J. ACM*, 20 (1973), 439-455.

- [259] Vaidya, P.M., *Space-time tradeoffs for orthogonal range queries*, SIAM J. Comput., 18 (1989), 748–758.
- [260] Vapnik, V.N., Chervonenkis, A. Ya. *On the uniform convergence of relative frequencies of events to their probabilities*, Theory Probab. Appl., 16 (1971), 264–280.
- [261] Vegter, G. *Kink-free deformations of polygons*, Proc. 5th Ann. ACM Symp. Comput. Geom. (1989), 61–68.
- [262] Vegter, G., Yap, C.K. *Computational complexity of combinatorial surfaces*, Proc. 6th Ann. ACM Symp. Comput. Geom. (1990), 102–111.
- [263] Weispfenning, V. *The complexity of linear problems in fields*, J. Symbolic Comput., 5 (1988), 3–27.
- [264] Welzl, E. *Partition trees for triangle counting and other range searching problems*, Proc. 4th Ann. ACM Symp. Comput. Geom. (1988), 23–33.
- [265] Welzl, E., *On spanning trees with low crossing numbers*, Tech. Rep. TR B 92–02, Free University, Berlin, 1992.
- [266] Whitney, H. *Elementary structure of real algebraic varieties*, Annals of Math., 66 (1957).
- [267] Willard, D.E. *Polygon retrieval*, SIAM J. Comput., 11 (1982), 149–165.
- [268] Yao, A.C. *On the complexity of maintaining partial sums*, SIAM J. Comput., 14 (1985), 277–288.
- [269] Yao, A.C. *A lower bound to finding convex hulls*, J. ACM, 28 (1981), 780–787.
- [270] Yao, A.C. *Lower bounds for algebraic computation trees with integer inputs*, Proc. 30th Ann. IEEE Symp. Foundat. Comput. Sci. (1989), 308–313.
- [271] Yao, F.F. *Computational Geometry*, in: Algorithms and Complexity, Handbook of Theoretical Computer Science, ed. J. van Leeuwen, Vol. A, Elsevier (1990), 343–389.
- [272] Yao, A.C., Yao, F.F. *A general approach to d-dimensional geometric queries*, Proc. 17th Ann. ACM Symp. Theory of Comput. (1985), 163–168.
- [273] Yap, C.K. *Symbolic treatment of geometric degeneracies*, J. Symbolic Comput., 10 (1990), 349–370.
- [274] Yap, C.K. *A geometric consistency theorem for a symbolic perturbation scheme*, J. Comput. Sys. Sci., 40 (1990), 2–18.
- [275] Yap, C.K. *Towards exact geometric computation*, Proc. 5th Canad. Conf. Comput. Geom. (1993), 405–419.