Voronoi diagrams
Marko Täht
Content

• Voronoi diagram
• Delauney triangulation
• How to create Voronoi diagram
• Uses of voronoi diagrams
Voronoi diagram

• Voronoi diagram is partitioning of space into areas based on distance to points in a specific subset of the space.
Voronoi properties

• Is a dual graph for Delauney Triangulation
• Closest pair of points correspond to adjacent cells in diagram
• Small change in a site location corresponds to a small change in Diagram (http://blog.ivank.net/voronoi-diagram-in-javascript.html)
Centroidal Voronoi Tessellation

- Voronoi tessellation is centroidal if each of the point is also the center of mass for its cell.
Weighted Voronoi Diagram

• Multiplicative weighted Voronoi diagram
  • Distance between points is multiplied by positive weights
  • Also known as circular Dirichlet tessellation
• Additively weighted Voronoi Diagram
  • Positive weights are subtracted from distance between points
  • Also known as hyperpolic Dirichlet tessellation
Delauney triangulation

• On a plane triangulation of a set of points, in such way that no point is in a circumcircle of any triangle.
Delauney properties

• Connecting the circumcircle centers you get the Voronoi graph for the given set of points
• Maximises the minimum angle not the edge length

Maximise minimum angle
Terrain generation using Delauney

- https://straypixels.net/delaunay-triangulation-terrain/
How to create voronoi diagram

- Brute force method
- Fortunes algorithm
- Bowyer-Watson algorithm
Brute force

• Choose a point
• Calculate distance to every other point
• Choose the point closest
• Make a bisecting edge between the 2 points
• Remove all the points in the direction of the other point that are further away than the point.
• Repeat until no more points can be removed
• Find intersections of adjacent edges
• Connect edges on intersection points
Brute force

• Easy to implement
• Sufficient for small number of points
• Too slow for large number of points
Fortune algorithm

- Also known as sweepline algorithm
- Runs in $O(n \log n)$

Required data structures:
- Binary tree (beach line)
- Sweepline ($y$ coordinate of sweepline)
- Priority queue (sort events by $y$ coordinate)

Event types:
- Site event (new arc added to beach line)
- Circle event (Arc is removed from beach line)
Beachline

Used to easily find arc under a new site and to check for circle events.
Site event

- New point from the queue is added to the beachline.
Circle event

• Arc is removed from the beachline
Algorithm

Q = queue
P = set of points

For each p in P:
    Q.add(new SiteEvent(P))

While Q is not empty:
    q = Q.pop()
    if q is site event: handleSiteEvent(q)
    else handleCircleEvent(q)
• Pros:
  • Fast

• Cons:
  • Difficult to understand
  • Difficult to implement
  • Implementation can have many special cases
  • Lose the benefits when extended to higher dimensions
Bowyer-Watson algorithm

- Used to create Delauney triangulation in N-dimension
- Algorithm:
  - List triangulation
    Add triangle that envelops all the points in the triangulation list
  1. Add point
  2. Find all triangles where the new point is in (bad triangles)
  3. Find all edges that are between bad triangle and good triangle
  4. Construct new triangles with the edges
  5. Remove bad triangles
  6. Repeat until no more points can be added

Remove all triangles that have a vertex from the super triangle
Pros:
- Works in N-dimensions
- Easy to implement
- With optimizations runs in $O(N \log N)$

Cons:
- Worst case performance $O(N^2)$

Delauney -> Voronoi:
- Find all the circumcenters of the triangles
- Connect adjacent triangle circumcenters with edge.
Centroidal Voronoi Diagram Creation

• Created using Lloyds algorithm
• Random points are generated
• Voronoi diagram Is calculated
• For each cell calculate center of mass
• Move point to the center of mass
• Repeat
Uses of Voronoi diagram

- Use for AI
- Procedual generation
- Art
- Calculate cracks in objects
Voronoi diagram for AI control

- Find nearest health pack, safe point, etc.
- Find safest route to finish. Closeness to points adds weight to voronoi edges. Use some pathfinding algorithm over the edges to find the path with minimum weight.
Terrain generation using Voronoi diagram


• http://www-cs-students.stanford.edu/~amitp/game-programming/polygon-map-generation/

• https://watabou.itch.io/medieval-fantasy-city-generator
Art

- Convert images into voronoi diagrams.
- Edges and darker colors get more points. Each cell is colored with the mean color of the original pixels in the cell.
• Or having more control on where the points are generated or how they are colored, can be used to generate art.
Calculate cracks in objects

- Calculate Voronoi diagram.
- Clip it into shape.
Used literature

- https://en.wikipedia.org/wiki/Fortune%27s_algorithm
- https://en.wikipedia.org/wiki/Lloyd%27s_algorithm
- https://www.desmos.com/calculator/ejatebvup4
- https://www.codeproject.com/Articles/882739/Simple-approach-to-Voronoi-diagrams