Simplex Range Reporting on a Pointer Machine

Bernard Chazelle
Department of Computer Science
Princeton University

Burton Rosenberg
Department of Mathematics
and Computer Science
University of Miami

Abstract

We give a lower bound on the following problem, known as *simplex range reporting:* Given a collection P of n points in d-space and an arbitrary simplex q, find all the points in $P \cap q$. It is understood that P is fixed and can be preprocessed ahead of time, while q is a query that must be answered on-line. We consider data structures for this problem that can be modeled on a pointer machine and whose query time is bounded by $O(n^\delta + r)$, where r is the number of points to be reported and δ is an arbitrary fixed real. We prove that any such data structure of that form must occupy storage $\Omega(n^{d(1-\delta)-\varepsilon})$, for any fixed $\varepsilon > 0$. This lower bound is tight within a factor of n^ε.

1 Introduction

Given a set of n points in d-space, precompute a data structure capable of counting or reporting all points inside an arbitrary query simplex. This problem, known as *simplex range searching*, has been extensively studied in recent years [4, 5, 6, 8, 9, 11, 14, 15, 17, 19, 20]. On the practical side, the problem relates to fundamental questions in computer graphics, (e.g., hidden surface removal), while theoretically it touches on some of the most central issues in algorithm design and combinatorial geometry (e.g, derandomization, geometric graph separation, k-sets). In spite of all the attention, however, only recently have optimal or quasi-optimal solutions been discovered. If m is the amount of storage available, it is possible to achieve a query time of roughly $n/m^{1/d}$, where "roughly" means that an extra factor of the form n^ε [4] or $(\log n)^{O(1)}$ [14] must be added to the complexity bound. What allows us to brand these solutions quasi-optimal is an (almost) matching lower bound established in the arithmetic model of computation [2]. This lower bound is very general and holds for any realistic computing model, but it is limited to the case where searching is interpreted as *counting* or more generally computing the cumulative weight of weighted points inside the query.

This has left open the question of proving the optimality of the known algorithms in the *reporting* case: this is the version of the problem where the points inside the query must be found and reported one by one. To date, only the case of orthogonal range reporting has been satisfactorily resolved [3]. To prove lower bounds in the counting case is difficult enough, but the difficulty is compounded in the reporting case, because of the possibility for the algorithm to amortize the search over the output. This design paradigm, known as *filtering search* [1], is based on the idea that if many points must be reported then the search can be slowed down proportionately, which is then likely to result in a slimmer data structure.

We look here at the typical case where a query time of the form $O(n^\delta + r)$ is sought, where r is the size of the output and δ is any fixed constant. We show that on a pointer machine any data structure with a query time of that form must be of size $\Omega(n^{d(1-\delta)-\varepsilon})$, for any fixed $\varepsilon > 0$. This lower bound is quasi-optimal. Despite the apparent restrictions we place on the model, we must mention that the overwhelming majority of data structures proposed in the literature for range searching fall in that category. The magnitude of our lower bound is striking. It says, for example, that in E^{20} to achieve a query time even as inefficient as $O(\sqrt{n}+r)$ still requires approximately n^{10} storage!

The result combines graph theory with some slight integral geometry. The next section defines the model and proves a technical lemma regarding the spread of information across the data structure. Section 3 contains the proof of the main result. Concluding thoughts are given in Section 4.

2 The Complexity of Navigation on a Pointer Machine

We assume some familiarity with the *pointer machine model* [18]. As in [3] the data structure is modeled as a directed graph $G = (V, E)$ of outdegree at most 2. Let $P = \{ p_1, \ldots, p_n \}$ be a set of n points in E^d. To each node v of the data structure, an integer $f(v)$ is attached. If $f(v) = i$ is not zero, then node v is associated with point p_i. A *query* q is a simplex in E^d, and the algorithm must report all points in $P \cap q$. When presented with q, the algorithm begins at a starting node and, after following pointers across the data structure, terminates with a working set $W(q)$ consisting of all the visited vertices that is required to contain the answer, namely,

$$\{ i \,|\, p_i \in q \} \subseteq \{ f(v) \,|\, v \in W(q) \}.$$

The *size* of the data structure G is n, the number of nodes in the graph. Note that our model accommodates static as well as self-adjusting data structures.

A data structure G is termed (a,δ)-*effective* with a and δ positive real numbers, if for any query q, we have $|\, W(q) \,| \le a(|\, P \cap q \,| + n^\delta)$. A collection of queries $Q = \{ q_i \}$ is called (c,k,δ)-*favorable* if for all i, $|\, P \cap q_i \,| > n^\delta$ and for all $i_1 < \cdots < i_k$, $|\, P \cap q_{i_1} \cap \cdots \cap q_{i_k} \,| < c$. We want to show that if δ is small, an (a,δ)-effective data structure must be large. Using the following result, which generalizes a lemma given in [3], we can lower-bound a data structure's size by exhibiting a (c,k,δ)-favorable set of queries.

Lemma 2.1 *For any fixed $a, \delta > 0$ and $c \ge 2$, if G is (a,δ)-effective and Q is (c,k,δ)-favorable, then*

$$|\, V \,| > |\, Q \,|\, n^\delta / (3(k-1)2^{8ac^2}),$$

for n large enough.

Proof: We exploit the fact that the data structure can quickly answer a large number of very different queries to show that the data structure is itself large. More precisely, we look at the *c-sets* of V,

$$V^{(c)} = \{ W \subseteq V \,\big|\, |W| = c \}.$$

Recall that a tree is *rooted* if its edges are directed and the root is the only node with no incoming edge. Given any subset $W \subseteq V$, we define the *spanning-size* of W in G, denoted $\Lambda_G(W)$, as the minimum number of edges in any rooted tree that spans W and is a subgraph of G. It is ∞ if no such tree exists. This definition applies to any directed graph, in particular to subgraphs of G. Below we shall need Λ_T, where T is a rooted tree and a subgraph of G.

The number of c-sets in G of spanning-size smaller than r is bounded by,

$$\begin{aligned} \Big| \{ W \in V^{(c)} \,|\, \Lambda_G(W) < r \} \Big| &\leq \Big| \{ (z, W) \in V \times V^{(c)} \,|\, \forall w \in W,\ d(z,w) < r \} \Big| \\ &\leq |V| \, 2^{rc}, \end{aligned}$$

because of the limitation on the outdegree of G. Suppose now that query q is presented to the algorithm. Fix a rooted tree $T' \subseteq G$ which contains exactly the vertices of $W(q)$. Because the algorithm reaches all the nodes in $W(q)$, such a tree exists. We can select from $W(q)$ a subset W that contains exactly one $w \in W$ with $f(w) = i$ for each $p_i \in P \cap q$.

Let T be the Steiner minimal tree of W inside of T'. Note that $\Lambda_T(Z) \geq \Lambda_G(Z)$ for any $Z \subseteq G$. Embed the tree T in the plane and number the vertices of W in a natural order around the border of T. Then, $W = w_1, w_2, \ldots, w_s$, where $s = |P \cap q|$, and,

$$\sum_{j=1}^{s-1} \Lambda_T\big(\{w_j, w_{j+1}\} \big) \leq 2\,|T|\,.$$

Consider the c-sets,

$$W_i = \{ w_i, \ldots, w_{c+i-1} \}, \;\; i = 1, \ldots, s-c+1.$$

It is clear that,

$$\Lambda_T(W_i) \leq \sum_{j=i}^{c+i-2} \Lambda_T\big(\{w_j, w_{j+1}\} \big).$$

Summing over all i,

$$\sum_{i=1}^{s-c+1} \Lambda_T(W_i) \leq (c-1) \sum_{j=1}^{s-1} \Lambda_T\big(\{w_j, w_{j+1}\} \big) \leq 2(c-1)\,|T|\,.$$

Since, $|T| \leq |W(q)|$, if we assume that G is (a, δ)-effective and Q is (c, k, δ)-favorable (thus $|P \cap q| > n^\delta$):

$$\sum_{i=1}^{s-c+1} \Lambda_T(W_i) < 4a(c-1)\,|P \cap q|\,,$$

for n large enough. By Markov's inequality,

$$\left|\left\{\, i \,\middle|\, \Lambda_T(W_i) \geq 8a(c-1) \,\right\}\right| \leq |\, P \cap q \,|/2,$$

and therefore,

$$\left|\left\{\, i \,\middle|\, \Lambda_T(W_i) < 8a(c-1) \,\right\}\right| \geq |\, P \cap q \,|/2 - c + 1 > |\, P \cap q \,|/3.$$

Because $\Lambda_T(W_i) \geq \Lambda_G(W_i)$, this is also a lower bound on the number of c-sets with spanning-size in G less than $8a(c-1)$.

This argument is valid for any q in Q. Since $|\, P \cap q_{i_1} \cap \cdots \cap q_{i_k} \,| < c$, for appropriate indices $i_1 < \cdots < i_k$, a c-set of small spanning-size will be counted at most $k-1$ times. Thus,

$$\left|\left\{\, W \in V^{(c)} \,\middle|\, \Lambda_G(W) < 8a(c-1) \,\right\}\right| > |\, Q \,| \ \ |\, P \cap q \,|/(3(k-1)) > |\, Q \,|\, n^{\delta}/(3(k-1))$$

for large enough n.

In view of the upper bound given at the beginning of this proof, the result follows easily. □

3 A Lower Bound for Simplex Range Reporting

According to the discussion of the previous section, any algorithm for solving simplex range reporting in time $O(n^{\delta}+r)$ can be modeled as an (a,δ)-effective data structure, for some suitable constant a. An $\Omega\big(\, n^{d(1-\delta)-\varepsilon} \,\big)$ lower bound on the size of the data structure follows, according to Lemma 2.1, from the existence of a set P of n points along with a $(c, \lceil \log n \rceil, \delta)$-favorable query set Q of size $\Omega\big(\, n^{d(1-\delta)-\delta-\varepsilon} \,\big)$. The strictly positive real ε can be chosen as small as desired.

Let q be any nonzero vector in Euclidean d-space and μ any strictly positive real. The *slab* $H_{q,\mu}$ is a "thick" hyperplane, derived by taking the hyperplane perpendicular to the vector through q and passing through the point q and translating it continuously by small amounts parallel to q. The exact translations are λq for all $-\mu < \lambda < \mu$. Summarizing, for $q \in E^d$, $q \neq 0$, $\mu > 0$,

$$H_{q,\mu} = \left\{\, x \in E^d \,\middle|\, \ |\, \langle\, x, q \,\rangle - |\, q \,|^2 \,| \leq \mu |\, q \,| \,\right\}.$$

The point q is the *defining point* of the slab $H_{q,\mu}$. Although our final result is stated for a collection of simplex queries, the query set we construct is a collection of slabs. Once a favorable query set has been constructed, using slabs for queries, we can replace the slabs by very long flat simplices using elementary perturbation techniques. The value μ is fixed later on in the proof. Hence our attention is focused on the choice of a set of defining point for slabs making up the query collection. Reflecting this attention, from now on the set Q will be a set of points, the set of defining points for μ width slabs. It will be understood that the collection of queries is actually,

$$\left\{\, H_{q,\mu} \,\middle|\, q \in Q \,\right\}.$$

Let $C_d = [0,1]^d$ be the unit d-cube in E^d. We construct a favorable query set in two steps. First we position the slabs so that their arrangement has certain geometric properties: their intersection with C_d must be large, but their k-wise intersections with each other must be small. Next, n points are thrown at random into C_d and we verify that with high probability the slabs are favorable for this point set.

Further on we shall demonstrate that a sufficient condition for any k of the slabs to intersect in a small volume is that any k of the defining points have a large convex hull. This relates to Heilbronn's problem [12, 13, 16]: what is the largest area, over all point-sets $P = \{\, p_1, \ldots, p_m \,\} \subset C_2$, of the smallest triangle with vertices in P? Here we require that the convex hull of k points in d dimensions should have volume $\Omega(1/m)$. This can be achieved if $k \geq \log m$:

Theorem 3.1 (Chazelle[2]) *For any $d > 1$ there exists a constant $c > 0$ such that a random set of m points in C_d has, with probability greater than $1 - 1/m$, the property that the convex hull of any $k \geq \log m$ of these points has volume greater than ck/m.*

Hence a random point set is likely to be "good" for the construction of a favorable query set.

Let Q_0 be a random set of m points uniformly distributed in C_{d-1}. Theorem 3.1 assures that with high probability any $k \geq \log m$ points will "enclose" a large volume. We create a set of points Q in C_d from Q_0 in C_{d-1} as follows. Shrink C_{d-1} by a factor of two and paste it to the top face of C_d, that is, the face with coordinate d constant 1. Paste so that the $(1, \ldots, 1)$ corner of C_{d-1} contacts the $(1, \ldots, 1)$ corner of C_d. This carries the points of Q_0 to a set of points Q' on the top face of C_d. Send rays from the origin through each $q' \in Q'$, and select a series of points along each such ray. Fixing a real $(0 < \mu < 1)$, each q' gives rise to $\Theta(1/\mu)$ points along the ray by the map,

$$q' \mapsto 2\mu i q',$$

where i ranges over all integers which make coordinate d of $2\mu i q'$ lie within $[1/2, 3/4]$. To be precise, Q is the image of $Q_0 \times I$ under the map,

$$\begin{array}{ccc} C_{d-1} \times \mathbf{Z} & \longrightarrow & C_d \\ (x_1, \ldots, x_{d-1}, i) & \mapsto & \mu i\,(x_1 + 1, \ldots, x_{d-1} + 1, 2). \end{array}$$

where,

$$I = \left\{\, i \in \mathbf{Z} \,\middle|\, 1/(4\mu) \leq i \leq 3/(8\mu) \,\right\},$$

(see Figure 1).

Lemma 3.1 *Assume μ goes to zero with increasing m. Then,*

1. *Q is a set of size $\Theta\big(\, m/\mu \,\big)$.*
2. *For all $q \in Q$ the slabs $H_{q,\mu}$ have an intersection with C_d of volume $\Theta\big(\, \mu \,\big)$.*
3. *Any $k = \lceil \log m \,\rceil$ of these slabs have an intersection of volume,*

$$O\big(\, \mu^d m (\log m)^{d-2} \,\big).$$

Figure 1: Building the query set.

Proof: The first claim is trivial. The second follows from the fact that each coordinate of any $q \in Q$ is in the interval $[1/4, 3/4]$. So a ball of radius $1/4 - \mu$ and center q intersects H_q in a hyperdisk D which lies entirely inside C_d. The cylinder of height 2μ and cross section D at its midpoint is inside C_d. Here we assume, by increasing m if necessary, that $\mu \ll 1/4$. This gives the lower bound on the volume of $H_{q,\mu} \cap C_d$. The upper bound follows by placing a sufficiently large ball around q, say of radius $\sqrt{d}$, so as to contain the piece of $H_{q,\mu}$ that lies in C_d.

The third claim is substantiated as follows. Let $H_{q_1,\mu}, \ldots, H_{q_k,\mu}$ be the $k = \lceil \log m \rceil$ slabs, where the q_i are all distinct. If q_i and q_j are collinear with the origin, the intersection is empty. If they are not, let $p_1, \ldots, p_k$ be the points in C_{d-1} which gave rise to $q_1, \ldots, q_k$. The convex hull of the p_i has $(d-1)$-dimensional volume at least $c_1 k/m$ for some appropriate positive constant c_1. Triangulate the convex hull using $O\big(k^{d-1}\big)$ simplices and choose one among the simplices of largest $(d-1)$-volume. After renumbering, the vertices of this simplex are $p_1, \ldots, p_d$ and it has volume at least $c_2/(k^{d-2}m)$. We conclude that,

$$|\det(q_1, \ldots, q_d)| \geq c_3/(k^{d-2}m),$$

where the q_i have been renumbered according to the same pattern as the p_i and c_2 and c_3 are positive constants depending only on the dimension. The lemma follows from the next result. $\square$

Figure 2: Intersection parallelotope.

Lemma 3.2 *Given $k = \lceil \log m \rceil$, from every set $q_1, \ldots, q_k \subset Q$ a subset $q_{i_1}, \ldots, q_{i_d}$ can be selected such that,*

$$Vol\big(H_{q_1,\mu} \cap \cdots \cap H_{q_k,\mu} \big) \leq Vol\big(H_{q_{i_1},\mu} \cap \cdots \cap H_{q_{i_d},\mu} \big) = O\big(\mu^d m (\log m)^{d-2} \big).$$

Proof: We can still assume that no two q_i's are collinear with the origin. The first inequality is trivial. In general, let $q_1, \ldots, q_d$ be linearly independent vectors. The polytope $H_{q_1,\mu} \cap \cdots \cap H_{q_d,\mu}$ is a translate of the parallelotope defined by d vectors w_j where,

$$\langle\, w_j, q_i \,\rangle = \begin{cases} 2\mu \,|\, q_i \,| & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

To be more precise,

$$H_{q_1,\mu} \cap \cdots \cap H_{q_d,\mu} = \big\{ \textstyle\sum_{i=1}^{d} \alpha_i w_i \,\big|\, 0 \leq \alpha_i \leq 1,\ i = 1, \ldots, d \big\} + x_o,$$

where x_o is the unique point of E^d satisfying,

$$\langle\, x_o, q_i \,\rangle - |\, q_i \,|^2 = -\mu \,|\, q_i \,|$$

for all $i = 1, \ldots, d$ (see Figure 2). Denote by $[w]$ the matrix $(w_1, \ldots, w_d)$, by $[q]$ the matrix $(q_1, \ldots, q_d)$, and by Λ the diagonal matrix with $\Lambda_{ii} = |\, q_i \,|$. Note that $\det[w]$ is the volume of the parallelotope $H_{q_1,\mu} \cap \cdots \cap H_{q_d,\mu}$. From $[w]^T[q] = (2\mu)^d \Lambda$ we have

$$\det[w] \det[q] = (2\mu)^d \,|\, q_1 \,| \cdots |\, q_d \,| \,.$$

Recall that from the set $q_1, \ldots, q_k \subset Q$ we can select d vectors such that $|\det[q]| \geq c_3/(k^{d-2} m)$, and $\sqrt{d}/4 \leq |\, q_i \,| \leq 3\sqrt{d}/4$. This gives the bound. □

We finish the proof of the lower bound with a probabilistic analysis of the interaction between the query set Q and n points P chosen randomly in the unit cube

C_d. For any real $0 < \delta < (d-1)/d$ and any fixed $\varepsilon > 0$, set,

$$\mu = 1/(\tau n^{1-\delta}), \quad m = n^{d(1-\delta)-1-\varepsilon},$$

where τ depends only on d and will be selected appropriately in Lemma 3.3. Note that μ tends to zero and m tends to infinity as n tends to infinity. Set $k = \lceil \log m \rceil$ and $c = \lceil d^2/\varepsilon \rceil$. We claim that with high probability the collection of slabs $H = \{ H_{q,\mu} \mid q \in Q \}$ is (c, k, δ)-favorable for the point set P, where Q is as in Lemma 3.1.

Lemma 3.3 *Let the n points $P = \{ p_1, \ldots, p_n \}$ be independently and uniformly distributed in the unit cube C_d. With probability $1 - o(1)$, for all $q \in Q$,*

$$| H_{q,\mu} \cap P | > n^{\delta}.$$

Proof: The points $p_i \in H_{q,\mu}$, $i = 1, \ldots, n$, are independent Bernoulli random variables with common probability,

$$p = \mathrm{Vol}\,(H_{q,\mu} \cap C_d) > K\mu = K/(\tau n^{1-\delta}),$$

for an appropriate K which depends only on d. We can make τ small enough so that $np > 2n^{\delta}$. The expected number of points in q is therefore $E\big(| H_{q,\mu} \cap P | \big) = np > 2n^{\delta}$. The Chernoff bound [7, 10] states that, for $X = \{ x_1, \ldots, x_n \}$ a Bernoulli random variable where $x_i = 1$ with probability p and $x_i = 0$ with probability $1 - p$,

$$\mathrm{Prob}\left(\sum_{i=1}^{n} x_i \le (1-\kappa) np \right) \le \left(\frac{e^{-\kappa}}{(1-\kappa)^{1-\kappa}} \right)^{np},$$

for $0 < \kappa < 1$. Therefore, the probability that $| H_{q,\mu} \cap P | \le np/2$ is less than $(2/e)^{np/2}$. Taking the disjunction over all $q \in Q$,

$$\begin{aligned} \mathrm{Prob}\,(\, \exists\, q \in Q \text{ s.t. } | H_{q,\mu} \cap P | \le np/2 \,) &\le | Q |\ \mathrm{Prob}\,(\, | H_{q,\mu} \cap P | \le np/2 \,) \\ &< (m/\mu)\,(2/e)^{np/2} \\ &< n^{d(1-\delta)-\delta-\varepsilon} (2/e)^{n^{\delta}}. \end{aligned}$$

It is not difficult to see that this probability goes to 0 as n goes to infinity. Therefore, with probability approaching 1, every $H_{q,\mu}$ has more than $np/2 > n^{\delta}$ points in it. □

Lemma 3.4 *Let P be a set of n random points chosen uniformly in the unit cube C_d. With probability $1 - o(1)$, for all distinct $q_1, \ldots, q_k \in Q$,*

$$| H_{q_1,\mu} \cap \cdots \cap H_{q_k,\mu} \cap P | < c.$$

Proof: The events $p_i \in H_{q_1,\mu} \cap \cdots \cap H_{q_k,\mu}$, for $i = 1, \ldots, n$, are independent Bernoulli random variables with common probability,

$$p = \mathrm{Vol}\,(H_{q_1,\mu} \cap \cdots \cap H_{q_k,\mu}) < K \mu^d m (\log m)^{d-2},$$

for an appropriate constant K, (Lemma 3.1(3)), We again refer to the Chernoff bound: for any positive real κ,

$$\text{Prob}\left(\sum_{i=1}^{n} x_i \geq (1+\kappa)np\right) \leq \left(\frac{e^{\kappa}}{(1+\kappa)^{1+\kappa}}\right)^{np},$$

thus if $np < 1$ then for any integer $b \geq 1$,

$$\text{Prob}\left(\sum_{i=1}^{n} x_i \geq b\right) \leq \left(\frac{enp}{b}\right)^{b}.$$

The expected number of points in $H_{q_1,\mu} \cap \cdots \cap H_{q_k,\mu}$ is less than 1 for n sufficiently large, hence,

$$\text{Prob}\left(\,|\, H_{q_1,\mu} \cap \cdots \cap H_{q_k,\mu} \cap P \,| \geq c\,\right) \leq \left(\frac{K'(\log m)^{d-2}}{cn^{\varepsilon}}\right)^{c},$$

where K' is a positive constant. Recall from Lemma 3.2 that the upper bound on the volume of a k-wise intersection of query slabs is derived by considering a subset of only d of them. Therefore,

$$\begin{aligned}&\text{Prob}\left(\,\exists q_1,\ldots,q_k \in Q \text{ s.t. } |\, H_{q_1,\mu} \cap \cdots \cap H_{q_k,\mu} \cap P \,| \geq c\,\right)\\ &\qquad\qquad\leq \binom{|\,Q\,|}{d}\left(\frac{K'(\log m)^{d-2}}{cn^{\varepsilon}}\right)^{c},\end{aligned}$$

which goes to 0 as n increases. $\square$

What has been shown is the existence of a collection H of $\Theta\big(\,n^{d(1-\delta)-\delta-\varepsilon}\,\big)$ slabs and a set of n points P such that H is $\big(\lceil d^2/\varepsilon \rceil, \lceil\,(d(1-\delta)-1-\varepsilon)\log n\,\rceil, \delta\,\big)$-favorable with respect to P. We can now apply Lemma 2.1 and derive,

Theorem 3.2 *Simplex reporting on a pointer machine in E^d with a query time of $O\big(\,n^{\delta}+r\,\big)$, where r is the number of points reported and $0 < \delta \leq 1$, requires space $\Omega\big(\,n^{d(1-\delta)-\varepsilon}\,\big)$, for any fixed $\varepsilon > 0$.*

4 Conclusion

Our bound implies that if the search time is to be in $O\big(\,(\log n)^b + r\,\big)$, for b arbitrarily large, then the space must be in $\Omega\big(\,n^{d-\varepsilon}\,\big)$ for all fixed $\varepsilon > 0$. The factor of $n^{-\varepsilon}$ was introduced in the proof during the construction of an example query set in order that certain probabilities could be driven to zero. We believe that an improvement of the method might reduce the $n^{-\varepsilon}$ factor to a polylogarithmic one. To get rid of this extra factor altogether, however, appears more difficult. Finally, we would like to approach the question of halfspace range searching, where we expect similar techniques to give a bound of $\Omega\big(\,n^{\lfloor d/2 \rfloor-\varepsilon}\,\big)$ for polylog-time queries.

Acknowledgments

The authors would like to thank Laszlo Lovàsz for helpful discussions. They acknowledge the National Science Foundation for supporting this research in part under Grant CCR–9002352.

References

[1] B. Chazelle. Filtering search: A new approach to query-answering. *SIAM Journal on Computing*, 15(3):703–724, 1986.

[2] B. Chazelle. Lower bounds on the complexity of polytope range searching. *Journal of the American Mathematical Society*, 2(4):637–666, 1989.

[3] B. Chazelle. Lower bounds for orthogonal range searching: I. The reporting case. *Journal of the ACM*, 37(2):200–212, April 1990.

[4] B. Chazelle, M. Sharir, and E. Welzl. Quasi-optimal upper bounds of simplex range searching and new zone theorems. In *The Proceedings of the Sixth Annual Symposium on Computational Geometry*, pages 23–33, 1990.

[5] B. Chazelle, M. Sharir, and E. Welzl. Quasi-optimal upper bounds for simplex range searching and new zone theorems. *Algorithmica*, 8:407–429, 1992.

[6] B. Chazelle and E. Welzl. Quasi-optimal range seaching in spaces of finite Vapnik Chervonenkis dimension. *Discrete and Computational Geometry*, 4(5):467–489, 1988.

[7] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23:137–142, 1952.

[8] H. Edelsbrunner, D. G. Kirkpatrick, and H. A. Maurer. Polygonal intersection searching. *Information Procesing Letters*, 14(2):74–79, 1982.

[9] H. Edelsbrunner and E. Welzl. Halfplanar range seach in linear space and $O(n^{0.695})$ query time. *Information Processing Letters*, 23:289–293, 1986.

[10] P. Erdös and J. Spencer. *Probabilistic Methods in Combinatorics.* Academic Press, New York, 1974.

[11] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete & Computational Geometry*, 2(3):237–256, 1987.

[12] J. Komlós, E. Szemerédi, and J. Pintz. On Heilbronn's triangle problem. *Journal of the London Mathematical Society*, 24(2):385–396, 1981.

[13] J. Komlós, E. Szemerédi, and J. Pintz. A lower bound for Heilbronn's problem. *Journal of the London Mathematical Society*, 25(2):13–24, 1982.

[14] J. Matoušek. Efficient partition trees. In *Proceedings of the Seventh Annual Symposium on Computational Geometry*, pages 1–9, 1991.

[15] J. Matoušek. Range searching with efficient hierarchical cuttings. In *Proceedings of the Eighth Annual Symposium on Computational Geometry*, pages 276–285, 1992.

[16] W. O. J. Moser. Problems on extremal properties of a finite set of points. *Discrete Geometry and Convexity*, 440:52–64, 1985.

[17] M. Paterson and F. F. Yao. Point retrieval for polygons. *Journal of Algorithms*, 7:441–447, 1986.

[18] R. E. Tarjan. A class of algorithms which require nonlinear time to maintain disjoint sets. *Journal of Computing System Science*, 18:110–127, 1979.

[19] E. Welzl. Partition trees for triangle counting and other range searching problems. In *Proceedings Fourth Annual Symposium on Computational Geometry*, pages 23–33, 1988.

[20] D. E. Willard. Polygon retrieval. *SIAM Journal on Computing*, 11:149–165, 1982.