Buletinul Stiintific al Universitatii din Baia-Mare
Seria B, Matematica-Informatica, vol.IX(1993),95-102

A METHOD OF DECOMPOSING A POLYGON IN TRIANGLES

Ioana ZELINA and Remus ZELINA

In this paper we shall present a method for decomposing a polygon in triangles, an algorithm for this method and some of its applications.

We shall begin with some definitions.

Let $P=[P_1,P_2,\ldots,P_n]$ be a polygon with the vertices P_i, $i=\overline{1,n}$ and the sides $[P_iP_{i+1}]$, $i=\overline{1,n}$ where $P_{n+1}=P_1$. We say the vertices P_i and P_{i+1} are consecutive vertices(P_n and P_1 are consecutive vertices, $P_{n+1}=P_j$)and the sides $[P_iP_{i+1}]$ and $[P_{i+1}P_{i+2}]$ are consecutive sides ($[P_{n-1}P_n]$ and $[P_nP_1]$ are consecutive sides).

Definition 1. we call "normal decomposition" of the polygon P a set of triangles $M=\{T_1,T_2,\ldots,T_n\}$ so that:

a1) the vertices of the triangles in M are vertices of P.

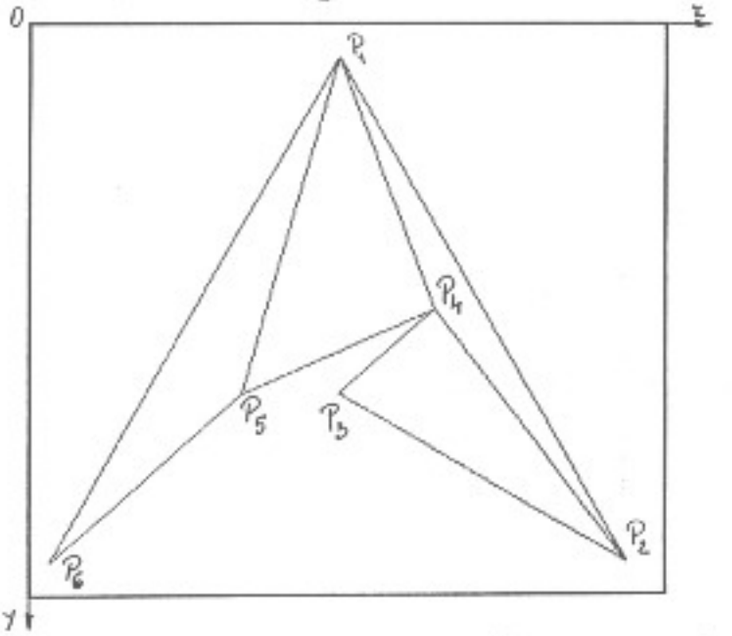
b1) any two triangles in M have disjoint interiors:

$$intT_i \cap intT_j = \Phi,\ i\neq j,\ i,\ j\in\{1,\ldots,m\}$$

c1) the union of the triangular surfaces in M is the polygonal surface P:

$$\bigcup_{i=1}^{m}\overline{intT_i} = \overline{intP}$$

An example of normal decomposition is described in Figure 1.



$M=\{P_2P_3P_4,\ P_1P_2P_4,\ P_1P_4P_5,\ P_1P_5P_6\}$ is a normal decomposition for P.

Figure 1.

Definition 2. We call a "border triangle" for the polygon P a triangle that has the vertices consecutive vertices of the polygon.

It is obvious that a border triangle is a triangle with two sides consecutive sides of the polygon.

For example, in Figure 1, the triangles $P_5P_6P_1$, $P_2P_3P_4$ and $P_3P_4P_5$ are border triangles for P. The triangle $P_1P_2P_4$ is not a border triangle for P.

Forward we shall use only the border triangles even if we do not call them explicitly border triangles. We are interested only in border triangles and if we use the triangle denoted $P_iP_{i+1}P_{i+2}$ obviously it is border triangle.

Definition 3. We call the border triangle $P_iP_{i+1}P_{i+2}$ a "good triangle"(it is useful to our method) if it has the properties:

a3) the triangle is "well-oriented" that means the intersection between any neighbourhood of P_{i+1}, the interior of the triangle and the interior of the polygon is not the null set.

b3) none of the others vertices of P belongs to the interior of the triangle.

Observation 1. a) a border triangle $P_iP_{i+1}P_{i+2}$ is well-oriented if the measure of the angle of the polygon $\widehat{P_iP_{i+1}P_{i+2}}$ is less than 180°.

b) we call a triangle that is not well-oriented triangle bad-oriented; we shall speak about the orientation of a triangle when we are interested if it is well or bad-oriented.

c) obviously a good triangle is a triangle with the interior included in the interior of the polygon. It is not posible to verify if a triangle is good or not using this property; the method search the good triangles between the well-oriented triangles. We shall give a simple method to find the orientation of a triangle in the next property.

Property 1. Let $P_iP_{i+1}P_{i+2}$ and $P_{i+1}P_{i+2}P_{i+3}$ be two border triangles with a common side. Then:

A1) If the vertices P_i and P_{i+3} are on the same side of $P_{i+1}P_{i+2}$ then they have the same orientation.

B1) If the vertices P_i and P_{i+3} are on the opposite sides of $P_{i+1}P_{i+2}$ then the triangles have different orientations.

Proof:

If the vertices P_i and P_{i+3} are on the same side of $P_{i+1}P_{i+2}$ then the measures of the angles of the polygon $\widehat{P}_{i+1}$ and $\widehat{P}_{i+2}$ are both less or greater than 180° that means the triangles have the same orientation(they are both well-oriented or bad-oriented, see figure 2).

If P_i and P_{i+3} are on opposite sides of $P_{i+1}P_{i+2}$ then only one of the measures of the angles of the polygon $\widehat{P}_{i+1}$ and $\widehat{P}_{i+2}$ is greater than 180°. So, only one of the triangles is well-oriented(see figure 3).

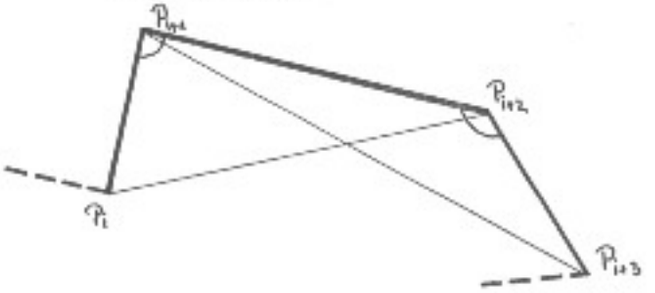


Figure 2.

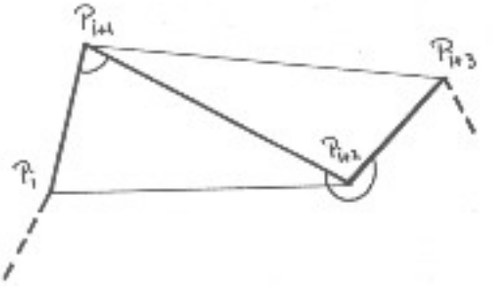


Figure 3.

So if we know the orientation of a border triangle $P_iP_{i+1}P_{i+2}$ we can find the orientation of the triangle $P_{i+1}P_{i+2}P_{i+3}$ using known mathematic formulas.

We shall describe now the method for decomposing the polygon P in triangles using good triangles:

- we search a good triangle in P. This triangle is one of the triangles in the normal decomposition. Such a triangle always exists because in any polygon there is a triangle with the interior included in the interior of the polygon.

- when we have found such a triangle we put it in the normal decomposition and we form a new polygon with less vertices than the polygon P and we use the same method for this polygon. We assume that the good triangle for P is $P_iP_{i+1}P_{i+2}$. The new polygon is the polygon that we obtain eliminating the vertex P_{i+1} and the sides $[P_iP_{i+1}]$ and $[P_{i+1}P_{i+2}]$ and forming the side $[P_iP_{i+2}]$. So the new polygon is $P'=[P_1,P_2,\ldots,P_{i-1},P_{i+1},\ldots,P_n]$. Renumbering the vertices (it is necessary for the algorithm) we obtain $P'=[P_1,\ldots,P_{n-1}]$(we need to remember the old index of each vertex). We repeat this procedure until the polygon P' has only three vertices and we put this last polygon in the normal decomposition.

If we consider the cartesian system xOy with O on the left-up corner of the screen then it is obvious that if P_{i0} is the vertex with the ordinate $y_{i0}= \min\{ y_i/ i=\overline{1,n}\}$ then the triangle $P_{i0-1}P_{i0}P_{i0+1}$ is well-oriented(the measure of the angle of the polygon $\widehat{P}_{i0}$ is less than 180°). So we start the examination for a good triangle with this triangle.

We denote : P'= the current polygon that is to be decomposed.
nr= the number of vertices in the polygon P'.
M = the current set of good triangles; finally M will be the normal decomposition.

This is the algorithm:

Step 1. We make the initialisations: P':=P; nr:=n; M:=Φ.
Step 2. If nr=3 then M:=M+{P'} and STOP.
Step 3. We choose the vertex P_i that has the minimum ordinate(it is a well-oriented).
Step 4. If the triangle $P_{i-1}P_iP_{i+1}$ is a good triangle then goto Step 5 else goto Step 6.

Step 5. M:=M+{$P_{i-1}P_iP_{i+1}$}; we consider the new polygon P' as we described when we presented the method; nr:=nr-1;
goto Step 2.

Step 6. i:=i+1; goto Step 4.

If i=1 then $P_{i-1}=P_n$ and $P_{i-2}=P_{n-1}$.

For example, here is the algorithm for decomposing the polygon in Figure 1, where $P_1(200,50)$, $P_2(350,350)$, $P_3(200,250)$, $P_4(250,200)$, $P_5(150,250)$, $P_6(50,350)$ are the coordinates of the vertices reported at the screen cartesian system:

Step 1. M:=0; P':=[$P_1P_2P_3P_4P_5P_6$]; nr:=6;

Step 3. $P_i:=P_1$; i:=1;

Step 4. $P_6P_1P_2$ is well oriented but it contains P_3, P_4, P_5.

Step 6. i:=2;

Step 4. $P_1P_2P_3$ is well-oriented($P_6P_1P_2$ was well-oriented and P_6 and P_3 are on the same side of P_1P_2) but contains P_4.

Step 6. i:=3;

Step 4. $P_2P_3P_4$ is well-oriented(P_1,P_4 are on the same of P_2P_3) and contains no other vertices of P' so it is a good triangle.

Step 5. M:={$P_2P_3P_4$}; P'=[$P_1P_2P_4P_5P_6$]; nr:=5;

Step 3. $P_i:=P_1$; i:=1;

Step 4. $P_6P_1P_2$ is not a good triangle;

Step 6. i:=2;

Step 4. $P_1P_2P_4$ is a good triangle;

Step 6. M:={$P_2P_3P_4$, $P_1P_2P_4$}; P'=[$P_1P_4P_5P_6$]; nr:=4;

Step 3. $P_i:=P_1$; i:=1;

Step 4. $P_1P_4P_5$ is a good triangle;

Step 6. M:={$P_2P_3P_4$, $P_1P_2P_4$, $P_1P_4P_5$}; P':=[$P_1P_5P_6$]; nr:=2;

Step 2. M:={$P_2P_3P_4$, $P_1P_2P_4$, $P_1P_4P_5$, $P_1P_5P_6$}; STOP.

Property 2. If P is a polygon with n(n>2) vertices then the cardinal number of the normal decomposition M is n-2.

Proof:

We prove that using the induction for the number of the vertices and, of course, our method.

If n= 3 then P is a good triangle so M={P} and |M|=2.

We suposse the cardinal number of the normal decomposition of any polygon with n-1 vertices is n-3.

Let P be a polygon with n vertices. We search a good triangle T for P and we put it in M, M={T}. Then we form the new polygon that has n-1 vertices as we described. The new polygon P' has a normal decomposition M' with the cardinal number n-3. Obviously T belongs not to M' because a vertex of T is not vertex of P'.

So, the normal decomposition M of P is M'U{T} and it has the cardinal n-3+1=n-2. The property is demonstrated.

The decomposition of a polygon in triangles is useful in some applications.

1. For example if we want to calculate the area of a concave polygon with a considerable number of vertices, we decompose the polygon in triangles and we calculate the aria of the triangle using mathematic formulas.

2. Also we can use this method when we have to find the convex cover of a concave polygon P. This is useful when we need to decupate a concave figure with minimum lost. The procedure is simple: we search the bad-oriented triangles $P_{i-1}P_iP_{i+1}$ wich contain no other vertices of P, we eliminate the vertex P_i and the coresponding sides and we form the side $[P_{i-1}P_{i+1}]$. We repeat this for the new polygon until there are no more bad-oriented triangles. The final polygon is the convex cover of P.

3. This could be also a method to fill the polygons.

We shall present now the functions and procedures used for describing the Pascal program for this algorithm. We shall not describe the simple functions. We use the types:

```
type polygon= record
                nv: integer; { number of vertices}
                vf: array[1..30] of pointtype; { the vertices}
              end;
     triangle= array[1..3] of pointtype;
```

We use an array M to remember the triangles in the normal decomposition in place of the set M.

```
Function Parte( p1,p2,p3,p4:pointtype): boolean;
```

Parte is true if p1 and p4 are on the same side of p2p3 and false otherwise.

```
Function VfMin( p:poligon): integer;
```

VfMin returns the index of the vertex with the minimum ordinate in the polygon P.

```
Function Contains(tr:triangle; P:polygon):boolean;
```

Contains returns true if there are vertices of the polygon P in the interior of the triangle tr and false otherwise.

```
Function GoodTr(p:polygon): integer;
var tr triangle;
    or,f: boolean;
begin
  i:=VfMin(p);
  tr[1]:=p.vf[i-1]; tr[2]:=p.vf[i]; tr[3]:=p.vf[i+1];
  or:=true; f:=false;
  repeat
   if or then
    if not Contains(tr,p) then
                          begin
                           Goodr:=i; f:=true
                          end
     else
       begin
       i:=i+1;
       if not Parte(p.vf[i-2],p.vf[i-1],p.vf[i],p.vf[i+1])then
          or:=not or;
       end
     else begin end
   until f;
end; { GoodTr}
```

GoodTr returns the index i of the vertex with the property that $P_{i-1}P_iP_{i+1}$ is a good triangle for P. The boolean variable or is true if the triangle is well-oriented and false otherwise. The variable f is true if we found a good-triangle and false otherwise.

```
Procedure ChangePol(var p:polygon; i:integer);
```

This procedure eliminate the vertex P_i from the polygon as we described in the method, we renumber the vertices, we remember the first index(in the initial polygon) of each vertex and we decrement the number of vertices.

```
Procedure Attrib(tr: triangle; j:integer; p1:polygon);
```

This is the procedure that assign to tr the good-triangle $P_{j-1}P_jP_{j+1}$ in the current polygon p1 using as names(indeces) for the vertices P_{j-1}, P_j, P_{j+1} the indeces in the initial polygon.

```
Procedure DivPol;
var p1: polygon;
    k,v,i: integer;
begin
     k:=0; { number of triangles in M}
     p1.nv:=pol.nv
     for i:=1 to pol.nr do p1.vf[i]:=pol.vf[i];
     while p1.nv>3 do
      begin
       v:=GoodTr(p1);k:=k+1;
       Attrib(M[k],v,p1);
       ChangePol(p1,v);
      end;
     k:=k+1;
     Attrib(M[k],1,p1)
end;{ DivPol}
```

We shall not describe here the procedures for introducing and printing the vertices of the polygon and for the initialisations.

REFERENCES

1. H.S.M. Coxeter, Introduction to Geometry, New-York, 1961.
2. V. Cristea, Turbo Pascal 6.0, Ed. Teora, Bucuresti, 1992.

UNIVERSITY OF BAIA-MARE
DEPARTAMENT OF MATHEMATICS
4800, BAIA-MARE
ROMANIA