A General Randomized Incremental Reconstruction Procedure

Kenneth L. Clarkson
Bell Laboratories, Lucent Technologies
Murray Hill, New Jersey 07974
`clarkson@research.bell-labs.com`
`http://cm.bell-labs.com/who/clarkson/`

July 30, 1997

Abstract

The technique of randomized incremental construction allows a variety of geometric structures to be built quickly. This note shows that once such a structure is built, it is possible to store the geometric input data for it so that the structure can be built again by a randomized algorithm even more quickly. Except for the randomization, this generalizes the technique of Snoeyink and van Kreveld that applies to planar problems.

1 Introduction

Given a set S of n points in the plane, Snoeyink and van Kreveld have shown that the points can be stored so that their Delaunay triangulation can be found in $O(n)$ time.[8] Snoeyink and van Kreveld have generalized this technique to a variety of problems, but restricted to geometric structures that are planar graphs. This note generalizes the technique even more, to the setting of randomized incremental construction.

For concreteness, the ideas will first be described in the setting of Delaunay triangulation, and then in general.

2 Delaunay triangulations

Given a set S of n sites (points) in the plane, their Delaunay triangulation $\mathcal{D}(S)$ can be built by maintaining a Delaunay triangulation $\mathcal{D}(R)$ of a random subset $R \subset S$, adding sites one by one to R and updating the triangulation. The expected number of triangles generated site is added is $O(1)$, and a simple auxiliary data structure allows the addition of each site to be done in $O(\log r)$ expected time, where r is the number of sites in R. (Such an algorithm was proposed by Boisannat *et al.*[2] the first analysis of a similar, offline, algorithm was given in Clarkson and Shor[4]. This readily shows that the expected overall time of an online algorithm is $O(n \log n)$,[5] and subsequent analyses showed that the expected cost of adding site $r+1$ is $O(\log r)$.[6, 1, 3, 7].) The dominant cost of inserting a site p into R (and so into the triangulation) is the search problem of finding the set of triangles of $\mathcal{D}(R)$ that conflict with p, in the sense that p is inside the circumcircle of such a triangle. Such triangles will not be in $\mathcal{D}(R \cup \{p\})$. By walking over the triangulation from a triangle that conflicts with p, it is easy to find the remaining triangles that conflict with p, and so the search problem reduces to that of finding a single triangle that conflicts with p.

Many methods for solving the search problem involve finding all triangles, over the history of the triangulation, that conflict with p; the expected number of such triangles is $O(\log r)$. A simple data structure allows the work for finding these triangles to be constant per triangle.[1, 3]

Snoeyink and van Kreveld have shown that it is possible to order the points in such a way that when the Delaunay triangulation is rebuilt, the search problem is easy to solve, requiring constant time per point. Their approach involves finding large sets of independent vertices of the triangulation, each with small degree, following Kirkpatrick.

We give an alternative approach, as follows. Let R_i denote the random subset R after i sites have been inserted, so that the sites can be numbered in a permutation π as $x_1 \ldots x_n$ with $R_i = \{x_1 \ldots x_i\}$. It will be convenient to define R_i as equal to S, for $i \geq n$.

Consider the sites as being inserted in "batches" of exponentially increasing size: let $M_1 \equiv R_1$, and for $j = 1 \ldots \lceil \lg n \rceil$, let

$$M_j \equiv R_{2^j} \setminus R_{2^{j-1}}.$$

We will build a different permutation π' of the sites $x'_1 \ldots x'_n$ such that the corresponding sets $R'_i \equiv \{x'_1 \ldots x'_i\}$ have the property that $R'_i = R_i$ when i is a power of two; that is, the batch sets M_j are the same for either permutation.

By picking the permutation π' properly, the sites within M_j will be ordered in a way that allows the search problem for M_j to be solved quickly, with respect to the Delaunay triangulation of $R_{2^{j-1}}$: that is, for each $p \in M_j$, the triangles in $\mathcal{D}(R_{2^{j-1}})$ conflicting with p can be found quickly. The construction of $\mathcal{D}(R_{2^j})$ then proceeds incrementally. The sites of M_j are added in random order, resulting in random subsets R''_i for $2^{j-1} < i \leq 2^j$. Note that $R''_i = R'_i = R_i$ for i a power of 2. The searching problem is solved by finding all triangles that conflict with p, for such triangles that appear in one of the triangulations $\mathcal{D}(R''_i)$, for $2^{j-1} < i \leq 2^j$. Theorem 1 below implies that the expected number of such triangles is $O(1)$.

The "encoding" algorithm produces an ordering of M_j, as given in π', by constructing an ordered list $\mathcal{L}$ of the sites in M_j in the following way. List the Delaunay triangles of $\mathcal{D}(R_{2^{j-1}})$ in lexicographic order, where a triangle with vertices $x'_a, x'_b, x'_c \in R'_{2^{j-1}}$ with $a < b < c$ is given a sort key (a, b, c). (For our purposes, supporting halfspaces of $R_{2^{j-1}}$ can be viewed as triangles with a vertex at infinity; this "vertex" can be numbered as site $n+1$ for the sort keys.)

Walk through the list of triangles in order, and for each triangle, look at the sites of M_j with which it conflicts. If such a site has not yet been put into $\mathcal{L}$, append it to the end of $\mathcal{L}$.

The ordering of $\mathcal{L}$ gives the ordering of M_j for π'.

Using radix sort, the encoding can be found in time proportional to the number of triangle/site conflicts.

The "decoding," or reconstruction, algorithm sorts the triangles of $\mathcal{D}(R_{2^{j-1}})$ lexicographically, as in the encoding procedure, and walks simultaneously through that sorted list of triangles and through M_j in order of π'. Suppose a triangle T and a site p are currently under consideration. If they do not conflict, then no site after p conflicts with T, and the next triangle on the triangle list can be considered. If they do conflict, then the search problem has been solved for p, and next site in the permutation π' can be considered for T.

Plainly the work done for reconstruction, as described so far, is proportional to the number of triangles plus the number of sites in M_j. It remains to show that the rest of the searching requires $O(1)$ expected time for each site in M_j; this will be shown in the general setting.

The rest of the reconstruction procedure, for sites in M_j, is randomized incremental construction, as applied to a random ordering of the points of M_j. Note that the result is a randomized incremental construction procedure using a permutation π'' that is random, but related to the random permutation π only in that the sets R''_i and R_i agree when i is a power of two.

3 The general case

The general setting for the reconstruction algorithm is the same as for "randomized incremental construction," for which the terminology and analysis of Clarkson *et al.* will be used.[3] The setting of general randomized incremental construction generalizes that of Delaunay triangulation, by generalizing from point sites to other geometric *objects*, and from Delaunay triangles to geometric *regions* determined by d or fewer objects, for some d. If a region is determined by some set of objects, those objects will to *support* it. The conflict between a Delaunay triangle and a site generalizes to some conflict relation between regions and objects. For a given set of objects S, the randomized incremental paradigm gives a way to construct $\mathcal{F}_0(S)$, which is the set of all regions determined by objects of S that do not conflict with any objects of S. The randomized incremental paradigm builds $\mathcal{F}_0(S)$ by maintaining $\mathcal{F}_0(R)$ as a random subset R of S is built up to S. The dominant cost is the searching problem, finding a region of $\mathcal{F}_0(R)$ with which a newly added object x conflicts.

This search problem can be solved in the randomized incremental paradigm by examining the history of the construction, examining all regions of

$$\mathcal{H}_{r-1} \equiv \cup_{1\leq i\leq r-1}\mathcal{F}_0(R_i)$$

that conflict with x.

The procedure described in the last section, for Delaunay triangulations, can be generalized in a straightforward way to this setting. This implies a low cost for finding a member of $\mathcal{F}_0(R_{2^{j-1}})$ with which a given object $x \in M_j$ conflicts. In most cases, the remaining cost of search is proportional to the number of regions of

$$\cup_{2^{j-1} \leq i \leq 2^j} \mathcal{F}_0(R_i)$$

with which a random $x \in M_j$ conflicts. The following theorem holds.

Theorem 1 *Let $z \equiv 2^{j-1}$. Let $\sigma = (x_1 \ldots x_n)$ be a random permutation , giving random subsets $T_i \equiv \{x_1 \ldots x_i\}$. For given j and r with $z < r \leq 2^j$, the expected number of regions in*

$$\cup_{z<i<r} \mathcal{F}_0(T_i)$$

with which object x_r conflicts is no more than

$$\frac{d}{z+1} f_{z+1} + \sum_{z<i\leq r} \frac{d(d-1)}{i(i-1)} f_i,$$

where f_i is the expected number of regions of $\mathcal{F}_0(T_i)$.

Proof. The proof can be readily adapted from the proof of Theorem 4 of [3]: in that proof, the history

$$H \equiv H(x_1, \ldots, x_{r-1}) \equiv \cup_{1\leq i<r} \mathcal{F}_0(T_i)$$

is compared with the history

$$H' \equiv H(x_r, x_1, \ldots, x_{r-1}) \equiv \cup_{1\leq i<r} \mathcal{F}_0(\{x_r\} \cup T_i),$$

the "alternate history" of the construction, where x_r is added first. Here we can change the comparison to that between H and

$$\hat{H} \equiv H(x_1, \ldots, x_z, x_r, x_{z+1}, \ldots, x_{r-1}),$$

the alternate history where x_r is added at the beginning of M_j. We have

$$|H| + |\hat{H} \setminus H| = |\hat{H}| + |H \setminus \hat{H}|,$$

where $H \setminus \hat{H}$ is the set of regions we want to count, those in H that conflict with x_r and appear in some $\mathcal{F}_0(T_i)$ for $i > z$. The set $\hat{H} \setminus H$ comprises regions supported by x_r, and appear after those in $\mathcal{F}_0(T_z)$. Hence the expected size $E|H \setminus \hat{H}|$ is bounded by

$$E|H| - E|\hat{H}| + E|\hat{H} \setminus H|. \tag{1}$$

The rest of the proof follows analogously to the proof of Theorem 4 of [3]. For completeness, we briefly complete the proof here. A region of $\hat{H} \setminus H$ may be in $\mathcal{F}_0(T'_z)$ and supported by x_r, where $T'_i \equiv T_i \cup \{x_r\}$. The expected number of such regions is $df_{z+1}/(z+1)$, since the probability that x_r, a random member of T'_z, supports a given region of $\mathcal{F}_0(T'_z)$ is $d/(z+1)$.

A remaining region of $\hat{H} \setminus H$ first appears at some i, for $z < i < r$; that is, such a region is in $\mathcal{F}_0(T'_i)$ but not in $\mathcal{F}_0(T'_{i-1})$ or $\mathcal{F}_0(T_i)$. Hence it is supported by x_r and by x_i. Since x_r and x_i are random elements of T'_i, the probability that they support a given member of $\mathcal{F}_0(T'_i)$ is at most $d(d-1)/(i+1)i$: for a given member of $\mathcal{F}_0(R'_i)$, there are at most $\binom{d}{2}$ pairs from R'_i that support it, and $\binom{i+1}{2}$ pairs altogether. Hence the expected size

$$\begin{aligned} E|\hat{H} \setminus H| &\leq \frac{d}{z+1} f_{z+1} + \sum_{z<i<r} \frac{d(d-1)}{i(i+1)} E|\mathcal{F}_0(T'_i)| \\ &= \frac{d}{z+1} f_{z+1} + \sum_{z<i<r} \frac{d(d-1)}{i(i+1)} f_{i+1}. \end{aligned} \tag{2}$$

To bound $E|H \setminus \hat{H}|$ using (1), it remains to bound the expected size $E|H|$. This is $\sum_{m<r} df_m/m$, as in Theorem 3[3]: count each region of H by its first appearance, at which, for some m, the region is in $\mathcal{F}_0(T_m)$ but not in $\mathcal{F}_0(T_{m-1})$; this implies that x_m supports the region. The probability that a random member of T_m supports the region is d/m, and therefore the expected number of new regions when x_m is added is df_m/m. A similar bound holds for $E|\hat{H}|$, and so $E|H| - E|\hat{H}|$ is $-df_r/r$.

Putting this bound with (1) and (2), theorem follows. □

In the case of Delaunay triangulation, $d = 3$ and $f_i = O(i)$, so the above implies that the search cost in reconstruction is expected $O(1)$ per site.

References

[1] J.-D. Boissonnat, O. Devillers, R. Schott, M. Teillaud, and M. Yvinec. Applications of random sampling to on-line algorithms in computational geometry. *Discrete Comput. Geom.*, 8:51–71, 1992.

[2] J.-D. Boissonnat and M. Teillaud. A hierarchical representation of objects: The Delaunay tree. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 260–268, 1986.

[3] K. L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom. Theory Appl.*, 3(4):185–212, 1993. http://cm.bell-labs.com/who/clarkson/4res.html.

[4] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989. http://cm.bell-labs.com/who/clarkson/rs2m.html.

[5] L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7:381–413, 1992.

[6] K. Mulmuley. Randomized multidimensional search trees: Further results in dynamic sampling. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 216–227, 1991.

[7] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1994.

[8] J. Snoeyink and M. van Kreveld. Good orders for incremental (re)construction. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 400–405, 1997.