5. Delaunay Triangulation: Incremental Construction

Lecture on Thursday 8th October, 2009 by Bernd Gärtner <gaertner@inf.ethz.ch>

In the last lecture, we have learned about the Lawson flip algorithm that computes a Delaunay triangulation of a given n-point set $P \subseteq \mathbb{R}^2$ with $O(n^2)$ Lawson flips. One can actually implement this algorithm to run in $O(n^2)$ time, and there are point sets where it may take $\Omega(n^2)$ flips.

In this lecture, we will discuss a different algorithm. The final goal is to show that this algorithm can be implemented to run in $O(n \log n)$ time; this lecture, however, is concerned only with the correctness of the algorithm. Throughout the lecture we assume that P is in general position (no 3 points on a line, no 4 points on a common circle), so that the Delaunay triangulation is unique (Theorem 4.10). There are techniques to deal with non-general position, but we don't discuss them here.

5.1 Incremental construction

The idea is to build the Delaunay triangulation of P by inserting one point after another. We always maintain the Delaunay triangulation of the point set R inserted so far, and when the next point s comes along, we simply update the triangulation to the Delaunay triangulation of $R \cup \{s\}$. Let $\mathcal{DT}(R)$ denote the Delaunay triangulation of $R \subseteq P$.

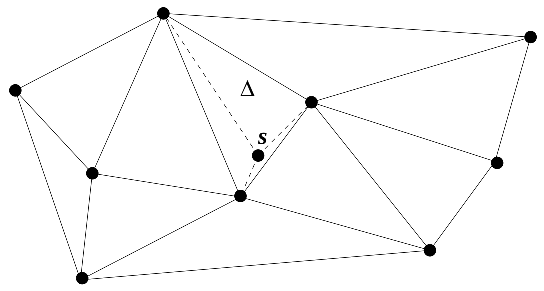


Figure 5.1: *Inserting s into $\mathcal{DT}(R)$: Step 1*

To avoid special cases, we enhance the point set P with three artificial points "far out". The convex hull of the resulting point set is a triangle; later, we can simply remove the extra points and their incident edges to obtain $\mathcal{DT}(P)$. The incremental algorithm starts off with the Delaunay triangulation of the three artificial points which consists of one big triangle enclosing all other points. (In our figures, we suppress the far-away points, since they are merely a technicality.)

Now assume that we have already built $\mathcal{DT}(R)$, and we next insert $s \in P \setminus R$. Here is the outline of the update step.

1. Find the triangle $\Delta = \Delta(p, q, r)$ of $\mathcal{DT}(R)$ that contains s, and replace it with the three triangles resulting from connecting s with all three vertices p, q, r; see Figure 5.1. We now have a triangulation $\mathcal{T}$ of $R \cup \{s\}$.

2. Perform Lawson flips on $\mathcal{T}$ until $\mathcal{DT}(R \cup \{s\})$ is obtained; see Figure 5.2

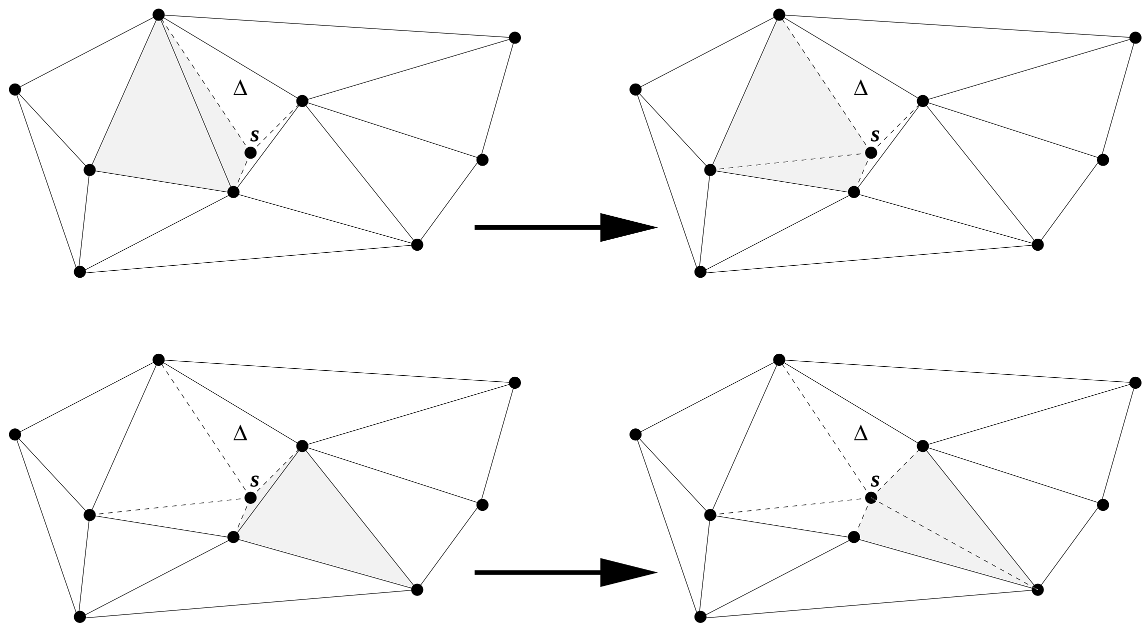


Figure 5.2: *Inserting* s *into* $\mathcal{DT}(R)$*: Step 2*

How to organize the Lawson flips. The Lawson flips can be organized quite systematically, since we always know the candidates for "bad" edges that may still have to be flipped. Initially (after step 1), only the three edges of Δ can be bad, since these are the only edges for which an incident triangle has changed (by inserting s in Step 1). Each of the three new edges is good, since the 4 vertices of its two incident triangles are not in convex position.

Now we have the following invariant (part (a) certainly holds in the first flip):

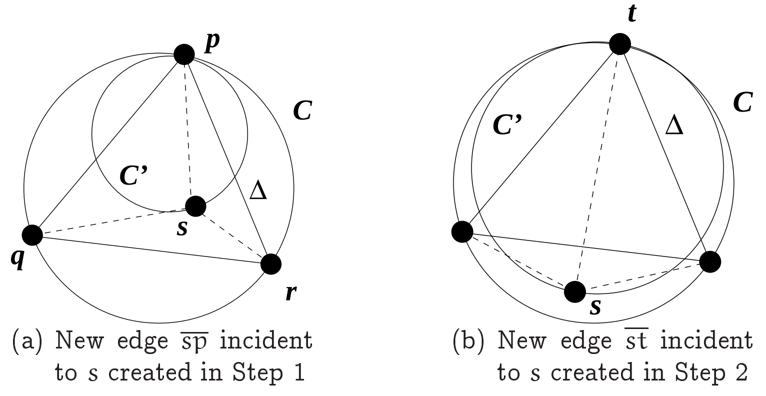
(a) In every flip, the convex quadrilateral Q in which the flip happens has exactly two edges incident to s, and the flip generates a new edge incident to s.

(b) Only the two edges of Q that are *not* incident to s can become bad through the flip.

We will prove part (b) in the next lemma. The invariant then follows since (b) entails (a) in the next flip. This means that we can maintain a queue of potentially bad edges that we process in turn. A good edge will simply be removed from the queue, and a bad edge will be flipped and replaced according to (b) with two new edges in the queue. In this way, we never flip edges incident to s; the next lemma proves that this is correct and at the same time establishes part (b) of the invariant.

Lemma 5.1 *Every edge incident to* s *that is created during the update is an edge of the Delaunay graph of* $P \cup \{s\}$ *and thus an edge that will be in* $\mathcal{DT}(R \cup \{s\})$. *It easily follows that edges incident to* s *will never become bad during the update step.*[1]

Proof. Let us consider one of the first three new edges, $\overline{sp}$, say. Since the triangle Δ has a circumcircle C strictly containing only s (Δ is in $\mathcal{DT}(R)$), we can shrink that circumcircle to a circle C' through s and p with no interior points, see Figure 5.3 (a). This proves that $\overline{sp}$ is in the Delaunay graph. If $\overline{st}$ is an edge created by a flip, a similar argument works. The flip destroys exactly one triangle Δ of $\mathcal{DT}(R)$. Its circumcircle C contains s only, and shrinking it yields an empty circle C' through s and t. Thus, $\overline{st}$ is in the Delaunay graph also in this case. □



(a) New edge $\overline{sp}$ incident to s created in Step 1 (b) New edge $\overline{st}$ incident to s created in Step 2

Figure 5.3: *Newly created edges incident to* s *are in the Delaunay graph*

5.2 The History Graph

What can we say about the performance of the incremental constuction? Not much yet. First of all, we did not specify how we find the triangle Δ of $\mathcal{DT}(R)$ that contains the point s to be inserted. Doing this in the obvious way (checking all triangles) is not good, since already the find steps would then amount to $O(n^2)$ work throughout the whole algorithm. Here is a smarter method, based on the *history graph*.

Definition 5.2 *Given* $R \subseteq P$ *(regarded as a sequence that reflects the insertion order), the history graph of* R *is a directed acyclic graph whose vertices are all triangles*

[1]if such an edge was bad, it could be flipped, but then it would be "gone forever" according to the lifting map interpretation from the previous lecture.

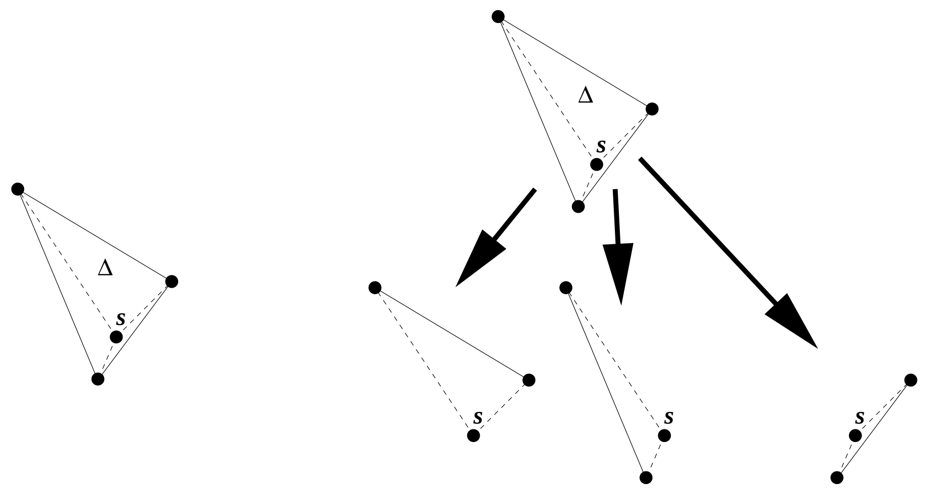


Figure 5.4: *The history graph: one triangle gets replaced by three triangles*

that have ever been created during the incremental construction of $\mathcal{DT}(R)$. There is a directed edge from Δ to Δ' whenever Δ has been destroyed during an insertion step, Δ' has been created during the same insertion step, and Δ overlaps with Δ' in its interior.

It follows that the history graph contains triangles of outdegrees 3, 2 and 0. The ones of outdegree 0 are clearly the triangles of $\mathcal{DT}(R)$.

The triangles of outdegree 3 are the ones that have been destroyed during Step 1 of an insertion. For each such triangle Δ, its three outneighbors are the three new triangles that have replaced it, see Figure 5.4.

The triangles of outdegree 2 are the ones that have been destroyed during Step 2 of an insertion. For each such triangle Δ, its two outneighbors are the two new triangles created during the flip that has destroyed Δ, see Figure 5.5.

The history graph can be built during the incremental construction at asymptotically no extra cost; but it may need extra space since it keeps all triangles ever created. Given the history graph, we can search for the triangle Δ of $\mathcal{DT}(R)$ that contains s, as follows. We start from the big triangle spanned by the three far-away points; this one certainly contains s. Then we follow a directed path in the history graph. If the current triangle still has outneighbors, we find the unique outneighbor containing s and continue the search with this neighbor. If the current triangle has no outneighbors anymore, it is in $\mathcal{DT}(R)$ and contains s—we are done.

5.3 The structural change

Concerning the actual update (Steps 1 and 2), we can make the following

Observation 5.3 *Given $\mathcal{DT}(R)$ and the triangle Δ of $\mathcal{DT}(R)$ that contains s, we can*

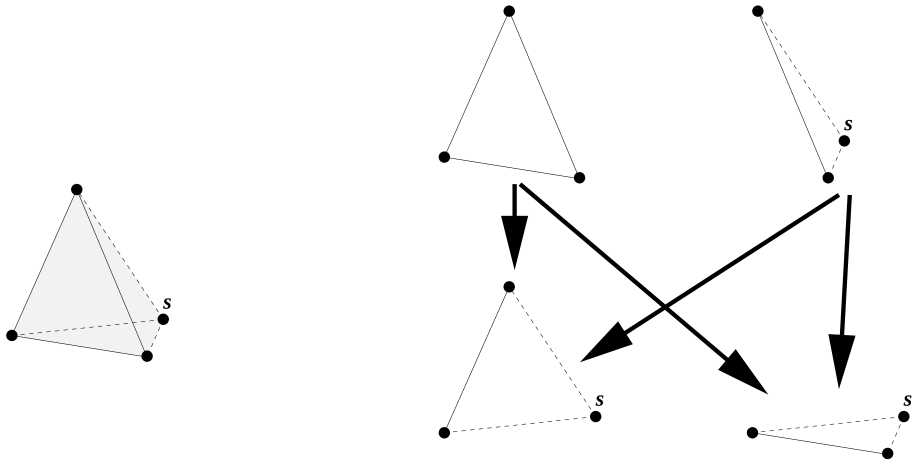


Figure 5.5: *The history graph: two triangles get replaced by two triangles*

build $\mathcal{DT}(R \cup \{s\})$ *in time proportional to the degree of* s *in* $\mathcal{DT}(R \cup \{s\})$*, which is the number of triangles of* $\mathcal{DT}(R \cup \{s\})$ *containing* s*.*

Indeed, since every flip generates exactly one new triangle incident to s, the number of flips is the degree of s minus three. Step 1 of the update takes constant time, and since also every flip can be implemented in constant time, the observation follows.

In the next lecture, we will show that a clever insertion order guarantees that the search paths traversed in the history graph are short, and that the structural change (the number of new triangles) is small. This will then give us the $O(n \log n)$ algorithm.

Questions

17. *Describe the algorithm for the incremental construction of* $\mathcal{DT}(P)$: how do we find the triangle containing the point s to be inserted into $\mathcal{DT}(R)$? How do we transform $\mathcal{DT}(R)$ into $\mathcal{DT}(R \cup \{s\})$? How many steps does the latter transformation take, in terms of $\mathcal{DT}(R \cup \{s\})$?