Advanced algorithms
Delaunay Triangulation Notes

Søren Dahlgaard

June 15, 2011

0.1 Disposition

1. Motivation - Terrains, maximize minimum angle
2. Thales's Theorem - Flipping edges and legal edges
3. Voronoi diagrams and the duality
4. Delaunay graphs are plane. (use disc theorem.)
5. Correctness of Delaunay triangulation.
6. Incremental method.
7. Finding the triangle containing p_r
8. p_{-1} and p_{-2}.
9. Running time.

0.2 Summary

When modeling terrains we often want to express a 3D space by a 2D surface. If we have n points in the continous space we would also like to approximate the height of the intermediate points. Essentially we want to create a function $f : A \subset \mathbb{R}^2 \to \mathbb{R}$ that gives a point a height.

We could simply assign a point (x, y) the height of the closest point $p \in A$. This would however result in a ugly discrete terrain. The idea is therefore to divide the terrain into triangles where each triangle vertices are in A. When we have done this we raise each point in A to its height.

We also want to create a good such triangulation. Our intuition (see fig 9.3) in the material tells us that a good triangulation is one that maximizes the minimum angle. Thus given a triangulation of m triangles we define its angle-vector as $(\alpha_1, \alpha_2, \ldots, \alpha_{3m})$ the sorted list of the $3m$ angles in the triangulation. Order such vectors lexicografically:

$$(\alpha_1, \ldots, \alpha_{3m}) < (\alpha'_1, \ldots, \alpha'_{3m}) \Leftrightarrow \exists i : \alpha_j = \alpha'_j, j < i \wedge \alpha_i < \alpha'_i$$

Euler's characteristic[1] gives us that there are $2n-2-k$ triangles and $3n-3-k$ edges, where k is the amount of vertices on the convex hull. This is because there is $m+1$ faces (1 unbounded) and $(3m+k)/2$ edges - 3 edges per triangle plus k of the unbounded. Furthermore all edges are incident to two triangles). Thus

$$2 = n - ((3m+k)/2) + (m+1)$$

We use Thales's Theorem to prove most of the things for delaunay triangulations. It states that if we have a circle C and two points a, b on the circle. Let l be the line passing through a, b. Let p, q be two points on the circle on the same side of l, then $\angle apb = \angle aqb$. If r and s are two points on the same side of l as p, q with r inside C and s outside C, then:

$$\angle arb > \angle apb = \angle aqb > \angle asb$$

If we have an edge $\overline{p_i p_j}$ that is not on the convex hull, then we have that it is incident to two triangles $p_i p_j p_k$ and $p_i p_j p_l$. We call flipping edges $\overline{p_i p_j}$ with $\overline{p_k p_l}$ for an edge flip. It changes maximum 6 angles. We call $\overline{p_i p_j}$ illegal if:

$$\min_{1 \leq i \leq 6} \alpha_i < \min_{1 \leq i \leq 6} \alpha_i'$$

It is obvious that if we have an illegal edge in a triangulation, we can make a better triangulation by flipping it.

Rather than checking all three angles when comparing an edge to its flipped version we can use Thales's Theorem:

If $\overline{p_i p_j}$ is an edge incident to triangles with p_k, p_l. Let C be the circle with p_i, p_j, p_k on its circumference. Then $\overline{p_i p_j}$ is illegal iff p_l lies within C.

Thus we can find an optimal triangulation by starting with any triangulation and flipping edges until no edges are illegal. Note that this algorithm terminates because we improve the triangulation at each step and there is a finite amount of triangulations.

1 Delaunay triangulation

A voronoi diagram is a where each point p has its own face such that for all points in the face, p is the closest point.

Let us define a graph where we connect each such point p with each of the points it shares an edge. Such a graph is called a Delaunay graph. We claim that it is planar (no edges cross except at endpoints). We prove this by contradiction.

Firstly: *The edge $\overline{p_i p_j}$ is in the delaunay graph iff there is a circle C_{ij} with p_i, p_j on its boundary and no other point contained in it. The center of this disc will lie on the voronoi edge between p_i, p_j*

We know this to be true and will use it: Let t_{ij} be the triangle with p_i, p_j and the center of C_{ij}. Assum another edge $\overline{p_k p_l}$ intersects with $\overline{p_i p_j}$, then both of these points must lie outside C_{ij}, and thus the edge must cross one of the edges of t_{ij} that goes from the center of C_{ij}. The same goes the other way, but then they can't be disjoint voronoi cells because the edge from p_i to center of C_{ij} lies entirely in $\text{Vor}(p_i)$.

We can extend this:

[1] `http://en.wikipedia.org/wiki/Euler_characteristic`

1. points $p_i, p_j, p_k \in P$ are vertices of the same face in the delaunay graph of P iff the circle through the points contains no other points in P.

2. Points p_i, p_j form an edge in the delaunay graph iff there is a closed disc containing p_i, p_j on its boundary such that no $p \in P$ is in this disc.

This implies:
T is a delaunay triangulation of P iff the circumcircle of any triangle of T contains no point of P in its interior.

We have that *A triangulation T of P is legal iff it is a delaunay triangulation.*
Proof: It is easy to see that any delaunay triangulation is legal.

Assume for contradiction that T is legal, but there is a triangle $\angle p_i, p_j, p_k$ such that p_l is in their circumcircle. If The triangle $\angle p_i, p_j, p_l$ is incident to $\angle p_i p_j p_k$ then the triangulation is not legal and we are done. Otherwise look at $\angle p_i p_j p_m$ the triangle incident. We assume without loss of generality that it is the edge $\overline{p_i p_j}$ that separates p_l from p_k. We also assume that $(p_i p_j p_k, p_l)$ is the pair maximizing $\angle p_i p_l p_j$. Then we have an illegal edge $\overline{p_j p_m}$ that we could flip with $\overline{p_i p_l}$.

Therefore we have that any angle-optimal terrain is a delaunay triangulation. Also any Delaunay triangulation maximizes the minimum angle over all triangulations.

1.1 Computing the delaunay triangulation

We could calculate the voronoi diagram and then create the triangulation, but we will do it in an incremental way:

1. Start with a huge triangle $\angle p_{-2} p_{-1} p_0$ that contains all of P.

2. For each point $p_i \in P$ at random add it and find the triangle of T that contains p_i.

3. If p_i lies on an edge $\overline{p_j p_k}$ add an edge from p_i to the last vertex of both the triangles incident to $\overline{p_j p_k}$. If p_i lies in a triangle add edges between p_i and each of the triangle's vertices.

4. The edges of the triangles we have changed might be illegal now, so legalize them.

If the triangle $\angle p_i p_j p_k$ was changed by adding a new point we have to legalize the edges: $\overline{p_i p_j}, \overline{p_i p_k}, \overline{p_j p_k}$. If we flip these edges we have to legalize the triangles affected recursively. Note that each flip increases the angles so it cannot loop infinitely. All new edges are incident to the added point p_l.

The idea behind proving the correctness of the algorithm is to see that each edge added is legal. This is done by seeing that the original circumcircle contains only p_l in its interior, thus the new circumcircle cannot contain any other points.

In order to find which triangle contains the new point p_l we use a directed acyclic graph (NOT A TREE!), such that:

1. The "leaves" are exactly the triangles of T

2. The internal nodes are previous triangles of T that were removed.

When we split a triangle it results in at most 3 new triangles:

- Insert causes either $1 \to 3$ or $2 \to 4$
- Flipping causes $2 \to 2$.

Each deleted triangle gets a pointer to the new triangles created from it.

Picking p_{-1}, p_{-2} and handling them. Instead of actually creating these vertices we just create them conceptually. Conceptually we have:

1. Let l_{-1} be a horizontal line below all points in P. We add p_{-1} as a point on this line with high enough x-coordinate such that no circumcircle contains p_{-1} and the clockwise ordering of P around p_{-1} is the same as their lexicographical order.
2. p_{-2} is the opposite except we include p_{-1} in P for circumcircles and ordering.

With these choices we can find out the position of p_k wrt. $\overline{p_i p_j}$. The following are equivalent:

1. p_j lies left of $\overline{p_i p_{-1}}$
2. p_j lies left of $\overline{p_{-2} p_i}$
3. p_j is lexicographically larger than p_i.

How do we check if an edge containing p_{-1} or p_{-2} is legal:

- An edge on $\angle p_{-2} p_{-1} p_0$ is always legal
- if $j, k, i, l \geq 0$ no special case is needed.
- An edge $\overline{p_i p_j}$ is legal iff $\min(k, l) < \min(i, j)$. At most one of i, j is negative (otherwise it's handled by case 1) and at most one of k, l is negative because we just inserted one of those. If $\min(k, l)$ is smallest we have that it lies outside the circle defined by the other 3 points.

1.1.1 Running time

The expected number of triangles created by the procedure is $9n + 1$. At each step we create k edges and all of these are connected to the newly inserted point, thus k is bounded above by the degree of this vertex. We have that the graph has at most $3(r + 3) - 6$ where $r + 3$ is the amount of vertices (3 initial plus r iterations). Thus the expected degree is

$$\frac{\text{total degree}}{r+3} = \frac{2(3(r+3)-9)}{r+3} = O(\frac{6r}{r})$$

We create at most $2(k - 3) - 3 = 9$ new triangles in each iteration so a total of $9n$ triangles plus the first one.

We know that the total amount of triangles created is linear, so the bounding factor must be the running time of the triangle locations. We do $O(n)$ of these. The idea is that if we visit a triangle $\angle p_i p_j p_k$ during our search for a leaf, we can attribute this to some triangle destroyed during the same iteration as $\angle p_i p_j p_k$ (either the triangle itself or an incident triangle.) If $K(\Delta)$ denotes the amount of points in the circumcircle of δ. Each triangle can be charged for each such point at most once (because we search for each point once), so we have the total running time is:

$$O(n + \sum_{\Delta} |K(\Delta)|$$

This is claimed to be $O(n \lg n)$.