6. Delaunay Triangulations

Lecture on Monday 13^{th} October, 2008 by Bernd Gärtner <gaertner@inf.ethz.ch>

6.1 Point Set Triangulations Revisited

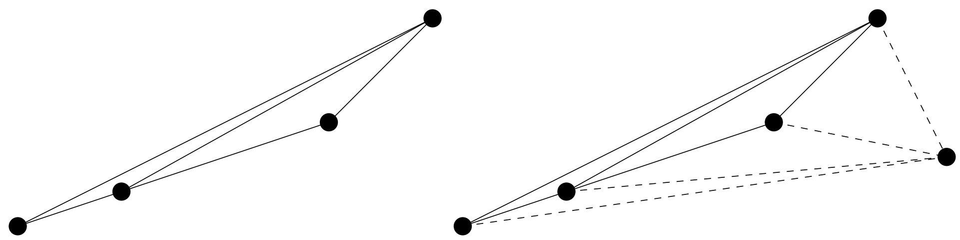
In the previous chapter, we have defined triangulations (straight-line embeddings of graphs where all bounded faces are triangles). In Figure 5.3, we have depicted three classes of triangulations: point set triangulations, simple polygon triangulations, and convex polygon triangulations. This chapter is about point set triangulations, and in particular about *Delaunay* triangulations that are specific point set triangulations.

Definition 6.1 *Given a finite point set* $P \subseteq \mathbb{R}^2$. *A triangulation* of P *is a triangulation whose vertices are exactly the points of* P.

While it was clear that one can always find a triangulation of a convex polygon (we were even able to count the number of possibilities), this is maybe less clear but still easy for point sets.

Observation 6.2 *Given a finite point set* $P \subseteq \mathbb{R}^2$ *with the property that not all points of* P *are on a common line. Then* P *has a triangulation.*

To construct one, we assume that no two points of P have the same x-coordinate (we can always rotate to achieve this). Let $p_1, \ldots, p_n$ be the list of points, ordered by x-coordinate. Let $p_1, \ldots, p_m$ be the smallest prefix such that $p_1, \ldots, p_m$ are not on a common line. We triangulate $p_1, \ldots, p_m$ by connecting p_m to $p_1, \ldots, p_{m-1}$ (which *are* on a common line), see Figure 6.1 (left).

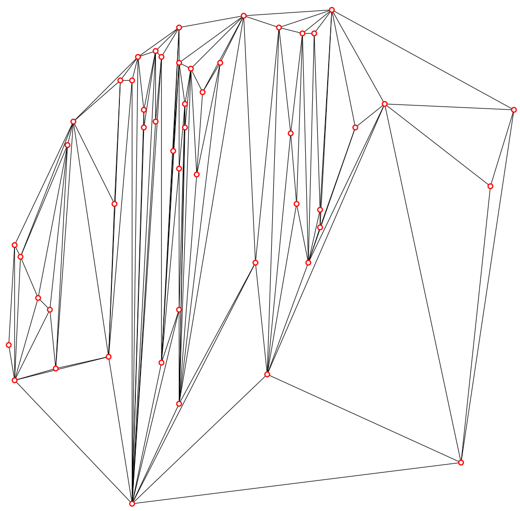


(a) getting started (b) adding a point

Figure 6.1: *Constructing the scan triangulation of* P

Then we add $p_{m+1}, \ldots, p_n$. In adding $p_i, i > m$, we connect p_i with all vertices of $\text{conv}(\{p_1, \ldots, p_{i-1}\})$ that it "sees". Since there are always at least two such vertices, we always add triangles (Figure 6.1 right). When we are done, exactly the points from P show up as vertices of the triangulation.

The triangulation that we get in this way is called a *scan triangulation*. Such a triangulation (Figure 6.2 (left) shows a larger one) is usually "ugly", though, since it tends to have many long and skinny triangles. In contrast, the *Delaunay triangulation* of the same point set (Figure 6.2 (right)) looks much nicer, and we discuss next how to get this triangulation.



(a) Scan triangulation (b) Delaunay triangulation

Figure 6.2: *Two triangulations of the same 50-point set*

6.2 The Empty Circle Property

The *circumcircle* of a triangle is the unique circle passing through the three vertices of the triangle, see Figure 6.3.

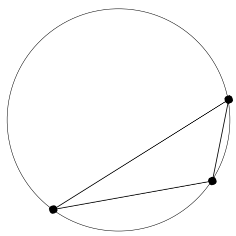


Figure 6.3: *Circumcircle of a triangle*

Definition 6.3 *Given a finite point set* $P \subseteq \mathbb{R}^2$*. A triangulation of* P *is called* Delaunay triangulation *if every triangle has an empty circumcircle, meaning that the* interior *of the circle does not contain any point of* P*.*

Figure 6.4 illustrates this: it shows a Delaunay triangulation of a set of 6 points: the circumcircles of all five triangles are empty (we also say that the triangles satisfy the empty circle property). The dashed circle is not empty, but that's ok, since it is not a circumcircle of any triangle.

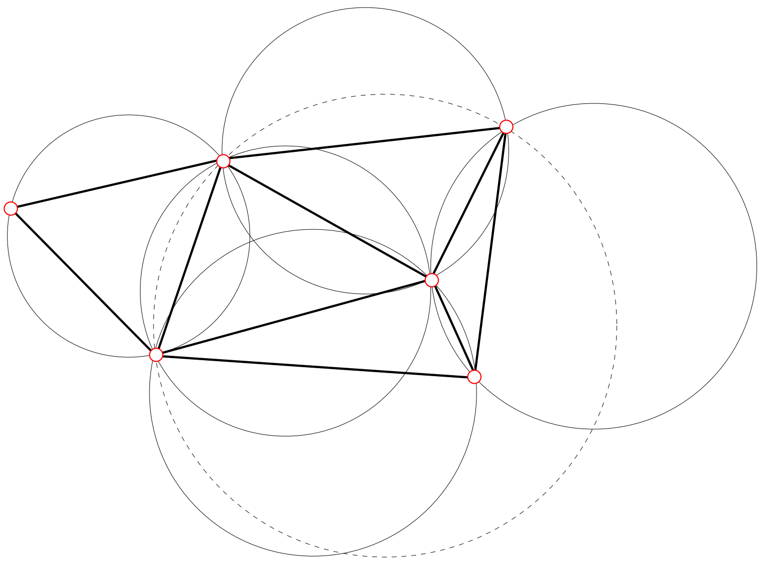
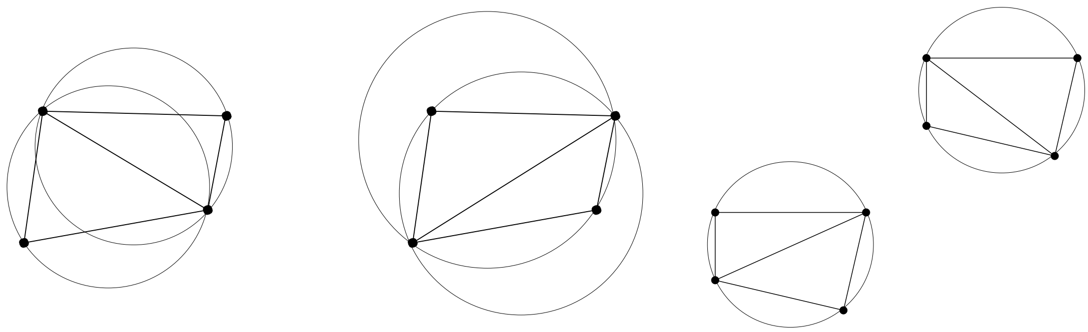


Figure 6.4: *All triangles satisfy the empty circle property*

It is instructive to look at the case of four points in convex position. We already know that there are two possible triangulations, but in general, only one of them will be Delaunay, see Figure 6.5 (a) and (b). If all four points are on a common circle, though, this circle is empty; at the same time it is the circumcircle of *all* possible triangles; therefore, both triangulations of the point set are Delaunay, see Figure 6.5 (c).



(a) Delaunay triangulation (b) Non-Delaunay triangulation (c) Point set with two Delaunay triangulations

Figure 6.5: *Triangulations of 4-point sets*

6.3 The Lawson Flip algorithm

It is not clear yet that every point set actually has a Delaunay triangulation (given that not all points are on a common line). In this and the next two sections, we will prove that this is the case. The proof is algorithmic. Here is the *Lawson flip algorithm* for a set P of n points.

1. Compute some triangulation of P (for example, the scan triangulation)
2. While there exists a subtriangulation of four points in convex position that is not Delaunay (like in Figure 6.5 (b)), replace this subtriangulation by the other triangulation of the four points (Figure 6.5 (a)).

We call the replacement operation in Step 2 a *(Lawson) flip*.

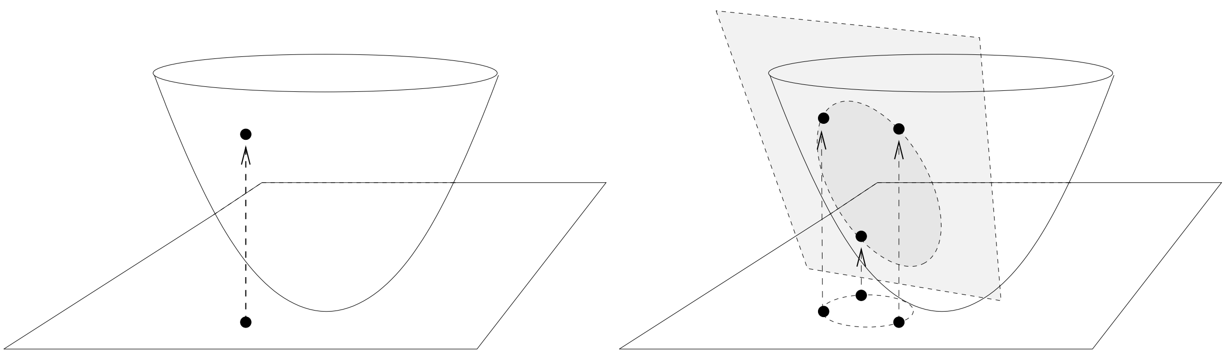
Theorem 6.4 *Let $P \subseteq \mathbb{R}^2$ be a set of n points, equipped with some triangulation $\mathcal{T}$. The Lawson flip algorithm terminates after at most $\binom{n}{2} = O(n^2)$ flips, and the resulting triangulation $\mathcal{D}$ is a Delaunay triangulation of P.*

6.4 Termination of the Lawson Flip Algorithm: The Lifting Map

In order to prove Theorem 6.4, we invoke the *lifting map*. This is the following: given a point $p = (x, y) \in \mathbb{R}^2$, its *lifting* $\ell(p)$ is the point

$$\ell(p) = (x, y, x^2 + y^2) \in \mathbb{R}^3.$$

Geometrically, ℓ "lifts" the point vertically up until it lies on the *unit paraboloid* $\{(x, y, z) \mid z = x^2 + y^2\} \subseteq \mathbb{R}^3$, see Figure 6.6 (a).



(a) The lifting map

(b) Points on/inside/outside a circle are lifted to points on/below/above a plane

Figure 6.6: *The lifting map: circles map to planes*

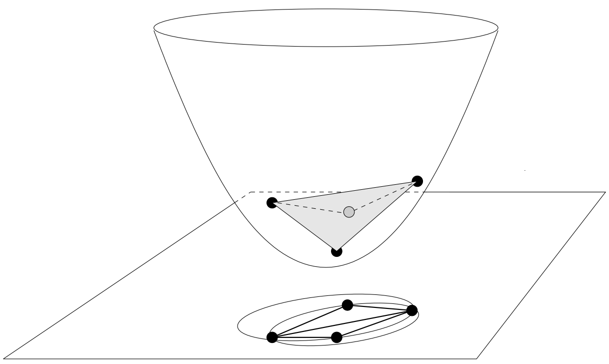
Here is the important property of the lifting map that is illustrated in Figure 6.6 (b) (proof left as an exercise).

Lemma 6.5 *Let* $C \subseteq \mathbb{R}^2$ *be a circle of positive radius. The "lifted circle"* $\ell(C) = \{\ell(p) \mid p \in C\}$ *is contained in a unique plane* $h_C \subseteq \mathbb{R}^3$*. Moreover, a point* $p \in \mathbb{R}^2$ *is strictly inside (outside, respectively) of* C *if and only if the lifted point* $\ell(p)$ *is strictly below (above, respectively)* h_C.

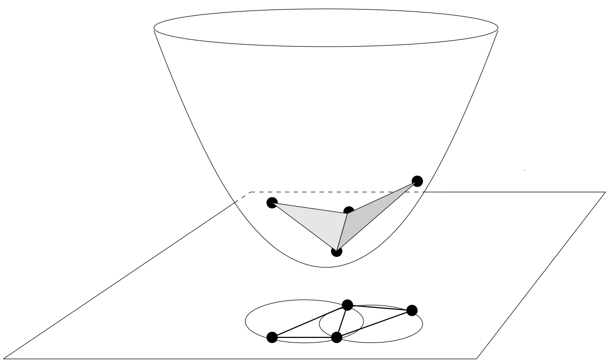
Using the lifting map, we can now prove Theorem 6.4. Let us fix the point set P for this and the next section. First, we need to argue that the algorithm indeed terminates (if you think about it a little, this is not obvious). So let's interpret a flip operation in the lifted picture. The flip involves four points in convex position in $\mathbb{R}^2$, and their lifted images form a tetrahedron in $\mathbb{R}^3$ (think about why this tetrahedron cannot be "flat").

The tetrahedron is made up of four triangles; when you look at it from the top, you see two of the triangles, and when you look from the bottom, you see the other two. In fact, what you see from the top and the bottom are the lifted images of the two possible triangulations of the four-point set in $\mathbb{R}^2$ that is involved in the flip.

Here is the crucial fact that follows from Lemma 6.5: The two top triangles come from the non-Delaunay triangulation before the flip, see Figure 6.7 (a). The reason is that both top triangles have the respective fourth point below them, meaning that in $\mathbb{R}^2$, the circumcircles of these triangles contain the respective fourth point—the empty circle property is violated. In contrast, the two bottom triangles come from the Delaunay triangulation of the four points: they both have the respective fourth point above them, meaning that in $\mathbb{R}^2$, the circumcircles of the triangles do not contain the respective fourth point, see Figure 6.7 (b).



(a) Before the flip: the top two triangles of the tetrahedron and the corresponding non-Delaunay triangulation in the plane



(b) After the flip: the bottom two triangles of the tetrahedron and the corresponding Delaunay triangulation in the plane

Figure 6.7: *Lawson flip: the height of the surface of lifted triangles decreases*

In the lifted picture, a Lawson flip can therefore be interpreted as an operation that replaces the top two triangles of a tetrahedron by the two bottom ones. If we consider the lifted image of the current triangulation, we therefore have a surface in $\mathbb{R}^3$ whose pointwise height can only decrease through Lawson flips. In particular, once an edge has been flipped, this edge will be strictly above the resulting surface and can therefore never be flipped a second time. Since n points can span at most $\binom{n}{2}$ edges, the bound on the number of flips follows.

6.5 Correctness of the Lawson Flip Algorithm: Convex Hulls in $\mathbb{R}^3$

It remains to show that the triangulation of P that we get upon termination of the Lawson flip algorithm is indeed a Delaunay triangulation. Here is a first observation telling us that the triangulation is "locally Delaunay".

Observation 6.6 *Let Δ, Δ' be two adjacent triangles in the triangulation $\mathcal{D}$ that results from the Lawson flip algorithm. Then the circumcircle of Δ does not have any vertex of Δ' in its interior, and vice versa.*

If the two triangles together form a convex quadrilateral, this follows from the fact that the Lawson flip algorithm did not flip the common edge of Δ and Δ'. If the four vertices are not in convex position, this is basic geometry: given a triangle Δ and its circumcircle C, any point in $C \setminus \Delta$ forms a convex quadrilateral with the vertices of Δ.

Now we show that the triangulation is also "globally Delaunay". Together with Lemma 6.5, this is immediately implied by the following

Claim 6.7 *Consider the surface of lifted triangles of the triangulation $\mathcal{D}$ obtained when the Lawson flip algorithm terminates. None of these triangles has a point $\ell(p), p \in P$, below it.*

To prove this, consider any lifted triangle $\ell(\Delta)$ and a lifted point $\ell(p)$ not being a vertex of $\ell(\Delta)$. Choose an open line segment $s \subseteq \mathbb{R}^2$ that connects some interior point of Δ with p. We may choose s in general position, meaning that it does not contain any point of P. Then s traverses a unique sequence of triangles $\Delta = \Delta_1, \Delta_2, \ldots, \Delta_k$, where Δ_k has p as one of its vertices, see Figure 6.8.

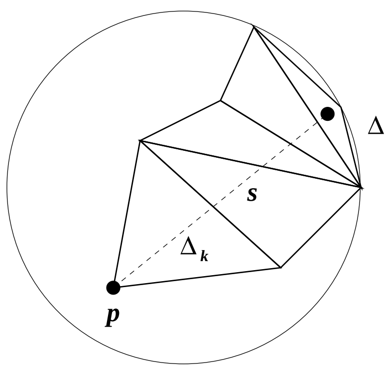


Figure 6.8: *Correctness of the Lawson flip algorithm*

Since we are in Switzerland, we can easily imagine ourselves hiking in a mountainscape (the surface of lifted triangles), heading straight for $\ell(p)$, and starting from a train station in $\ell(\Delta)$. This triangle might not be horizontal, but by rotating the situation, we can imagine $\ell(\Delta)$ to be horizontal.

Now, while hiking from $\ell(\Delta)$ to $\ell(p)$ along $\ell(s)$ on the surface, we cross some edges where the path changes slope (initially, the slope is 0). But whenever we cross an edge, the path becomes steeper [1] by Lemma 6.5, since the triangulation is locally Delaunay by Observation 6.6. Thus, when we look back, we will always see $\ell(\Delta)$. This in particular holds when we reach $\ell(p)$, and this shows that $\ell(p)$ cannot be below $\ell(\Delta)$.

This concludes the proof of Theorem 6.4, but let's put this into a broader perspective: we now know that the lifted triangulation is a surface (consisting of triangles spanned by the lifted points) with the property that for every triangle, the plane spanned by the triangle has no lifted points below it. Equivalently, the upper halfspace associated to this plane contains all the lifted points. This should ring a bell, see Theorem 1.9.

Without a formal proof (this and the previous section should serve as informal proofs), we state the following

Theorem 6.8 *Let $P \subseteq \mathbb{R}^2$ be a set of n points, not all of them on a common line, and let $\mathcal{D}$ be a Delaunay triangulation of P. Then the surface $\ell(\mathcal{D})$ of lifted triangles is the* lower convex hull *of the lifted point set $\ell(P)$. The lower convex hull consists of exactly those facets of the polytope* $\text{conv}(\ell(P))$ *that have no points below them (see Figure 6.9).*

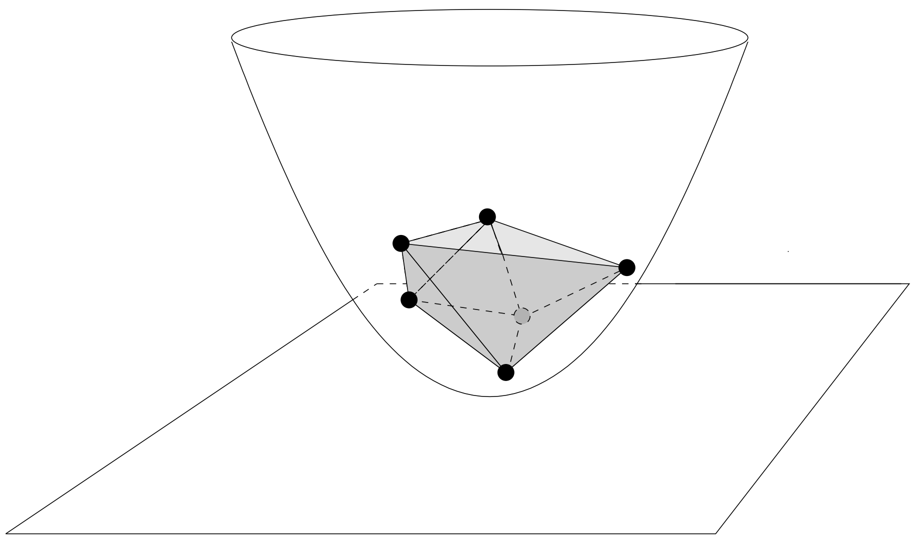


Figure 6.9: *Lower convex hull of lifted points*

[1] or keeps its slope; but it will never become more flat

This also yields the following result that we leave as an exercise: given the facets of the polytope $\text{conv}(\ell(P))$ (these may be triangles, but also polygons of higher order), we can construct a Delaunay triangulation of P in time $O(n)$. We have therefore reduced the problem of computing Delaunay triangulations in $\mathbb{R}^2$ to the problem of computing convex hulls in $\mathbb{R}^3$. This will be important when we talk about *efficient* algorithms for computing Delaunay triangulations (it turns out that the Lawson flip algorithm is not the best possible choice).

6.6 The Delaunay Triangulation Maximizes the Smallest Angle

Why are we interested in Delaunay triangulations at all? After all, having empty circumcircles is not a goal in itself. But it turns out that Delaunay triangulations satisfy a number of interesting properties. Here we just show one of them.

Given a triangulation $\mathcal{T}$ of P, we consider the sorted sequence $A(\mathcal{T}) = (\alpha_1, \alpha_2, \ldots \alpha_{3m})$ of interior angles, where m is the number of triangles (we have already remarked earlier that m is a function of P only and does not depend on $\mathcal{T}$. Being sorted means that $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_{3m}$. Let $\mathcal{T}, \mathcal{T}'$ be two triangulations. We say that $A(\mathcal{T}) < A(\mathcal{T}')$ if there exists some i for which $\alpha_i < \alpha'_i$ and $\alpha_j = \alpha'_j, j < i$. This means that $A(\mathcal{T})$ is *lexicographically smaller* than $A(\mathcal{T}')$.

Here is the result.

Theorem 6.9 *Let $P \subseteq \mathbb{R}^2$ be a finite set of points, not all on a line, let $\mathcal{D}$ be a Delaunay triangulation of P, and let $\mathcal{T}$ be any triangulation of P. Then*

$$A(\mathcal{T}) \leq A(\mathcal{D}).$$

In particular, $\mathcal{D}$ *maximizes the smallest angle* among all triangulations of P. Filling in the details of the following proof remains an exercise.

Proof. We know that $\mathcal{T}$ can be transformed into a Delaunay triangulation $\mathcal{D}'$ through the Lawson flip algorithm. It can be shown that each such flip lexicographically increases the sorted angle sequence, so that

$$A(\mathcal{T}) \leq A(\mathcal{D}').$$

Moreover, one can show that $A(\mathcal{D}) = A(\mathcal{D}')$, and this finishes the proof. □

References