Incremental Construction along Space-Filling Curves

Kevin Buchin*

Abstract

For the incremental construction of a Delaunay triangulation, we prove that inserting points in rounds and walking along a space-filling curve in each round yields an algorithm running in linear expected time for uniformly distributed points. We complement this result by a simpler incremental construction running in linear expected time in any dimension.

1 Introduction

Motivation When devising an insertion order for the incremental construction of the Delaunay triangulation there are two seemingly conflicting goals: Inserting points randomly from the data avoids creating artificial triangles during the construction. In contrast, inserting points nearby allows taking advantage of geometric locality and locality of reference.

Randomized incremental construction follows the first approach. It is asymptotically optimal but performs poorly with modern memory hierarchies when used for large data sets as observed by Amenta, Choi and Rote [1]. They showed how randomness can be reduced without changing the asymptotic performance by a *biased randomized insertion order*: Points are randomly assigned to rounds of insertion of increasing sizes, and within a round the order of insertion can be chosen freely.

This allows us to use locality within the rounds by traversing the points of a round in an order along a space-filling curve [11]. We chose a space-filling curve order because it combines locality of reference with geometric locality by linearizing space, adapts well to irregularities of the point distribution, is fast to compute, is applicable in higher dimensions, and gives a good bound on the length of the resulting tour.

Related Algorithms Some linear expected time algorithms for constructing the Delaunay triangulation of uniformly distributed points from a bounded convex area in the plane are known [2, 6, 8]. Dwyer [6] gives an algorithm running in linear expected time for points from a sphere in any fixed dimension.

*Freie Universität Berlin, Institut für Informatik, buchin@inf.fu-berlin.de. This research was supported by the Deutsche Forschungsgemeinschaft within the European graduate program 'Combinatorics, Geometry, and Computation' (No. GRK 588/2).

Two incremental constructions running in linear time in practice on uniformly distributed points are known [9, 12]. In both cases, the analysis does not treat the irregularities near the boundary. The boundary case can be avoided by considering points from a Poisson point process.

Inserting near the Boundary For algorithms based on incremental construction, points near the boundary seem difficult to handle, because long and thin triangles slow down the point location. Figure 1 shows a typical case of this: Near the boundary, triangles with a large circumcircle are likely to occur in the triangulation, because a large part of the circumcircle may lie outside the region with points.

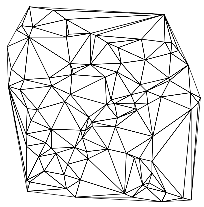

Figure 1: Delaunay triangulation of points in a square

Our main effort is to prove that the boundary case does not change overall linearity. While the analysis is done for our algorithm it seems possible to adapt the analysis to treat the algorithms mentioned above.

Surprisingly, we found another simple incremental construction which has no problems near the boundary and constructs the Delaunay triangulation in linear expected time in any fixed dimension. We include an analysis of this algorithm.

Contributions Our main contribution is to prove that a biased randomized insertion order together with a local insertion scheme runs in linear expected time on uniform points in a bounded convex region. This result complements the good practical performance of biased randomized insertion orders and resolves an open problem posed by Amenta et al. [1] This algorithm is the first completely analyzed linear expected time incremental construction algorithm for Delaunay triangulations.

The main technical contribution is the explicit analysis of point location near the boundary. Furthermore, we present an incremental construction running in linear expected time in any fixed dimension.

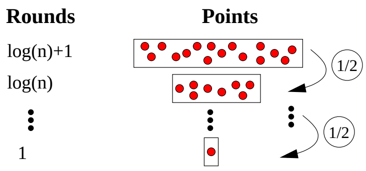

Figure 2: Assigning points to rounds

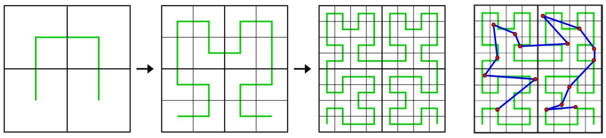

Figure 3: First steps in the construction of the Hilbert curve and a space-filling curve tour

2 Walking along a Space-Filling Curve

Incremental Construction The basic concept of incremental construction is simple to state: Insert the points into the Delaunay triangulation one by one, updating the data structure after each insertion step. The time needed to insert a point consists of the time needed for locating the point in the current triangulation and the time for updating the triangulation. If the points are inserted in random order the expected total time needed for updating is in $O(n)$ and for point location is in $O(n \log n)$.

Biased Randomized Insertion Orders The order of insertion is allowed to deviate from a random order as long as randomness dominates. Sufficient randomness can be introduced to the insertion order by assigning the points independently at random to rounds as illustrated in Figure 2: A point is independently assigned to the last round with the probability of 1/2. Each of the remaining points is assigned to the next to last round with the probability of 1/2, and so on [1].

After a logarithmic number of rounds an expected constant number of points remain, and we can stop sampling and assign the remaining points to the first round. The points are inserted round by round. In a round points can be inserted in an arbitrary order.

Biased randomized insertion orders were originally introduced to reduce random memory access. We make use of the fact that they do not change the update cost, which, in our case, is linear. Therefore we can focus on the point location time.

Space-Filling Curves Within a round we construct a short tour through the points by the space-filling curve heuristic for the traveling salesman [10]. To see how the tour is constructed, consider the first steps of the geometric construction of the Hilbert Curve shown in Figure 3. The space is successively subdivided. The cells are ordered in such a way that consecutive cells in the order are adjacent.

The limit of this process yields a *space-filling curve*, i. e. a surjective mapping from the unit interval to the unit square or, more generally, to the d–dimensional unit cube. Formally, the space-filling curve heuristic sorts the points by selecting a preimage for every point and by sorting the points according to the preimages.

In practice, the process can be stopped after a finite number of subdivisions. The maximal number of subdivisions necessary is the number of bits of precision. In order to achieve an $O(\sqrt{n})$ bound on the tour length a subdivision with as many cells as points is sufficient. Points within a cell can then be ordered in an arbitrary order.

Walking We traverse the tour and insert the points along the way. The next point is located by *walking* [5], i. e. by a local search starting at the current point and traversing the triangles stabbed by the line segment between the two points. This point location scheme does not need a point location data structure.

The heuristic can be used not only for points in the unit square but also in an arbitrary rectangle. The bound on the tour length changes by a factor of the length of the longer side. For points in a bounded convex region a bounding rectangle is used.

3 Analysis

Space-Filling Curve Heuristic The heuristic constructs a tour through a given set of points in the unit square by visiting them in the order of their preimages under a space-filling curve ψ. The order of preimages is not unique since ψ cannot be injective [11]. For the heuristic to be effective the images of nearby points on the side of the preimages should be near to each other in space. For space-filling curves this follows from their Lipschitz continuity of order 1/2, i. e. that for any s, t in the unit interval $|\psi(s) - \psi(t)| \leq c_\psi |s-t|^{1/2}$.

The space-filling curve heuristic was popularized by Platzman and Bartholdi [10]. A general treatment and probabilistic analysis is given by Gao and Steele [7]. We summarize the result we need in the following lemma:

Lemma 1 *For a space-filling curve that is Lipschitz of order* $1/2$ *and can be generated by subdivision, an order on the points can be computed in linear time in such a way that for any* k*-subset of points the length of the tour through these points along the order is bounded by* $O(k^{1/2})$.

For this lemma no assumption on the point distribution is used. A stronger bound holds for points distributed uniformly in the unit square [7]. In d dimensions the bound generalizes to $O(n^{(d-1)/d})$.

Counting Intersections To analyze the running time it is sufficient to analyze the time required in the last round using an induction. Assume $m+n$ points distributed independently and uniformly at random in a bounded convex region C of area 1, where n points are already inserted in the Delaunay triangulation. To insert the m remaining points, a tour through the points is constructed using the space-filling curve heuristic.

The points are located by traversing the triangulation along the tour. Therefore, the time needed for locating the points is proportional to the number of intersections between the tour and the triangulation.

A bound on the expected number of intersections is obtained by considering *exclusion regions* for possible edges of the triangulations, i. e. if the region contains points on both sides of the possible edges the edge cannot be in the triangulation. For Delaunay triangulations the disc with the edge as diameter is an exclusion region. For uniformly distributed points the edges of a triangulation with exclusion regions typically are expected to be either short or near to the boundary. This can be strengthened to the following:

Lemma 2 (Devroye, Mücke and Zhu [5]) *The expected number of intersections between a Delaunay triangulation of points distributed independently and uniformly in a compact convex area C and a fixed line segment L that is at least distance $c_0\sqrt{\log n/n}$ from the boundary of C is bounded by*

$$c_1 + c_2|L|\sqrt{n},$$

where c_0 is a constant, and c_1 and c_2 depend only on the geometrical properties of C.

The bound on the tour length and the bound on the number of intersections together give a linear bound for all line segments that have a distance of at least $c_0\sqrt{\log n/n}$ from the boundary ∂C. The expected number of points near the boundary is bounded by $m' := c_0|\partial C|m\sqrt{\log n/n}$ and the number of line segments by $2m'$, and therefore, by Jensen's inequality and Lemma 1, the total length of these line segments by $c\sqrt{2m'}$ for suitable c.

To treat these segments we quantify what it means that the edges of the triangulation are likely to be short or near to the boundary:

Lemma 3 *Let T be the Delaunay triangulation of n points distributed independently and uniformly in a convex area C. Denote by $D_{w,l}$ the event that a Delaunay edge with an endpoint with a distance of at least w to the boundary of C is longer than l.*
For $c > 1$ and $l \geq cw$

$$\Pr(D_{w,l}) \leq n^2 e^{-(n-2)wl\sqrt{1-1/c^2}/2}.$$

In particular, if $l \geq 3w$ and $wl \geq 6\sqrt{2}\log n/n$, then $\Pr(D_{w,l}) \in o(1/n)$.

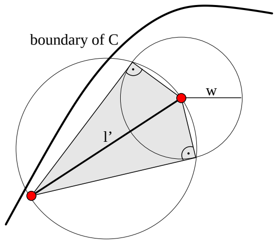

Figure 4: Exclusion region for a Delaunay edge that is contained in C

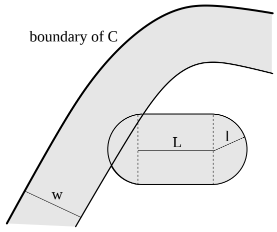

Figure 5: Area for endpoints of Delaunay edges intersecting L

Proof. Consider the edge in Figure 4 with length $l' \geq l$ and a vertex with a distance of more than w to the boundary of C. The two rectangular triangles form an exclusion region for the edge that is contained in C. The area of a triangle is bounded by $1/2 \cdot w\sqrt{l^2 - w^2} \geq wl\sqrt{1-1/c^2}/2$. There are $\binom{n}{2}$ possible edges and therefore

$$\begin{aligned}\Pr(D_{w,l}) &\leq \binom{n}{2} 2(1 - wl\sqrt{1-1/c^2})^{n-2} \\ &\leq n^2 e^{-(n-2)wl\sqrt{1-1/c^2}/2}\end{aligned}$$

□

This gives us a bound on the number of Delaunay edges that can intersect the line segments of the tour:

Lemma 4 *The expected number of intersections of a Delaunay triangulation and a tour along a Lipschitz-1/2 space-filling curve with a total number of N points which are distributed independently and uniformly in a convex area is linear in N.*

Proof. Assume $l \geq 3w$ and $wl \geq 6\sqrt{2}\log n/n$. With high probability only edges with endpoints with distance of at most l to one of the line segments, or with distance at most w from the boundary can intersect. For a single line segment this area is shown in Figure 5. The expected number of endpoints of edges that intersect a line segment L is therefore bounded by $n(|\partial C|w + \pi l^2 + 2l|L|) + o(1)$. Because of planarity there are at most three times that many edges intersecting L.

For k line segments of total length λ this yields a $3n(|\partial C|kw + \pi l^2 k + 2l\lambda) + o(k)$ bound on the number of intersecting edges. In our case, we have $k = c_0|\partial C|m\sqrt{\log n/n}$ and $\lambda = \sqrt{k}$. Choosing $w := \max(k^{-1/4}\sqrt{\log n/n}, (\log n/n)^{2/3})$ and $l := 6\sqrt{2}\log n/(nw)$ the number of intersections can be bounded by $O(n^{1/8}m^{3/4}\log^{7/8} n + n^{-1/6}m\log^{7/6} n)$. Adding up the bound for segments near the boundary and far away from the boundary yields a linear bound on the expected number of intersections. □

Inserting Points We now extend the analysis to the case where the triangulation changes during a tour because points are inserted. The points of the triangulation occur in two different roles in the analysis: They may contribute to the number of intersections as an endpoint of an intersecting edge but they may also block other edges because they lie in their exclusion region. The analysis can be extended by taking all points as possible endpoints but only the points of the original triangulation as blocking points.

For a fixed line segment of the tour the remaining points of the tour are not independent of this segment but their density can be bounded if we use a *bi-measure preserving* curve, which allows us to work on uniform distributions on the preimage and the image exchangeably [7]. The cost resulting from the fact that the starting point of a line segment is a vertex of the triangulation can be bounded by the update cost. In total this yields the following theorem:

Theorem 5 *Using a biased randomized insertion order and, in each round, walking along a Lipschitz-* $1/2$*, bi-measure preserving space-filling curve, the incremental construction algorithm runs in linear expected time for points distributed independently and uniformly in a bounded, convex area.*

4 Seeking a Conflict with the Neighbor

The main problem in the average case analysis of incremental constructions seems to be the boundary. Here we give an algorithm for which this is not so.

The points are inserted in random order. The algorithm maintains a dynamic bucketing scheme. This allows us to find the nearest neighbor in the triangulation for a new point in constant expected time using spiral search [2]. Now a d–simplex incident to the nearest neighbor is found which conflicts with this point. From this triangle all conflicting d–simplices are found as in the Bowyer-Watson algorithm.

Theorem 6 Seeking a Conflict with the Neighbor *constructs the Delaunay triangulation in linear expected time when the points are distributed independently and uniformly in a d–dimensional bounded convex open region for which the expected complexity of the Delaunay triangulation is linear. In particular, this is the case for the unit d–ball.*

Proof. The expected time required for searching the nearest neighbor and for updating the triangulation is linear [2]. It remains to bound the expected number of d–simplices of the triangulation containing the nearest neighbor of a new point. The difficulty is that the nearest neighbor is not a random point of the triangulation but a constant bound can be obtained by using that the in-degree of the nearest neighbor graph is bounded in any fixed dimension.

The special case of the d–ball follows directly from the linear expected complexity [6]. □

5 Discussion

We presented two incremental construction algorithms for the Delaunay triangulation. The first algorithm constructs a spatial order of the points. Ideally, the Delaunay triangulation should be stored in the same order to make use of the locality of reference. A possible way to achieve this is presented by Blandford et al. [3]. For this, it is important to use one ordering for all points.

Two advantages of the first algorithm which we have not addressed in the analysis are its good performance on surface points and on large data sets. Furthermore, the algorithm runs in higher dimensions.

References

[1] N. Amenta, S. Choi, and G. Rote. Incremental constructions con BRIO. In *Proc. 19th Annual Symposium on Computational Geometry*, pages 211–219. ACM Press, 2003.

[2] J. L. Bentley, B. W. Weide, and A. C.-C. Yao. Optimal expected-time algorithms for closest point problems. *ACM Trans. Math. Softw.*, 6(4):563–580, 1980.

[3] D. K. Blandford, G. E. Blelloch, D. E. Cardoze, and C. Kadow. Compact representations of simplicial meshes in two and three dimensions. In *Proc. 12th Int. Meshing Roundtable*, 2003.

[4] L. Devroye, C. Lemaire, and J.-M. Moreau. Expected time analysis for Delaunay point location. *Comput. Geom. Theory Appl.*, 22:61–89, 2004.

[5] L. Devroye, E. Mücke, and B. Zhu. A note on point location in Delaunay triangulations of random points. *Algorithmica*, 22:477–482, 1998.

[6] R. A. Dwyer. Higher-dimensional Voronoi diagrams in linear expected time. *Discrete Comput. Geom.*, 6(4):343–367, 1991.

[7] J. Gao and J. M. Steele. General spacefilling curve heuristics and limit theory for the traveling salesman problem. *J. Complex.*, 10(2):230–245, 1994.

[8] J. Katajainen and M. Koppinen. Constructing Delaunay triangulations by merging buckets in quadtree order. *Fundam. Inform.*, 11:275–288, 1988.

[9] T. Ohya, M. Iri, and K. Murota. Improvements of the incremental method for the Voronoi diagram with a comparison of various algorithms. *J. Operations Res. Soc. Japan*, 27:306–337, 1984.

[10] L. K. Platzman and J. J. Bartholdi, III. Spacefilling curves and the planar travelling salesman problem. *J. ACM*, 36(4):719–737, 1989.

[11] H. Sagan. *Space-Filling Curves*. Springer, 1994.

[12] P. Su and R. L. S. Drysdale. A comparison of sequential Delaunay triangulation algorithms. *Comput. Geom. Theory Appl.*, 7(5-6):361–385, 1997.