TIGHT BOUNDS ON THE STABBING NUMBER OF SPANNING TREES IN EUCLIDEAN SPACE

Bernard Chazelle

CS-TR-155-88

May 1988

Tight Bounds on the Stabbing Number of Spanning Trees in Euclidean Space

BERNARD CHAZELLE

Department of Computer Science
Princeton University
Princeton, NJ 08544

Abstract: We tighten the analysis of a data structure for simplex range searching discovered recently by Welzl. Our main result is that any set of n points in E^d admits a spanning tree which cannot be cut by any hyperplane (or hypersphere) through more than roughly $n^{1-1/d}$ edges. This result yields quasi-optimal solutions to simplex range searching in the arithmetic model of computation. We also look at circular and polygonal range searching on a random access machine. Given n points in E^2, we derive a data structure of size $O(n \log n)$ for counting how many points fall inside a query convex k-gon (for arbitrary values of k). The query time is $O(\sqrt{kn} \log n)$. If k is fixed once and for all (as in triangular range searching) then the storage requirement drops to $O(n)$. We also describe an $O(n \log n)$-size data structure for counting how many points fall inside a query circle in $O(\sqrt{n} \log^2 n)$ query time.

The author wishes to acknowledge the National Science Foundation for supporting this research in part under Grant CCR–8700917.

1. Motivation

Consider the general *range searching* problem: given a finite collection P of points in E^d and a region $q \subseteq E^d$, report (or count) the points of $P \cap q$. It is understood that the points are given once and for all and that the region q is a query to be answered on-line. There is usually a prescribed set of allowable queries, called the *query domain.* A typical example is to take the set of all simplices (simplex range searching), the set of all halfspaces (halfspace range searching), or the set of all d-balls (spherical range searching). To achieve greater generality, it is customary to assign a weight to each point of P and ask for the cumulative weight of $P \cap q$. Assume that P consists of n points, $p_1, \ldots, p_n$, and that only $O(n)$ space is available for storing the data structure. Here is a simple idea. Let $p_{i_1}, \ldots, p_{i_k}$ $(i_1 < \cdots < i_k)$ be the points of $P \cap q$. Rewrite the sequence $i_1, \ldots, i_k$ as a collection of maximal intervals, $s_1, \ldots, s_\ell$. For example, the sequence $(2, 3, 5, 7, 8, 9)$ yields $s_1 = [2, 3]$, $s_2 = [5, 5]$, and $s_3 = [7, 9]$. If the interval decomposition is given to us for free, answering the query q boils down to adding up together the answers to ℓ partial sum queries.

As a reminder, here is a quick run-down on the partial sum problem. Given n elements, $w_1, \ldots, w_n$, in a group or semigroup $(S, +)$, a *partial sum query* is an interval $[i, j]$: its answer is the value of $w_i + \cdots + w_j$. In the group model, there is a trivial constant time solution using linear storage. The same idea provides an optimal solution in the reporting case, where the problem is to report the elements $w_i, \ldots, w_j$. The case of a general semigroup is more subtle. Yao [9] has given a linear-size data structure with $O(\alpha(n))$ query time, where $\alpha(n)$ is a functional inverse of Ackermann's function (see Appendix). The algorithm is optimal in the *arithmetic model of computation.* In that model (Fredman [6], Yao [9]) a data structure is merely a collection of precomputed sums. The query time counts only the number of arithmetic operations needed to answer the query; it says nothing about the time to find the operands in the data structure.

Returning to our original discussion, the query is answered by computing ℓ partial sums and adding them up, which takes time $O(\ell\alpha(n))$. Of course, this solution is incomplete since it brushes aside the problem of computing the interval decomposition. In the arithmetic model, however, none of that work would be charged anyway. Therefore our solution, though unrealistic as it may be, can be meaningfully compared against the best lower bounds obtained in the arithmetic model.

The efficacy of this approach depends on the maximum size of an interval decomposition. Given a certain permutation of the input set P, we must consider the maximum value of ℓ over all queries q. The idea is to choose a permutation which minimizes this quantity (or at least makes it small enough). A permutation of the input points can be regarded as a one-path spanning tree, $(p_1, p_2), \ldots, (p_{n-1}, p_n)$. We say that a query q *stabs* an edge of the tree if the endpoints of the edge do not lie both inside or both outside q. If ℓ is the size of the interval decomposition, the number of edges stabbed by q lies between $2\ell - 2$ and 2ℓ. Remark: the term *stabbing number* is from (Edelsbrunner et al [5]); the notion originates in (Welzl [8]), where the term *crossing number* is used.

As it turns out, a spanning tree T is just as useful as a spanning path. Let $\sigma(q)$ denote the number of edges of T stabbed by q. The maximum value of $\sigma(q)$ over all queries q is called the *stabbing number* of T and is denoted $\sigma(T)$. Answering a query can be reduced to the computation of at most $\sigma(T)+1$ partial sums. Why is that so? Connect together the vertices of T in the order given by a depth-first traversal of the tree. This gives us a spanning path whose stabbing number is at most twice that of T. Indeed, let e be an edge of the spanning path that is stabbed by q. If e is not an edge of T then it creates a cycle in T, at least two of whose edges are stabbed by q. Because of the depth-first labeling, no edge of T can lie on more than two cycles, which proves our claim. The upper bound of $\sigma(T)+1$ on the number of partial sums follows readily.

The problem is now to compute spanning trees of low stabbing number. Welzl [8] recently discovered an ingenious method for building trees with stabbing numbers in $O(n^{1-1/d}\log n)$. His result holds for a general class of query domains, which includes, in particular, halfspace, simplex, and spherical range searching. A probabilistic variant of this result has also been obtained in (Edelsbrunner et al [5]). We will show here that the factor of $\log n$ can be removed in the three cases just mentioned. Using our previous remarks, this result has a simple geometric interpretation: it says that n points in E^d can always be made the vertices of a polygonal curve which no hyperplane (and no hypersphere) can cut in more than roughly $n^{1-1/d}$ pieces. As shown in [8] this bound is asymptotically optimal. Our algorithm is similar to Welzl's. What differs is the proof of a certain combinatorial lemma. Roughly speaking, the lemma says that if we have n points and m hyperplanes (or hyperspheres) in E^d, then two of the points are separated from each other by at most $m/n^{1/d}$ hyperplanes (or hyperspheres). Whereas Welzl uses probabilistic techniques, namely, ε-net theory, to tackle this problem, our approach remains purely geometric. It relies on packing properties of certain Euclidean-like metrics.

Besides deriving quasi-optimal solutions to simplex range searching in the arithmetic model, we also look at circular and polygonal range searching on a random access machine. Given n points in E^2, we derive a data structure of size $O(n\log n)$ for counting how many points fall inside a query convex k-gon (for arbitrary values of k). The query time is $O(\sqrt{kn}\log n)$. If k is fixed once and for all (as in triangular range searching) then the storage requirement drops to $O(n)$. We also describe an $O(n\log n)$-size data structure for counting how many points fall inside a query circle in $O(\sqrt{n}\log^2 n)$ query time.

2. Computing Spanning Trees of Low Stabbing Number

We begin with halfspace range searching. Then we generalize the technique to spherical range searching. As alluded to earlier, our results are predicated on the Euclidean-like behavior of certain noncontinuous metrics. Let $p_1,\ldots,p_n$ be n points of E^d, called *sites*, and let $\pi_1,\ldots,\pi_m$ be a finite collection of hyperplanes. Each hyperplane π_i is assigned a positive real number w_i (its weight). The sum of all the weights is denoted Δ. Given any two points p and q, we define the distance

$\delta(p,q)$ as the sum $\sum_i w_i$, taken over all hyperplanes π_i that split the segment pq. We say that a hyperplane *splits* a segment if it intersects it but does not contain either of its endpoints. Note that Δ is the (finite) diameter of the entire space. As it turns out, the distance δ is a little awkward to work with. One reason is that it does not satisfy the triangular inequality. So, to make our life easier, we define the distance $\delta^+(p,q)$ by extending the sum to all the hyperplanes intersecting pq.

Lemma 1. *Given m weighted hyperplanes and n points $p_1,\dots,p_n$ in E^d, if n is large enough, there exist two sites p_i and p_j $(i<j)$ such that $\delta(p_i,p_j)\le 3^{d+4}\Delta/n^{1/d}$.*

By "n is large enough", we mean that n should exceed some appropriate constant which depends only on d and not on m or Δ. No assumptions are made on the positions of the sites and hyperplanes, nor for that matter on the number of hyperplanes. For this reason, we begin the proof by tackling a restricted version of the problem and then increase the generality in stages. Assume that the m hyperplanes are in general position and have unit weight ($w_i=1$). The metrics δ and δ^+ share some fundamental properties with the Euclidean metric. For example, Lemma 2 says that, for r not too large, a ball of radius r (with respect to δ^+) has volume $\Omega(r^d)$, where the "volume" of a region V is understood as the number of d-wise hyperplane intersections in V. More important, Lemma 3 asserts that in the metric δ the two nearest sites are only $O(\Delta/n^{1/d})$ apart. A similar, well-known fact in E^d is that if a set of n points has Euclidean diameter Δ, then the Euclidean distance between the two nearest points is $O(\Delta/n^{1/d})$.

Let H be the arrangement formed by the m hyperplanes. Because of general positioning, any vertex v of H is the intersection of exactly d hyperplanes. Given a point p and a real r, we define the ball $B(p,r)$ as the set of vertices v of H such that $\delta^+(p,v)\le r$. The *volume* of $B(p,r)$ is its cardinality.

Lemma 2. *Given m hyperplanes in general position in E^d, let p be a point of E^d and let r be a real such that $2^{d+3}\le r\le m$. If the hyperplanes are assigned unit weight, then the volume of $B(p,r)$ exceeds $r^d/3^{d^2}$.*

Proof: We proceed by induction on d. Let $g_d(m,r)$ be the minimum volume of any ball $B(p,r)$ in E^d. If $d=1$, then we have $g_1(m,r)\ge\lfloor r\rfloor>r/3$. Assume now that $d>1$. Let $\pi_1,\dots,\pi_m$ denote the m hyperplanes and let H be the arrangement which they form. Because of our assumptions there exists a line L passing through p which does not intersect any $\pi_i\cap\pi_j$ $(i<j)$ outside of p. Let $q_1,\dots,q_m$ be the sequence of intersections between L and the hyperplanes. The sequence is chosen so that the Euclidean distance between p and q_i is non-decreasing. Let π_{i_k} be the hyperplane associated with q_k and let H_k denote the arrangement formed by the $(d-2)$-flats $\pi_{i_k}\cap\pi_\ell$ $(\ell\ne i_k)$. Each H_k is an arrangement of $m-1$ unit-weight hyperplanes in general position in E^{d-1}. Except perhaps for $q_1,\dots,q_d$, each point q_i is distinct. Therefore, using the monotonicity of $g_{d-1}(m,r)$ in r and the fact that δ^+ satisfies the triangular inequality, we have

$$dg_d(m,r)\ge\sum_{d<k\le\lfloor r\rfloor}g_{d-1}(m-1,r-k)>\sum_{d<k\le\lfloor r/2\rfloor}(r-k)^{d-1}/3^{(d-1)^2}\ge\frac{r^d}{2^{d-1}3^{(d-1)^2+1}},$$

from which it follows that $g_d(m,r) > r^d/3^{d^2}$. ∎

Lemma 3. *Given m unit-weight hyperplanes in general position in E^d and n points $p_1,\ldots,p_n$ in E^d, if n is large enough, there exist two points p_i and p_j $(i<j)$ such that $\delta(p_i,p_j) \le 3^{d+4}m/n^{1/d}$.*

Proof: Let $r = 3^{d+3}m/n^{1/d}$ and assume that

$$3^{d+3}m \ge 2^{d+3}n^{1/d}. \tag{2.1}$$

We easily verify that $2^{d+3} \le r \le m$, for n large enough. Assume that the balls $B(p_i,r)$ $(1 \le i \le n)$ are pairwise disjoint. The combined volume of the n balls cannot exceed the volume of, say, $B(p_1,+\infty)$. Using Lemma 2, we find that $nr^d/3^{d^2} < \binom{m}{d}$, which gives a contradiction. It follows that two of the balls must intersect. By the triangular inequality, the δ^+-distance, and hence the δ-distance, between their centers is at most $2r$. Suppose now that (2.1) does not hold. We can assume that $m \ge 2d$. Otherwise, for n large enough and $j \le d$, some j-face of the arrangement H will contain at least two sites p_i and p_j, and hence $\delta(p_i,p_j)=0$. If $m \ge 2d$, the number ϕ of d-faces in H satisfies (Edelsbrunner [4])

$$\phi = \sum_{0\le k\le d}\binom{m}{k} \le d\binom{m}{d} + 1 \le m^d + 1.$$

This implies that $n > (3/2)^{d(d+3)}(\phi - 1)$. Since $m > 0$, we have $\phi \ge 2$, and therefore $n \ge \phi + 1$. This shows that at least two points p_i and p_j $(i<j)$ lie in the closure of the same d-face of the arrangement, therefore $\delta(p_i,p_j) = 0$. ∎

Assume that the hyperplanes π_i are still in general position. We now turn to the case of positive integral weights $w_1,\ldots,w_m$. The idea is to make w_i copies of each π_i and perturb them in such a way that the arrangement formed by any sample of hyperplanes, one for each π_i, remains isotopic to H. Also, we must ensure that if a site does not lie on π_i then it lies on the same side of π_i and its duplicates. Finally, we require that the new hyperplanes be in general position. It is clear that the transformation can only increase the distance between any two sites (in the metric δ). This allows us to generalize Lemma 3 to the case of positive integral weights.

How about arbitrary positive real weights? Pick some large integer k and replace each w_i by the integral weight $w_i' = \lfloor 2^k w_i \rfloor$. Our previous generalization shows that, for n large enough, there exist two sites p_i and p_j $(i<j)$ such that $\delta'(p_i,p_j) \le 3^{d+4}\Delta'/n^{1/d}$, where $\Delta' = \sum_{1\le i\le m}\lfloor 2^k w_i\rfloor$ and δ' is the metric δ modified in the obvious way. Let J be the set of indices ℓ such that π_ℓ intersects the segment pq (but not either of the two endpoints). We have

$$\sum_{\ell\in J}\lfloor 2^k w_\ell\rfloor \le 3^{d+4}\Delta'/n^{1/d},$$

therefore

$$\delta(p_i,p_j) = \sum_{\ell\in J} w_\ell \le 3^{d+4}\Delta/n^{1/d} + |J|/2^k.$$

Since k can be arbitrarily large, it follows that $\delta(p_i, p_j) \leq 3^{d+4}\Delta/n^{1/d}$, which generalizes Lemma 3. Finally, we must tackle the case where the hyperplanes are not in general position. In that case, we can slightly perturb them without decreasing the δ-distance between any two sites. This completes the proof of Lemma 1.

We are now ready to exhibit spanning trees of low stabbing number. Let $p_1, \ldots, p_n$ be n sites in E^d and let $\pi_1, \ldots, \pi_m$ be the hyperplanes passing through each d-tuple of points (if more than one hyperplane contains a given d-tuple, pick any one of them). Assign unit weight to each of the $m = \binom{n}{d}$ hyperplanes. From Lemma 1, two sites p_i and p_j $(i < j)$ lie within a distance $3^{d+4}\Delta/n^{1/d}$ of each other. Let us call the segment p_ip_j a *special edge*: it is the first edge of our spanning tree. For this reason, the hyperplanes crossing the special edge are no longer nearly as fresh and young as the others. To reflect this situation, we borrow a clever idea from (Welzl [8]). We shall multiply by two the weight of each hyperplane that splits p_ip_j, and then remove the site p_i from the set. (There is nothing magic about 2; any constant multiplier would work just the same.) Next, we iterate the whole procedure ρ times, where

$$n^{1/d}(\log m)/3^{d+4} \leq \rho < n. \tag{2.2}$$

At all times the weight of the system stays within

$$\left(1 + \frac{3^{d+4}}{n^{1/d}}\right)^{\rho} m \leq e^{3^{d+4}\rho/n^{1/d}} m < e^{3^{d+5}\rho/n^{1/d}}.$$

The assignment $\rho = \lfloor n^{1/d} \log n \rfloor$ ensures (2.2) and guarantees that no hyperplane can split more than $3^{d+6} \log n$ special edges.

Lemma 4. *Let P be a set of n points in E^d $(d > 1)$. If n is large enough, it is possible to form a forest consisting of at least $\lfloor n^{1/d} \log n \rfloor$ distinct edges with endpoints in P so that no hyperplane splits more than $3^{d+6} \log n$ edges.*

The pairs of Lemma 4 form a forest of trees. Keep one point per tree and apply the lemma to the remaining points. Iterate on this process until the number of remaining points falls below $n^{1-1/d}$. Finally, connect the remaining points via an arbitrary spanning path. This procedure produces a spanning tree of the n sites which any hyperplane can split at only $O(n^{1-1/d})$ edges. (A quick-and-dirty way to see this is to break down the iteration into phases where the number of points shrinks by roughly half each time.) It follows that if the sites are in general position, the tree has a stabbing number in $O(n^{1-1/d})$ with respect to halfspace range searching. If we do not have general positioning, we face the difficulty that a halfspace might cut through a large number of edges which are not split by its bounding hyperplane. Actually, a simple perturbation argument shows that this cannot happen. In light of our observations in the first section, we conclude that any set of n points in E^d can be made the vertices of a polygonal curve which any hyperplane splits through $O(n^{1-1/d})$ edges. We will see in the next section how to generalize this result to hyperspheres.

Since a simplex is the intersection of $d+1$ halfspaces, what we have proven has a direct application to simplex range searching. Applying the strategy outlined in section 1, we have the following result.

Theorem 5. *Simplex range searching on n points in E^d can be performed in $O(n^{1-1/d}\alpha(n))$ query time and $O(n)$ storage in the arithmetic model of computation.*

Recall that $\alpha(n)$ is the inverse of the function $A(n,n)$, the diagonal of Ackermann's function (see Appendix). It follows from (Chazelle [1]) that the bounds of the theorem are tight within a factor of $\alpha(n)$, if $d=2$, and $\alpha(n)\log n$, if $d>2$.

3. Spherical Range Searching

Can we generalize Lemma 4 to more complicated range searching problems? As shown in (Yao [10]) spherical range searching in E^d is a special case of halfspace range searching in E^{d+1}. Theorem 5 is thus ready for action. We will obtain better results, however, if we can treat hyperspheres as we did hyperplanes and stay in d-space. Let us go through the whole argument once again. Lemma 4 is not a problem: the number m of representative hyperspheres to consider is $O(n^{d+1})$, so the proof requires only minor modifications. Actually, it is quite clear that only Lemmas 2 and 3 need to be looked at. Once these hurdles have been cleared, dealing with arbitrary weights and hyperspheres which are not in general position will be no different from the previous case (and remorselessly omitted). In a nutshell, the difficulty with hyperspheres is that any d of them may not be expected to intersect always, as was the case with hyperplanes in general position. This will necessitate a revision of our volume-based argument.

So, we now have n sites $p_1,\dots,p_n$ and m hyperspheres $\pi_1,\dots,\pi_m$ in E^d. For consistency, we shall assume that the sites lie on a d-sphere S^d in E^{d+1}. This does not really matter since S^d can always be chosen very big (why this is so is left as a homework). On the other hand, it allows us to define each π_i as the intersection of S^d with some hyperplane. We shall assume that the set of hyperspheres is in general position. The real reason for switching to S^d is to salvage the induction used in the proof of Lemma 2. Each of the m hyperspheres is assigned unit weight. Given two points p and q in S^d, the distance $\delta(p,q)$ is now defined as the number of hyperspheres that split the segment pq. We say that a hypersphere π *splits* a segment s if the endpoints of s lie in distinct connected components of $S^d \setminus \pi$, or equivalently, if the d-flat supporting π intersects the segment s but not its endpoints. As before, the distance $\delta^+(p,q)$ differs from $\delta(p,q)$ by counting all the hyperspheres whose supporting d-flats intersect the segment pq. Trivially, δ^+ satisfies the triangular inequality. Note that the sum of the weights (that is, m, in this case) is no longer the diameter of the space (at least not always). Let H be the arrangement formed by the m hyperspheres. The ball $B(p,r)$ and its volume are defined just as before, by considering the vertices of H within a δ^+-distance of r from p.

One serious difference with before is that the volume of S^d might no longer be on the order of m^d. If things went nicely, then every set of d hyperspheres would intersect in exactly two points and the volume of S^d would be precisely $2\binom{m}{d}$. Unfortunately, we might have much fewer intersections. To cope with this problem, we put H in normal form by adding dummy vertices to it. We define the notion of a *μ-complete* arrangement. For the sake of the definition, let us make no assumption on m except that it is a nonnegative integer. Let μ be any integer $\geq \max(d,m)$. To begin with, add $\binom{\mu}{d}$ dummy vertices into S^d. Then, for each $C \subseteq \{\pi_1, \ldots, \pi_m\}$ such that $1 \leq |C| \leq d-1$ and $\bigcap_{\pi \in C} \pi \neq \emptyset$, place $\binom{\mu-|C|}{d-|C|}$ dummy vertices in the $(d-|C|)$-sphere $\bigcap_{\pi \in C} \pi$. Adding vertices is just a trick to increase the volume of $\bigcap_{\pi \in C} \pi$. The precise location of the vertices in their various hyperspheres is irrelevant. Likewise, dummy vertices do not affect the distances δ and δ': they simply modify the volume distribution of H. The new arrangement H is said to be μ-complete. The volume of S^d in a μ-complete arrangement may vary: in all cases, however, it falls between $\binom{\mu}{d}$ and

$$3\binom{\mu}{d} + \sum_{1 \leq c \leq d-1} \binom{\mu}{c}\binom{\mu-c}{d-c} < (3+d)\mu^d. \tag{2.3}$$

A crucial property of μ-completeness is a form of invariance under cross-section. Assume that $d \geq 2$ and $m > 0$, and consider the arrangement on π_k formed by the $(d-2)$-spheres $\pi_k \cap \pi_j$ $(j \neq k)$. Of course, all intersections might be empty, but that is all right. We easily check that in all cases the arrangement is $(\mu-1)$-complete. Well, actually, we may have a few more vertices than needed. To preserve μ-completeness, we shall get rid of the excess. We are now ready to revisit Lemma 2.

Lemma 6. *Given an m-complete arrangement of m hyperspheres in S^d in general position, let p be a point of S^d and let r be a real such that $2^{d+3} \leq r \leq m$. If the hyperspheres are assigned unit weight, then the volume of $B(p,r)$ exceeds $r^d/3^{d^2}$.*

Proof: We prove the lemma with respect to a μ-complete arrangement of m hyperspheres, where $0 \leq m \leq \mu$ and $2^{d+3} \leq r \leq \mu$. As usual, let $\pi_1, \ldots, \pi_m$ denote the m hyperspheres. We proceed by induction on d. Let $g_d(\mu, r)$ be the minimum volume of any ball $B(p,r)$ in S^d. If $d=1$, then the hyperspheres are pairs of points on S^1 and their "inside" is one of the two circular arcs which they define. If there is a point $p' \in S^1$ such that $\delta^+(p,p') \geq \lfloor r \rfloor$, then obviously the volume of $B(p,r)$ is at least $\lfloor r \rfloor > r/3$. If there is no such point then the volume of $B(p,r)$ is equal to the volume of $B(p,+\infty) \geq \mu > r/3$. Assume now that $d > 1$. Let p' be a point of S^d which maximizes the distance $\rho = \delta^+(p,p')$. If $\rho \leq r$ then the volume of $B(p,r)$ is that of S^d, which is at least $\binom{\mu}{d} > \mu^d/(2d)^d > r^d/3^{d^2}$. Suppose now that $\rho > r$ and consider a circle (that is, the intersection of S^d with a two-dimensional flat of E^{d+1}) which contains p and p' and at the same time avoids any $\pi_i \cap \pi_j$ $(i < j)$ outside of p and p'. Think now of a point q moving continuously along the circle from p to p' in some given direction. The distance $\delta^+(p,q)$ goes from $j \leq d$ to ρ not necessarily monotonically. However, it varies from d to $\rho - d$ by increments of ± 1. This implies that q crosses at least $\ell = \lfloor r/2 \rfloor - d$ *distinct* hyperspheres at points $q_1, \ldots, q_\ell$, such that $\delta^+(p,q_k) \leq \lfloor r/2 \rfloor$

$(1 \le k \le \ell)$. The hypersphere passing through each q_k is now regarded as the underlying space S^{d-1} of an arrangement H_k of unit-weight $(d-2)$-spheres. From a previous remark, we know that H_k is $(\mu - 1)$-complete (after perhaps removal of some dummy vertices). Since δ^+ satisfies the triangular inequality, we have

$$dg_d(\mu, r) \ge \sum_{d<k<\rho-d} g_{d-1}(\mu - 1, r - k) > \sum_{d<k\le\lfloor r/2\rfloor} (r-k)^{d-1}/3^{(d-1)^2} \ge \frac{r^d}{2^{d-1}3^{(d-1)^2+1}},$$

from which it follows that $g_d(\mu, r) > r^d/3^{d^2}$. ∎

Lemma 7. *Given m unit-weight hyperspheres in general position in E^d and n points $p_1, \ldots, p_n$ in E^d, if n is large enough, there exist two points p_i and p_j $(i < j)$ such that $\delta(p_i, p_j) \le 3^{d+4}m/n^{1/d}$.*

Proof: Almost strictly identical to that of Lemma 3. Here it is. We shall assume that the arrangement of hyperspheres is m-complete. Let $r = 3^{d+3}m/n^{1/d}$ and assume that

$$3^{d+3}m \ge 2^{d+3}n^{1/d}. \tag{2.4}$$

We easily verify that $2^{d+3} \le r \le m$, for n large enough. Assume that the balls $B(p_i, r)$ $(1 \le i \le n)$ are pairwise disjoint. The combined volume of the n balls cannot exceed the volume of, say, $B(p_1, +\infty)$. Using Lemma 6 and relation (2.3), we find that $nr^d/3^{d^2} < (3+d)m^d$, which gives a contradiction. It follows that two of the balls must intersect. By the triangular inequality, the distance between their centers is at most $2r$. Suppose now that (2.4) does not hold. We can assume that m is larger than any appropriate constant. Otherwise, for n large enough, some j-face of the arrangement $(j \le d)$ will contain at least two sites p_i and p_j, and hence $\delta(p_i, p_j) = 0$. We need to evaluate the maximum number $f(d, m)$ of d-regions in an arrangement of m hyperspheres in S^d. We have the recurrence $f(1, m) = 2m$ and $f(d, m) = f(d, m-1) + f(d-1, m-1)$. Using a path-counting method (e.g., Monier [7]), we easily find that

$$f(d, m) = 2\sum_{0\le k\le d} \binom{m-1}{k} \le 2(1 + m^d).$$

Let Φ be the maximum number of d-regions in an arrangement of m hyperspheres in E^d. For n large enough, we have

$$\Phi \le f(d, m) + 1 < \frac{2^{d(d+3)+1}n}{3^{d(d+3)}} + 3.$$

For Φ large enough, we have $n > \Phi$ and, therefore, at least two sites lie in the closure of the same d-face of the arrangement. Their distance to each other in the metric δ is null. ∎

As we said earlier, relaxing the assumptions of Lemma 7 is identical to the hyperplane case. The same reasoning leads to the generalization of Lemma 4. We conclude with the main results of this section.

Theorem 8. *Any set of n points in E^d can be made the vertices of a polygonal curve which any hypersphere can split through only $O(n^{1-1/d})$ edges. This upper bound is optimal in the worst case.*

Theorem 9. *Spherical range searching on n points in E^d can be performed in $O(n^{1-1/d}\alpha(n))$ query time and $O(n)$ storage in the arithmetic model of computation.*

What happens if we have a more realistic model of computation such as a random access machine or a pointer machine? Let us look at the two-dimensional case. Given n points in E^2, count how many points lie inside a query circle. By lifting the problem to E^3 we reduce it to intersecting a query plane with a polygonal curve $C = (p_1, \ldots, p_n, p_1)$. This reduction was made in (Yao [10]): map the site $(x, y, 0)$ to the point (x, y, z), where $z = x^2 + y^2$, and map the query circle $(x-a)^2 + (y-b)^2 = r^2$ to the plane $z = a(2x - a) + b(2y - b) + r^2$. The sites contained in a query circle are precisely those mapping below the plane associated with the circle.

The question is now: Given a plane π, find all the edges of C split by π. To do so, we set up a complete binary tree of n leaves. The leaves are associated with $p_1, \ldots, p_n$, from left to right. Each internal node v of the tree is associated with the set $S(v)$ consisting of all the points whose corresponding leaves are descendents of v. The idea is to store in v a pointer to the convex hull of $S(v)$. Using a linear-size data structure (Dobkin and Kirkpatrick [3]) we can check if a query hyperplane separates the set $S(v)$ in $O(\log n)$ time. We say that a plane π *separates* a set of points if both connected components of $E^3 \setminus \pi$ contains at least one of the points. Using straightforward arguments, we conclude to the existence of an $O(n \log n)$-size data structure for computing all k intersections between C and a query plane in time $O((k+1)\log^2 n)$. In light of Theorem 8, we easily derive

Theorem 10. *Spherical range searching on n points in E^2 can be performed in $O(\sqrt{n}\log^2 n)$ query time and $O(n \log n)$ storage on a random access machine or a pointer machine.*

4. Polygon Range Searching

We now consider the problem of counting the number of points in a convex k-gon K (the problems of reporting or summing up weights can be handled the same way and give similar upper bounds). As in the previous section, we assume that the underlying model of computation is a random access machine or a pointer machine. Let $p_1, \ldots, p_n$ be the input points arranged in the order provided by the curve of Theorem 8. This gives us an n-gon Π which, unfortunately, might not be simple. It is obvious, however, that the diagonal-switching trick of the traveling salesman problem will make the polygon Π simple without increasing the stabbing number. The idea is to take all intersecting edges (a, b) and (c, d) and replace them by either (a, c) and (b, d), or (a, d) and (b, c), whichever choice keeps the polygonal line connected.

It has been shown in (Chazelle and Guibas [2]) that ray-shooting can be performed in Π in $O(\log n)$ time, using $O(n)$ space. (Ray-shooting means finding the first hit of a ray directed towards Π.) This allows us to find the intersections of Π and K in time proportional to $k\sqrt{n}\log n$, which gives us the desired interval decomposition of the vertices of Π. From there, we complete the computation in an overall time of $O(k\sqrt{n}\log n)$. This is a simple exercise, so let us move on.† The factor $k\sqrt{n}$ is an asymptotic upper bound on the maximum number of intersections between the boundaries of Π and K. We will show below that a clever choice of Π can limit this number to $O(\sqrt{kn})$.

Let k be a fixed integer between 1 and n. We subdivide the set of n points into k subsets of size $\lceil n/k \rceil$ or less. The first subset includes the $\lceil n/k \rceil$ leftmost points, the second subset includes the points whose x-coordinates have ranks ranging between $\lceil n/k \rceil + 1$ and $2\lceil n/k \rceil$, etc. Apply Theorem 8 to each subset and turn the polygonal curves into simple polygons $C_1, \ldots, C_k$.

We claim that the edges of any convex k-gon K can split only a total of $O(\sqrt{kn})$ edges among $C_1, \ldots, C_k$. Why is that so? Imagine $k-1$ vertical lines separating the C_i's. We can cut up edges of K so as to fit between two consecutive vertical lines without introducing more than $2(k-1)$ new edges. Then we can ray-shoot each edge separately and collect the pieces in the obvious way. Our claim follows trivially, as well as the fact that the query time is $O(\sqrt{kn}\log n)$. Note that the convexity of K is used only to keep the number of preprocessing cuts small. The same method works if, say, K is monotone in a fixed direction. By building $\lfloor \log n \rfloor$ data structures, one for each value of $k = 2^i \leq n$, we achieve the following result.

Theorem 10. *It is possible to preprocess n points in E^2, using $O(n\log n)$ storage, so that given any convex k-gon K ($k \leq n$), the number of points inside K can be computed in time $O(\sqrt{kn}\log n)$ on a random access machine (or a pointer machine). The same result holds if the polygon K is monotone in a fixed direction.*

5. A Proof of Optimality

We have shown that for a fixed value of k, a set of n points in E^2 admits a spanning tree of stabbing number $O(\sqrt{kn})$ with respect to polygon range searching. We can generalize an argument of Welzl [8] to prove the optimality of this result in the asymptotic sense. We construct a set of n points by carefully arranging k building blocks. A building block is any set affinely equivalent to the m^2 vertices of an m-by-m square grid, with $m = \lfloor \sqrt{n/k} \rfloor$. Given a direction ℓ and a small angle θ, we define an *(ℓ, θ)-block* as a building block whose diagonal has direction ℓ and whose grid axes form an angle θ (Figure 1).

† If we are shooting towards a vertex with both adjacent edges on the same side of the shooting line, we should not report that intersection, or we might expose ourselves to big trouble in case we have a linear number of collinear vertices.

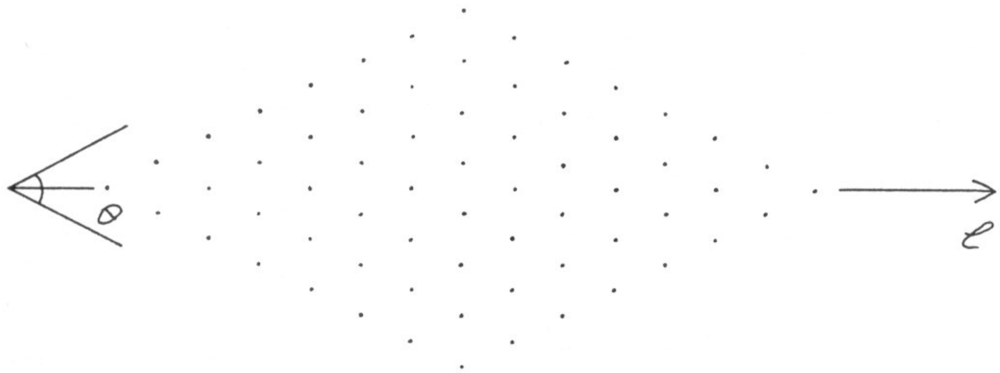

Figure 1. An (ℓ, θ)-block is affinely equivalent to an m-by-m grid: the angle between its axes is θ and its long diagonal is parallel to ℓ.

Now take a regular $9k$-gon centered at the origin and pick k consecutive edges in the north-east quadrant. On each edge chosen, place a small (ℓ, θ)-block, where ℓ is the direction of the edge and θ is a very small angle, say, $\pi/9^{9^k}$. The block should extend over, say, a ninth of the edge (Figure 2). We define the *canonical lines* of the grid $\{(i,j) \mid 0 \leq i,j \leq m-1\}$ to be the $m+1$ horizontal lines $y = i - 1/2$ $(0 \leq i \leq m)$ and the $m+1$ vertical lines $x = i - 1/2$ $(0 \leq i \leq m)$. The two extreme vertical and horizontal lines form a square enclosing the grid, which we call its *box*. By affinity, every block has a box and a set of canonical lines.

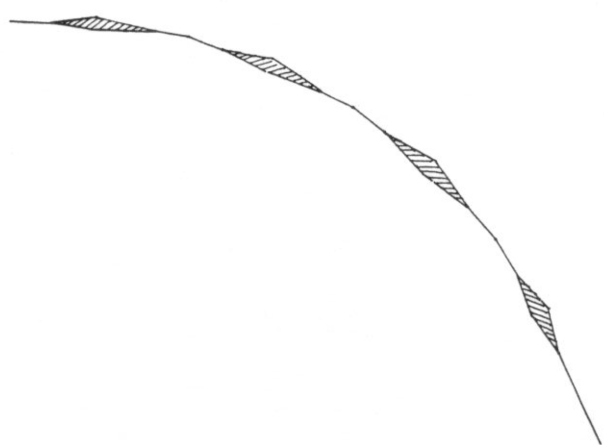

Figure 2. The set of sites is obtained by arranging k blocks along a regular convex polygon.

Let T be a spanning tree of the set of km^2 sites. Consider the sites of one of the blocks. Each of them is entirely separated from the other sites by canonical lines. On the other hand, each site is connected to at least one other site by an edge of T. This implies that the $2(m+1)$ canonical lines of the block have a total of at least $m^2/2$ intersection points with T. By the pigeon-hole principle, one canonical line intersects at least $m/5$ edges of T (for m large enough). Note that the intersections lie in the box of the block. Let us mark this canonical line and do the same for each block. The crucial observation is that these chosen lines form the edges of an unbounded convex k-gon K, each line contributing exactly one edge. Furthermore, all the intersections with T associated with each canonical line happen to lie on its contributed edge. This implies that the boundary of K splits at least $mk/5$ edges of T, which is on the order of $\sqrt{kn}$.

Theorem 11. *Let k and n be two integers, with $k \leq n$. Any set of n points in E^2 can be made the vertices of a polygonal curve such that the boundary of any convex k-gon can split only $O(\sqrt{kn})$ edges. This upper bound is optimal in the worst case.*

6. Conclusions

We have carefully evaded the issue of preprocessing. It is elementary to check that all the data structures given in this paper can be contructed in polynomial time. The real challenge is to determine how efficiently the construction can be made to be. But perhaps the most interesting open problem is to make the result of Theorem 5 hold on a random access machine.

Appendix

Below is a definition of the functional inverse of Ackermann's function, denoted $\alpha(n)$. Let $A(i,j)$ be the function defined recursively as follows:

$A(0,j) = 2j$, for any $j \geq 0$.
$A(i,0) = 0$ and $A(i,1) = 2$, for any $i \geq 1$.
$A(i,j) = A(i-1, A(i,j-1))$, for any $i \geq 1$ and $j \geq 2$.

We define $\alpha(n) = \min\{\, i \mid i \geq 1, A(i,i) > n \,\}$. For any $m \geq n \geq 1$, we also define the function $\alpha(m,n)$ by $\alpha(m,n) = \min\{\, i \mid i \geq 1, A(i, 4\lceil m/n \rceil) > \log n \,\}$. Yao [9] has given a linear-size data structure for partial sum computation: each query is answered in time $O(\alpha(cn,n))$, where c is an appropriate constant. We easily check that $\alpha(cn,n) = O(\alpha(n))$, therefore we are justified to say that the query time is $O(\alpha(n))$.

REFERENCES

1. Chazelle, B. *Polytope range searching and integral geometry*, Proc. 28th Ann. IEEE Sympos. Found. Comput. Sci. (1987), 1–10.
2. Chazelle, B., Guibas, L.J. *Visibility and intersection problems in plane geometry*, Proc. 1st Ann. ACM Sympos. Comput. Geom. (1985), 135–146. To appear in *Disc. Comput. Geom.*
3. Dobkin, D.P., Kirkpatrick, D.G. *Fast detection of polyhedral intersection*, Theoret. Comput. Sci. 27 (1983), 241–253.
4. Edelsbrunner, H. *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, Germany, 1987.
5. Edelsbrunner, H., Guibas, L.J., Hershberger, J., Seidel, R., Sharir, M., Snoeyink, J., Welzl, E. *Implicitly representing arrangements of lines or segments*, Proc. 4th Ann. ACM Sympos. Comput. Geom. (1988), 56–69.
6. Fredman, M.L. *Lower bounds on the complexity of some optimal data structures*, SIAM J. Comput. 10 (1981), 1–10.
7. Monier, L. *Combinatorial solutions of multidimensional divide-and-conquer recurrences*, J. Algorithms 1 (1980), 60–74.
8. Welzl, E. *Partition trees for triangle counting and other range searching problems*, Proc. 4th Ann. ACM Sympos. Comput. Geom. (1988), 23–33.
9. Yao, A.C. *On the complexity of maintaining partial sums*, SIAM J. Comput. 14 (1985), 277–288.
10. Yao, F.F. *A 3–space partition and its applications*, Proc. 15th Ann. ACM Sympos. Theory Comput. (1983), 258–263.