Algorithmica (1998) 22: 477–482

Algorithmica

A Note on Point Location in Delaunay Triangulations of Random Points[1]

L. Devroye,[2] E. P. Mücke,[3] and Binhai Zhu[4]

Abstract. This short note considers the problem of point location in a Delaunay triangulation of n random points, using no additional preprocessing or storage other than a standard data structure representing the triangulation. A simple and easy-to-implement (but, of course, worst-case suboptimal) heuristic is shown to take expected time $O(n^{1/3})$.

Key Words. Voronoi diagram, Delaunay triangulation, Point location, Probabilistic analysis of algorithms, Computational geometry.

1. Introduction and Main Result. Assume that we are given a Delaunay triangulation $\mathcal{D}$ of n points $X_1, \ldots, X_n$ in the plane, represented by one of the standard data structures for triangulations (see, e.g., Okabe et al., 1992). That is, in PASCAL terminology, the information is stored as points, edges, and triangles, linked by neighborhood information:

```
point:      RECORD            x,y: real
                              neighbors: edgelist END
edgelist:   RECORD            next: ↑edgelist
                              key: ↑edge END
edge:       RECORD            pt1,pt2: ↑point
                              tr1,tr2: ↑triangle END
triangle:   RECORD            ed1,ed2,ed3: ↑edge END
delaunay:   ARRAY[1..n] OF    ↑point
```

It should be noted that this is not the most space-efficient way to store planar triangulations (see, e.g., Guibas and Stolfi, 1985), but it is sufficient for the sake of this discussion. The crucial point is that $\mathcal{D}$ is represented with $O(1)$ storage per triangle, and no other structure is assumed on top of this simple graph-like object. The objective is to investigate how fast we can perform point location for a query point q without any further preprocessing of the data. We remark that to apply our point-location heuristic, the data structure must support $O(1)$ time access to a triangle from a neighboring triangle. Certainly, the above data structure supports this; moreover, almost all commonly

[1] The first author's work was supported by NSERC Grant A3456 and by FCAR Grant 90-ER-0291. Part of this work was done while the second and third authors were at CIC-3, Los Alamos National Laboratory.
[2] School of Computer Science, McGill University, Montreal, Quebec, Canada H3A 2A7.
[3] ANSYS, Inc., 201 Johnson Road, Houston, PA 15342-1300, USA.
[4] Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong.

Received November 27, 1997; revised February 15, 1998. Communicated by H. Prodinger and W. Szpankowski.

used data structures storing planar triangulations support this operation: DCEL (Doubly Connected Edge List) or simply a linked list of triangles with all the necessary local information (three vertices, three edges, and three pointers pointing to its neighboring triangles), etc.

A simple method is mentioned in the literature (see, e.g., Green and Sibson, 1978; Bowyer, 1981) as performing quite well in practice; however, a rigorous expected-time analysis was never given: start with one of the points of $\mathcal{D}$, say, $Y = X_j$, and "walk from neighbor to neighbor" across the triangulation toward the query point q. A recent implementation for spatial Delaunay triangulations (Mücke, 1993) uses the same method in three dimensions, together with a heuristic to find a good starting point, and reports it as extremely efficient in practice.

Here, we formalize this heuristic for the planar case, and analyze its expected time complexity. Note that $d(\cdot, \cdot)$ denotes the Euclidean distance between points, and $d(x, A)$ is $\inf_{y \in A} d(x, y)$ whenever A is a set.

STEP 1. Select m points $Y_1, \ldots, Y_m$ at random and without replacement from $X_1, \ldots, X_n$.

STEP 2. Determine the index j such that $d(Y_j, q)$ is minimal for $j \in \{1, \ldots, m\}$. Set $Y = Y_j$.

STEP 3. Locate the triangle containing q by traversing all triangles crossed by the line segment (Y, q).

Step 3 is easy to implement given the adjacency list implementation mentioned above. The time T taken by the algorithm is $\Theta(m)$ (Steps 1 and 2), plus $\Theta(k)$, where k is the number of triangles crossed by the line segment (Y, q) in Step 3. Here, the Θ, O, and Ω notation is used as in standard textbooks on data structures (see, e.g., Cormen et al., 1990). Since Y is random, T is random as well. Our main result is the following:

THEOREM. *If $X_1, \ldots, X_n$ are independently drawn from a distribution with the uniform density f on a convex compact set $C \subseteq \mathbb{R}^2$ of unit area, $m = o(n)$, and $m \to \infty$, and if the query point q is independent of $X_1, \ldots, X_n$ and is at distance at least $2\sqrt{\log n/m}$ from the boundary ∂C of C, then the expected time of the simple algorithm given above is bounded by*

$$c_1 m + c_2 \sqrt{n/m},$$

where $c_1, c_2 > 0$ are universal constants depending upon the geometrical properties of C only. In particular, the expected time is $O(n^{1/3})$ if $m = \lceil n^{1/3} \rceil$ and q is at least distance $2\sqrt{\log n}/n^{1/6}$ away from ∂C.

As a technical report, the present paper predated a later paper by Mücke et al. (1996), who generalized the present results to three dimensions at the expense of an extra polylog factor in the complexities. The latter factor became necessary after noting that Delaunay triangulations in dimensions greater than two are not planar. Throughout the present paper, c_i, $i \geq 1$, denote positive constants depending upon the geometrical properties of C.

2. Proof of the Theorem. The proof rests on the following lemma, which can be thought of as the simplification of Theorem 2 of Mücke et al. (1996), which in turn uses an argument from Bose and Devroye (1997).

LEMMA. *Let C and $X_1, \ldots, X_n$ be as in the theorem. If $\mathcal{L}$ is a fixed line segment of length $|\mathcal{L}|$ and is at distance $\geq 3\sqrt{\log n/n}$ from the boundary of C, and if $\mathcal{L}$ is independent of $X_1, \ldots, X_n$, then the expected number of triangles or edges of the Delaunay triangulation for $X_1, \ldots, X_n$ crossed by $\mathcal{L}$ is bounded by*

$$c_3 + c_4|\mathcal{L}|\sqrt{n},$$

where c_3, c_4 are universal positive constants not depending upon $\mathcal{L}$ or n.

To use this lemma for a random line segment $\mathcal{L}$, we must first make sure that $\mathcal{L}$ is independent of $X_1, \ldots, X_n$. This is not the case here. For this reason, we make a small detour. Let $\mathcal{D}$ be the Delaunay triangulation for $X_1, \ldots, X_n$, and let $\mathcal{D}_m$ be the Delaunay triangulation for $\{X_1, \ldots, X_n\} - \{Y_1, \ldots, Y_m\}$. Then $\mathcal{L} = (Y, q)$, the line segment connecting Y and q, is independent of the $n - m$ data points defining $\mathcal{D}_m$. We need to make sure that $\mathcal{L} = (Y, q)$ is at distance $\geq 3\sqrt{\log n/n}$ from ∂C, under the assumption that $d(q, \partial C) \geq \xi\sqrt{\log n/n}$, where $\xi \geq 6$ may depend upon n. This follows from the convexity of C, the triangle inequality, and $d(Y, q) \leq (\xi/2)\sqrt{\log n/n}$. We show below that this claim holds with high probability.

Let B be a probability event defined as $B \stackrel{\text{def}}{=} \{d(Y, \partial C) \geq 3\sqrt{\log n/n}\}$. Let I_Q be the indicator function for a probability event Q. Let N denote the number of triangles in $\mathcal{D}_m$ crossed by $\mathcal{L}$. We have $\mathbf{E}\{N\} = \mathbf{E}\{NI_B\} + \mathbf{E}\{NI_{\bar{B}}\}$ where $\bar{B}$ denotes the complement of B. Here $\mathbf{E}\{NI_B\}$ ($\mathbf{E}\{NI_{\bar{B}}\}$) denotes the number of triangles in $\mathcal{D}_m$ crossed by $\mathcal{L}$ when the event B ($\bar{B}$) occurs. We provide upper bounds for the two terms on the right-hand side.

We begin with $\mathbf{E}\{NI_{\bar{B}}\}$. Because of the planarity of two-dimensional Delaunay triangulations, $N \leq 3n$. Hence

$$\mathbf{E}\{NI_{\bar{B}}\} = \mathbf{E}\{N|\bar{B}\}\mathbf{P}\{\bar{B}\} \leq 3n\mathbf{P}\{\bar{B}\}.$$

By the triangle inequality,

$$\begin{aligned}
\mathbf{P}\{\bar{B}\} &\leq \mathbf{P}\{d(Y, q) \geq (\xi/2)\sqrt{\log n/n}\} \\
&= (\mathbf{P}\{d(Y_1, q) \geq (\xi/2)\sqrt{\log n/n}\})^m \\
&= (1 - \mathbf{P}\{d(Y_1, q) < (\xi/2)\sqrt{\log n/n}\})^m \\
&\leq \exp(-m\mathbf{P}\{d(Y_1, q) < (\xi/2)\sqrt{\log n/n}\}) \\
&\leq \exp(-\xi^2\pi m \log n/(4n)).
\end{aligned}$$

Thus,

$$\mathbf{E}\{NI_{\bar{B}}\} \leq 3ne^{-\xi^2\pi m \log n/(4n)}.$$

We now turn to $\mathbf{E}\{NI_B\}$. On B, $d^2(Y, q)\pi$ is the probability contents of the circle at q of radius $d(Y, q)$, and is therefore distributed as the minimum of m i.i.d. (independently identically distributed) uniform $[0, 1]$ random variables, which we call Z. Clearly,

$\mathbf{E}\{Z\} = 1/(m+1)$. Our lemma, along with the fact that $\{X_1, \ldots, X_n\} - \{Y_1, \ldots, Y_m\}$ constitutes an i.i.d. set of random variables implies that

$$\begin{aligned}
\mathbf{E}\{NI_B\} &\le c_3\mathbf{P}\{B\} + c_4\sqrt{n-m}\,\mathbf{E}\{d(Y,q)I_B\} \\
&\le c_3 + c_4\sqrt{n}\,\mathbf{E}\{d(Y,q)I_B\} \\
&\le c_3 + c_4\sqrt{n}\sqrt{\mathbf{E}\{d^2(Y,q)I_B\}} \\
&\quad \text{(by the Cauchy–Schwarz inequality)} \\
&\le c_3 + c_4\sqrt{n}\sqrt{\mathbf{E}\{(Z/\pi)I_B\}} \\
&\le c_3 + c_4\sqrt{n}\sqrt{\mathbf{E}\{Z/\pi\}} \\
&= c_3 + c_4\sqrt{n/(\pi(m+1))}.
\end{aligned}$$

Thus, the expected number of triangles or edges of $\mathcal{D}_m$ crossed by $\mathcal{L}$ is

$$\mathbf{E}\{N\} \le 3ne^{-\xi^2\pi m \log n/(4n)} + c_3 + c_4\sqrt{n/(\pi(m+1))},$$

where we recall that $\xi \ge 6$ can still be selected by us. The number of triangles or edges of $\mathcal{D}$ crossed by $\mathcal{L}$ is not more than that for $\mathcal{D}_m$ plus the sum S of the degrees of $Y_1, \ldots, Y_m$ in the Delaunay triangulation $\mathcal{D}$. To see this, note that $\mathcal{L}$ either crosses a triangle without one of the Y_i's as a vertex (in which case the triangle is identical in $\mathcal{D}$ and $\mathcal{D}_m$) or with one of the Y_i's as a vertex. The total number of the latter kind of triangles does not exceed S. The expected value of S is, by symmetry, m times the expected degree of Y_1. By the planarity of $\mathcal{D}$, we know that the sum of all degrees of $X_1, \ldots, X_n$ is twice the number of edges, which does not exceed $6n$. Therefore, the expected degree of X_1 or Y_1 does not exceed 6. Combining all this shows that

$$\mathbf{E}\{T\} \le O(m) + 3ne^{-\xi^2\pi m \log n/(4n)} + c_4\sqrt{n/(\pi(m+1))}.$$

Now, take $\xi = \max(6, \sqrt{12n\log(3n)/m\pi\,\log n})$ to make the second term at most 1. The bound then becomes $O(m + \sqrt{n/m})$. Note that with this choice of ξ, the condition on q becomes

$$d(q, \partial C) \ge \max(6\sqrt{\log n/n}, \sqrt{12\log(3n)/m\pi}).$$

If $m = o(n)$, for n large enough, this is equivalent to $d(q, \partial C) \ge \sqrt{12\log(3n)/m\pi}$. This in turn is implied for large n by $d(q, \partial C) \ge 2\sqrt{\log n/m}$. This concludes the proof of the theorem.

3. Remarks. 1. The theorem above is easily generalized to arbitrary densities f bounded away from 0 and ∞ on C, and the area of C does of course not need to be one. However, these trivial points make the proofs less readable. The constants c_1 and c_2 would then also depend upon the area of C, and the upper and lower bounds of f on C. The boundary condition on q would also change by a constant factor.

2. In Delaunay triangulations, the boundary effect is considerable, and requires conditions such as the ones seen in the theorem. The boundary effect was circumvented by Bern et al. (1991), in the analysis of the maximal degree in a Delaunay triangulation, by

considering an infinite Poisson point process in the plane and studying its restriction to finite sets.

3. The theorem may also be used to obtain a very simple on-line algorithm for insertion and deletion in a Delaunay triangulation with $O(n^{1/3})$ expected time per operation. This result uses the fact that after a triangle is located, we can find out in which Voronoi cell q falls, and update the local structure in time bounded by the number of faces of the Voronoi cell. Now, the expected number of faces of the Voronoi cell to which a random q belongs (drawn independently and according to the density f from which the data were drawn) is $O(1)$. Clearly, the expected complexity is eclipsed by the $O(\log n)$ expected-time fully dynamic algorithm (Devillers et al. 1992), but the data structure is also less complicated. We would also like to point out that when a query point is very close to the boundary of the triangulation the bound in this theorem might vary. In fact, the trivial linear bound is the only known one when q is very close to the boundary.

4. Using the given point-location scheme, a Delaunay triangulation can be constructed in $O(n^{4/3})$ expected time. Again, this is theoretically slower than some well-known $O(n \log n)$ algorithms (Shamos and Hoey, 1975; Lee and Schachter, 1980; Guibas and Stolfi, 1985) or some $O(n \log n)$ expected-time randomized algorithms (Guibas et al., 1990; Boissonnat and Teillaud, 1993).

5. It should also be noted that we do not make use of the power of truncation and bucketing, so that the algorithm cannot be expected to compete against fine-tuned bucketing methods (see, e.g., Maus, 1984; Dwyer, 1986, 1987; Katajainen and Koppinen, 1988; Ohya et al., 1984a,b; Sugihara et al., 1990; Ooishi and Sugihara, 1991) which all achieve $O(n)$ expected time under certain conditions on the distribution of the data.

6. The algorithm has been implemented by the authors. The program is only about 200 lines long (including comments) and it is very efficient. With respect to a random Delaunay triangulation of 3200 (32,000) points the average number of triangles visited, over a sufficient large number of trials, is about 36 (50) when 32 random points are selected. We strongly believe that the method discussed above is also efficient in practice to locate a query point in arbitrary triangulations. We also need to mention the recent work of Lemaire (1997), which includes novel algorithms based on the paradigm of Green and Sibson, as well as many experiments, including experiments with our method, which was coined the *Jump-and-Walk* method. All the new algorithms of Lemaire require additional preprocessing beyond the simple structure assumed in this paper.

7. Finally, a d-dimensional version of this simple point-location scheme seems to require expected time $O(m + (n/m)^{1/d})$, which is $O(n^{1/(d+1)})$ if we set $m = \Theta(n^{1/(d+1)})$. The constant in front of the polynomial factor grows exponentially quickly with d however.

References

M. Bern, D. Eppstein, and F. F. Yao, The expected extremes in a Delaunay triangulation, *International Journal of Computational Geometry and Its Applications*, vol. 1, pp. 79–91, 1991.

J.-D. Boissonnat and M. Teillaud, On the randomized construction of the Delaunay tree, *Theoretical Computer Science*, vol. 112, pp. 339–354, 1993.

P. Bose and L. Devroye, Intersections with random Voronoi diagrams, *Computational Geometry*: *Theory and Applications*, 1997. To appear.

A. Bowyer, Computing Dirichlet tessellations, *Computer Journal*, vol. 24, pp. 162–166, 1981.

T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Boston, MA, 1990.

O. Devillers, S. Meiser, and M. Teillaud, Fully dynamic Delaunay triangulation in logarithmic expected time per operation, *Computational Geometry*: *Theory and Applications*, vol. 2, pp. 55–80, 1992.

R. A. Dwyer, A simple divide-and-conquer algorithm for constructing Delaunay triangulations in $O(n \log \log n)$ expected time, in: *Proceedings of the* 2*nd Annual Symposium on Computational Geometry*, pp. 276–284, ACM, New York, 1986.

R. A. Dwyer, A faster divide-and-conquer algorithm for constructing Delaunay triangulations, *Algorithmica*, vol. 2, pp. 137–151, 1987.

P. J. Green and R. Sibson, Computing Dirichlet tessellations in the plane, *Computer Journal*, vol. 21, pp. 168–173, 1978.

L. Guibas and J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, *ACM Transactions on Graphics*, vol. 4, pp. 74–123, 1985.

L. J. Guibas, D. E. Knuth, and M. Sharir, Randomized incremental construction of Delaunay and Voronoi diagrams, in: *ICALP* 90, pp. 414–431, Lecture Notes in Computer Science, vol. 443, Springer-Verlag, Berlin, 1990.

J. Katajainen and M. Koppinen, Constructing Delaunay triangulations by merging buckets in quadtree order, *Fundamenta Informaticae*, vol. 11, pp. 275–288, 1988.

D. Kirkpatrick: Optimal search in planar subdivisions, *SIAM Journal on Computing*, vol. 12, pp. 28–35, 1983.

D.-T. Lee and B. J. Schachter, Two algorithms for constructing the Delaunay triangulation, *International Journal of Computer and Information Sciences*, vol. 9, pp. 219–242, 1980.

C. Lemaire, Triangulation de Delaunay et Arbres Multidimensionnels, Ph.D. Dissertation, Université Jean Monnet, Saint-Etienne, 1997.

A. Maus, Delaunay triangulation and the convex hull of n points in expected linear time, *BIT*, vol. 24, pp. 151–163, 1984.

E. P. Mücke, Shapes and implementations in three-dimensional geometry, Technical report UIUCDCS-R-93-1836, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, 1993.

E. P. Mücke, I. Saias, and B. Zhu, Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations, in: *Proceedings of the* 12*th Annual Symposium on Computational Geometry*, pp. 274–283, ACM, New York, 1996.

T. Ohya, M. Iri, and K. Murota, Improvements of the incremental method for the Voronoi diagram with computational comparison of various algorithms, *Journal of Operations Research*, vol. 27, pp. 306–336, 1984a.

T. Ohya, M. Iri, and K. Murota, A fast Voronoi-diagram algorithm with quaternary tree bucketing, *Information Processing Letters*, vol. 18, pp. 227–231, 1984b.

A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations*: *Concepts and Applications of Voronoi Diagrams*, Wiley, Chichester, 1992.

Y. Ooishi and K. Sugihara, Numerically robust divide-and-conquer algorithm for constructing Voronoi diagrams, *Transactions of the Information Processing Society of Japan*, vol. 32, pp. 709–720, 1991.

M. I. Shamos and D. Hoey, Closest-point problems, in: *Proceedings of the* 16*th IEEE Symposium on the Foundations of Computer Science*, pp. 151–162, 1975.

K. Sugihara, Y. Ooishi, and H. Imai, Topology-oriented approach to robustness and its applications to several Voronoi-diagram algorithms, in: *Abstracts of the Second Canadian Conference on Computational Geometry, Ottawa*, pp. 36–39, 1990.