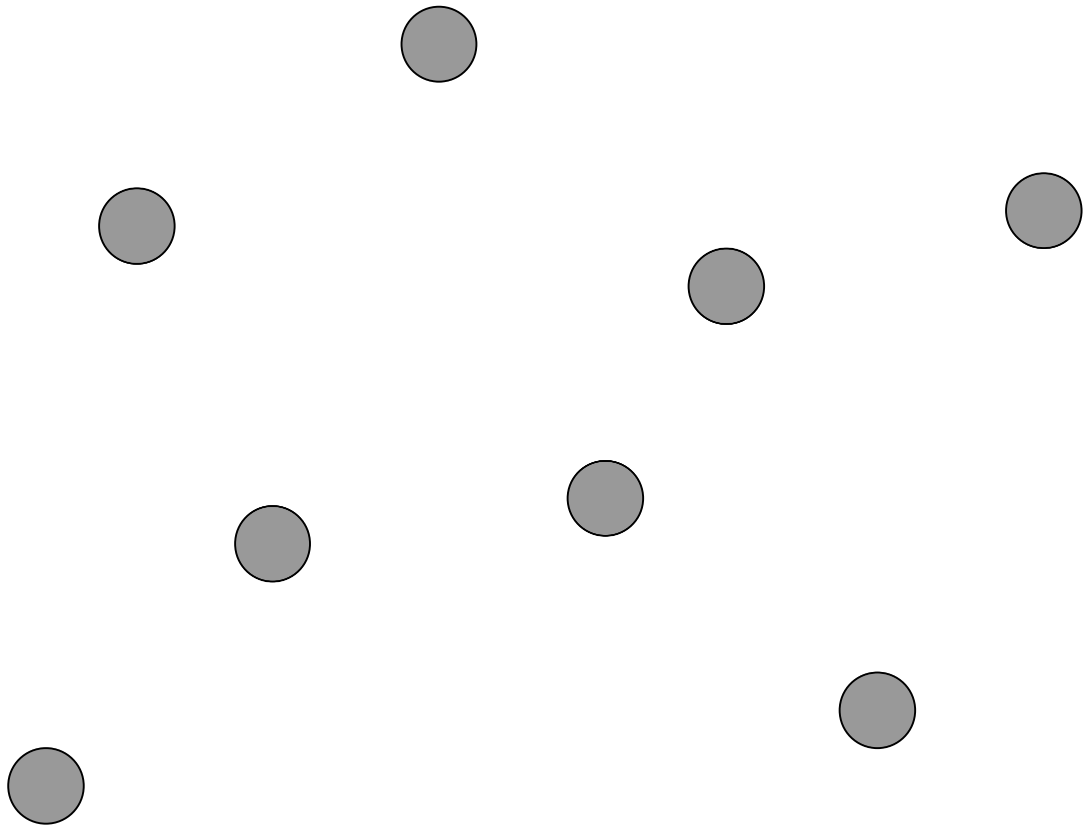
Computational Geometry

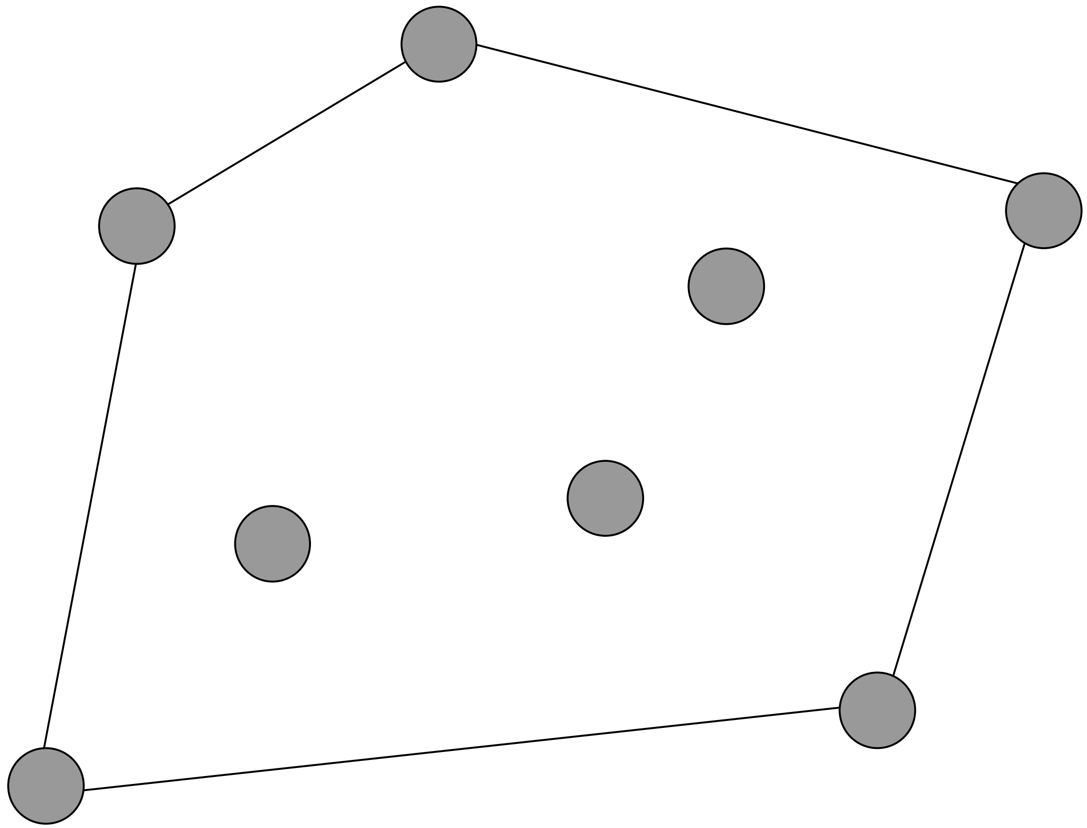
Convex Hulls
Robust Geometric Primitives
Degeneracy and Stability

Nick Pilkington

Convex Hull



Convex Hull



Identifying

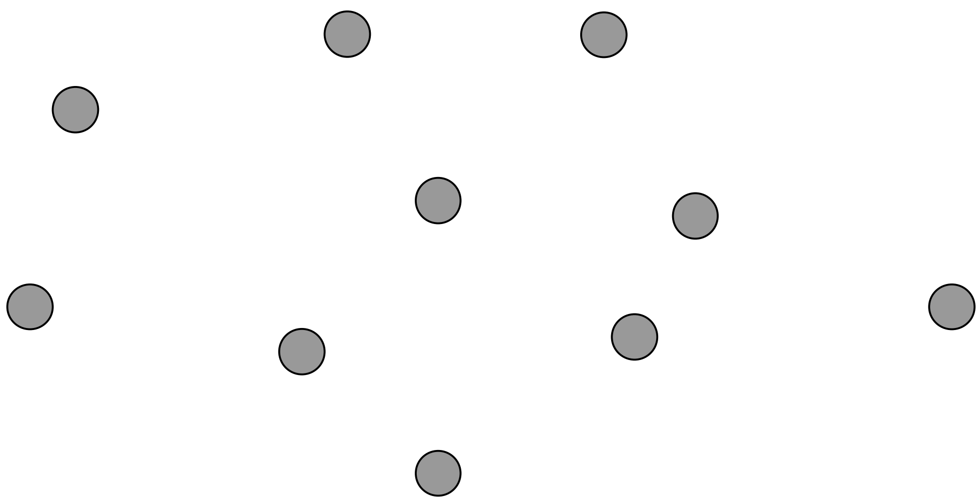
- Two points furthest distance from each other.
- Minimum set of points enclosing…
- Ideas of fencing, encompassing or enclosing
- Diameter, furthest pair, longest distance between…

Algorithms

- Jarvis March
  - O(n.h)
- Graham Scan
  - O(n.log n)

Jarvis March

- Pick a point that you know to be on the convex hull.
- To determine the next point on the hull, loop through all points and find the one that forms the minimum sized anticlockwise angle off the horizontal axis from the previous point.
- Continue until you encounter the first point



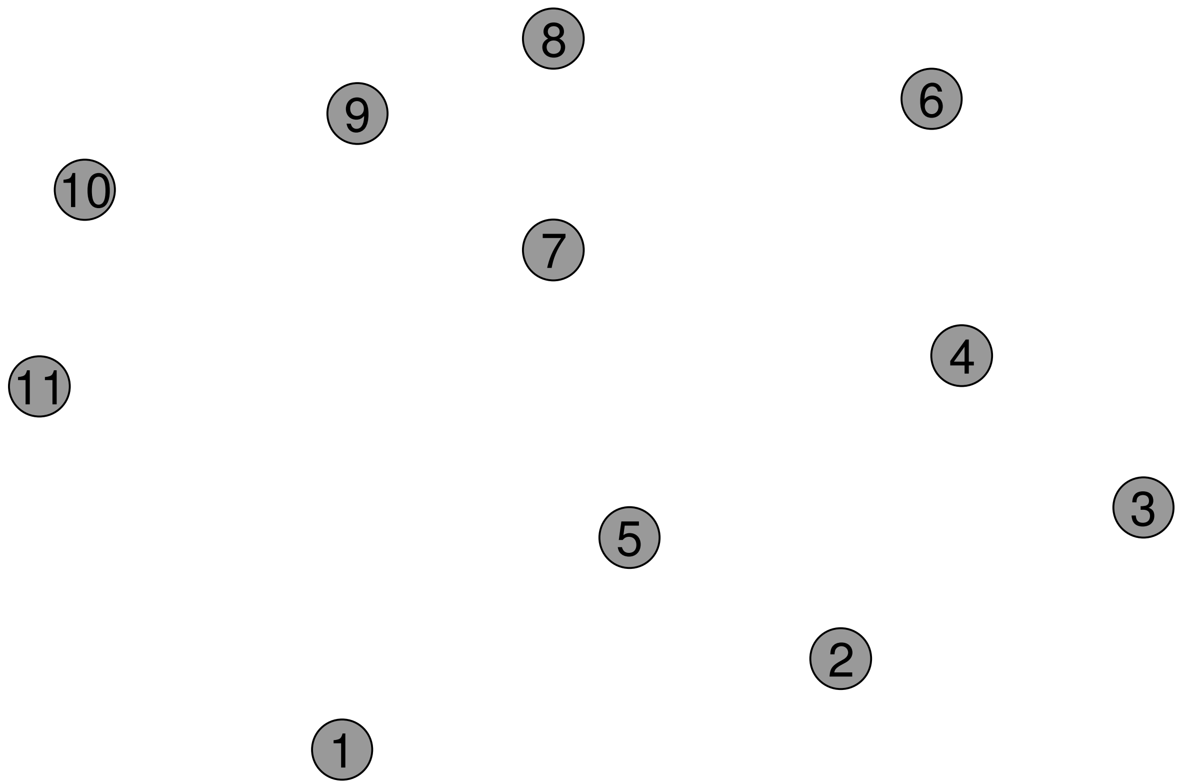
Pros and Cons

- Pros
  - Easy to program
- Cons
  - Slow! Possibly $O(n^2)$

Graham Scan

- Find a point you know to be on the convex hull. (Lower y coordinate)
- Sort all other points angularly around this point (anti clockwise), by calculating the angle that each point makes with the x axis (within the range 0 to 360 degrees)
- Add the first two points to the hull.
- For every other point except the last point
- Make it the next point in the convex hull
- Check to see if the angle it forms with the previous two points is greater than 180 degrees
  - As long as the angle formed with the last two points is greater than 180 degrees, remove the previous point
- To add the last point
  - Perform the deletion above,
  - Check to see if the angle the last point forms with the previous point and the first point is greater than 180 degrees or if the angle formed with the last point and the first two points is greater than 180 degrees.
  - If the first case is true, remove the last point, and continue checking with the next-to-last point.
  - If the second case is true, remove the first point and continue checking.
  - Stop when neither case is true.

Graham Scan



Pros and Cons

- Pros
  - Fast
  - Relatively Easy to Program

- Cons
  - Bit fiddly to program

Robust Geometric Primitives

- Area of a Triangle
- Line / Line Segment Intersection
- Point in Polygon
- Area of a Polygon
- Pitfalls, Arithmetic Accuracies, Exceptions.

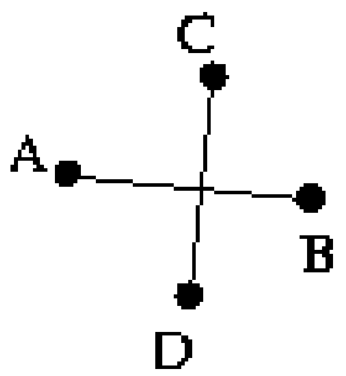
Area of a Triangle

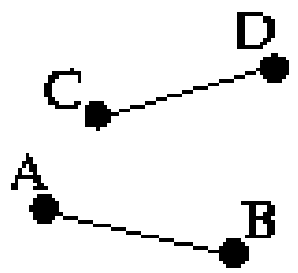
- Heron's Formula
- Area = Sqrt(s(s-a)(s-b)(s-c)), where s = (a+b+c)/2
- Determinant / Cross Product

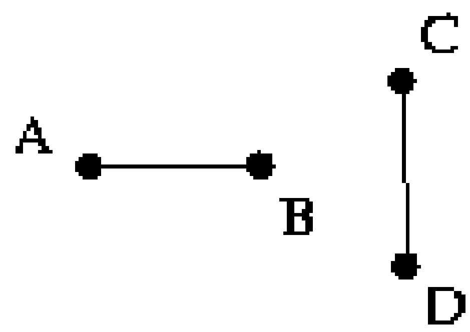
Line/Line Segment

Line / Line Segment
Above and Below Test
Use the signed area of the triangle

Line Segment / Line Segment
Same a s above, however TWO checks are necessary!

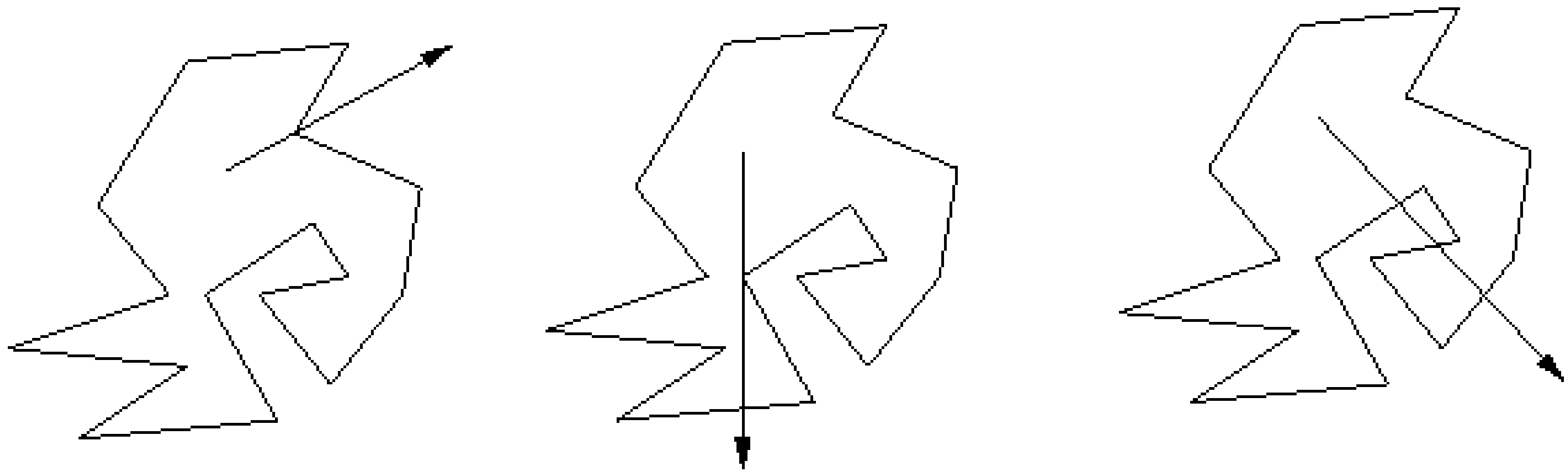






Point in Polygon

- Extend the point in a random direction forming a ray.
- Find the number of times the ray intersects an edge of the polygon.
- Even number of intersections = outside
- Odd number of intersection = inside



Algorithm generalizes to 3 dimensions!

Area of Polygon

Area of polygon

The area of a polygon with vertices (*x* 1, *y* 1), ..., (*x* n, *y* n) is equal to the determinant:

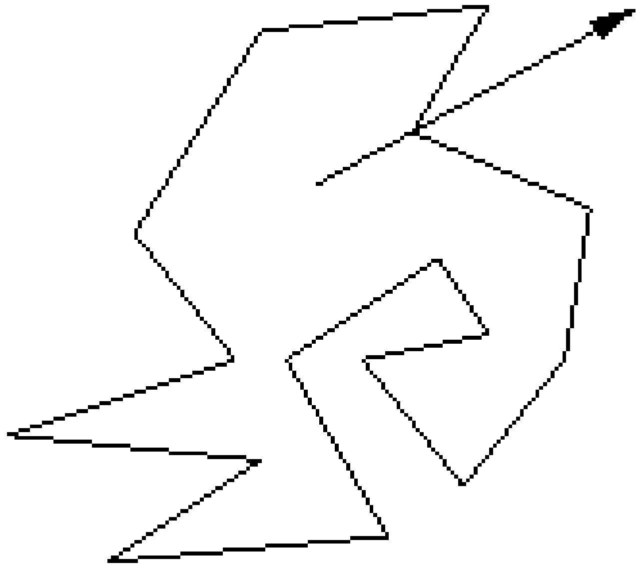
$$\frac{1}{2} \begin{vmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{vmatrix}$$

This formula generalizes to compute d! times the volume of a simplex in d dimensions.

Areas and volumes are signed!

Area of a Polygon

- Triangulation
- Monte Carlo Methods



Pitfalls

- Degeneracy
  - Ignore It !
  - Deal with It!

Numerical Stability

  - Precision!
  - Rounding Errors!

Questions

- IOI 2003 Day 2 :: Boundary

  Given a rectangular field with fence posts 1 meter apart along its perimeter. An number of rocks represented by polygons in the field. A viewers position. Calculate the number of fence posts that a person at the viewers position can see, unobstructed by rocks in the field.

O(N.R)	72~80%
O(R log R + N Log d)	100%

***Looking at the top 20 places from IOI 2003 only 6 of them scored more than 72% in this problem. ***