*

September 13, 2010

Abstract

This lecture gives an introduction to Convex Hull and to the algorithms that are used to construct the convex hull.

1 Convex Hull (CH)

1.1 Convex Set

Convex Set is a set $X \subseteq R^2$ which satisfies the following property:

For any two points $p, q \epsilon X$, the convex linear combination,

λ p + $(1 - \lambda)$ q ϵ X , $0 \leq \lambda \leq 1$

ie., the entire segment $\bar{pq} \subset X$

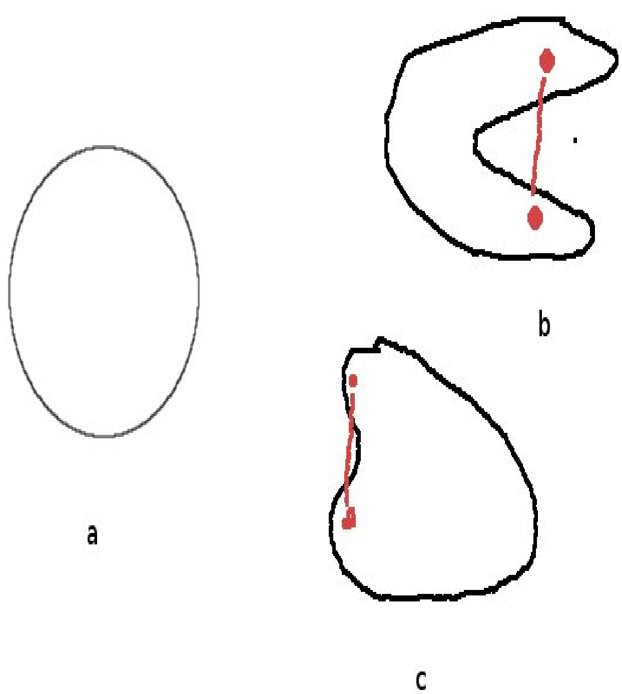

Figure1 : (a) Convex set, (b) and (c) non convex set

*Department of Computer Science and Engineering, IIT Delhi, New Delhi 1100116, India.

1.2 Convex Hull: Definition

Suppose S is a finite set of points. The convex hull of S is the smallest convex set that contains S.

The smallest convex set happens to be a convex polygon. So CH(S) is the smallest convex polygon containing S in terms of (i)area and (ii)perimeter

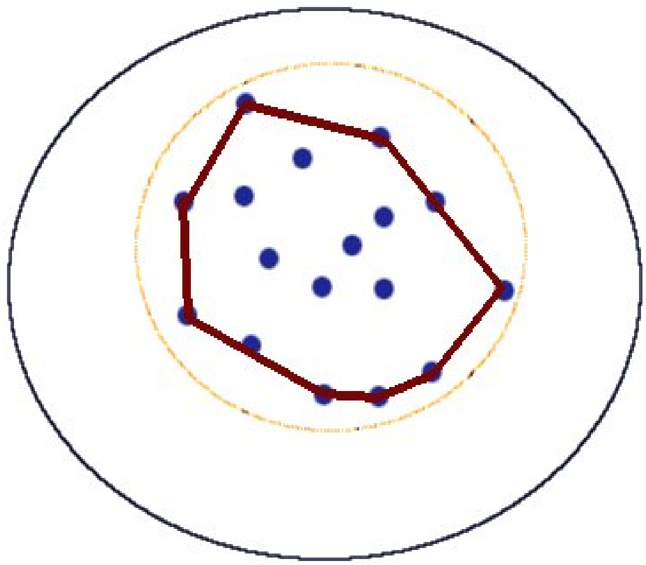

Figure 2

The points on the boundary of the convex hull are corner points / boundary points. The boundary of a planar convex hull is an ordered chain of vertices or edges.

A boundary point supports a tangent(a line passing throught the point, keeping the remaining points on one side)

The intersection of two convex hulls is a convex hull.

2 Algorithms

2.1 Rope

The idea is to tie a rope at all points possible. The algorithm starts from the left most point. The point which makes the maximum of all angles with the current point, is the point to which the rope is to be tied next. But to find the angles, inverse trignometric functions are required.

To avoid the use trignometric functions, an alternate technique is used. An observation is made here on how the rope is getting tied. It is noted that the rope always makes a right turn to get tied to the next point(clockwise direction assumed). Now the rope can be tied to the points in increasing order of say X dirextion. Whereever the rope makes a left turn, that point is not a boundary point.

To find the turn, the triangle property is used. Consider the signed area of the triangle defined by $(x_1, y_1), (x_2, y_2)$ and (x_3, y_3)

$$area = (1/2)\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

If the points are clockwise(right turn), the area is negative.
If the points are anticlockwise(left turn), the area is positive.

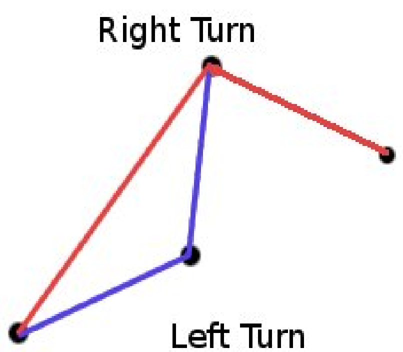

Figure 3

2.2 Graham Scan

Boundary point can also be defined as a point p ϵ S than cannot be expressed as a clc of two points in the convex hull of S. No point that is clc of two other points in CH(S) is a corner point.

A point is a clc of points $q_1, q_2, ...q_k$, if
p $= \lambda_1\ q_1 + \lambda_2\ q_2 + ... + \lambda_k\ q_k$, $0 < \lambda_i < 1$ and $\Sigma\ \lambda_i = 1$
Two sets of points can be separated iff the convex hulls of the two sets are disjoint.

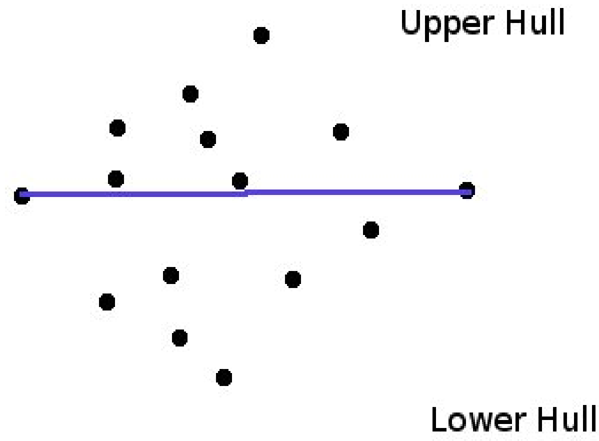

Figure 4

2.2.1 Algorithm

The set of points is divided along the line joining the leftmost and rightmost points, into upper and lower hulls. The upper and lower convex hulls are disjoint and can be merged later.

For each hull,

(i) Sort the points in one direction, say X direction

(ii) Connect all the points to their corresponding neighbouring points.This forms a monotone chain.

(iii) At every point, if a left turn occurs remove the point and if a right turn occurs continue.

The final set of points that are not removed from the list are the corner points of the convex hull.

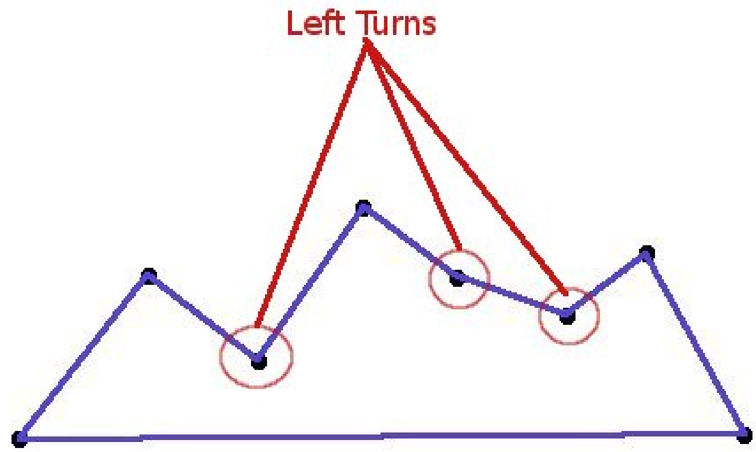

Figure 5: Monotone Chain

2.2.2 Analysis

Running time for sort is O(nlogn).

A point can be added or deleted once using a stack. Running time is (number of left turns + number of right turns), which is O(n).

The effective running time of the algorithm is O(nlogn).

3 Lower Bound Running time

3.1 Proof

Given, S = $\{x_1, x_2, ..., x_n\}$

S' = $\{(x_1, x_1^2), (x_2, x_2^2), ...(x_n, x_n^2)\}$

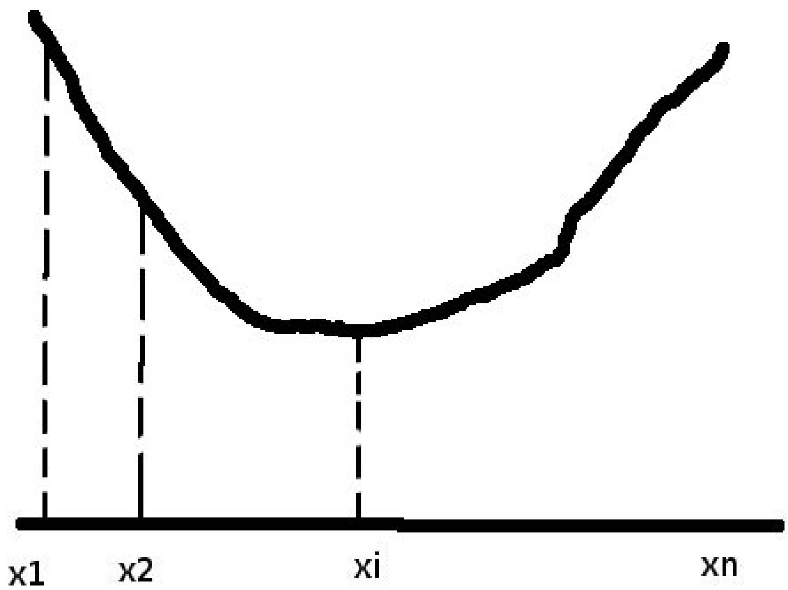

Figure 6

CH(S') gives an ordered set of boundary points which implies that we can sort S using this ordering.

Sorting(S) $\alpha_{O(n)}$ CH(S')

Therefore, CH must have a lower bound for sorting which is O(nlogn)

3.2 Observation

It is assumed that the CH algorithm must output ordered set of convex vertices, instead of outputting the corner points in any order.

4 Motivation for Quick Hull

Consider upper hull. Identify the lefmost and rightmost points and the point with the highest perpendicular distance to the line joining leftmost and rightmost points. All the points within the triangle formed by these three points are not corner points and can be ignored for the convex hull construction.

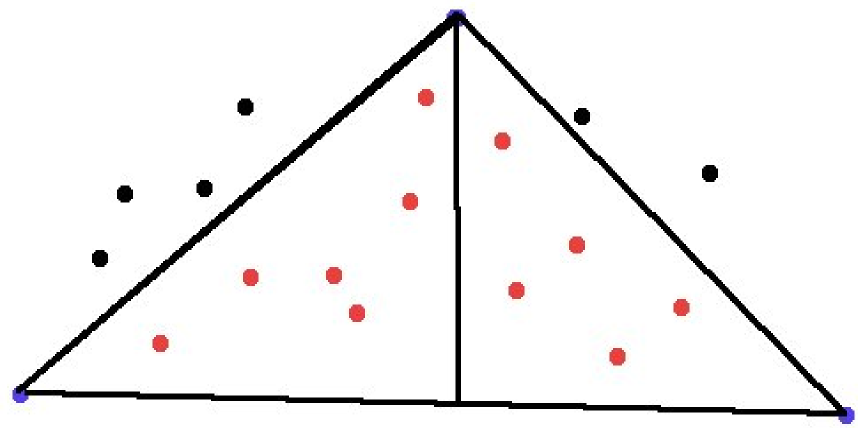

Figure 7