On the Sectional Area of Convex Polytopes*

David Avis[1] Prosenjit Bose[2] Thomas C. Shermer[3]
Jack Snoeyink[2] Godfried Toussaint[1] Binhai Zhu[4]

June 26, 1996

Abstract

Let K be a convex polytope in $\mathcal{R}^d$, let $h(x)$ be the hyperplane consisting of all points with first coordinate equal to x, and let $A(x)$ be the area (or volume, if $d > 3$) of the section $K \cap h(x)$. Using the Brunn-Minkowski inequality, we show that $A(x)$ is a strictly unimodal function and give an algorithm to determine a hyperplane that achieves the maximum. Our algorithm runs in linear time, if the facial lattice of K is given.

1 Introduction

A function $f : \mathcal{R} \rightarrow \mathcal{R}$ is said to be *unimodal* if it increases to a maximum value and then decreases. It is *strictly unimodal* if the increase and decrease are strict. To be precise, f is strictly unimodal iff for all reals $x < y$, the minimum value $v = \min\{f(x), f(y)\}$ is either the global minimum or maximum of f or $v < f(z)$ for all $z \in (x, y)$.

Unimodality of functions is important for the design of efficient search algorithms because it permits *prune-and-search* strategies such as binary search or Fibonacci search [11]. For example, it was shown by Chazelle and Dobkin [4, 6] that the perpendicular distance from a line ℓ to an n-vertex

*The National Science and Engineering Research Council is gratefully acknowledged for grants partially supporting this research.

[1]School of Computer Science, McGill University, Montreal, PQ, Canada

[2]Dept. of Computer Science, Univ. of British Columbia, Vancouver, BC, Canada

[3]School of Computer Science, Simon Fraser Univ., Burnaby, BC, Canada

[4]Los Alamos National Lab, Los Alamos, NM, USA and Dept. of Computer Science, City Univ. of Hong Kong.

convex polygon Q is bimodal (i.e., with exactly two local minima). From this, unimodal functions can be constructed and the farthest point from ℓ can be computed in $O(\log n)$ time. Unimodality can also simplify proofs of geometric properties. For example, Pach [13] gave a 9-case combinatorial argument proving that the minimal-area triangle determined by three vertices of a convex polygon must share two edges with the polygon. This follows easily from the unimodality of distance.

It was shown in 1973 [8] that if $A(x)$ is the length of the intersection of a convex polygon Q with the vertical line through x, then $A(x)$ is a strictly unimodal function. We generalize this to higher dimensions in Section 2. Specifically, we show that the area in $\mathcal{R}^3$ (or volume in $\mathcal{R}^d$) of the intersection of a convex polytope K and a hyperplane $h(x) = \{(x, x_2, \ldots, x_d) \mid \forall x_i \in \mathcal{R}\}$ is a strictly unimodal function $A(x)$. (If the intersection plane is defined by rotation instead of translation, then there are convex polytopes for which sectional area is not unimodal.)

In the plane, the knowledge that $A(x)$ is unimodal leads easily to a binary search algorithm that computes the maximum-length intersection in $O(\log^2 n)$ time. (With a "tentative prune-and-search" paradigm, one can obtain an $O(\log n)$ time algorithm [12].) In higher dimensions, we show that prune-and-search can be used to compute the intersection with maximum area (volume) in time proportional to the size of K, if K is stored with its complete facial lattice.

We begin in Section 3 with an algorithm specifically for $\mathcal{R}^3$; this algorithm computes the maximum-area section of a convex polyhedron K in time proportional to the number of vertices or faces of K. This algorithm has an application to shape matching: Given convex polygons P and Q and a direction in which to translate P, one can find the translation having maximum overlap with Q in linear time.

In Section 4 we extend this algorithm to $\mathcal{R}^d$ by representing K as a sum of positive and negative simplices and computing the maximum-volume section of such a sum, knowing that the volume function is unimodal. Shermer [16] developed a similar algorithm for computing a bisector of a triangulated polygon; we observe that the simplices need not form a decomposition of the polytope and simplify his algorithm as well.

2 Unimodality of Sectional Area

Let K be a convex polyhedron in $\mathcal{R}^d$. Let $h(x)$ be the hyperplane consisting of all points with first coordinate equal to x. Let $A(x)$ denote the area (or volume, if $d > 3$) of the intersection $K \cap h(x)$. We show that $A(x)$ is a strictly unimodal function; it need not be convex or concave. This has also been observed by B. Chazelle and D. Dobkin (private communication).

The proof of strict unimodality rests on the Brunn-Minkowski Theorem, a powerful inequality for *mixed volumes* in $\mathcal{R}^{d-1}$ [15]. In this theorem, polytopes are considered as sets of vectors and linear combinations of polytopes give new polytopes by scalar multiplication and Minkowski sums of sets of vectors. Let $A(Q)$ denote the area (or volume) of a polytope Q.

Theorem 2.1 (Brunn-Minkowski) *If Q_1 and Q_2 are convex polygons in $\mathcal{R}^{d-1}$, then*

$$\sqrt[d-1]{A(\lambda Q_1 + (1-\lambda)Q_2)} \geq \lambda \sqrt[d-1]{A(Q_1)} + (1-\lambda) \sqrt[d-1]{A(Q_2)},$$

for all $0 < \lambda < 1$. Equality holds if and only if Q_1 and Q_2 are homothetic— that is, if $Q_2 = \lambda Q_1 + p$ for some $\lambda > 0$ and $p \in \mathcal{R}^{d-1}$.

An easy corollary establishes strict unimodality.

Corollary 2.2 *The sectional area $A(x)$ is a strictly unimodal function.*

Proof: To establish strict unimodality, we show that the minimum of $A(x)$ for x in any interval $[x_1, x_2]$ is $\min\{A(x_1), A(x_2)\}$. If $A(x_1) \neq A(x_2)$, then the minimum value is unique.

Assume that $A(x_1) \geq A(x_2)$, since the opposite inequality can be handled in a similar manner. Notice that any $x \in (x_1, x_2)$ can be written as $\lambda x_1 + (1-\lambda)x_2$ for $0 < \lambda < 1$.

Let S_1 and S_2 be the polygons in which K intersects the planes $h(x_1)$ and $h(x_2)$, respectively. (If either polygon is empty, say $S_i = \emptyset$, then $A(x_i) = 0$ is the unique minimum value.) Because K is convex, both S_1 and S_2 are convex. Furthermore, the intersection of K with the plane $x = \lambda x_1 + (1-\lambda)x_2$, for $0 < \lambda < 1$ contains the polygon $\lambda S_1 + (1-\lambda)S_2$. Applying the inequalities first for containment and second for Brunn-Minkowski, we obtain

$$\begin{aligned}\sqrt[d-1]{A(\lambda x_1 + (1-\lambda)x_2)} &\geq \sqrt[d-1]{A(\lambda S_1 + (1-\lambda)S_2)} \\ &\geq \lambda \sqrt[d-1]{A(S_1)} + (1-\lambda) \sqrt[d-1]{A(S_2)} \\ &= \sqrt[d-1]{A(S_2)} + \lambda(\sqrt[d-1]{A(S_1)} - \sqrt[d-1]{A(S_2)}).\end{aligned}$$

In the last line, the coefficient of λ is non-negative, so $A(x) \geq A(x_2)$ for $x \in (x_1, x_2)$. Furthermore, if $A(x_1) > A(x_2)$, then this coefficient is positive, which implies that $A(x) > A(x_2)$. Thus, $A(x)$ is strictly unimodal. ∎

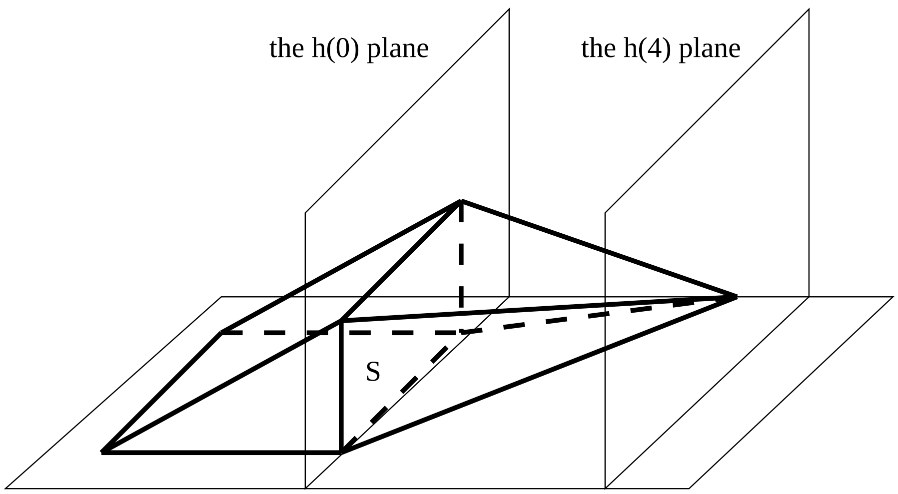

Figure 1: A convex polyhedron whose section area is neither convex or concave.

We observe that $A(x)$ need not be either convex or concave. Recall that a function $f(x)$ is concave if $f(\lambda x_1 + (1-\lambda)x_2) \geq \lambda f(x_1) + (1-\lambda)f(x_2)$. Consider a pyramid in $\mathcal{R}^3$ whose base polygon S is a square of area 4 in the $h(0)$ plane and whose apex is in the $h(4)$ plane. All sections of this polyhedron are homothets (squares). The Brunn-Minkowski theorem (or a simple geometric argument) tells us that the area $A(2) = (\sqrt{A(0)}/2 + \sqrt{A(4)}/2)^2 = 1$. Since $A(4)/2 + A(0)/2 = 2 > A(2)$, the sectional area function $A(x)$ is not convex. If we now form the convex polyhedron by taking the union of S and its reflection across its base, we obtain a polyhedron for which $A(x)$ is neither convex nor concave, as illustrated in figure 1.

Finally, if instead of translating a plane across a convex polyhedron, we rotate a plane about some axis, then sectional area is no longer unimodal. We can take a planar construction [1], which places n vertices around the unit circle in the yz plane, and add a point in $\mathcal{R}^3$ at $(1, 0, 0)$ to obtain a pyramid with a regular n-gon as its base. The sections formed by planes rotating about the x axis are unit-height triangles whose base edge length is a function of θ, the angle of the plane. This length has local maxima at each vertex [1], thus the sectional area has $\Omega(n)$ local maxima and minima.

3 An Algorithm for Sectional Area in $\mathcal{R}^3$ and 2-d Shape Matching

By using the unimodality property, we can find the maximum-area intersection of a convex, n-vertex polyhedron K with a plane perpendicular to the first coordinate axis. If the vertices, edges, and faces of K are stored in a standard data structure, such as the winged-edge [2], quad-edge [9], or doubly-connected edge list [14], then our algorithm runs in $O(n)$ time.

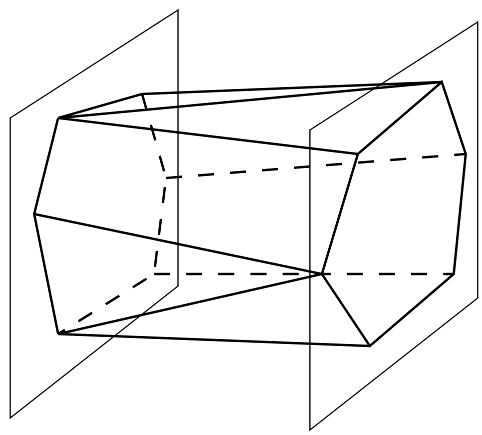

Figure 2: A drum.

We begin by defining a special class of polyhedra that have quadratic sectional-area function $A(x)$. A polyhedron is a *drum* if all of its vertices lie on two planes perpendicular to the x axis. Figure 2 depicts a convex drum; drums can also be non-convex.

Lemma 3.1 *Suppose that polyhedron P is a drum with vertices lying on planes $x = x_1$ and $x = x_2$. Then $A(x)$ is a quadratic function for $x \in [x_1, x_2]$.*

Proof: The section of a drum by a plane $x = \lambda x_1 + (1 - \lambda)x_2$ for $0 < \lambda < 1$ is a simple polygon, which we call S. Suppose the vertices of S are the points $s_1, s_2, \ldots, s_k = s_0$, where $s_i = (s_i.x, s_i.y, s_i.z)$. Then the area $A(S)$ is given by the formula

$$A(S) = \frac{1}{2} \sum_{1 \leq i \leq k} (s_{i-1}.y \cdot s_i.z - s_{i-1}.z \cdot s_i.y).$$

Vertex s_i of S is actually the intersection of the plane $x = \lambda x_1 + (1-\lambda)x_2$ with some edge of the drum. If the endpoints of that edge are q_i on the plane $x = x_1$ and x_i on $x = x_2$, then $s_i = \lambda q_i + (1-\lambda)x_i$. Plugging the coordinates of these points into the area $A(S)$ gives us a quadratic function of λ. (Note: the q_i need not all be distinct and the x_i need not all be distinct.) ∎

Corollary 3.2 *For a convex polyhedron K, the sectional area $A(x)$ is a piecewise quadratic function.*

Proof: Intersect K with each plane $h(x)$ that contains a vertex of K and define a set of $O(n)$ drums by taking the closures of the portions of K between adjacent planes. Lemma 3.1 says that area is quadratic on each drum. ∎

The simple structure of $A(x)$ within drums suggests several possible algorithms for computing the maximum sectional area:

- When K is a drum, for example, we can maximize by computing the derivative $A'(x)$, which is a linear function by Lemma 3.1. This takes $O(n)$ time.
- When K is not a drum, we can decompose K into drums, as suggested in the proof of corollary 3.2, and maximize in each drum. Unfortunately, the drums of this decomposition can have $\Theta(n^2)$ vertices.
- We could sweep through the drums of K in order of increasing x coordinate and take advantage of the fact that the changes in the area formula are proportional to the number of edges of K, which is $O(n)$. This would necessarily sort the vertices of K by x coordinate and require $\Theta(n \log n)$ time.
- The strict unimodality of $A(x)$ allows us to apply binary search on x coordinates of vertices to locate the drum containing the maximum area section. Computing the area function at $\Theta(\log n)$ sections, however, could still take $\Theta(n \log n)$ time.

To obtain a linear-time algorithm, we employ binary search, but we ensure that the complexity of the polyhedron being searched decreases by a constant factor at each search step. We define near-drums, which are the polyhedra that we will search, and a cutting operation to reduce their complexity.

A polyhedron P is a *near-drum* if it lies between the planes $x = x_1$ and $x = x_2$. The vertices of P strictly between these two planes are called *inner vertices*. Edges of P that are neither contained in the planes nor incident on inner vertices are called *long edges*. We can decompose a convex near-drum into a few (non-convex) drums and one convex near-drum that has size proportional to the sum of the degrees of the inner vertices of P.

Lemma 3.3 *Let P be an n-vertex, convex near-drum whose inner vertices have total degree k. In $O(n)$ time we can decompose P into a set of drums with $O(n)$ vertices total and a convex near drum with $O(k)$ vertices.*

Proof: We continue to assume that P is represented by a standard data structure. We can therefore traverse the representation for P, and mark edges that bound faces that are incident to inner vertices.

Because faces of a convex near-drum are convex, each face incident on an inner vertex has at most one long edge. By charging the degree of an inner vertex for marking incident edges and faces, we see that at most $2k$ edges are marked. In fact, our decomposition is done if all edges are marked (we have a near-drum of $O(k)$ vertices) or if all edges are unmarked (we have a drum).

Otherwise, consider the long edges in sequence around P. Let $E = \{e_1, e_2, \ldots, e_m\}$ be a sequence of unmarked long edges that is bounded by marked edges (q_1, q_2) and (r_1, r_2), where points q_i and r_i lie on the plane $x = x_i$ for $i = 1, 2$. (It is possible that edges of E end at q_i or r_i, or even that $q_i = r_i$ for $i = 1$ or $i = 2$.)

We cut a drum off the near-drum P as follows. Cutting along the boundary of the (possibly degenerate) quadrilateral $\square(q_1, q_2, r_2, r_1)$ separates the surface of the near-drum into two pieces. The piece that contains the edges E has no inner vertices, because its faces are incident on unmarked edges. Taking the convex hull of the other piece adds at most one diagonal—either (q_1, r_2) or (r_1, q_2)—and makes it into a convex near-drum. Adding the same diagonal into quadrilateral $\square(q_1, q_2, r_2, r_1)$ forms two triangles in space that were inside the convex polyhedron P. These triangles cut a drum from P whose edges consist of E with (q_1, q_2), (r_1, r_2), and the diagonal. (If $m = 1$ then e_1 was the diagonal and the resulting drum has no area and can be ignored.)

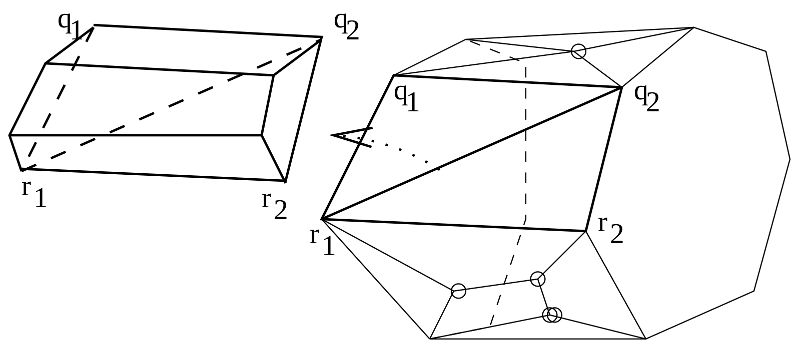

Figure 3: Removing a sequence of unmarked long edges.

We apply this cutting operation to all sequences of unmarked edges. Because each edge can participate in at most one cutting, the resulting drums have $O(n)$ total complexity. The final near-drum has at most $2k$ marked edges and k new diagonals. This completes the decomposition. ∎

We can now describe the algorithm that takes a convex polyhedron K and computes the maximum of the sectional area function, $A(x)$. The algorithm maintains a near-drum P whose area is $A_P(x)$ for $x \in [x_1, x_2]$ and a quadratic *drum area function* $A_D(x)$ valid over $[x_1, x_2]$. The algorithm also maintains two invariants:

1. The complexity of P is proportional to the total degree of its inner vertices.
2. The drum area function $A_D(x) = A(x) - A_P(x)$ over $[x_1, x_2]$.

To initialize, we set the quadratic function $A_D(x) = 0$ and turn K into near-drum by letting $x = x_1$ and $x = x_2$ be the supporting planes parallel to the yz plane. (If an initial support plane contains more than one vertex of K we may need to perform a cutting operation to establish the first invariant, as below.)

Now, given the near-drum P, list the x coordinates of the inner vertices with multiplicity equal to the vertex degree. The median x coordinate, $\tilde{x}$, can be found in linear time [3]. We can evaluate the derivative of sectional area at the median, $A'(\tilde{x})$ by summing $A'_D(\tilde{x})$ and $A'_P(\tilde{x})$—the latter we obtain by traversing P. If $A'(\tilde{x}) = 0$, then we have found the maximum sectional area. Otherwise, if $A'(\tilde{x})$ is positive we search $[x_1, \tilde{x}]$, and if negative we search $[\tilde{x}, x_2]$ for the drum containing the maximum.

Suppose that we need to search $[x_1, \tilde{x}]$. We set $x_2 = \tilde{x}$, which shrinks the near-drum. To re-establish invariant 1, we perform the cutting operation of Lemma 3.3, obtaining some drums $D_1, D_2, \ldots, D_m$ and a near-drum P'. By Lemma 3.1, the area functions for drums $D_1, D_2, \ldots, D_m$ are quadratic over $[x_1, x_2]$, so they can be accumulated into $A_D(x)$, restoring invariant 2.

The total degree of internal vertices of P' is less than half that of P. Since the initial sum of vertex degrees is $O(n)$ by Euler's formula, after $O(\log n)$ iterations P' is empty, and the maximum of $A_D(x)$ over $[x_1, x_2]$ is the maximum of $A(x)$ by invariant 2. Furthermore, the amount of work per iteration is proportional to the total degree of internal vertices of P. Because this decreases by a geometric series, the running time of the algorithm is linear in n.

We conclude

Theorem 3.4 *For a convex polyhedron K with n vertices, the maximum-area cross-section orthogonal to a given direction can be computed in $O(n)$ time.*

A corollary of this result and Chazelle's linear-time algorithm for intersecting convex polyhedra [5] has consequences in shape matching in the plane. Given two convex polygons, P and Q, and a direction v, we can compute the translation of P in the direction v that has maximum intersection with Q or, equivalently, that minimizes the symmetric difference of $P + \alpha v$ and Q. De Berg et al. [7] have recently developed an $O(n \log n)$ algorithm to find the translation of P in any direction that has the maximum overlap with Q; their algorithm depends on this subroutine.

Corollary 3.5 *Given convex polygons P and Q, the magnitude α of the translation along a given vector v that maximizes the area of intersection $(P + \alpha v) \cap Q$ can be computed in linear time.*

Proof: Add an α coordinate perpendicular to the xy-plane. Extend Q to a right cylinder in the direction $(0, 0, 1)$ and P to a slanted cylinder in the direction $(v.x, v.y, 1)$. The intersection of these two cylinders is a polyhedron K.

Each vertex of K comes from a line through a vertex of P or Q that is parallel to its cylinder axis and that intersects a plane of the other cylinder. Because the cylinders are convex, each line gives rise to at most two vertices; the size of K is proportional to the sizes of P and Q. We can compute K in linear time by Chazelle's algorithm [5]. (In fact, since we are dealing with cylinders, Chazelle's algorithm can be simplified.)

The maximum section of K that is parallel to the xy plane has α coordinate with the desired magnitude. ∎

One can replace the median finding with randomly selecting a simplex and one of its vertices to use for partitioning. The expected running time would remain linear.

4 An Algorithm for Sectional Area in $\mathcal{R}^d$

Given a convex polytope $K \subset \mathcal{R}^d$ that is described by its facial lattice, we can again compute the maximum sectional volume $A(x)$ in linear time: We represent K as a sum of positive and negative simplices, each with sectional volume functions that are piecewise polynomials of degree d. Unimodality allows us to use prune-and-search to narrow the interval for the maximum section and replace simplices that have no vertices in the interval by their volume functions.

Shermer [16] gave a similar algorithm for bisecting a polygon P in linear time, which relied on a trapezoidation of P. We observe that the trapezoids (or simplices) can be both positive and negative; they need not be a partition of P. This allows for easier generalization of his algorithm to higher dimensions. Furthermore, we replace his two median searches per step by a single one.

To begin, we need the formula for the sectional volume of a simplex.

Lemma 4.1 *In $\mathcal{R}^d$, the sectional volume of a simplex is a degree d polynomial between planes containing vertices.*

Theorem 4.2 *The maximum sectional volume $A(x)$ of a convex polytope $K \subset \mathcal{R}^d$ can be computed in linear time, if the facial lattice of K is given.*

Proof: Form a linear-size sum of simplices $K = \sum_i \pm S_i$ by bottom vertex triangulation or any other way of connecting facets to an origin.

Let interval $[x_1, x_2]$ contain the x coordinates of vertices of K. Let $\mathcal{S} \subset \{S_i\}$ denote the set of simplices with vertices having x coordinates in the open interval (x_1, x_2). Initialize a degree-d polynomial, $A_D(x) = 0$, to record the volume function for simplices of $\overline{\mathcal{S}} = \{S_i\} \setminus \mathcal{S}$.

List the x-coordinates of vertices in $[x_1, x_2]$ with multiplicity equal to the number of times the vertex appears in $\mathcal{S}$. Find the median x

coordinate, $\tilde{x}$, and evaluate the derivative $A'(x)$ by evaluating $A'_D(x)$ and $\sum_{S_i \in \mathcal{S}} A'(S_i)$.

Next, assign $x_1 = \tilde{x}$ or $x_2 = \tilde{x}$ so that $[x_1, x_2]$ still contains the maximum. Some simplices may no longer have vertices in the open interval (x_1, x_2); we remove them from $\mathcal{S}$ and add their volume functions to polynomial A_D. If $\mathcal{S}$ becomes empty, then we maximize $A_D(x)$ over $x \in [x_1, x_2]$, otherwise we continue with the previous paragraph.

The amount of work in each step is proportional to the number of vertices of $\mathcal{S}$ in $[x_1, x_2]$, since each simplex has constant complexity. At each step, this number is halved. Thus, the total is a geometric series that sums to linear. ∎

This algorithm can also be modified to compute the plane $h(x)$ that bisects a (possibly non-convex) polyhedron.

References

[1] D. Avis, G. T. Toussaint, and B. K. Bhattacharya. On the multimodality of distances in convex polygons. *Comput. Math. Appl.*, 8:153–156, 1982.

[2] B. Baumgart. A polyhedral representation for computer vision. In *Proc. of the AFIPS National Computer Science*, pages:589-596, 1975.

[3] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan. Time bounds for selection. *J. Comp. Sys. Sci.*, 7:448-461, 1973.

[4] B. Chazelle. Computational geometry and convexity. PhD Thesis, Dept. of Computer Science, Yale University, 1980.

[5] B. Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. *SIAM J Comp.*, 21(4):671–696, 1992.

[6] B. Chazelle and D. P. Dobkin. Intersection of convex objects in two and three dimensions. *JACM*, 34:1–27, 1987.

[7] M. de Berg, O. Devillers, M. van Kreveld, O. Schwarzkopf, and M. Teillaud. Computing the maximum overlap of two convex polygons under translations. Manuscript, 1995.

[8] S. W. Dharmadhikari and K. Jogdeo. A characterization of convexity and central symmetry for planar polygonal sets. *Israel J. Math.*, 15:356–366, 1973.

[9] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computationa of Voronoi diagrams. *ACM Trans. on Graphics*, 4(2):74–123, 1985.

[10] L. Khachiyan. Complexity of polytope volume computation. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, pages 91–101. Springer-Verlag, 1993.

[11] J. Kiefer. Sequential minimax search for a maximum. *Proc. Amer, Math. Soc.*, 4:502–506, 1953.

[12] D. Kirpatrick and J. Snoeyink. Tentative prune-and-search for computing fixed-points with applications to geometric computation. *Fundamenta Informaticæ*, 22(4):353–370, 1995.

[13] J. Pach. On an isoperimetric problem. *Studia Sci. Math. Hungar.*, 13:43–45, 1978.

[14] F. P. Preparata and M. R. Shamos. *Computational Geometry—An Introduction*. Springer-Verlag, New York, 1985.

[15] J. R. Sangwine-Yager. Mixed volumes. In P. M. Gruber and J. M. Wills, editors, *Handbook of Convex Geometry*, volume A, pages 43–71. Elsevier, 1993.

[16] T. C. Shermer. A linear algorithm for bisecting a polygon. *IPL*, 41:135–140, 1992.