Can Nearest Neighbor Searching be Simple and Always Fast?

Victor Alvarez[1]*, David G. Kirkpatrick[2], and Raimund Seidel[3]

[1] Fachrichtung Informatik, Universität des Saarlandes, alvarez@cs.uni-saarland.de
[2] Department of Computer Science, University of British Columbia, kirk@cs.ubc.ca
[3] Fachrichtung Informatik, Universität des Saarlandes, rseidel@cs.uni-saarland.de

Abstract. Nearest Neighbor Searching, *i.e.* determining from a set S of n sites in the plane the one that is closest to a given query point q, is a classical problem in computational geometry. Fast theoretical solutions are known, e.g. point location in the Voronoi Diagram of S, or specialized structures such as so-called Delaunay hierarchies. However, practitioners tend to deem these solutions as too complicated or computationally too costly to be actually useful.

Recently in ALENEX 2010 Birn *et al.* proposed a simple and practical randomized solution. They reported encouraging experimental results and presented a partial performance analysis. They argued that in many cases their method achieves logarithmic expected query time but they also noted that in some cases linear expected query time is incurred. They raised the question whether some variant of their approach can achieve logarithmic expected query time in all cases.

The approach of Birn *et al.* derives its simplicity mostly from the fact that it applies only one simple type of geometric predicate: which one of two sites in S is closer to the query point q. In this paper we show that any method for planar nearest neighbor searching that relies just on this one type of geometric predicate can be forced to make at least $n-1$ such predicate evaluations during a worst case query.

1 Introduction

Nearest Neighbor Searching is a classical problem in computational geometry. In the simplest non-trivial case it asks to preprocess a set S of n sites in the plane so that for any query point q the site in S that is closest to q can be determined quickly. The problem appears already in Shamos' 1975 seminal paper on "Geometric Complexity" [9]. Already there the problem is reduced to point location in the Voronoi diagram of S, which was then solved in logarithmic time, although using quadratic space, by what is now commonly known as the Dobkin-Lipton slab method. Of course later on various logarithmic time, linear space methods for planar point location were proposed, e.g. [6, 4], providing a solution to the Nearest Neighbor Searching problem that was completely satisfactory

* Partially supported by CONACYT-DAAD of México.

and asymptotically optimal from a theoretical point of view. However, practitioners have considered these solution too complicated or the constants hidden in the asymptotics too large and have frequently employed solutions based on kd-trees [1, 8] that do not guarantee optimal worst case performance but appear to be very fast in practice.

Recently Birn *et al.* [2] proposed a new randomized method for Nearest Neighbor Searching that is simple and elegant and appears to be fast in practice in many cases. The method does not rely on a straightforward reduction of Nearest Neighbor Searching to planar point location. Birn *et al.* presented an analysis showing that under certain plausible circumstances their method achieves logarithmic expected query time. However they also noted that in some circumstances their method incurs linear expected query time. They also reported on some attempts to modify their approach to achieve logarithmic query time in all cases, while maintaining simplicity and practicality.

Our paper shows that such attempts must be futile unless additional types of geometric predicates are employed or the input is restricted in some form. The method of Birn *et al.* derives some of its simplicity from the fact that they use only one type of geometric predicate: which of the two sites $a, b \in S$ is closer to the query point q? Or, in other words, where does q lie in relation to the perpendicular bisector $\mathcal{L}_{a,b}$ of sites a and b.

In this paper we consider the problem of Nearest Neighbor searching under the condition that the only geometric predicates used are such tests against perpendicular bisectors between sites, let us call them *bisector tests.* Note that all the fast planar point location methods when applied to Voronoi diagrams use besides such bisector tests (which typically are just tests against Voronoi edges) also some other tests, typically how a query point lies in relation to a vertical or horizontal line through a Voronoi vertex.

Is it possible to achieve fast Nearest Neighbor Searching by just using bisector tests? At first you are inclined to answer no: consider a set S of n sites with one site having a Voronoi region V_a with $n-1$ edges. Testing whether a query point q is contained in this region seems to require a test against each of its $n-1$ edges. But on further thought you notice that V_a can be crossed by many bisectors and containment of q in V_a could perhaps be certified by relatively few tests of q against such bisectors.

In this paper we show that this is in general not the case.

Theorem 1. *There is a set S of n sites in the plane so that any method for Nearest Neighbor Search in S that relies exclusively on bisector tests (*i.e. *testing the location of the query point against the perpendicular bisector between two sites $a, b \in S$) can be forced to use $n-1$ such tests in the worst case.*

At first sight this theorem seems to convey only a negative message: With bisector tests alone you cannot do guaranteed fast Nearest Neighbor Searching. But there is also a positive take to this, namely advice to the algorithm designer: You want to do guaranteed fast Nearest Neighbor Searching? Then you better use other geometric predicates besides bisector tests, or — your algorithm or its analysis should assume small coordinate representations for the sites and must exploit this assumption. The latter alternative derives from the fact that the set of sites S constructed in the proof of Theorem 1 has exponential spread, i.e. the ratio of smallest and largest inter-site distance in S is exponential in n. This exponential behavior does not seem to be a fluke of our method but in some sense seems inherent in such a lower bound construction.

2 Preliminaries

All of the following will be set in the plane $\mathbb{R}^2$. By *comparing a (query) point q against a (test) line L* we mean determining whether q lies on L or, if not, which of the two open halfplanes bounded by L contains q. We will denote the halfplane that contains q by $L[q]$. We will use the expression "testing q against L" synonymously with "comparing q against L."

Let P be a convex polygon with n edges and let $\mathcal{T}$ be a finite set of lines. We are interested in the computational complexity of methods that ascertain containment of a query point q in polygon P by just comparing q against testlines in $\mathcal{T}$. We will measure this complexity only in terms of number of comparisons of the query point against lines in $\mathcal{T}$. We will completely ignore all other costs of querying, all costs incurred by preprocessing, and also all space issues.

Let us consider a few examples: If $\mathcal{T}$ consists of all lines that support edges of P, then determining whether $q \in P$ can be achieved by comparing q against each line in $\mathcal{T}$. If one of these comparisons is omitted, say the one against the line that supports edge e, then you cannot be sure of actual containment in P since a query point q' close to edge e but outside P would show the same comparison outcomes as a point q close to edge e but inside P. If, on the other hand, $\mathcal{T}$ consists of all lines through pairs of vertices of P then for any query point q only $\log_2 n + O(1)$ comparisons against lines in $\mathcal{T}$ will suffice to determine whether $q \in P$ or not – essentially you just need to perform a binary search of q among all lines in $\mathcal{T}$ that contain the same vertex.

We want to say that a test set $\mathcal{T}$ is k-bad for polygon P if no method that determines whether a query point q is contained in P and that employs just comparisons of q against lines in $\mathcal{T}$ can always give correct answers using fewer than k such comparisons. More formally we say that a test set $\mathcal{T}$ is k-bad for P if there is a point c so that for every $\mathcal{S} \subset \mathcal{T}$ with $|\mathcal{S}| < k$ we have that $\bigcap\{L[c] | L \in \mathcal{S}\}$ intersects P as well as its complement $\overline{P}$.

Lemma 1. *If a test set $\mathcal{T}$ is k-bad for a polygon P, then any method that determines membership of a query point in P by just using comparisons against lines in $\mathcal{T}$ can be forced to make at least k such comparisons.*

Proof. Let c be the point mentioned in the definition of k-badness of $\mathcal{T}$. We will use an adversary argument with the following strategy: during a query answer all line comparisons as if c were the query point. Assume $k-1$ tests had been made during a query and $\mathcal{S}$ were the set of lines in $\mathcal{T}$ against which the query point was compared. Let q and q' be points in $\bigcap\{L[c] | L \in \mathcal{S}\}$ with $q \in P$ and $q' \notin P$. Such points must exist by the definition of k-badness. For the query algorithm the points q and q' are indisdinguishable from c, since all three points behave the same on all comparisons against lines in $\mathcal{S}$. But since $q \in P$ and $q' \notin P$ the query needs at least one more comparison against some line in $\mathcal{T}$ in order to produce a correct answer.

Let us call a line L a *shaving line* of polygon P if one of the two closed halfspaces bounded by L contains exactly one edge e of P completely. We say in this case that L *"shaves off e."* Note that every line that contains an edge of P is a shaving line of P.

Lemma 2. *Let P be a convex polygon with $n > 6$ sides and let $\mathcal{T}$ be a set of shaving lines of P. The set $\mathcal{T}$ is $\lfloor n/2 \rfloor$-bad for P.*

Proof. Let $\mathcal{T}$ be such a set of shaving lines. For $L \in \mathcal{T}$ let L^+ be the open halfspace bounded by L that does **not** contain the edge shaved off by L. The halfspace L^+ contains $n - 2 > 4$ vertices. Thus, since $n > 5$ for any three lines $L_1, L_2, L_3 \in \mathcal{T}$ the intersection $L_1^+ \cap L_2^+ \cap L_3^+$ must be non-empty. Thus by Helly's theorem the intersection $\bigcap\{L^+ | L \in \mathcal{T}\}$ is non-empty. Let c be a point in this intersection.

Now let $\mathcal{S} \subset \mathcal{T}$ with $|\mathcal{S}| = k - 1 < \lfloor n/2 \rfloor$. There must be two consecutive edges of P so that neither is shaved off by any line in $\mathcal{S}$. Let v be the vertex[4] joining those two edges. Certainly $\bigcap\{L[c] | L \in \mathcal{S}\}$ contains c but also v and also an entire neighborhood of v, and therefore also points that are in P and points that are not in P. Thus $\mathcal{T}$ is $\lfloor n/2 \rfloor$-bad for P.

Lemma 2 allows to prove $\Omega(n)$ lower bounds for anwering point containment queries in an n-sided convex polygon using comparisons against a set of test lines. We want to strengthen this to be able to claim that in the worst case actually at least n such comparisons are needed.

We will strengthen the notion of "shaving" to "closely shaving." For this purpose pick from the edge e of P some point m_e in its relative interior, and we will refer to m_e as the *midstpoint of* e and we will denote the set of all chosen midstpoints, one for each edge, by M_P. We will call a line L that shaves off edge e of P *closely shaving* iff the closed halfplane bounded by L that contains e contains no midstpoint except for m_e, or, in other words, the open halfspace bounded by L that does not contain e contains all midstpoints except for m_e.

Lemma 3. *For $n > 6$ let P be an n-sided convex polygon with set M_P of chosen midstpoint, and let $\mathcal{T}$ be a set of closely shaving test lines.*

The set $\mathcal{T}$ is n-bad for P.

Proof. Let c be the point as in the proof of Lemma 2, and let $\mathcal{S}$ be a subset of $\mathcal{T}$ containing fewer than n lines. There must be some edge e of P that is not shaved off by any line in $\mathcal{S}$. Because of the closely shaving property the midstpoint m_e

[4] We ignore here the case that P is unbounded and those two edges are unbounded and hence v is a vertex "at infinity." This case can be taken care of either by arguing it separately or by just claiming the slightly weaker $(\lfloor n/2 \rfloor - 1)$-badness.

must be contained in the open halfplane $L[c]$ for each $L \in \mathcal{S}$ and the same must be true for an entire neighborhood of m_e. Thus $\bigcap\{L[c] | L \in \mathcal{S}\}$ contains m_e and also an entire neighborhood of m_e, and therfore also points that are in P and points that are not in P. Thus $\mathcal{T}$ is n-bad for P.

3 A Voronoi Diagram Construction

With the results of the preliminary section the strategy for proving the main Theorem 1 should be clear: Construct a set S of n sites, so that in its Voronoi diagram there is a Voronoi cell V_p with $n-1$ edges so that the set $\mathcal{L}_S$ of all perpendicular bisectors between pairs of points in S forms a set of closely shaving lines of V_p (for appropriately chosen midstpoints). Lemma 3 then immediately implies Theorem 1.

For our set S we will choose points from the non-negative part $\mathbb{P}$ of the unit parabola described by $\{u(t) | t \geq 0\}$ with $u(t) = (t, t^2)$. Let $0 \leq t_1 < t_2 < \cdots < t_n$ and let $S = \{u(t_i) | 1 \leq i \leq n\}$. It is well known that the structure of the Voronoi diagram of S is completely determined and independent of the actual choices of the t_i's. To see this note that any circle can intersect $\mathbb{P}$ in at most 3 points, which is a consequence of the fact that these intersection points are given by the non-negative roots of a polynomial $(a-t)^2 + (b-t^2)^2 - r^2$, which has coefficient 0 for t^3, but this coefficient is the sum of the four roots, and hence at most 3 of them can be non-negative (except for the uninteresting case that all are 0). If there are 3 intersection points between $\mathbb{P}$ and a circle, then $\mathbb{P}$ must cross the circle in those points and some parts of $\mathbb{P}$ must lie inside the circle. From this, and the fact that for every circle $u(t)$ is outside for sufficiently large t you can characterize which triples of points from S span Delaunay triangles: all triples of the form $(u(t_1), u(t_{i-1}), u(t_i))$ with $3 \leq i \leq n$. This is akin to the well known Gale's evenness condition for the description of the structure of cyclic polytopes [10, page 14].

The structure of the Voronoi diagram of S is now as follow: let V_i denote the Voronoi region of $u(t_i)$; the Voronoi region V_1 neighbors every region $V_2, V_3, \ldots, V_n$ in this counterclockwise order; V_2 neighbors V_1, V_3, for $3 \leq i < n$ the region V_i neighbors V_{i-1}, V_1, V_{i+1}, and V_n neighbors V_1, V_{n-1}. All Voronoi regions are unbounded (since S is in convex position). See Fig. 1 for an example.

For $a, b \geq 0$ consider the perpendicular bisector between the points $u(a)$ and $u(b)$. It is described by the equation

$$y = \frac{-1}{a+b}x + (a^2 + b^2 + 1)/2\,.$$

Note that its slope is always negative and for a or b sufficiently large the slope can be made arbitrarily close to 0.

Let $\mathcal{L}_{i,j}$ be the perpendicular bisector between $u(t_i)$ and $u(t_j)$. By the discussion above each bisector $\mathcal{L}_{1,j}$ contributes an edge e_j to the Voronoi region $P = V_1$. Our goal will be to show that the t_i's can be chosen so that for each

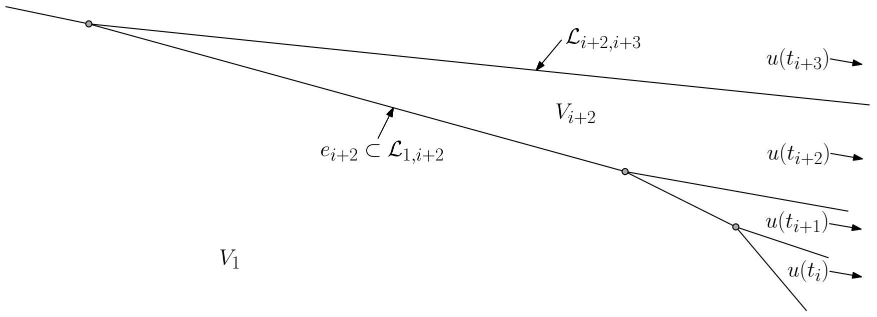

Fig. 1.

$j > 2$ each bisector $\mathcal{L}_{ij}$, with $1 \leq i < j$, closely shaves off edge e_j from polygon P (of course with an appropriate choice of midstpoints m_i on e_i, for $1 < i \leq n$).

We will prove this by induction on n. So assume that for some $n \geq 3$ values $0 = t_1 < t_2 < \cdots < t_n$ have been chosen and for each j with $1 < j \leq n$ a midstpoint m_j on Voronoi edge e_j has been chosen, so that for $1 \leq i < j$ the bisector $\mathcal{L}_{i,j}$ closely shaves off edge e_j from V_1. Now we want to choose $t_{n+1} > t_n$ so that the bisectors $\mathcal{L}_{i,j}$ with $1 \leq i < j \leq n$ remain closely shaving, and the "new" bisectors $\mathcal{L}_{i,n+1}$ with $1 \leq i \leq n$ are closely shaving as well, in particular they closely shave off the new edge e_{n+1} of V_1 contributed by $\mathcal{L}_{1,n+1}$.

Since "closely shaving off" is a local condition, in that it depends only on edge e_j and the midstpoints m_{j-1} and m_{j+1} all bisectors $\mathcal{L}_{i,j}$ with $i < j < n$ definitely remain closely shaving, and the bisectors $\mathcal{L}_{i,n}$ with $i < n$ remain closely shaving, provided the new Voronoi vertex v_n between e_n and e_{n+1} (which is the intersection of $\mathcal{L}_{1,n}$ and $\mathcal{L}_{1,n+1}$) is to the left of midstpoint m_n. It can easily be checked that the x-coordinate of v_n is given by $-(t_n^2 t_{n+1} + t_n t_{n+1}^2)/2$. Thus by making t_{n+1} large enough the Voronoi vertex v_n can be moved as far left on $\mathcal{L}_{1,n}$ as desired.

We further need that all the bisectors $\mathcal{L}_{i,n}$ with $i \leq n$ intersect the new Voronoi edge e_{n+1}. This happens if $\mathcal{L}_{1,n+1}$ has slope closer to 0 than the slopes of all the $\mathcal{L}_{i,n}$'s. Since the slope of $\mathcal{L}_{i,j}$ is given by $-1/(t_i + t_j)$ this can also be achieved by making t_{n+1} suitably large, in particular

$$t_{n+1} \geq t_n + t_{n-1}\,. \tag{1}$$

The midstpoint m_{n+1} for e_{n+1} can then be chosen as any point on e_{n+1} to the left of all of those intersections $\mathcal{L}_{i,n} \cap \mathcal{L}_{1,n+1}$.

Finally we need to ensure that "new" bisectors $\mathcal{L}_{i,n+1}$ with $1 < i \leq n$ intersect edge e_n between midstpoint m_n and the new Voronoi vertex v_n, moreover their slope should be closer to 0 than the slope of $\mathcal{L}_{1,n+1}$ so that no $\mathcal{L}_{i,n+1}$ can intersect e_{n+1}. Fortunately this slope condition holds automatically, since for $i > 1$ we have $t_i > t_1 = 0$.

Now consider the intersection $\mathcal{L}_{1,n} \cap \mathcal{L}_{i,n+1}$ for some i with $1 < i < n$. Its x-coordinate is given by

$$-\frac{t_n(t_{n+1}^3 + t_{n+1}^2 t_i + t_{n+1}(t_i^2 - t_n^2) + t_i^3 - t_n^2 t_i)}{2(t_i + t_{n+1} - t_n)},$$

which clearly can be made as small as desired by making t_{n+1} sufficiently large. Thus all these intersections can be moved to the left of m_n.

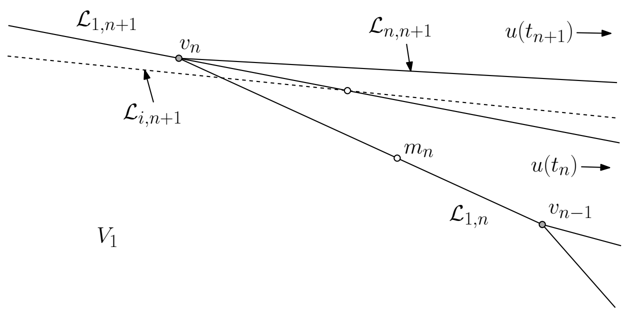

Fig. 2.

It remains to show that all these intersections occur to the right of v_n. Since the slope of $\mathcal{L}_{i,n+1}$ is closer to 0 than the slope of $\mathcal{L}_{1,n+1}$, it suffices to show that the intersections $\mathcal{L}_{1,n+1} \cap \mathcal{L}_{i,n+1}$ happen on $\mathcal{L}_{1,n+1}$ to the right of $v_n = \mathcal{L}_{1,n+1} \cap \mathcal{L}_{n,n+1}$. The x-coordinate of such an intersection is given by

$$-(t_i^2 t_{n+1} + t_i t_{n+1}^2)/2\,,$$

which, since $t_i < t_n$, is clearly larger than the x-coordinate of $v_n = \mathcal{L}_{1,n+1} \cap \mathcal{L}_{n,n+1}$, which is

$$-(t_n^2 t_{n+1} + t_n t_{n+1}^2)/2\,.$$

4 Exponentiality of the Construction

The Fibonacci type Inequality (1) implies that t_n, and hence some site coordinate in S, is at least exponential in n. Sites with such large coordinates do not seem to occur naturally in actual inputs, and thus our lower bound construction seems artificial and not really relevant "in practice." We therefore tried to alter our construction in order to avoid such exponential behavior. The most natural approach seemed to be to replace the parabola from which we chose the sites of S by some other curve. We tried several curves, e.g. $(t, t^{1+\varepsilon})$ for $t \geq 0$, or $(t, t \log t)$ for $t \geq 1$, or the hyperbola $(t, 1/t)$ for $t \geq 1$. Each admitted the same inductive construction we used in the proof in the previous section, but in each case we arrived at exponentially large coordinates (particularly bad in the case of the hyperbola). This raised the suspicion that this exponential behavior is inherent in this inductive construction. This turns out to be indeed the case.

Assume we choose S from some curve γ. By translation and rotation[5] we may assume without loss of generality that the curve starts at the origin, lies in the positive quadrant, is monotonically increasing and convex. Let $p_1, p_1, \ldots, p_n$ be the points chosen from the curve in their natural order with p_1 being the origin.

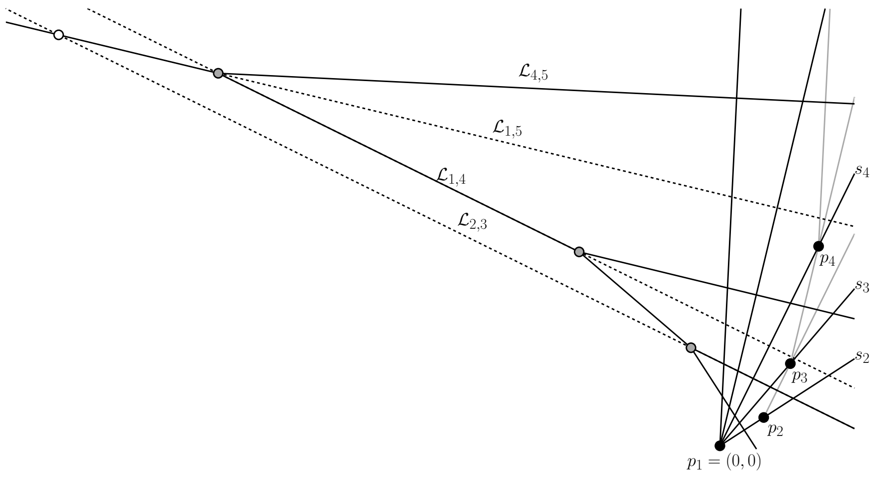

Fig. 3.

The construction of the previous section places a number of constraints on the p_i's. Here we will be interested in only one of them, namely the one that led to inequality (1). It requires that for each i the bisector $\mathcal{L}_{1,i+1}$ between p_1 and p_{i+1} has slope closer to 0 than the bisector $\mathcal{L}_{i-1,i}$ between p_{i-1} and p_i. Since the bisector $\mathcal{L}_{k,\ell}$ is normal to the line through p_ℓ and p_k this implies the following condition: for each i the line through p_{i-1} and p_i must have slope at most as large as the slope of the line between p_1 and p_{i+1}. If we set $p_i = (x_i, s_i \cdot x_i)$, i.e. the slope of the line through $p_1 = (0,0)$ and p_i is s_i, then this slope condition is expressed algebraically by

$$\frac{s_i x_i - s_{i-1} x_{i-1}}{x_i - x_{i-1}} \leq s_{i+1}\,,$$

which implies that

$$x_i \geq \frac{s_{i+1} - s_{i-1}}{s_{i+1} - s_i} x_{i-1}\,.$$

[5] Such a rotation can be realized with "small" rational numbers in the transformation matrix [3].

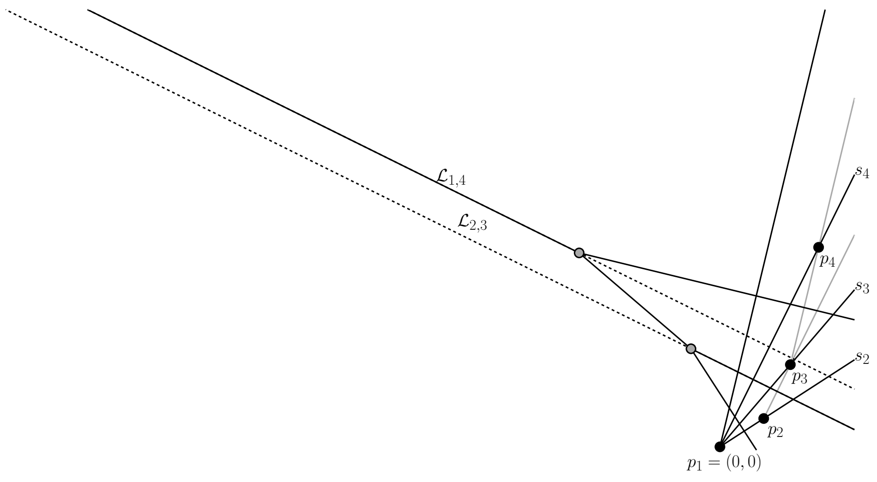

Fig. 4.

This must hold for $3 \leq i < n$. Now let d_i denote the slope difference $s_{i+1} - s_i$. We obtain that

$$x_i \geq \frac{d_i + d_{i-1}}{d_i} x_{i-1} = \left(1 + \frac{d_{i-1}}{d_i}\right) x_{i-1} \,,$$

and putting all these inequalities together we get that

$$x_{n-1} \geq \left(1 + \frac{d_2}{d_3}\right)\left(1 + \frac{d_3}{d_4}\right) \cdots \left(1 + \frac{d_{n-2}}{d_{n-1}}\right) x_2 \,. \tag{2}$$

Below we show that inequality (2) implies that x_{n-1}/x_2 or d_{n-1}/d_2 must be exponentially large. Both can be seen as constant sized rational expressions of the coordinates of p_2, p_3, p_{n-1}, and p_n. If those coordinates are rational numbers then therefore at least one of the involved numerators or denominators must be exponentially large also.

Lemma 4. *Let $D_0, D_1, \ldots, D_N$ be positive numbers and let*

$$P = \left(1 + \frac{D_0}{D_1}\right)\left(1 + \frac{D_1}{D_2}\right) \cdots \left(1 + \frac{D_{N-1}}{D_N}\right) \,.$$

Then $P \geq \varphi^N$ or $D_N/D_0 \geq \varphi^N$, where $\varphi = 1.618...$ is the golden ratio, the larger root of $x^2 = x + 1$.

Proof. It is an easy exercise to show that for $\prod_{1 \leq i \leq N} \rho_i = A$, with $\rho_i > 0$ for all i, the product $\prod_{1 \leq i \leq N}(1 + \rho_i)$ is minimized when all ρ_i are the same, i.e. $\rho_i = A^{1/N}$.

With $\rho_i = D_{i-1}/D_i$ we get that $\prod_{1\le i\le N} \rho_i = D_0/D_N$, and hence

$$P \ge \left(1 + (D_0/D_N)^{1/N}\right)^N .$$

Now let $X = (D_N/D_0)^{1/N}$. We have $P \ge (1 + 1/X)^N$ and $D_N/D_0 = X^N$. Clearly for $X \ge \varphi$ we have

$$D_N/D_0 \ge \varphi^N$$

.
For $0 < X \le \varphi$ we have

$$P \ge (1 + 1/X)^N \ge (1 + 1/\varphi)^N = \varphi^N,$$

since by the definition of φ we have $1 + 1/\varphi = \varphi$.

5 Conclusions and Remarks

We have shown that Nearest Neighbor Searching in the plane must be slow in the worst case if the only type of primitive predicate allowed is comparing the query point against bisectors of sites. This raises the question which additional type of predicates must be used in order to facilitate guaranteed fast query time. Certainly comparing query points against horizontal or vertical lines through Voronoi points would do the job. However these predicates are a bit more complicated than you would like, since algebraically they are realized by evaluating the sign of a degree-3 polynomial in the coordinates of the sites and the query point. Note that in contrast comparing a query point against a site bisector amounts to evaluating the sign of a degree 2 polynomial. There has been some work on reducing the degrees of the predicate polynomials that are used in a nearest neighbor query [5, 7], however they apply only to the case where the query points have integral coordinates.

In the context of this work an interesting question is whether comparing query points against horizontal or vertical lines through sites could be useful. Algebraically this is possibly the simplest of all imaginable predicates, since it amounts to the evaluation of a degree 1 polynomial. Unfortunately at this point we do not see any way of adapting our construction so that our lower bound also works if those types of predicates are allowed. We leave it as in interesting open problem to either prove that such predicates are not useful for nearest neighbor searching or to devise a method that profitably exploits such predicates.

Our lower bound constructions involve very large numbers. So it is conceivable that if all numbers involved have small rational representations then just bisector tests suffice for fast Nearest Neighbor Searching. At this point we do not see how such a restriction on the coordinates can be profitably exploited.

The results in this paper should be regarded as advice to the algorithm designer. If you want guaranteed fast Nearest Neighbor Search, then using just bisector tests cannot do the job, unless you make assumptions on your input coordinates and exploit these assumptions.

References

1. Jon L. Bentley, *Multidimensional Binary Search Trees used for Associative Searching*, Communications of the ACM, 18(9):509–517, 1975.
2. Marcel Birn, Manuel Holtgrewe, Peter Sanders, and Johannes Singler, *Simple and Fast*
   Nearest Neighbor Search, Proceedings of the Twelfth Workshop on Algorithm Engineering and Experiments (ALENEX), SIAM, 2010.
3. John F. Canny, Bruce R. Donald, and Eugene K. Ressler, *A Rational Rotation Method for Robust Geometric Algorithms*, Proc. of the Eigth ACM Symposium on Computational Geometry (SOCG), 251–260 (1992).
4. David G. Kirkpatrick, *Optimal Search in Planar Subdivisions*, SIAM J. Comput. Volume 12, Issue 1, pp. 28–35, 1983.
5. Giuseppe Liotta, Franco P. Preparata and Roberto Tamassia, *Robust Proximity Queries: An Illustration of Degree-driven Algorithm Design*, SIAM J. Comput. Volume 28, Issue 3, pp. 864–889, 1998.
6. Richard J. Lipton, Robert E. Tarjan, *Applications of a Planar Separator Theorem*, SIAM J. Comput. Volume 9, Issue 3, pp. 615–627, 1980.
7. David Millman and Jack Snoeyink, *Computing Planar Voronoi Diagrams in Double Precision: A Further Example of Degree-driven Algorithm Design*, Proceedings of the Annual Symposium on Computational Geometry (SoCG), 386–392, 2010.
8. David M. Mount and Sunil Arya. *ANN: A Library for Approximate Nearest Neighbor Searching.* CGC 2nd Annual Fall Workshop on Computational Geometry, 1997.
9. Michael I. Shamos, *Geometric Complexity*, Proceedings of Seventh Annual ACM Symposium on Theory of Computing (STOC), 224–233, 1975.
10. Günter M. Ziegler, **Lectures on Polytopes**, Springer 1995.