Convex Hull

CS16: Introduction to Data Structures & Algorithms
Seny Kamara - Spring 2017
Outline

- Overview
- Convex Hull
- Graham Scan Algorithm
- Incremental Algorithm
Convex Hull

- The convex hull of a set of points is...
  - ...the smallest convex polygon containing the points
- A convex polygon is...
  - ...a nonintersecting polygon whose internal angles are convex (i.e., less than 180 degrees)
  - In a convex polygon, segment joining any two points lies entirely inside polygon
Convex Hull

convex

non-convex

Segment not contained!
Convex Hull

- Kind of like a rubber band snapping around the points
- Special cases!
  - collinearity: a point on a segment is not part of the convex hull
Applications

- Motion planning
  - Find optimal route that avoids obstacles for a robot
- Bounding box
  - Obtain closer bounding box in computer graphics
- Pattern matching
  - Compare two objects using their convex hulls
Finding a Convex Hull

- Algorithm for determining a convex polygon
  - move counterclockwise & always produce left-turns
Calculating Orientation

- Orientation of 3 points $a, b, c$ is either:
  - clockwise (CW): right turn
  - counter clockwise (CCW): left turn
  - co-linear (COLL): no turn

- Orientation of $a, b, c$ determined...
  - ...by sign of the determinant

$$\Delta(a, b, c) = \begin{vmatrix} x_a & y_a & 1 \\ x_b & y_b & 1 \\ x_c & y_c & 1 \end{vmatrix}$$
function isLeftTurn(a, b, c):
    return (b.x - a.x)×(c.y - a.y)−(b.y - a.y)×(c.x - a.x) > 0
Calculating Orientation

- Using the `isLeftTurn()` method:
  - \((0.5-0) \times (0.5-0) - (1-0) \times (1-0) = -0.75 \text{ (CW)}\)
  - \((1-0) \times (1-0) - (0.5-0) \times (0.5-0) = 0.75 \text{ (CCW)}\)
  - \((1-0) \times (2-0) - (1-0) \times (2-0) = 0 \text{ (COLL)}\)
Calculating Orientation

- Let \(a, b, c\) be 3 consecutive vertices of polygon in CCW
  - \(b\) not included in hull if \(a, b, c\) non-CCW
    - \(\text{orientation}(a, b, c) = \text{CW or COLL}\)
  - \(b\) included in hull if \(a, b, c\) convex
    - \(\text{orientation}(a, b, c) = \text{CCW...}\)
  - ...and all non-hull points have been removed
Calculating Orientation

Activity #1

1 min
Calculating Orientation

Activity #1
Calculating Orientation

Activity #1
Graham Scan Algorithm

- Find **anchor** point
  - anchor point: point with smallest $y$ value
- Sort points in CCW order around anchor
  - can sort points by comparing angle between anchor & the point you’re looking at
  - the smaller the angle, the closer the point

**Note:** this algorithm is referred to as the static Graham Scan algorithm in Convexhull handout
Graham Scan Algorithm

- Polygon is traversed in sorted order & seq. \( H \) of vertices is maintained
- Add each point \( a \) to \( H \)
  - While last turn is a right turn, remove second to last point from \( H \)
- In example,
  - \( p, q, r \) forms right turn so \( q \) removed
  - \( o, p, r \) forms right turn so \( p \) removed
Graham Scan: Pseudo-code

```python
function graham_scan(pts):
    // Input: Set of points pts
    // Output: Hull of points
    find anchor point
    sort other points in CCW order around anchor
    hull = []
    for p in pts:
        add p to hull
        while last turn is a “right turn”
            remove 2nd to last point
        add anchor to hull
    return hull
```

Note: this is very high-level pseudocode; there are many special cases to consider!

Overall run time: $O(n \log n)$
Testing

Activity #2

1 min
Activity #2

1 min

Testing
Testing

0 min

Activity #2
Incremental Algorithm

‣ What if we already have hull & want to add point \( q \)?
  ‣ This is next project!

‣ Get angle from anchor to \( q \)
  ‣ find \( p \) and \( r \) in the hull on either side of \( q \)
    ‣ Note that calculating anchor when using incremental algorithm is different than for the static Graham Scan. More on this in the convexhull handout

‣ If \( p, q, r \) form a left turn then add \( q \) to hull

‣ Check if adding \( q \) creates a convex shape
  ‣ if there are right turns on either side of \( q \)…
  ‣ …remove vertices until shape is convex
  ‣ In same way as static Graham scan
Incremental Algorithm

Original Hull

Want to add point $q$

Find $p$ and $r$

$p, q, r$ form a left turn, so add $q$

$o, p, q$ form a right turn, so remove $p$

$n, o, q$ form a right turn, so remove $o$
Incremental Algorithm

Since $m, n, q$ is left turn, we’re done with that side

Now we look at the other side

Since $q, r, s$ is a left turn, we’re done!

› Remember
  › you can have right turns on either or both sides,
  › so make sure to check in both directions and remove concave points!
Incremental Analysis

- Let $n$ be the size of the convex hull
- Suppose it is stored in a binary search tree
  - sorted around the anchor
- To check if point $q$ should be in hull
  - insert into tree & get its neighbors $(p, r)$
  - $O(\log n)$
- The traverse the ring
  - possibly deleting $d$ points from hull
  - $O((1+d)\log(n))$
- So incremental insertion is $O(d\log(n))$
  - where $d$ is number of points removed by insertion