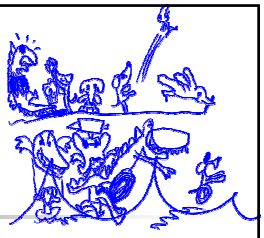
Computational Geometry for the Tablet PC



CSE 481b
Lecture 17

Announcements

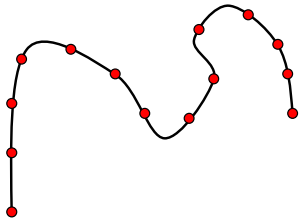
- Thursday, March 2
  - ABET Program review
    - Give important feedback on the program
    - Free Donuts!

Overview

- Computational Geometry on the Tablet PC
- Geometric primitives
- Intersections
- Polygons
- Convexity
- Voronoi Diagram

Tablet Geometry

- Basic structure – Stroke: sequence of points
  - Himetric coordinates
  - Sampled 150 times per second
  - Coordinates stored in an array Points

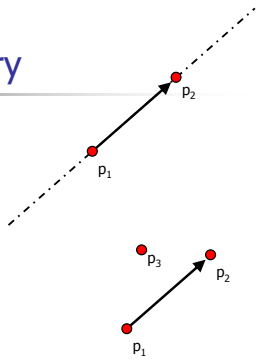


Computational Geometry

- Algorithms for geometric computation
- Numerical issues with coordinates
- Importance of degenerate cases
- Examples of degenerate cases
  - Three lines intersecting at a point
  - Segments overlapping
  - Three points co-linear

Basic geometry

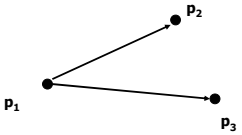
- Point p
  - p
- Line segment
  - (p_1, p_2)
- Distance
  - Dist(p_1, p_2)
- Basic Test
  - LeftOf(p_1, p_2, p_3)
  - CCW(p_1, p_2, p_3)



Counter Clockwise Test

- CCW(p_1, p_2, p_3)

```
public static bool CcwTest(Point p1, Point p2, Point p3){
      int q1 = (p1.Y - p2.Y)*(p3.X - p1.X);
      int q2 = (p2.X - p1.X)*(p3.Y - p1.Y);
      return q1 + q2 < 0;
}
```



Segment intersection

- Find intersection of (p_1,p_2) and (p_3,p_4)
  - $Q = \alpha p_1 + (1-\alpha)p_2$
  - $Q = \beta p_3 + (1-\beta)p_4$
- Solve for α, β
  - Two equations, two unknowns
  - Intersect if $0 < \alpha < 1$ and $0 < \beta < 1$
- Derived points
  - In general, try to avoid computing derived points in geometric algorithms

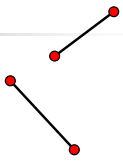
Problem

- Determine if two line segments (p_1, p_2) and (p_3,p_4) intersect just using CCW Tests

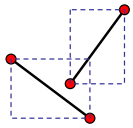
Student Submission

Making intersection test more efficient

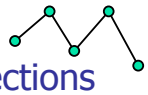
- Take care of easy cases using coordinate comparisons



- Only use CCW tests if bounding boxes intersect



Computing intersections

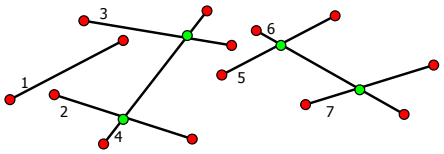


- How many self intersections can a single stroke with n points have?

Student Submission

Segment intersection algorithm

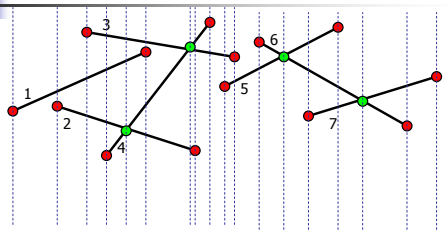
- Run time O(nlog n + Klog n) for finding K intersections
- Sweepline Algorithm



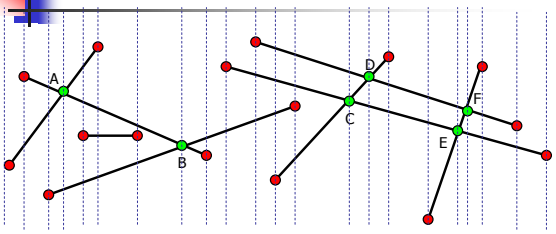
Sweepline Algorithm

- Event queue
  - Start Segment (S_2)
  - End Segment (E_2)
  - Intersection ($I_{2,4}$)
- Move sweepline to next event
- Maintain vertical order of segments as line sweeps across
- Start Segment
  - Insert in list
  - Check above and below for intersection
- End Segment
  - Remove from list
  - Check newly adjacent segments for intersection
- Intersection
  - Reorder segments
  - Check above and below for intersection

Sweepline example



Activity: Identify when each of the intersections is detected

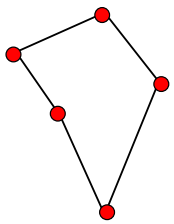


Student Submission

Polygons

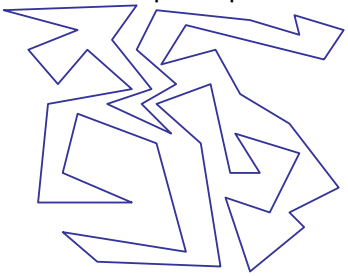
- Sequence of points representing a closed path
- Simple polygon – closed path with no self intersections

Describe an algorithm to test if a point q is in a polygon P



Polygon inclusion test

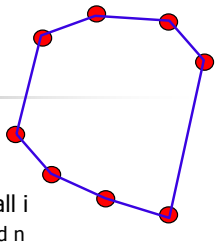
- Is the point q inside the Polygon P?



Convexity

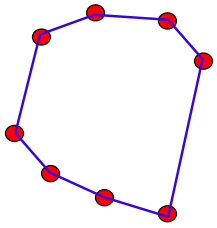
- Defn: Set S is convex if whenever p_1, and p_2 are in S, the segment (p_1, p_2) is contained in S

Convex polygons



- P = $\{p_0, p_1, \ldots p_{n-1}\}$
- P is convex if
  - CCW(p_i, p_{i+1}, p_{i+2}) for all i
    - Interpret subscripts mod n
    - Also holds for CW (depending on how points are ordered)

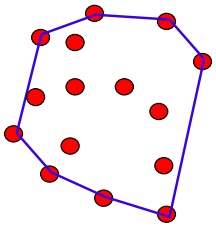
Problem: Test if a point q is inside a convex polygon P using CCW Tests



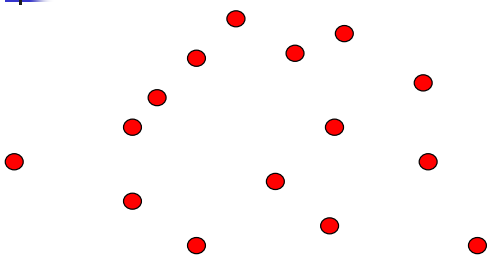
Student Submission

Convex hull

- Smallest enclosing convex figure
- Rubber band "algorithm"

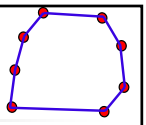


Compute the Convex Hull



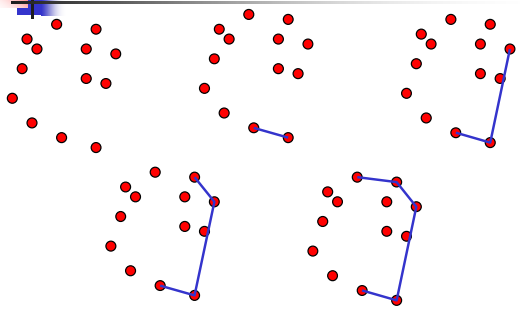
Student Submission

Algorithms

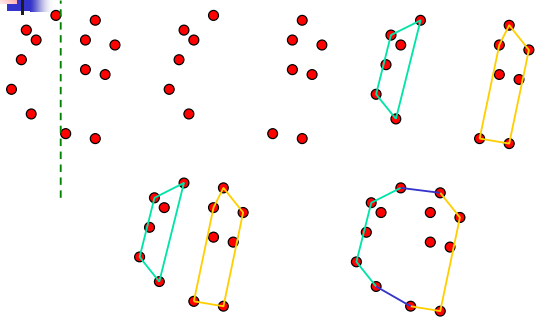


- Convex hull algorithms: O(nlog n)
  - Related to sorting
- Insertion algorithm
- Gift Wrapping (Jarvis's march)
- Divide and Conquer
- Graham Scan

Convex Hull Algorithms
Gift wrapping



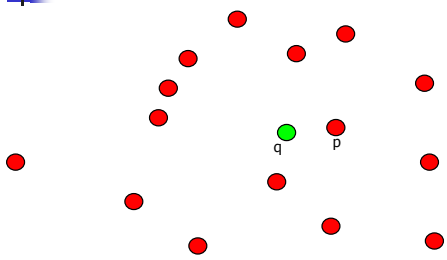
Divide and Conquer



Graham Scan

- Polar sort the points around a point inside the hull
- Scan points in CCW order
  - Discard any point that causes a CW turn
    - If CCW advance
    - If !CCW, discard current point and back up

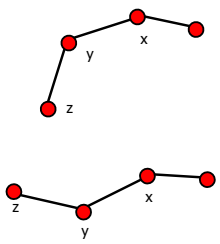
Polar sort the red points around q (Start with p, number the points in CCW order)



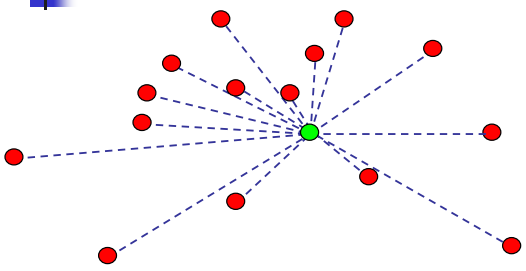
Student Submission

Graham Scan Algorithm

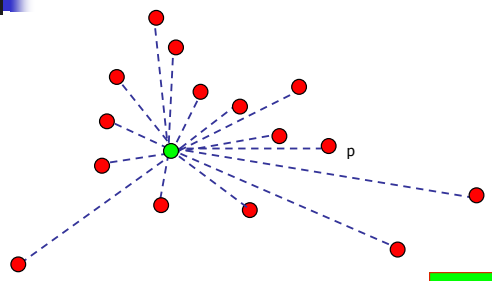
- Stack of vertices
  - Possible hull vertices
  - z – next vertex
  - y – top of stack
  - x – next on stack
- If CCW(x, y, z)
  - Push(z)
- If (! CCW(x, y, z))
  - Pop stack



GS Example to walk through



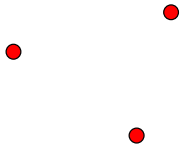
Student submission: Give order vertices are discarded in the scan



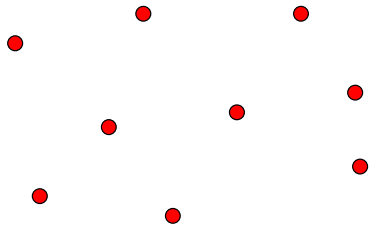
Student Submission

Voronoi Diagram

- Given a set of points: subdivide space into the regions closest to each point



Compute the Voronoi Diagram



Student Submission

Algorithms for Computing the Voronoi