"A convex quadrilateral has two diagonals, and in a triangulation one can be substituted for the other."

9 Edge flipping

Recall lemma C from section 8, which says if all edges of a triangulation are locally delone then the triangulation is the Delaunay triangulation of the point set. This result suggests an iterative algorithm that substitutes a locally delone edge for one that is not locally delone. The idea for this algorithm is due to Lawson [3, 4], and it predates its analysis based on lifting the points from two to three dimensions.

Flipping. Assume $\mathcal{K}$ is an arbitrary triangulation of a finite set $S \subseteq \mathbf{R}^2$ in general position. Unless $\mathcal{K} = \mathcal{D}_S$, there is an edge $xy \in \mathcal{K}$ that is not locally delone. Let axy and bxy be the two triangles that share xy. b lies inside the circle through a, x, y, so $axy \cup bxy$ is a convex quadrilateral, see figure 9.1. The other diagonal, ab, is locally delone.

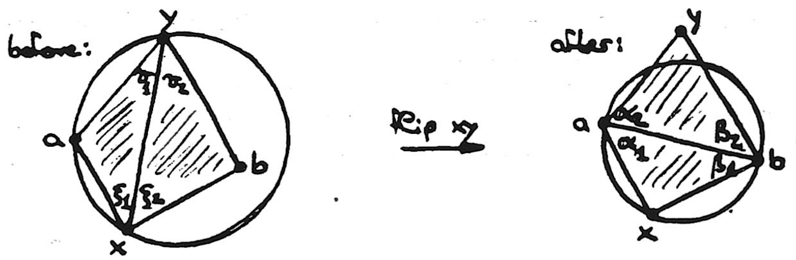

Figure 9.1: A flip substitutes one triangulation of a convex quadrilateral for the other.

To *flip* xy means to substitute $\mathcal{L} = \mathcal{K} - \{xy, axy, bxy\} \cup \{ab, xab, yab\}$ for $\mathcal{K}$. Note that a flip has a direction with tendency towards locally delone edges. There is no such flip that takes $\mathcal{L}$ to $\mathcal{K}$.

Algorithm. The details of the resulting iterative algorithm are now specified. It starts with an arbitrary triangulation $\mathcal{K}_0$ of S. A stack is used to quickly identify edges to be flipped. We mark edges to keep track of the ones that are on the stack. The stack is initialized to contain all edges of $\mathcal{K}_0$, so all edges are marked. The following invariants are maintained throughout the algorithm:

(I1) the stack contains only edges of the current triangulation, $\mathcal{K}$, and it contains at most one copy of each,

(I2) it contains all edges that are not locally delone in $\mathcal{K}$, and possibly others.

Edges are flipped until all are locally delone and the stack is empty.

```
while stack is non-empty do
        pop xy from the stack and unmark it;
        if xy is not locally delone then
            flip xy by substituting it with ab;
            for uv ∈ {ax, ay, bx, by} do
                 if uv is not marked then push uv and mark it endif
            endfor
        endif
endwhile.
```

It is easy to see that (I1) and (I2) hold initially, for $\mathcal{K} = \mathcal{K}_0$, and are maintained throughout. By Euler's relation, $\mathcal{K}$ has fewer than $3n$ edges, $n = \text{card}\, S$, so (I1) implies that the size of the stack never exceeds $3n$. For an efficient execution, we need a data structure for $\mathcal{K}$ that takes only constant time for local operations such as finding the triangles containing a given edge. A number of such data structures have been proposed in the literature, and the two most popular ones are possibly the winged-edge [1] and the quad-edge [2] data structures. Even with the assumption of such a data structure, it is not obvious how to bound the running time of the iteration. The main difficulty is that the stack shrinks and grows in a seemingly unpredictable manner. We need a better understanding of the changing geometry to get a handle on the time-complexity.

Lifting map. We introduce a map $\lambda : \mathbf{R}^2 \to \mathbf{R}^3$ and study the effect of the algorithm on $\lambda(S)$, which is a finite set in $\mathbf{R}^3$. For every $p = (\phi_1, \phi_2) \in \mathbf{R}^2$ we define $\lambda(p) = (\phi_1, \phi_2, \phi_1^2 + \phi_2^2)$. λ can be visualized by identifying $\mathbf{R}^2$ with the x_1x_2-plane in $\mathbf{R}^3$; it projects every point in this plane vertically into the paraboloid of revolution, $\pi : x_3 = x_1^2 + x_2^2$, see figure 9.2.

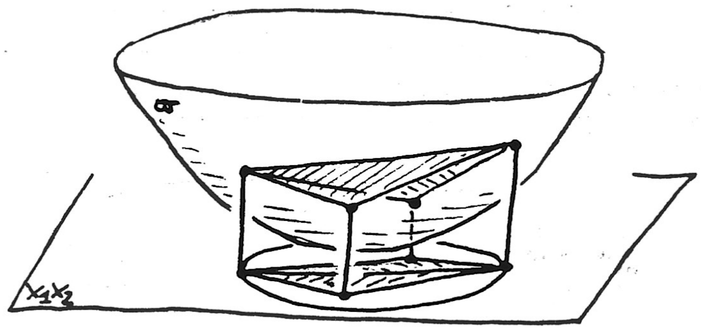

Figure 9.2: λ defines a bijection between the x_1x_2-plane and the paraboloid π. Two triangles that are not locally delone in the plane correspond to two lifted triangles forming a concave dihedral angle.

LEMMA A. For every circle c in $\mathbf{R}^2$ there is a plane h in $\mathbf{R}^3$ so that $c = \lambda^{-1}(h \cap \pi)$.

PROOF. We construct the plane from c. Let γ_1, γ_2 be the coordinates of the center and ρ the radius of c. The linear relation $x_3 = 2\gamma_1 x_1 + 2\gamma_2 x_2 - (\gamma_1^2 + \gamma_2^2 - \rho^2)$ specifies a plane, h, and the projection of $h \cap \pi$ into the first two coordinates is give by

$$(x_1 - \gamma_1)^2 + (x_2 - \gamma_2)^2 = \rho^2,$$

which is c. $\square$

REMARK. The *in-circle test* decides whether a point $x \in \mathbf{R}^2$ lies inside or outside the circle through three points $a, b, c \in \mathbf{R}^2$. The lemma implies that this test can be decided by checking whether $\lambda(x) \in \mathbf{R}^3$ lies below or above the plane through $\lambda(a), \lambda(b), \lambda(c)$. The corresponding algebraic test is a product of two determinants:

$$\det \begin{pmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_1^2 + \alpha_2^2 \\ 1 & \beta_1 & \beta_2 & \beta_1^2 + \beta_2^2 \\ 1 & \gamma_1 & \gamma_2 & \gamma_1^2 + \gamma_2^2 \\ 1 & \xi_1 & \xi_2 & \xi_1^2 + \xi_2^2 \end{pmatrix} \cdot \det \begin{pmatrix} 1 & \alpha_1 & \alpha_2 \\ 1 & \beta_1 & \beta_2 \\ 1 & \gamma_1 & \gamma_2 \end{pmatrix},$$

and x lies inside (outside) the circle iff the product is negative (positive). Degenerate cases correspond to vanishing determinants.

Time-complexity. For a triangulation $\mathcal{K}$ of $S \subseteq \mathbf{R}^2$, define $\mathcal{K}_\pi = \{\text{conv}\,\lambda(T) \mid \text{conv}\,T \in \mathcal{K}\}$. $\mathcal{K}_\pi$ is a simplicial 2-complex in $\mathbf{R}^3$, and its underlying space, $|\mathcal{K}_\pi|$, is a piecewise linear surface that intersects a vertical line in a point or not at all. An edge $xy \in \mathcal{K}$ corresponds to the edge $\lambda(x)\lambda(y) \in \mathcal{K}_\pi$. Lemma A implies xy is not locally delone in $\mathcal{K}$ iff the dihedral angle at $\lambda(x)\lambda(y)$ below $\mathcal{K}_\pi$ is less than 180°, see figure 9.2. Incidentally, this implies that $\mathcal{K} = \mathcal{D}_S$ iff $\mathcal{K}_\pi$ is a convex surface. We will elaborate on this observation later. The edge-flipping algorithm can now be visualized in $\mathbf{R}^3$. Flipping xy corresponds to gluing a tetrahedron, $\lambda(x)\lambda(y)\lambda(a)\lambda(b)$, underneath $|\mathcal{K}_\pi|$.

The new lower surface is the underlying space of the new $\mathcal{K}_\pi$ which projects to the new $\mathcal{K}$. Each flip creates a tetrahedron, and the entire algorithm creates a simplicial 3-complex in $\mathbb{R}^3$. The number of vertices of this complex is $n = \operatorname{card} S$.

LEMMA B. A simplicial complex with n vertices in $\mathbb{R}^3$ has fewer than $\frac{n^2}{2}$ tetrahedra.

PROOF. Consider all tetrahedra $abcx$ that share a common vertex x. The triangles abc form a 2-complex with at most $n-1$ vertices around x. The 1-skeleton of this 2-complex is a planar graph, so there are at most $2(n-1)-4 = 2n-6$ triangles. The total number of tetrahedra can thus not exceed $\frac{n}{4}(2n-6) < \frac{n^2}{2}$. □

Since each flip is encoded by a tetrahedron, the algorithm cannot perform more than $\frac{n^2}{2}$ flips, and therefore takes time at most $O(n^2)$. As mentioned earlier, the amount of storage is only $O(n)$ because the marking mechanism prevents the stack from growing beyond $3n$ edges.

Maxmin angle criterion. The edge-flipping algorithm can be used as a tool to proving that the Delaunay triangulation optimizes certain quality criteria, over all triangulations of some fixed point set. One such criterion is the smallest angle. For a triangulation $\mathcal{K}$ of S let $\alpha(\mathcal{K})$ be the smallest angle within any triangle in $\mathcal{K}$. We begin by arguing that a flip cannot decrease the size of the smallest angle. Consider the flip shown in figure 9.1. The six angles before the flip are $\alpha_1+\alpha_2, \beta_1+\beta_2, \xi_1, \xi_2, v_1, v_2$, and the angles after the flip are $\alpha_1, \alpha_2, \beta_1, \beta_2, \xi_1+\xi_2, v_1+v_2$. All angles are positive, so $\xi_1, \xi_2 < \xi_1+\xi_2$ and $v_1, v_2 < v_1+v_2$. Furthermore, $\xi_1 < \beta_2$, $\xi_2 < \alpha_2$, $v_1 < \beta_1$, and $v_2 < \alpha_1$ because b lies inside the circle of axy and a lies inside the circle of bxy. So no matter which one of the 6 angles after the flip is smallest, there is an even smaller angle before the flip. Sibson [6] was the first to realize this implies that Delaunay triangulations maximize smallest angles.

THM. 9.1 Let $\mathcal{K}$ be a triangulation of a finite set $S \subseteq \mathbb{R}^2$ in general position. Then $\alpha(\mathcal{K}) \leq \alpha(\mathcal{D}_S)$.

PROOF. $\mathcal{D}_S$ can be constructed from $\mathcal{K}$ be a finite sequence of edge flips. Each flip either increases α or leaves it unchanged. □

REMARK. The result in thm. 9.1 can be strengthened by considering the lexicographically ordered angle vector. Let $\mathcal{K}$ and $\mathcal{L}$ be two triangulations of S and let κ_i and λ_i be the i-smallest angle in $\mathcal{K}$ and $\mathcal{L}$, respectively. Write $\mathcal{K} \prec \mathcal{L}$ if there is a j so that $\kappa_j < \lambda_j$ and $\kappa_i = \lambda_i$ for $i < j$. With the same argument as for thm. 9.1 we get $\mathcal{K} \prec \mathcal{D}_S$ for all triangulations $\mathcal{K} \neq \mathcal{D}_S$ of S.

Homework exercises

Let S be a set of n points in general position in $\mathbb{R}^2$. The *flip graph* of S, $F = (T, A)$, is defined as follows: T is the set of triangulations of S, and $(\mu, \nu) \in A$ if ν can be obtained from μ by a single flip substituting a locally delone edge for one that is not locally delone. We have seen that $\omega = \mathcal{D}_S \in T$ is the only sink and every node has a path of length less than $\frac{n^2}{2}$ to ω. A map $\varphi : T \to \mathbb{R}$ is *monotone* if $\varphi(\mu) \leq \varphi(\nu)$ for all arcs $(\mu, \nu) \in A$, or $\varphi(\mu) \geq \varphi(\nu)$ for all arcs. Observe that if φ is monotone then the Delaunay triangulation maximizes or minimizes φ over all triangulations of S.

9.1 Define $\varphi_1(\mu)$ equal to the largest circumcircle radius of any triangle in μ. Argue that φ_1 is monotone.

9.2 Define $\varphi_2(\mu)$ equal to the smallest circumcircle radius of any triangle in μ. Argue that φ_2 is monotone.

9.3 Define the *minidisk* of a triangle σ as the smallest disk that contains σ; its bounding circle is not necessarily the circumcircle of σ. Define $\varphi_3(\mu)$ equal to the largest minidisk radius of any triangle in μ. Show that φ_3 is monotone.

(Remark. It is interesting that unlike for φ_1 and φ_2 the stronger result about the lexicographic optimization is not correct for φ_3. On the other hand, the result for φ_3 generalizes to three and higher dimensions [5], while φ_1 and φ_2 are not necessarily optimized by Delaunay triangulations already in 3-dimensional space.)

9.4 Define $\varphi_4(\mu)$ equal to the largest angle within any triangle in μ. Show that $\omega = \mathcal{D}_S$ does not minimize φ_4.

References

[1] B. G. Baumgart. A polyhedron representation for computer vision. *In* "Proc. AFIPS Natl. Comput. Conf. **44**, 1975", 589–596.

[2] L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Trans. Graphics* 4 (1985), 74–123.

[3] C. L. Lawson. Generation of a triangular grid with applications to contour plotting. Memo. 299, Calif. Inst. Jet Propulsion Lab., 1972.

[4] C. L. Lawson. Software for C^1 surface interpolation. *In* "Mathematical Software III", ed. J. Rice, Academic Press, 1977, 161–194.

[5] V. T. Rajan. Optimality of the Delaunay triangulation in $\mathbb{R}^d$. *In* "Proc. 7th Ann. Sympos. Comput. Geom., 1991", 357–363.

[6] R. Sibson. Locally equiangular triangulations. *Comput. J.* **21** (1978), 243–245.