Concerning on the convex hull of 2D set of points

Mircea Popovici Jr.

Department of Computer Science, *OVIDIUS* University of Constanta
Constanta,Romania

Abstract

An important problem in computational geometry is to determine the convex hull of a set of points, given in the 2D-space. The aim of this paper is to present an efficient algorithm which solves the problem bellow:

Let's consider $P = \{P_i | P_i(x_i, y_i), i = 1, \ldots, n\}$ as a set of arbitrary points in 2D-space, construct the smallest convex polygon, whose knots are set in P and the left points are situated within it. This polygon is called **convex hull**.

Keywords: *Computational geometry, convex hull.*

1 INTRODUCTION

Many solutions have been developed to date. Some of them use the position of a point relative to a line, other the scalar product between two vectors, but none of them introduce a method to reduce the time running.

So, one of these methods is to verify for each couple of points if all the points that are left are situated on the same side of the straight line which is determined by the two considered points. If the test is passed succesfully, the couple of points - the test was made in comparision with - will constitute two consecutive vertices of the polygon in demand.

By doing this for a large number of points, the algorithm grows inconvenient. The algorithm based on scalar product without any other sort elements is inefficient too.

2 PROPOSED METHOD

We ought to say that the basic idea of the following algorithm is one of the easier, being, in fact, the feigning of enveloping a set of points with a thread. The efficiency of the procedure proves notable espacially in the case of analysing a large number of points, whose distribution does not influence upon the running time.

So, we want to determine the convex polygon having the smallest area $Q = Q_1 Q_2 \ldots Q_m,\ m \leq n$ with following properties:

1. $Q_j \in \{P_i | i = 1, \ldots, n\}$,
2. $P_i \in Int(Q) \cup \overline{(Q)}$, for any $i \in \{1, \ldots, n\}$.

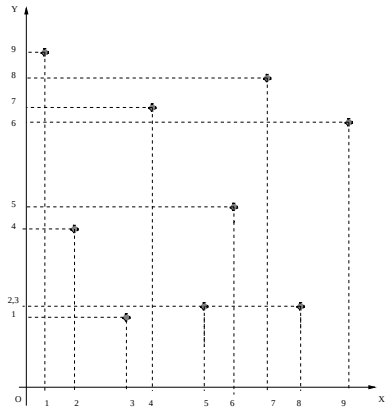

Figure 1:

were P_i are arbitrary points from xy-plane (from this reason the order of choise and analysis of points it isn't important, see figure 1)

To solve the problem we proceed as follows (we suppose to have four directions of analysis of the set P as it is show in figure 2):

- we'll order the set P ascending on x-axis, producing the PX vector of points;
- we order, again, the original set of points ascending on the y-axis, resulting the PY vector.

The four directions are:

1. N-E (north-east): this direction is caracterised by the descending trace of both vectors, PX and PY.
2. E-S (east-south): direction corespond to the ascending trace of PX vector and descending PY vector.
3. S-W (south-west) and E-S (east-south) directions are complementary with N-E and E-S.

These directions will give us a hint for covering the set of arbitrary points P, and suggest that the points whose coordinates are extremes valued are, obviously, on the convex hull, determining the rectangle from figure 2. This suggestion is confirm by theorem Klein-Millman: *The convex hull of a set of points in a Banach space is made by the extreme points of the set.*

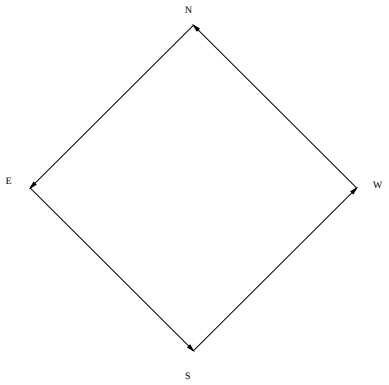

Figure 2:

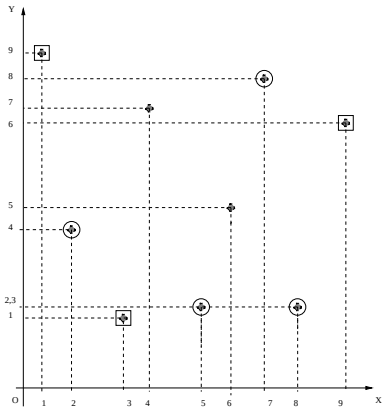

Figure 3:

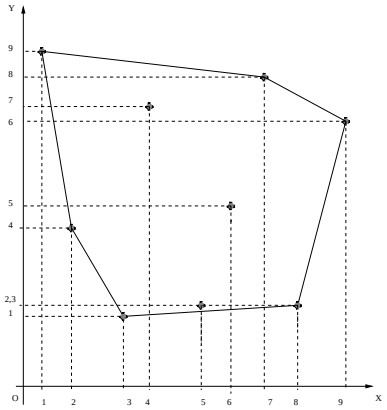

Figure 4:

Now we are able to proceed to a first analysis at set P, following the four directions we describe above, first will be taken arbitrary. We'll explain just one trace direction, the remainders been same structured, just a litle adapted.

Suppose that first direction of our process is S-W (in which the PX and PY vectors are ascending traced).

We have noted by CH, the vector who contain the convex hull vertices; by $oldx$($oldy$) the abscisae(ordinate) of the last point introduced in CH vector, and by $workx$($worky$) the abscisae(ordinate) of the current point from set P which is in process. So:

1. we'll put in the CH vector, in the first unoccupied position (which now is 1, of course), the point whose coordinates are $(ymin, x)$. If there exist more than one points with $y = ymin$, we will take the point with smallest abscisae from these points. In our case the point is (PX_3, PX_1),
2. we set $oldx = PX_3$, $oldy = PY_1$ and $workx = oldx$.
3. until $workx \leq PX_9$, ($PX_9 = maxx$):
   (a) we cover the set of points in the order in which they appear in PY vector, for each point making the test $x > oldx$, and then setting $workx = x$.
   (b) if the test is passed succesfully, then we're actuallizing $oldx$ by x of the just analysed point, put this point in CH vector and skip to the next point.

Acting in this way we'll obtain the following points in CH vector: (PX_3, PY_1), (PX_5, PY_2) and (PX_8, PY_3). The point (PX_9, PY_6) it will be considered in the next trace direction, W-N, where it exist a cycle with same structure, only the difference being that the vectors PX and PY role will be intercharged.

After all of above mentioned directions have been analysed, we'll have the following components of CH vector:

- (PX_9, PY_6) and (PX_7, PY_8) derived from W-N direction,
- none of the points are introduced by tracing N-E direction,
- (PX_1, PY_9) and (PX_2, PY_4) from E-S direction.

All these elements of CH vector can be found in figure 3. You can observe how near we are of the final result! (figure 4)

For the next step, and the last (who produce the final convex hull), we need two flag elements: $next(i)$ which contain the position of the next point of the point $CH(i)$, and $mesaj$, a logical variable whose role is to stop the sorting process. More, we'll use a procedure to determine the position of a point in 2D-plane against an oriented line, the sense being produced by the order of the two points lies on the line (this order is very important from this reason). The procedure will return an real value (positive or negative) and it is based on scalar product between two vectors.

For verifing that CH vector contains only extreme points we are using the above mentioned procedure, i.e. we analized the position of point $CH(i+2)$ against the edge $(CH(i), CH(i+1))$, evaluating the determinant:

$$d = \begin{vmatrix} x_i - x_{i+1} & x_{i+2} - x_{i+1} \\ y_i - y_{i+1} & y_{i+2} - y_{i+1} \end{vmatrix}$$

where $x_j = CH[j].x, y_j = CH[j].y, j \in \{i, i+1, i+2\}$, and $i = 1, \ldots, m, m = card(CH)$, till d becames greater than zero for all $(i, i+1, i+2)$ triples coresponding with points from CH vector. For programming reasons, we've assimilated CH vector with a circular list.

3 CONCLUSION

Some of you may wonder *There are too many types of sorting methods. For what reason?* It is simply to notice that for a great number of arbitrarypoints in 2D-plane, this method excludes a very large part of them, so, the expensive test based on the scalar product action just for a small subset of the initial one. More, it's less expensive to compute the determinant d for $n/3$ times than $2n$ times.

The convex hull can be used to determine the smallest area rectangle who include an arbitrary set of 2D-points. The solution of this problem, once introduced in an algorithm to save a graphic image on display, may produce an efficient saving method, the efficiency consist in the memory space needed for this kind of savings.

Another domain of aplicability is triangularization of an 2D-set of points, too. For this, is enough to proceed in the following way:

1. note the set of points not tested by P',
2. construct the convex hull, CH', of this set of points, P',
3. subtract CH' from P' and obtain $P'' = P' - CH'$,
4. construct the convex hull, CH'', of the set P'',
5. the region delimited by CH' and CH'' is triangularized.
6. rename the set P'' with P'
7. if the number of points of set P' is greater than 2 go to step 2.

The results of the algorithm can be used as input data of another triangulation process, based on Delaunay empty circumcircle criterion. The same principle can be used in the process of generation of 3D objects by planar sections and, based on this, follows the triangular faces generation phase.

Last, but not least, the convex hull can be succesfully use in finite element method process.

References

[1] Foley,van Dam, Feiner, Hughes - Computer Graphics. Principles and Practice, Addison Wesley, 1990.

[2] F.P. Preparata - Approximation algorithm for convex hulls, Communications of the ACM, Vol. 25, No. 1, 1982.

Author:

Mircea Popovici Jr.
Prof. Assistant at Department of Computer Science
OVIDIUS University of Constanta.
Address: B-dul Mamaia 124, 8700 Romania, Phone:00-40-41-618070
Email: dmpopovici@ovidius.ct.ro