# CONVEX HULL


## Submitted By:

Akshat Gupta (180123002)

Anish Kumar (180123003)


### *AIM:*

To construct the convex hull of a set of n points in 2 dimensions by improving upon the already existing techniques.

### *BACKGROUND:*

The **convex hull** of a set Q of *n* points, denoted by CH(Q), is the smallest convex polygon P for which each point in Q is either on the boundary of P or in its interior. Among various methods proposed, 3 of them are very popular namely “Graham’s Scan”, “Quick Hull” and “Jarvis’ March”.

Convex hulls have wide applications in many fields, like in mathematics convex hulls are used to study polynomials, matrix eigenvalues, and several theorems in discrete geometry involve convex hulls. Few real life examples are collision avoidance of cars on roads, shape analysis, image processing, medical simulations, etc.

Graham’s Scan was proposed by Ronald Graham in 1972[1] with running time O(n log n). Jarvis’ March, a.k.a. Gift Wrapping was created independently by D.R. Chand and S.S. Kapur in 1970[2] and R.A. Jarvis in 1973[3]. The algorithm runs in time O(nh), where h is the number of vertices of CH(Q). Quick Hull was proposed by Jonathan Greenfield in 1990[4] which uses divide and conquer and has worst time complexity of $O(n^2)$ when all points are hull vertices, but an expected time complexity of O (n log n).

In Graham’s Scan method we use stack data structure to work efficiently:

1. Sort the set of points by increasing order of x-coordinate.
2. Check orientation at each point with two most recent selected points, if the point forms positive orientation i.e. point forms left hand turn then push the point in the stack or else if right-hand turn then pop out the last point.
3. Join the points in order that are stored in stack. This will give our required convex hull.

Quick Hull method:

1. Find points with minimum and maximum x coordinates, as these will always be a part of the convex hull. If many points with the same minimum/maximum x-coordinate exist, use ones with minimum/maximum y-coordinate correspondingly.
2. Use the line formed by the two points to divide the set in two subsets of points, which will be processed recursively.
3. Determine the point, on one side of the line, with the maximum distance from the line. This point forms a triangle with the line.
4. The points lying inside of that triangle cannot be a part of the convex hull and can therefore be ignored in the next steps.
5. Repeat the previous two steps on the lines formed by the triangle (not the initial line).
6. Keep on doing so until no more points are left, the recursion has come to an end and the points selected constitute the convex hull.

Jarvis' March:

1. Choose the point with the minimum y-coordinate (say $p_0$).
2. Select the next vertex $p_1$ such that it has the smallest polar angle w.r.t. $p_0$. (In case of ties choose the point farthest from $p_0$).
3. Go on like this until the highest vertex (say $p_k$) is reached. The right chain of CH(Q) is thus constructed.
4. To construct the left chain, start at $p_k$ and choose $p_{k+1}$ as the point with the smallest polar angle with respect to $p_k$, but from the negative x-axis. Continue on, forming the left chain by taking polar angles from the negative x-axis, until we come back to our original vertex $p_0$.

Graham's Scan method works slow for a set of large points as it has to check the orientation of all points and the computational cost of such a large set will be really unaffordable, whereas Graham's method works faster for a set of smaller number of points.

## ***WORKS FOCUSED ON:***

We focused on a couple of research works done in this field by T.M. Chan[5] and Muhammad Sharif[6] respectively.

### Chan's Method

This is an output sensitive algorithm, i.e. the complexity is measured as a function of both *n* and output size *h*.

Jarvis' March computes the *h* vertices of the convex hull one at a time by a sequence of *h* wrapping steps. A wrapping step can be done faster if the points are pre-processed.

**Algorithm:**

**CH(Q, m, H)** where $3 \le m \le n$, and $H \ge 1$

1. partition Q into subsets $Q_1, \ldots, Q_{[n/m]+1}$ each of size at most m
2. for *i* = 1, . . . . .,[n/m]+1 do
3. compute conv($P_i$) by Graham's scan and store its vertices in an array
4. $p_0 \leftarrow (0,-\infty)$
5. $P_1 \leftarrow$ the rightmost point of Q
6. for *k* = 1, . . . . ., H do
7. for i = 1, . . . . . ,[n/m]+1 do
8. compute the point $q_i \in Q_i$ that maximizes $\angle p_{k-1} p_k q_i$ ( $q_i \neq p_k$ ) by performing a binary search on the vertices of conv(Pi)
9. $p_{k+1} \leftarrow$ the point q from $\{q_1, \ldots, q_{[n/m]+1}\}$ that maximizes $\angle p_{k-1} p_k q$
10. if $p_{k+1} = p_1$ then return the list $(p_1, \ldots, p_k)$
11. return incomplete

**Explanation of the algorithm and Time Analysis:**

1. Choose a parameter *m* between 1 and *n* and partition Q into ([*n*/*m*]+1) groups each of size at most *m*.
2. Compute convex hull of each group in O(m log m) time by Graham's Scan.
3. This gives [n/m]+1 possibly overlapping convex polygons each with at most *m* vertices, after a pre-processing time of O((n/m) (m log m)) = O(n log m).
4. Now, a wrapping step can be done by scanning all [n/m]+1 polygons and computing tangents or supporting lines of the polygons through the current vertex $P_0$. Finding tangent takes logarithmic time for a convex polygon by binary or Fibonacci search, so the time

required for a wrapping step is O((n/m) log m). As *h* wrapping steps are needed to compute the hull, total time of the algorithm becomes O(n logm + h ((n/m)logm)) = O(n (1+h/m) logm)

5. Since the value of h is not known in advance, we use the following algorithm:

   **CH(Q)**
   1. for $t$ = 1,2, . . . . do
   2. L ← CH(Q, m, H), where m = H = min{($2^2$)^t, n}
   3. if L ≠ incomplete then return L

   Total number of iterations is [log log h]+1 and each iteration takes O(n log H) = O(n $2^t$) time. So, total running time of Chan's algorithm is O( $\sum_{t=1}^{[\log\log h]+1} \mathrm{n}2^t$) = O(n log h).

### Sharif's Method:

In order to make it fast for larger number of points, the initial steps of Quick Hull algorithm are applied which discard many useless points and thus increasing the speed of the Graham's method.

Both the methods have some drawbacks which drive them slow and the proposed technique is composed of both the methods which overcomes the drawbacks of the two methods and makes the proposed technique much fast for a large set of points.

**Algorithm:**

```
1. Find out the points with Xmin, Xmax, Ymin and Ymax from set n.
2. Construct Quadrilateral using these points.
3. Discard the points inside it forming a new set nnew
4. Sort the points (respect to x-axis), denoted (P1, P2, .... Pn)
5. Push P1 and P2 onto U (stack)
6. While i≤n
{
  If (xi makes left turn relative to top 2 items on stack)
  {
    Push xi;
    i++;
  }
  Else
  {
    Pop (last two added points);
    Discard;
}}
```

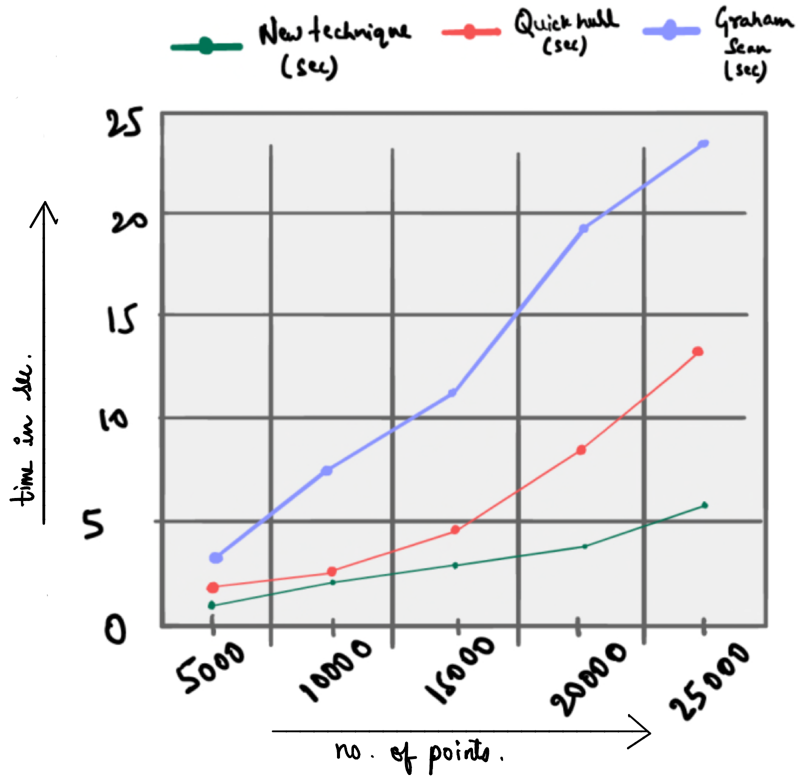
**Explanation of the algorithm:**

1. The first step starts with Quick hull and finding minimum and maximum X and Y coordinates from a set of n planar points.
2. Construct a quadrilateral using these points and a rectangle passing through these points.
3. Discard the points lying inside and on the quadrilateral and thus forming a new set of points.
4. In each triangle, sort the points (w.r.t x-coordinates) and name them ($P_1$, $P_2$, $P_3$...., $P_n$).
5. Push points $P_1$ and $P_2$ in stack.
6. Initiate a while loop for all $i \leq n$ (where total points inside that particular triangle is *n*).
7. Now we will check the orientation of the point joined relative to the last two points added to the stack. (Orientation can be calculated by taking cross product of the points).
8. If points take left turn which means they have positive orientation, they are pushed in the stack else the point from the top of the stack will be popped out.

Analysis of the new technique with the help of a graph:

From the graph below it can be clearly analysed that result of the proposed technique is much faster than the existing Graham's Scan and Quick Hull algorithms.

| | Points | Graham Scan (sec) | Quick Hull (sec) | New technique (sec) |
|---|---|---|---|---|
| 1 | 5000 | 3.73 | 1.26 | 0.86 |
| 2 | 10000 | 7.35 | 2.64 | 1.63 |
| 3 | 15000 | 11.96 | 4.68 | 2.65 |
| 4 | 2000 | 19.23 | 6.95 | 3.96 |
| 5 | 25000 | 23.6 | 7.99 | 5.86 |



## ***CONCLUSION:***

The above two techniques, which are modifications and improvements on some already existing techniques, perform faster than any of the individual methods. Both the techniques attempt to discard as many useless points as possible before applying the pre-existing methods.

Chan's method speeds up Jarvis' March and gift wrapping method by using grouping. While Jarvis' March constructs the convex hull in O(nh) time, Chan's method improves it to O(n log h).

Sharif's method uses Quick Hull's initial steps to remove unnecessary points and then apply Graham's Scan to a much smaller set of points, thereby speeding up the process.

However we noticed that the researcher did not provide any remedy to the problem when we have a point whose coordinates are (X-min, Y-min) (or similar type of coordinates). In these types of cases only 3 or 2 points will be available in the 1$^{st}$ step of the proposed algorithm and there will not be any quadrilateral formation. Instead of a quadrilateral, a triangle or a straight line will be formed.

## ***REFERENCES:***


[1] R.L. Graham (1972): "An efficient algorithm for determining the convex hull of a finite planar set" PDF available at- https://www.math.ucsd.edu/~ronspubs/72_10_convex_hull.pdf

[2] D. R. Chand and S. S. Kapur. An algorithm for convex polytopes. J. Assoc. Comput. Mach., 17:78-86, 1970.

[3] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. Inform. Process. Lett., 2:18-21, 1973.

[4] Greenfield, Jonathan Scott (1990): "A proof for a Quick Hull Algorithm" available at- https://surface.syr.edu/eecs_techreports/65/

[5] T.M. Chan (1996): **"**Optimal output-sensitive convex hull algorithms in two and three dimensions" PDF available at - https://link.springer.com/article/10.1007/BF02712873

[6] Muhammad Sharif (2011): "A new approach to compute convex hull" PDF available at- https://www.researchgate.net/publication/236898954_A_new_approach_to_compute_convex_hull

[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein: "Introduction to Algorithms"