Mircea Popovici Jr.
OVIDIUS University of Constanta
ROMANIA

Concerning on the convex hull of 2D set of points

International Conference Graphicon 1998,
Moscow, Russia, http://www.graphicon.ru/

dmpopovici@ovidius.ct.ro

dmpopovici@ovidius.ct.ro

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 $P=\{P_i \mid P_i(x_i,y_i),\ i=1,\dots,n\}$ 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.

dmpopovici@ovidius.ct.ro

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 successfully, the couple of points - the test was made in comparison 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.

dmpopovici@ovidius.ct.ro

Proposed method

We want to determine the convex polygon having the smallest area $Q=Q_1Q_2\ldots Q_m$, $m \leq n$ with following properties:

.$Q_j \in \{P_i \mid i=1,\ldots,n\}$,

.$P_i \in \mathrm{Int}(Q) \cup$, for any $i \in \{1,\ldots,n\}$.

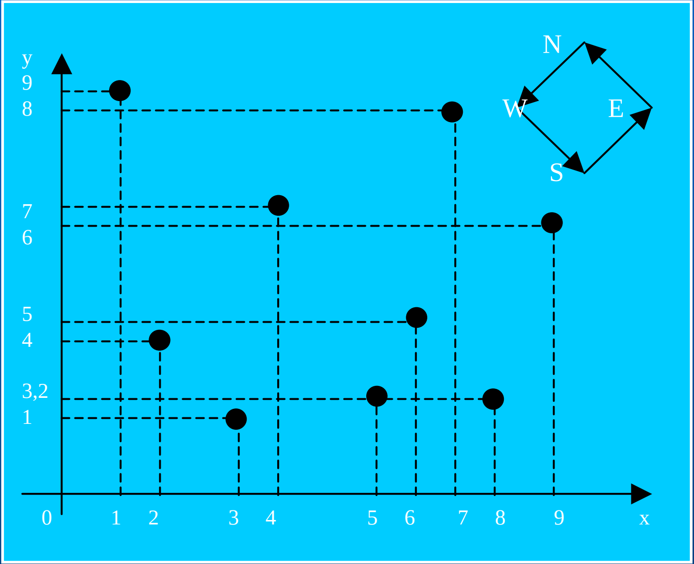
were P_i are arbitrary points from *xy*-plane (from this reason the order of choise and analysis of points it isn't important)

dmpopovici@ovidius.ct.ro

Step 1

To solve the problem we proceed as follows, supposing we have four directions of analysis of the set *P (N-W, W-S, S-E, and E-N*):

- **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.**



These four 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, and they are:

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

International Conference Graphicon 1998, Moscow, Russia, http://www.graphicon.ru/

dmpopovici@ovidius.ct.ro

Step 2

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

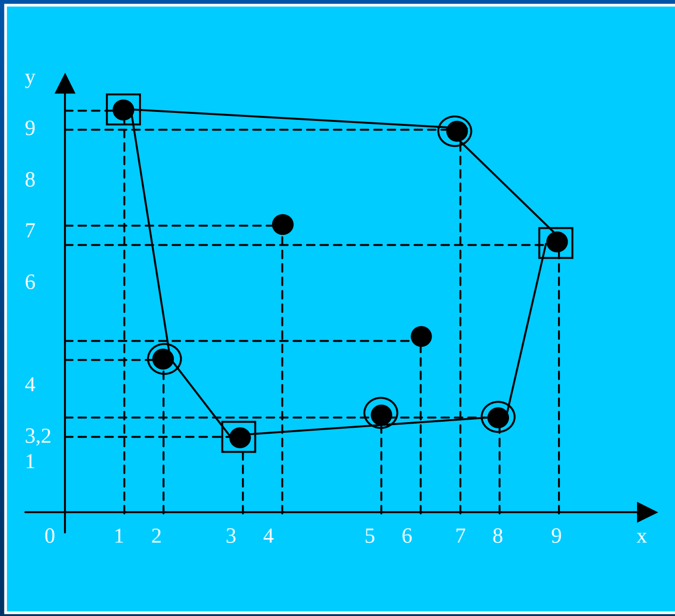
We have noted by *CH*, the vector who contain the convex hull vertices; by *oldx*(*oldy*) the ordinate of the last point introduced in *CH* vector, and by *workx*(*worky*) the ordinate of the current point from set *P* which is in process. So:

- 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),
- we set $oldx=PX_3$, $oldy=PY1$ and $workx=oldx$.
- until $workx \leq PX_9$, $(PX_9=maxx)$:
  - 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*.
  - 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, E-N, where it exist a cycle with same structure, only the difference being that the vectors *PX* and *PY* role will be intercharged.

dmpopovici@ovidius.ct.ro

Step 3

- 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 E-N direction,
  - none of the points are introduced by tracing N-W direction,
  - (PX_1,PY_9) and (PX_2,PY_4) from W-S direction.
- Now, you can observe how near we are of the final result!



- 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*∈*{i,i+1,i+2}*, and *i=1,…,m*, *m*=card(*CH*), till *d* becames greater than zero for all (*i,i+1,i+2*) triples coresponding with points from *CH* vector.

International Conference Graphicon 1998, Moscow, Russia, http://www.graphicon.ru/

Previous Next

dmpopovici@ovidius.ct.ro

Conclusions and ...

- It is simply to notice that for a great number of arbitrary points 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, solution which can be introduced in an algorithm to save a graphic image on display.

dmpopovici@ovidius.ct.ro

… future works

- Another domain of aplicability is triangularization of an 2D-set of points. 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 successfully use in finite element method process.

dmpopovici@ovidius.ct.ro

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.

[3] F.P.Preparata, M.L.Shamos – ***Computational Geometry. An introduction***. Springer, New York.

[4] T.P.Fang, L.A. Piegl – ***An algorithm for Delaunay triangulation and convex hull computation using a parse matrix***, Comput.-Aided Des. 1992, 24, 425-436.

About the author

***Mircea Popovici Jr*.,**

Professor assistant

Department of Computer Science and Numerical Methods

OVIDIUS University of Constanta.

Address: 124 Mamaia, 8700 Constanta, Romania.

E-mail: **dmpopovici@ovidius.ct.ro**

Web page: **www.imc.ro/~dmpopovici**