Introduction to Voronoi Diagrams and Delaunay Triangulations

Solomon Boulos
Voronoi Diagrams

Voronoi region:

\[ V(p_i) = \{ x \in \mathbb{R}^n \mid \| p_i - x \| \leq \| p_j - x \|, \forall j \leq n \} \]
Delaunay Triangulation

- Dual to Voronoi Diagram
- Connect Vertices Across Common Line
Circumcircle Property

\[ T_i \in D(S) \iff \text{The circumcircle of } T_i \text{ is empty} \]
General Position

What if you had points on the same circle?
Perturbation

What if we move a point just slightly?
Circumcircle Test

- Simple determinant test
  \[
  \begin{vmatrix}
  x^2 + y^2 & x & y & 1 \\
  x_1^2 + y_1^2 & x_1 & y_1 & 1 \\
  x_2^2 + y_2^2 & x_2 & y_2 & 1 \\
  x_3^2 + y_3^2 & x_3 & y_3 & 1 \\
  \end{vmatrix}
  \]

- Equal 0 → On Circle
- Less Than 0 → Outside Circle
- Greater Than 0 → Inside Circle
Symbolic Perturbation

- Determinant == 0 is very sensitive
- Sensitive enough for an $\epsilon$

$$\Delta(\epsilon) = \begin{vmatrix} 1 & \xi_{2i-1} + \epsilon_{2i-1} & \xi_{2i} + \epsilon_{2i} \\ 1 & \xi_{2j-1} + \epsilon_{2j-1} & \xi_{2j} + \epsilon_{2j} \\ 1 & \xi_{2k-1} + \epsilon_{2k-1} & \xi_{2k} + \epsilon_{2k} \end{vmatrix}$$

- Further breaks up into sub-determinants
Naive Algorithm

- Take 3 points at random
- See if Circumcircle is Empty (test against all points)
- If not, add triangle
Circumcircle Test is localized

- For nicely distributed points the circumcircles are small
- Suggests incremental algorithm
Incremental Algorithm

- Assume we start with a current delaunay triangulation
- Choose a new vertex to add at random
- Add new triangles, flip edges
Example

Triangle pab is illegal.

Triangle pad is okay.

Triangle pdb is illegal.
Example Continued

Triangle pbc is illegal.

Triangle peb is okay.

Triangle pde is okay.
Example Finished

Triangle pbf is okay.

Triangle pfc is okay.

Triangle pca is okay.
Complete Algorithm

- Find the triangle $abc$ containing the new point
- Add new edges linking the new point to the vertices of $abc$
- Perform Swap Test until Stack is Empty
The Swap Test

- If edge is on the convex hull, skip
- Otherwise check quad for circumcircle test
- If test fails, add new edges
Why does it work?

- You start with a Delaunay Triangulation
- In each step, only a local neighborhood needs fixup
- Each flip may break two more edges (ad and db)
The Initial Triangulation

- Ugliest Detail
- Create a “big triangle”
- Needs vertices that won’t be within the circumcircles
- Remove vertices from final triangulation
Total Running Time

- Time to find containing triangle (happens $n$ times)
  - There exist $O(\log n)$ methods for this
- Time to perform insertion (happens $n$ times)
  - Time to perform a swap ($O(1)$, but how many times?)
- Total Time = $O(n \log n + nu)$
Look at single flip

- Replaces link edge with new edge
- Adds back two link edges (net increase 1)
- Number Link Edges - 3 = Number of Flips
Upper bound

- Number Link Edges = Vertex Degree
- $3 \leq \text{Vertex Degree} \leq 6$ for planar graphs
- Max flips = 3
- Total Running Time = $O(n \log n)$
Limitations

- Edge Flips don’t extend past 2D
- Needs initial big triangle
Determinants and Orientation

A determinant is a signed volume

\[
\begin{vmatrix}
  x_1 & y_1 & 1 \\
  x_2 & y_2 & 1 \\
  x_3 & y_3 & 1 \\
\end{vmatrix}
= \pm \| (\vec{v}_2 - \vec{v}_1) \times (\vec{v}_3 - \vec{v}_1) \| 
\]

- Less than 0 → Clockwise
- Greater than 0 → Counter-Clockwise
Circumcircle as Orientation

- The Circumcircle test is an orientation test
- Let $p' = (p, \|p\|^2)$
Properties of the Space of Spheres

- Points below the cutting plane are inside the circle
- Also applies to triangular regions
Using Convex Hulls

- Simple Algorithm

1. Project onto paraboloid.
2. Compute convex hull.
3. Project hull faces back to plane.
Advantages of Convex Hull Approach

- Many good Convex Hull Algorithms (e.g. QuickHull)
- Simple extension to arbitrary dimensions
- No strange infinite triangle initialization
Questions?