Planar Convex Hulls

Barton, Alan

For the publisher's version, please access the DOI link below./ Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

https://doi.org/10.4224/8914427

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

NRC · CNRC

Planar Convex Hulls *

Barton, A.
November 2005

* published at NRC/ERB-1128. 4 Pages. November 2005. NRC 48290.

Canada

National Research Council Canada
Conseil national de recherches Canada

Institute for Information Technology
Institut de technologie de l'information

ERB-1128

NRC·CNRC

Planar Convex Hulls

Barton, A.
November 2005

Canada

NRC 48290

Planar Convex Hulls*

Alan J. Barton†
Integrated Reasoning Group
Institute for Information Technology
National Research Council Canada
Ottawa, Canada, K1A 0R6
alan.barton@nrc-cnrc.gc.ca

ABSTRACT
An attempt is made to understand some of the planar convex hull algorithms leading up to and including Chan's 1995 planar convex hull algorithm. The algorithms include: *i)* Graham's 1972 algorithm , *ii)* Jarvis' 1973 algorithm, and *iii)* Chan's 1995 algorithm.

Keywords
planar convex hull, design and analysis of algorithms, output-sensitive

1. INTRODUCTION
We should not rely on careful examinations, we should avoid the need for it. – *J.W. Tukey, 1977*[1]

Planar (2-dimensional) convex hulls may be used as a representation of the shape of a data set in application areas of Pattern Recognition. For example, one application of using higher dimensional convex hulls is reported in [8]. There are other possible conceptualizations of the meaning for shape. One possibility is a generalization of the convex hull called an α-shape [7]. Higher dimensional convex hulls, input spaces other than $\mathbb{R}^2$, and other shape representations will not be discussed further.

The objectives are:

1. to find an efficient algorithm for constructing convex hulls, where the measure of efficiency is asymptotic worst-case running time as a function of both n (input size) and m (output size) [2], and
2. to report hull vertices in a counter clockwise ordering (not required, but convenient).

The following assumptions are made:

1. Each point, p, lies in an $n = 2$-dimensional space.
2. $\exists$ metric ($d =$Euclidian distance) defined over the space
3. When $d = 2$, no 3 points lie along the same line. If they do, perturbation methods could be used [2], but may introduce other problems. For example, points on the hull after perturbation, may end up enlarging the hull, thereby changing the shape of the hull (See Fig.1.

The following definitions are here for clarity: X is a *metric space* [5] with metric d if X is a set and for $x, y \in X$, $d : X \to [0, \infty)$ the following axioms hold:

1. $d(x, y) = 0$ iff $x = y$
2. $d(x, y) = d(y, x)$ for each $x, y \in X$
3. $d(x, z) \le d(x, y) + d(y, z)$ for each $x, y, z \in X$

For example, d may be considered to be the Euclidean distance:

$$\left\| \overleftarrow{x} - \overleftarrow{y} \right\|_2 = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

which is an example from the family of metrics called the L_m norm [5], where $m \in (0..\infty]$

$$\begin{aligned} L_m(\overleftarrow{x}, \overleftarrow{y}) &= \left\| \overleftarrow{x} - \overleftarrow{y} \right\|_m \\ &= ((x_1 - y_1)^m + (x_2 - y_2)^m + \cdots \\ &\quad \cdots + (x_n - y_n)^m)^{1/m} \end{aligned}$$

In order to calculate the angle θ between 3 points, polar coordinates, which are defined based on Cartesian coordinates, could be used. The coordinate space transformations [9] are:

$$\begin{aligned} x &= r \cdot \cos\theta \\ y &= r \cdot \sin\theta \\ r &= \sqrt{x^2 + y^2} \\ \theta &= \tan^{-1}(y/x) \end{aligned}$$

*Report for the course COMP5730 entitled *Design and Analysis of Algorithms* at Carleton University

†Master of Computer Science (in progress)

[1] John W. Tukey was referring to the fact that plotted data (e.g. a scatter plot) should be clearly presented to a reader when attempting to argue for a particular perspective[6].

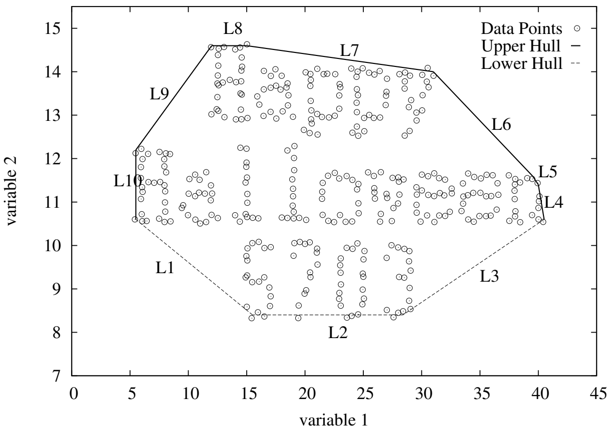

Figure 1: Example of a convex hull surrounding points in general position. This example shows the points after perturbation, and demonstrates the problem of points being pushed outside of the convex hull that would have enveloped the original data. For example, investigate line 8 (L8).

The **convex hull** of a set S of points ($\mathbf{CH}(S)$) is the smallest convex polygon C for which each point in S is either on the boundary of C or in its interior[2]. When considering general sets, the convex hull could potentially be considered as a specific instance of a definition from Topology. Namely, the **frontier** of a set S, denoted $\digamma(S)$, is the set of all frontier points of S[5]. The definition of convex hull implies that it (the $\mathbf{CH}(S)$) is not continuously smooth throughout the polygon, but only piece-wise smooth along its *faces*. In addition, the convex hull could be equivalently defined as the set of all convex combinations of S, i.e.

$$CH(S) = \left\{ \sum_{i=1}^{n} \alpha_i x_i : \sum_{i=1}^{n} \alpha_i = 1, \alpha_1, \alpha_2, \cdots, \alpha_n \geq 0 \right\}$$

with the vertices of $\mathbf{CH}(S)$ being called the *extreme points* of S yielding the fact that $\mathbf{CH}(S) \equiv \mathbf{CH}(\textit{extreme points})$ because points are in general position (i.e. no 3 are collinear), see Fig-1.

2. GRAHAM 1972

Graham's algorithm [3] is (asymptotically) optimal, since $\Omega(n \log n)$ lower bound can be obtained by a reduction from sorting [2]. The algorithm is:

Graham(point set $S \subseteq \mathbb{R}^2$)
1 Point $P \leftarrow$ InteriorPt(S)
2 $S^{(1)} \leftarrow$ To-Polar(S, P)
3 $S^{(2)} \leftarrow$ Sort-θ-Incr($S^{(1)}$)
4 $S^{(3)} \leftarrow$ DelClosePts($S^{(2)}$)
5 $CH(S) \leftarrow$ Scan-Del($S^{(3)}$)

For details of the algorithm, refer to [3]. The interesting points are: *i)* the $O(n)$ search for an explicit interior point. See Fig.2, which could be performed in other ways, such as selecting the minimum and maximum points in each dimension, and then constructing a centroid for those two extreme points. If they are the same, then all of the points are collinear, and the convex hull collapses to a line. *ii)* the polar transformations are straightforward and take $O(n)$ *iii)* the θ sort will take $\frac{n \log n}{\log 2}$ using the sorting routine referenced in the paper. This can easily be seen to be $O(n \log n)$ *iv)* the deletion of close points, see Fig.3 requires $O(n)$, and *v)* scanning and deleting points has two cases, as explicated in Fig.4. The total running time of Graham's algorithm is: $T(n) = O(n \log n)$.

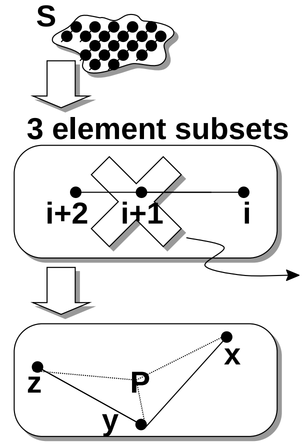

Figure 2: Step 1 of Graham's algorithm. Midpoints of collinear points are deleted in an attempt to construct a point P that is within the interior of 3 non-collinear points $\in S$.

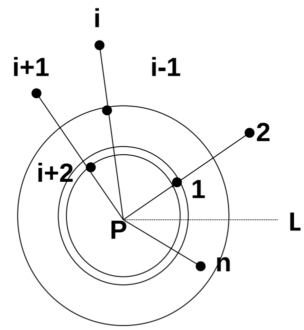

Figure 3: Step 4 of Graham's algorithm. Remove those points that lie on the same radial arm that are not maximally distant.

3. JARVIS 1973

The general idea of the algorithm, is to first select a point outside of the input set S, and then construct one hull boundary segment at a time, in a counter clockwise manner. Fig.5 demonstrates how the angle θ, with respect to a horizontal line, could be calculated from an initial point p_{out} outside the bounding box of S. The minimum angle would be selected, and that endpoint would become the first point of the convex hull. In which case, the second step performs similarly to the first, but with the origin of the comparisons emanating from the first convex hull point Fig.6. The procedure is repeated, from each of the points, with the

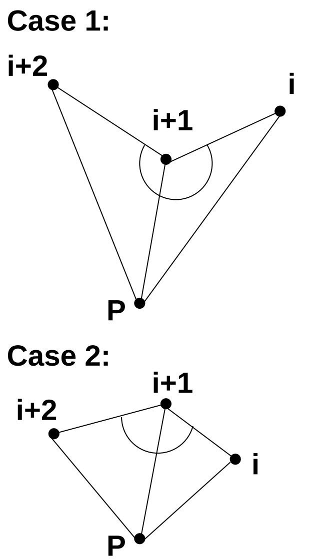

Figure 4: Two cases to consider in Graham's algorithm. Either *i)* $\theta > \pi$, in which case point i may be deleted, or *ii)* $\theta \leq \pi$, in which case point i becomes part of the hull and the next group of 3 points are considered by sliding forward by 1.

additional optimization that interior points of the hull under current construction no longer need to be considered Fig.7. The author [4] mentions an additional optimization in that the angle (θ) calculations may be substituted by sign tests (based on being located in one of the four quadrants). Jarvis' algorithm [4] has a running time of $T(n) = O(nm)$.

Jarvis(point set $S \subseteq \mathbb{R}^2$)

Point $p_{out} \leftarrow$ Outside(S)
Point $p_{curr} \leftarrow p_{out}$
for each point $p_i \in S$
  do
    if $p_{curr} \neq p_{out}$
      then
        DelCheck(S)
    $\theta_i \leftarrow$ Angle(p_{curr}, p_i)
    $p_{curr} \leftarrow$ MinAngle()
$CH(S) \leftarrow S$

4. CHAN 1995

Chan [1] uses the concept of running algorithm Chan() for a small value of H. If this fails, (poor randomly partitions sets) then the algorithm is stopped and restarted with a larger H value to a maximum of $H = n$, the number of points (since H can be when all of the points are on the convex hull). This algorithmic technique is called *doubling*. The Chan(P, m, H) basically attempts to find a convex hull of length H or less, if one exists, by artificially constructing some partitions of size at most m. These partitions are each run through a convex hull algorithm (say Graham's scan) and convex hulls are created containing those partition points Fig.9. The convex polygons are then used informally as a set of large points in a Graham

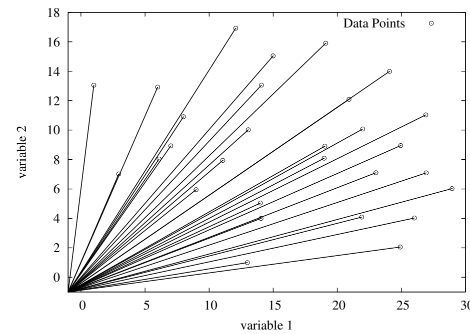

Figure 5: Jarvis' algorithm need to compare a point with all of the rest.

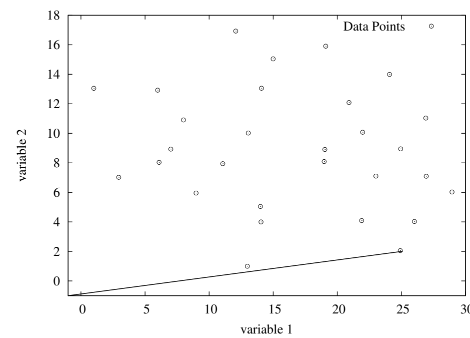

Figure 6: Jarvis' algorithm picks the minimum θ (angle).

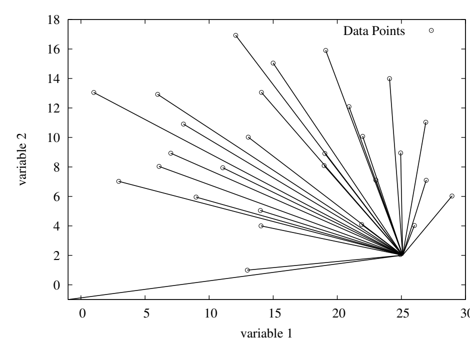

Figure 7: The second step in Jarvis' algorithm requires $n - 1$ evaluations.

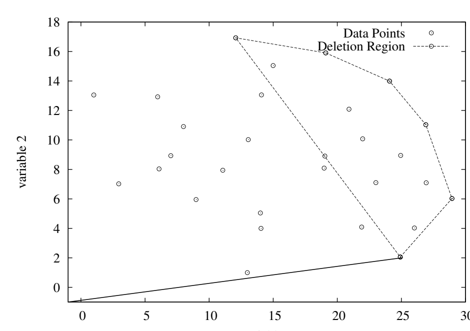

Figure 8: The i-th step in Jarvis' algorithm is able to delete those points inside the growing hull boundary.

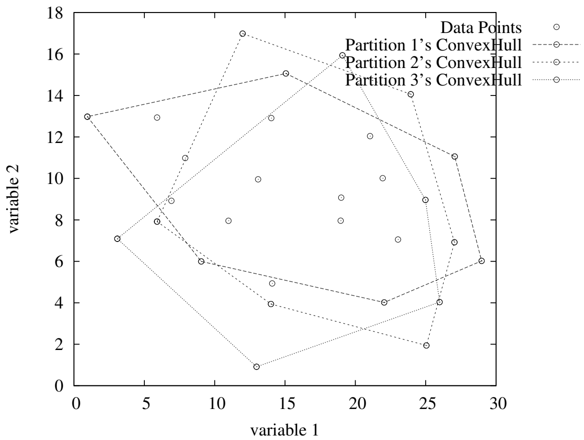

Figure 9: Example of 3 convex hulls each constructed by, for example, Graham's algorithm, surrounding the points in each of the 3 partitions of S as constructed by Chan's algorithm.

Scan type manner. If no convex hull is found, then incomplete is returned and the doubling technique is used to search further. There are $ceil(\frac{n}{m})$ possibly overlapping convex polygons, where each has at most m vertices. It takes $O(m \log m)$ for Graham's scan. The preprocessing time is $O(\frac{n}{m}(m \log m)) = O(n \log m)$ with h wrapping steps each costing $O(\frac{n}{m} \log m)$. Therefore, the total time is $O(n \log m + h(\frac{n}{m} \log m)) = O(n(1 + \frac{h}{m}) \log m)$ according to [1].

Chan(P, m, H)

partition P into subsets $P_1, \cdots, P_{ceil(\frac{n}{m})}$ each of size at most m
for $i = 1, \cdots, ceil(\frac{n}{m})$ **Do**
    compute CH(P_i) by Graham's scan
$p_0 \leftarrow (0, -\infty)$
$p_1 \leftarrow$ the rightmost point of P
for$k = 1, \cdots, H$ **Do**
    for$i = 1, \cdots, ceil(\frac{n}{m})$ **Do**
        compute the point $q_i \in P_i$ that maximizes
        $\angle p_{k-1} p_k q_i (q_i \neq p_k)$ by binSearch(p_i)
    $p_{k+1} \leftarrow$ point $q \in \{q_1, \cdots, q_{ceil(\frac{n}{m})}\}$
        maximizing $\angle p_{k-1} p_k q$
    if $p_{k+1} = p_1$ **Then** return the list $< p_1, \cdots, p_k >$
return incomplete

ChanWrapper(P)

for $t = 1, 2 \cdots)$ **Do**
    $L \leftarrow Chan(P, m, H)$, where $m = H = min(2^{2^t}, n)$
    if $L \neq$ incomplete then return L

5. CONCLUSIONS

Three planar convex hull algorithms have been explicated, along with their order of complexity in terms of their number of basic operations. Diagramatic descriptions of the correctness of each algorithm have also been presented. Within the time constraints imposed, a c-degree of depth of understanding has occurred, for some slightly large constant c.

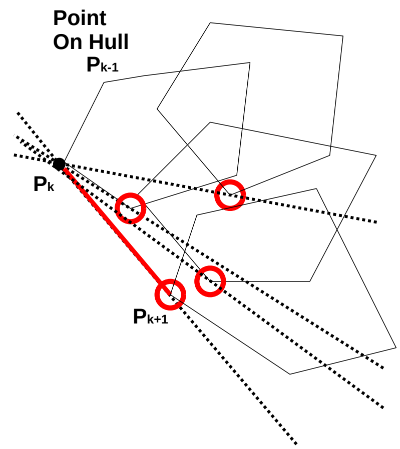

Figure 10: Find a point maximizing θ for a polygon, and choose the one maximizing θ over all polygons.

6. ACKNOWLEDGMENTS

The author would like to thank Prof. Maheshwari for teaching the course. NRC-IIT-IR group for supporting my attendance at this course, in particular, the following people from NRC: Julio J. Valdés, Bob Orchard and Fazel Famili. CISTI for supplying research materials. Professor Morin's CH2() program. Implementors of the unix *banner* command.

7. REFERENCES

[1] T. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. In *Discrete and Computational Geometry*, volume 16, pages 361–368, 1996.

[2] T. M.-Y. Chan. *Output-sensitive Construction of Convex Hulls*. PhD thesis, University of British Columbia, October 1995.

[3] R. Graham. An efficient algorith[sic] for determining the convex hull of a finite planar set. *Information Processing Letters*, 1:132–133, 1972.

[4] R. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2:18–21, 1973.

[5] L. C. Kinsey. *Topology of Surfaces*. Number ISBN 0-387-94102-9. Springer-Verlag, New York, 1993.

[6] J. W. Tukey. *Exploratory Data Analysis*. 1977.

[7] `http://biogeometry.cs.duke.edu/software/alphashapes/index.html`. Alpha shapes introduction. October 31 2005.

[8] J. J. Valdés and A. J. Barton. Relevant attribute discovery in high dimensional data based on rough sets applications to leukemia gene expressions. In *Lecture Notes in Computer Sciences / Lecture Notes in Artificial Intelligence*, Regina, Saskatchewan, September 1 2005.

[9] E. W. Weisstein. Polar coordinates. From MathWorld – A Wolfram Web Resource. `http://mathworld.wolfram.com/PolarCoordinates.html`, October 31 2005.