Portland State University

CS350

Algorithms And Complexity

ConvexHull Analysis: BruteForce vs. QuickHull

Author:
Wes Risenmay
Josh Willhite

Instructor:
Dr. Andrew Black

March 11, 2014

Contents

1 Objective

2 ConvexHull Overview

3 Algorithm Explaination

3.1 Brute Force

3.2 QuickHull

4 Algorithm Implementation

4.1 Brute Force

As the name implies the brute force solution to the convexhule problem is very straight forward, as shown below in Figure 1.

```java
public static LinkedList<Point2D> bruteForceConvexHull(Point2D points[]) {
    LinkedList<Point2D> convexHull = new LinkedList<Point2D>();

    double curr_x_prod, prev_x_prod;
    boolean on_hull;
    for(Point2D a: points) {
        for(Point2D b: points) {
            if(a != b) {
            prev_x_prod = Math.PI;
            on_hull = true;
            for(Point2D c: points) {
                if(c != a && c != b) {
                    curr_x_prod = ((a.getX() - b.getX())*(b.getY() - c.getY()) - (b.getX() - c.getX())*(a.getY() - b.getY()));
                    if(curr_x_prod > 0) {
                        curr_x_prod = 1;
                    }
                    else if(curr_x_prod < 0) {
                        curr_x_prod = -1;
                    }
                    else {
                        curr_x_prod = 0;
                    }
                    if (curr_x_prod == prev_x_prod || prev_x_prod == Math.PI || curr_x_prod == 0) {
                        prev_x_prod = curr_x_prod;
                    }
                    else {
                        on_hull = false;
                        break;
                    }
                }
            }
            if(on_hull) {
                if(!convexHull.contains(a)) {
                    convexHull.add(a);
                }
                if(!convexHull.contains(b)) {
                    convexHull.add(b);
                }
            }
            }
        }
    }

    return convexHull;
}
```

Figure 1: Brute Force Convex Hull