



A11103 710484

NIST  
PUBLICATIONS

Applied and  
Computational  
Mathematics  
Division

NISTIR 4716

---

Computing and Applied Mathematics Laboratory

---

*Computing Delaunay  
Triangulations for Comet-shaped  
Polygons*

*J. Bernal*

*November 1991*

---

U.S. DEPARTMENT OF COMMERCE  
National Institute of Standards and Technology  
Gaithersburg, MD 20899

QC  
100  
.U56  
4716  
1991  
C.2



NISTIR 4716

# Computing Delaunay Triangulations for Comet-Shaped Polygons

**J. Bernal**

**U.S. DEPARTMENT OF COMMERCE  
National Institute of Standards  
and Technology  
Computing and Applied  
Mathematics Laboratory  
Applied and Computational  
Mathematics Division  
Gaithersburg, MD 20899**

**November 1991**



**U.S. DEPARTMENT OF COMMERCE  
Robert A. Mosbacher, Secretary  
NATIONAL INSTITUTE OF STANDARDS  
AND TECHNOLOGY  
John W. Lyons, Director**



# Computing Delaunay triangulations for comet-shaped polygons

Javier Bernal

*National Institute of Standards and Technology, Gaithersburg, MD 20899*

**Abstract.** In this paper, we present two triangulation algorithms which combined produce an algorithm for computing Delaunay triangulations for comet-shaped polygons. The first algorithm constructs in linear time a triangulation for a comet-shaped polygon. The second algorithm constructs a Delaunay triangulation for a polygon from any triangulation for the polygon. The algorithms can be used for deleting vertices in a Delaunay triangulation and for computing constrained Delaunay triangulations.

**Key words.** algorithm, comet-shaped polygon, computational complexity, computational geometry, constrained Delaunay triangulation, Voronoi diagram

**AMS(MOS) subject classifications.** 68U05

## 1. Introduction

Given  $S$ , a finite set of points in the plane, a *triangulation for  $S$*  is any collection of triangles in the plane having pair-wise disjoint interiors, each of which intersects  $S$  exactly at its vertices, and the union of which is the convex hull of  $S$ . Given  $T$ , a triangulation for  $S$ , we say that  $T$  is a *Delaunay triangulation for  $S$*  if for each triangle in  $T$  there does not exist a point of  $S$  inside the circumcircle of the triangle. Delaunay triangulations have been studied and algorithms for computing them have been presented in [1, 4, 5, 6, 7, 8].

Let  $R$  be a polygon in the plane. By a *triangulation for  $R$*  we mean a collection of triangles in the plane having pair-wise disjoint interiors, the vertices of which are the vertices of  $R$ , and the union of which is  $R$ . Given  $T$ , a triangulation for  $R$ , we say that  $T$  is a *Delaunay triangulation for  $R$*  if for each triangle  $t$  in  $T$  there does not exist a vertex  $P$  of  $R$  inside the circumcircle of  $t$  such that the boundary of  $R$  does not intersect the interior of the convex hull of  $t \cup \{P\}$ . The problem of computing Delaunay triangulations for polygons has been addressed in [3].

Let  $R$  be a simple polygon. Here and in the sequel, we denote by  $I(R)$ ,  $B(R)$ , and  $V(R)$ , respectively, the interior of  $R$ , the boundary of  $R$ , and the set of vertices of  $R$ . In addition, given points  $P$  and  $Q$  in the plane,  $P \neq Q$ , we denote by  $[P, Q]$  and  $(P, Q)$ , respectively, the closed and open line segments having  $P$  and  $Q$  as end-points.

Given a simple polygon  $R$ , we say that  $R$  is *star-shaped* if there exists a point  $Q$  in  $I(R)$ , such that for each point  $P$  in  $R$ ,  $P \neq Q$ ,  $(P, Q)$  is contained in  $I(R)$ . An example of a star-shaped polygon is the union of the triangles in a triangulation having a given vertex in common.

Given a simple polygon  $R$ , and points  $P$  and  $Q$  in  $V(R)$  and  $B(R)$ , respectively,  $P \neq Q$ , such that  $(P, Q)$  is contained in  $R \setminus V(R)$ , we say that  $R$  is *comet-shaped relative to  $[P, Q]$*  if for each point  $U$  in  $R \setminus [P, Q]$ , there exists a point  $W$  in  $(P, Q)$  such that  $(U, W)$  is contained in  $I(R)$ . We say that  $R$  is *comet-shaped* if there exist points  $P$  and  $Q$  in  $V(R)$  and  $B(R)$ , respectively,  $P \neq Q$ , such that  $R$  is comet-shaped relative to  $[P, Q]$ . Clearly, star-shaped polygons are comet-shaped. Figure 1 illustrates several comet-shaped polygons.

In this paper, we present two algorithms that can be combined to produce an algorithm for computing a Delaunay triangulation for a comet-shaped polygon. The first algorithm, which we call CMTTRI, constructs in linear time a triangulation for a comet-shaped polygon. The second algorithm, which we call OPTTRI, constructs a Delaunay triangulation for a polygon from any triangulation for the polygon.

Algorithm CMTTRI makes use of a modified version of an algorithm in [2], called EDGSTR, that was designed for computing a Delaunay triangulation for a polygon that

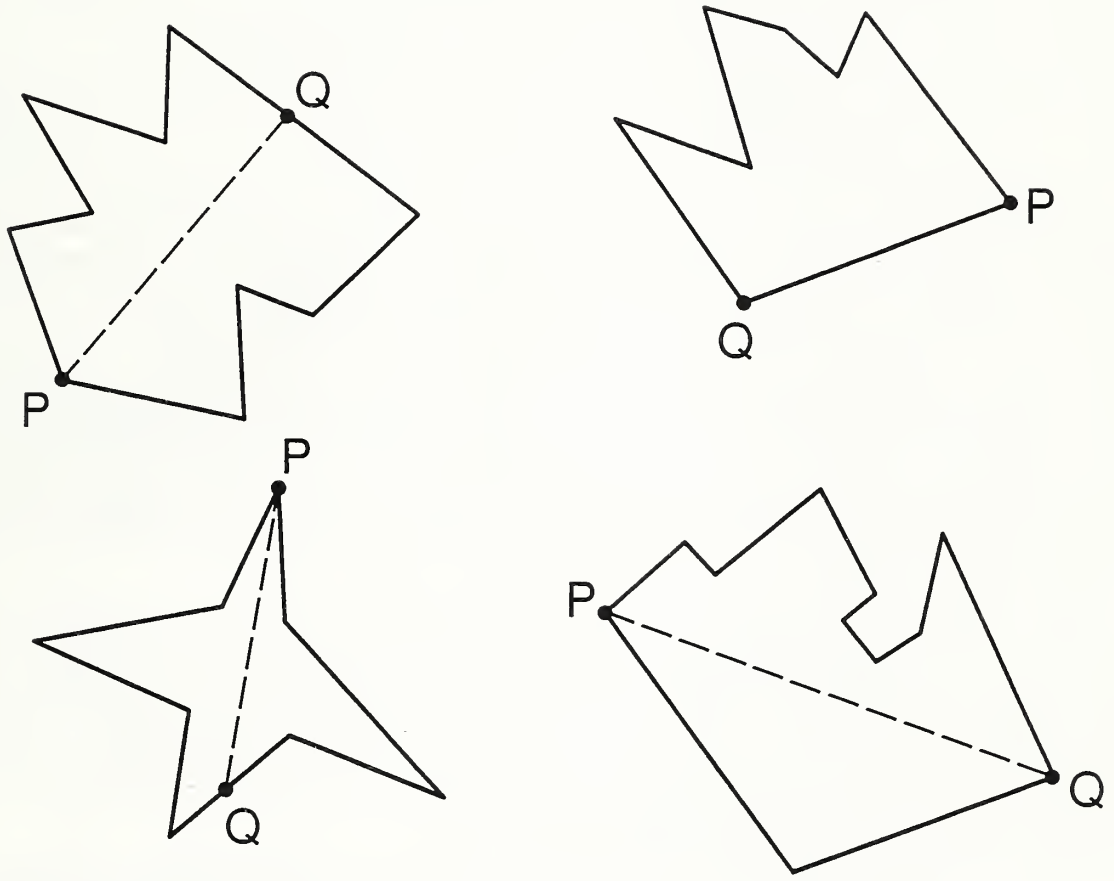


Figure 1: Several comet-shaped polygons. Each polygon is comet-shaped relative to  $[P, Q]$ .

is comet-shaped relative to a linear component of its boundary. This special type of comet-shaped polygon was called edge star-shaped in [2], and was used there in the context of Delaunay triangulations constrained by line segments. Let  $R$  be one such polygon. With  $r$  defined as the number of vertices of  $R$ , assume  $P_1$  and  $P_r$  are vertices of  $R$  such that  $R$  is comet-shaped relative to  $[P_1, P_r]$ , and let  $P_1, \dots, P_r$  be the vertices of  $R$  in the order in which they appear in  $B(R)$  in a counterclockwise direction around  $R$ . Given integers  $i, k, 1 \leq i < k \leq r$ , we define *the convex realization in  $R$  of  $P_i, \dots, P_k$* , as the subset of  $R$  which is the union of line segments of the form  $[P, Q]$  where  $P \in [P_l, P_{l+1}]$ ,  $Q \in [P_m, P_{m+1}]$  for integers  $l, m, i \leq l, m \leq k - 1$ , and  $[P, Q] \subseteq R$ . In addition, by letting  $R_{ik}$  be the convex realization in  $R$  of  $P_i, \dots, P_k$ , we define *the convex envelope in  $R$  of  $P_i, \dots, P_k$* , as the subset of  $R_{ik}$  that contains a point  $U$  if and only if either  $U \in [P_1, P_r]$ , or for some point  $W$  in  $(P_1, P_r)$ ,  $(U, W)$  is contained in  $I(R \setminus R_{ik})$ . We notice, by letting  $E_{ik}$  be the convex envelope in  $R$  of  $P_i, \dots, P_k$ , that for some integer  $s, 2 \leq s \leq k - i + 1$ , points  $Q_1, \dots, Q_s$  exist such that  $Q_1$  equals  $P_i$ ,  $Q_s$  equals  $P_k$ ,  $E_{ik}$  equals  $\cup_{l=1}^{s-1} [Q_l, Q_{l+1}]$ , and  $Q_1, \dots, Q_s$  are the points in  $\{P_i, \dots, P_k\} \cap E_{ik}$  in the order in which they appear in  $B(R)$  in a counterclockwise direction around  $R$ . Figure 2 illustrates a polygon that is comet-shaped relative to a linear component of its boundary, and the convex realization and convex envelope in the polygon of a subset of the set of its vertices.

Let  $R$  and  $P_1, \dots, P_r$  be as above. Given integers  $i, k, 1 \leq i < k - 1 < r$ , a close analysis of algorithm EDGSTR in [2] reveals that by undergoing some minor modifications it can also be used for computing a representation for  $E_{ik}$  and a triangulation for each component of non-empty interior of  $R_{ik}$ . The modified version of algorithm EDGSTR of which algorithm CMTTRI makes use, does exactly this and is essentially EDGSTR without the step that enforces the Delaunay requirement. This modified version of EDGSTR, which we call EDGTTRI, is also presented in this paper.

Let  $R$  be a comet-shaped polygon, and let  $P$  and  $Q$  be points in  $V(R)$  and  $B(R)$ , respectively,  $P \neq Q$ , such that  $R$  is comet-shaped relative to  $[P, Q]$ . In addition, without any loss of generality, assume that  $[P, Q]$  is parallel to the x-axis of the 2-dimensional Cartesian coordinate system and that it partitions  $R$  into two regions of non-empty interior. Under these assumptions, we let  $R_L$  represent the polygon which is the portion of  $R$  on or below  $[P, Q]$ , and  $R_U$  the polygon which is the portion of  $R$  on or above  $[P, Q]$ . Also, for some positive integer  $j_L$ , we let  $P_j^L, j = 1, \dots, j_L$ , represent the points that are vertices for both  $R$  and  $R_L$  in the order in which they appear in  $B(R_L)$  in a counterclockwise direction around  $R_L$  with  $P_1^L$  equal to  $P$ ; and for some positive integer  $j_U$ , we let  $P_j^U, j = 1, \dots, j_U$ , represent the points that are vertices for both  $R$  and  $R_U$  in the order in which they appear in  $B(R_U)$  in a counterclockwise direction around  $R_U$  with  $P_{j_U}^U$  equal to  $P$ . Clearly,  $[P, Q]$  is a linear component of the boundaries of  $R_L$  and  $R_U$ ;  $R_L$  and  $R_U$  are comet-shaped polygons relative



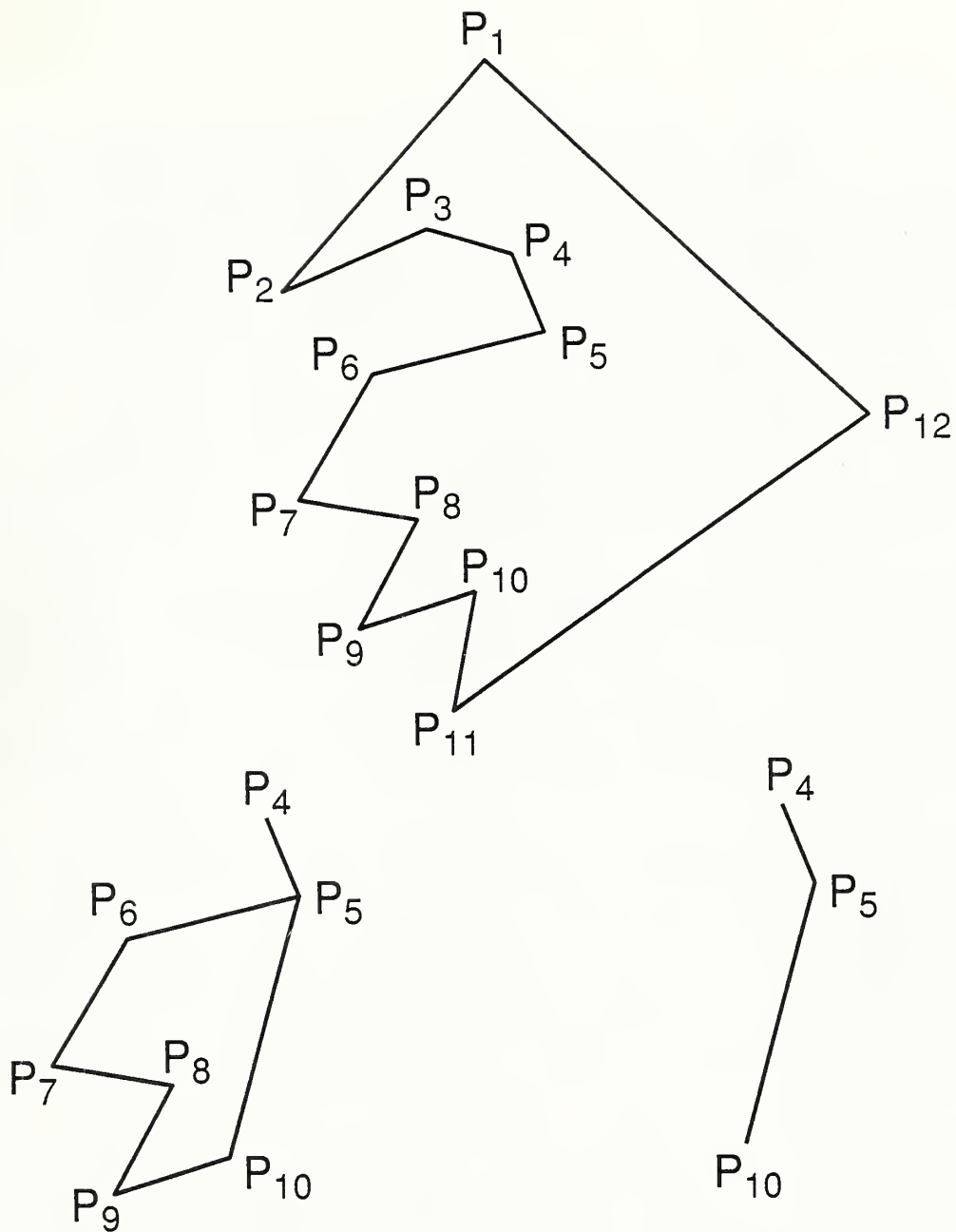


Figure 2: Above, a polygon  $R$  with vertices  $P_1, \dots, P_{12}$ .  $R$  is comet-shaped relative to  $[P_1, P_{12}]$ . Below, from left to right, the convex realization  $R_{4,10}$  and the convex envelope  $E_{4,10}$  in  $R$  of  $P_4, \dots, P_{10}$ .

to  $[P, Q]$ ; and the concepts of convex realization and convex envelope make sense in  $R_L$  for  $P_j^L$ ,  $j = 1, \dots, j_L$ , and in  $R_U$  for  $P_j^U$ ,  $j = 1, \dots, j_U$ . Algorithm CMTTRI computes a triangulation for  $R$  by essentially first partitioning  $R$  into three regions that have pair-wise disjoint interiors, and then triangulating each component of nonempty interior of each region. The first and second regions correspond to the convex realizations in  $R_L$  of  $P_j^L$ ,  $j = 1, \dots, j_L$ , and in  $R_U$  of  $P_j^U$ ,  $j = 1, \dots, j_U$ , respectively. Thus, algorithm EDGTRI is applicable for computing representations for the convex envelopes in  $R_L$  of  $P_j^L$ ,  $j = 1, \dots, j_L$ , and in  $R_U$  of  $P_j^U$ ,  $j = 1, \dots, j_U$ , and for computing a triangulation for each component of non-empty interior of the convex realizations in  $R_L$  of  $P_j^L$ ,  $j = 1, \dots, j_L$ , and in  $R_U$  of  $P_j^U$ ,  $j = 1, \dots, j_U$ . Finally, the third region corresponds to the closure of the complement in  $R$  of the union of the first two regions. This region is empty if  $Q$  is a vertex of  $R$ . Otherwise, it is the polygon whose boundary is composed of the line segment  $[P_1^U, P_{j_L}^L]$ , the convex envelope in  $R_L$  of  $P_j^L$ ,  $j = 1, \dots, j_L$ , and the convex envelope in  $R_U$  of  $P_j^U$ ,  $j = 1, \dots, j_U$ . Thus, the region is a comet-shaped polygon relative to  $[P_1^U, P_{j_L}^L]$ , and algorithm EDGTRI is also applicable for computing a triangulation for it. Figure 3 illustrates a comet-shaped polygon that has been partitioned into the three aforementioned regions. The shaded region is the interior of the third region. The first region is the rest of the polygon on or below  $[P, Q]$  minus the interior of the linear component of the polygon that contains  $Q$ . The second region can be similarly identified on or above  $[P, Q]$ .

Let  $T$  be a triangulation for a polygon  $R$ . Given a triangle  $t$  in  $T$ , we denote by  $A(t)$  the set of vertices of  $R$  that are vertices of triangles in  $T$  adjacent to  $t$ , and say that  $t$  satisfies the *circle criterion in  $T$*  if none of the vertices of  $R$  in  $A(t)$  is inside the circumcircle of  $T$ . Using arguments similar to those in [4], it can be shown that  $T$  is a Delaunay triangulation for  $R$  if each triangle in  $T$  satisfies the circle criterion. Algorithm OPTTRI is an incremental algorithm which, based on this result, computes a Delaunay triangulation for a polygon  $R$  from an arbitrary triangulation  $T$  for the polygon. OPTTRI starts by selecting an arbitrary triangle, which we call  $t_1$ , in  $T$ . Let  $m$  be the number of triangles in  $T$ . Given a positive integer  $n$ ,  $n < m$ , OPTTRI inductively selects triangles  $t_1, \dots, t_n$  in  $T$ , whose union, which we call  $R_n$ , is connected, and computes triangles  $t_1^n, \dots, t_n^n$ , the collection of which is a Delaunay triangulation for  $R_n$ . OPTTRI then proceeds to select a triangle  $t_{n+1}$  in  $T$  in such a way that  $t_{n+1}$  is different from the previously selected triangles and the union of  $R_n$  and  $t_{n+1}$ , which we call  $R_{n+1}$ , is connected. Because the vertices of the triangles in  $T$  are in  $B(R)$ ,  $t_{n+1}$  must have exactly one side in common with  $R_n$ . Thus, an iterative edge-swapping procedure based on the circle criterion can be applied to  $t_1^n, \dots, t_n^n, t_{n+1}$ , that starts by testing  $t_{n+1}$  for the circle criterion, and that ends with a possibly new collection of triangles  $t_1^{n+1}, \dots, t_{n+1}^{n+1}$ , each of which satisfies the circle criterion, and the union of which is  $R_{n+1}$ .

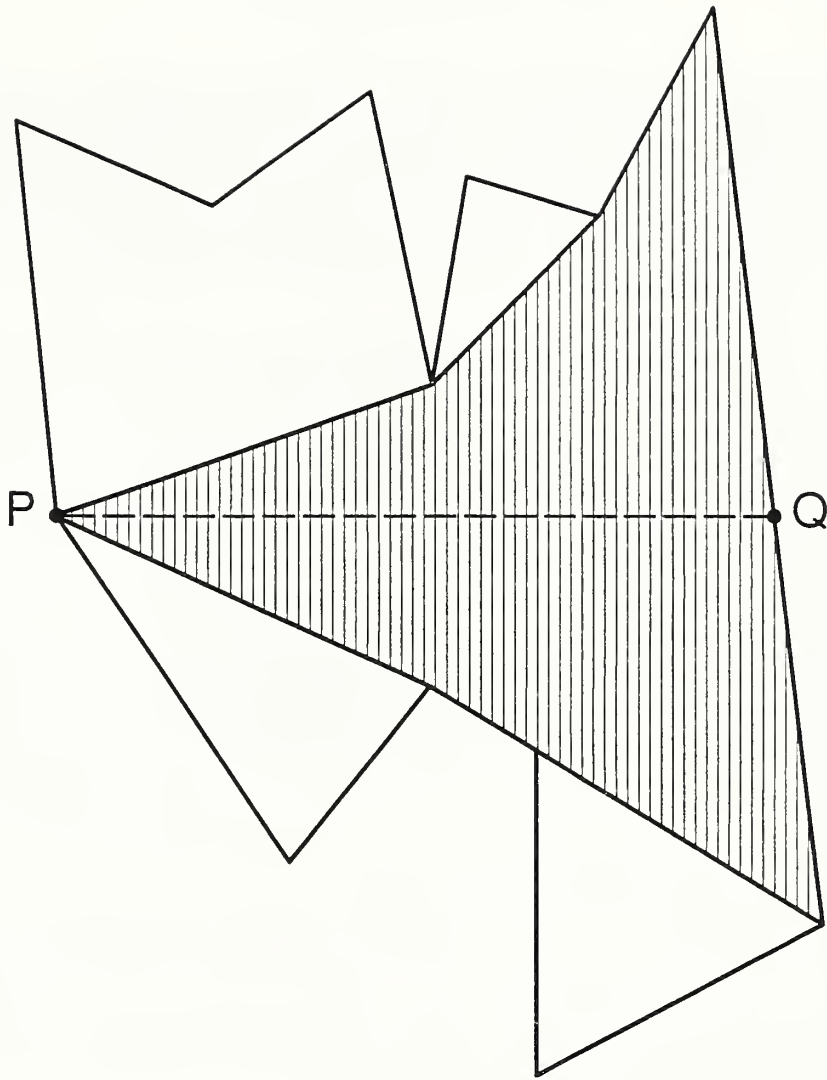


Figure 3: A comet-shaped polygon and the three regions into which it is partitioned by algorithm CMTTRI.

## 2. The EDGTRI algorithm

Let  $R$  be a simple polygon, and let  $e$  be a linear component of  $B(R)$  such that  $R$  is comet-shaped relative to  $e$ . Let  $r$  be the number of vertices of  $R$ , and let  $P_1, \dots, P_r$  be the vertices of  $R$  in the order in which they appear in  $B(R)$  in a counterclockwise direction around  $R$  with  $[P_1, P_r]$  equal to  $e$ . Let  $i, k$  be integers,  $1 \leq i < k - 1 < r$ , and let  $R_{ik}$  and  $E_{ik}$  be, respectively, the convex realization and the convex envelope in  $R$  of  $P_i, \dots, P_k$ . Set  $j$  equal to  $k - i + 1$ , and define a one-to-one function  $F$  from  $\{1, \dots, j\}$  onto  $\{P_i, \dots, P_k\}$  by setting  $F(l)$  equal to  $P_{i+l-1}$  for each  $l, l = 1, \dots, j$ .

In what follows, we present algorithm EDGTRI which computes in linear time a representation for  $E_{ik}$  and a triangulation for each component of non-empty interior of  $R_{ik}$ . The output from EDGTRI will consist of  $T$ , the collection of triangles computed by EDGTRI, tacitly in the form of a data structure that describes the triangles and their interrelations;  $J, 2 \leq J \leq j$ , the number of points in  $\{P_i, \dots, P_k\} \cap E_{ik}$ ; and  $G$ , the representation for  $E_{ik}$ , in the form of a one-to-one function from  $\{1, \dots, J\}$  into  $\{P_i, \dots, P_k\}$ , with  $G(1)$  equal to  $P_i$ ,  $G(J)$  equal to  $P_k$ ,  $E_{ik}$  equal to  $\cup_{l=1}^{j-1} [G(l), G(l+1)]$ , and  $G(1), \dots, G(J)$  equal to the points in  $\{P_i, \dots, P_k\} \cap E_{ik}$  in the order in which they appear in  $B(R)$  in a counterclockwise direction around  $R$ . Here, given points  $Q_1, Q_2, Q_3$  in the plane,  $Q_2 \neq Q_3$  and  $Q_2 \neq Q_1$ ,  $m(Q_2\vec{Q}_3, Q_2\vec{Q}_1)$  will represent the size in radians of the angle produced by a counterclockwise rotation around  $Q_2$  from ray  $Q_2\vec{Q}_3$  to ray  $Q_2\vec{Q}_1$ . The outline of EDGTRI follows.

```

procedure EDGTRI( $T, F, j, G, J$ )
  begin
1.    $T := \emptyset; G(1) := F(1); G(2) := F(2); J := 2;$ 
2.   for  $I := 3$  until  $j$  do
      begin
3.      $J := J + 1; G(J) := F(I);$ 
4.      $Q_1 := G(J - 2); Q_2 := G(J - 1); Q_3 := G(J);$ 
5.     while  $(m(Q_2\vec{Q}_3, Q_2\vec{Q}_1) < \pi$  and  $J \neq 2)$  do
          begin
6.        $T := T \cup \{\Delta Q_1 Q_2 Q_3\};$ 
7.        $J := J - 1; G(J) := Q_3;$ 
8.       if  $(J \neq 2)$  then
          begin
9.          $Q_1 := G(J - 2); Q_2 := G(J - 1)$ 
          end
        end
      end
  end

```

end  
end

The justification of EDGTRI is essentially that of EDGSTR in [2]. As for its complexity, it depends essentially on how often lines 6 through 9 of EDGTRI are executed. Since the latter depends essentially on how many triangles are created during the execution of EDGTRI, it follows that the complexity of EDGTRI depends linearly on  $j$ .

### 3. The CMTTRI algorithm

Let  $R$  be a simple polygon, and let  $P$  and  $Q$  be points in  $V(R)$  and  $B(R)$ , respectively,  $P \neq Q$ , such that  $R$  is comet-shaped relative to  $[P, Q]$ . Let  $r$  be the number of vertices of  $R$ , and let  $P_1, \dots, P_r$  be the vertices of  $R$  in the order in which they appear in  $B(R)$  in a counterclockwise direction around  $R$  with  $P_1$  equal to  $P$ . Define a function  $F$  from  $\{1, \dots, r+1\}$  onto  $V(R)$  by setting  $F(i)$  equal to  $P_i$  for each  $i$ ,  $i = 1, \dots, r$ , and  $F(r+1)$  equal to  $P_1$ .

In what follows, we present algorithm CMTTRI which computes in linear time a triangulation for  $R$ . The output from CMTTRI will consist of  $T$ , the triangulation for  $R$ , tacitly in the form of a data structure that describes the triangles and their interrelations. The outline of CMTTRI follows.

```

procedure CMTTRI( $T, F, P, Q$ )
  begin
1.    $F_1(1) := F(1); F_1(2) := F(2); j := 2;$ 
2.   while ( $Q \notin (F(j), F(j+1))$  and  $Q \neq F(j)$ ) do
      begin
3.      $j := j + 1; F_1(j) := F(j)$ 
      end
4.    $j_1 := j; flag := 0;$ 
5.   if ( $Q \neq F(j)$ ) then  $j := j + 1$ 
      else  $flag := 1;$ 
6.    $F_2(1) := F(j); F_2(2) := F(j+1); j := j + 1; j_2 := 2;$ 
7.   while ( $P \neq F(j)$ ) do
      begin
8.      $j := j + 1; j_2 := j_2 + 1; F_2(j_2) := F(j)$ 
      end

```

```

9.   if ( $j_1 \geq 3$ ) then EDGTRI( $T_1, F_1, j_1, G_1, J_1$ )
      else
          begin
10.       $T_1 := \emptyset$ ;  $G_1(1) := F_1(1)$ ;  $G_1(2) := F_1(2)$ ;  $J_1 := 2$ 
          end
11.   if ( $j_2 \geq 3$ ) then EDGTRI( $T_2, F_2, j_2, G_2, J_2$ )
      else
          begin
12.       $T_2 := \emptyset$ ;  $G_2(1) := F_2(1)$ ;  $G_2(2) := F_2(2)$ ;  $J_2 := 2$ 
          end
13.   if ( $flag = 0$ ) then
          begin
14.      for  $j := 1$  until  $J_2$  do  $F_3(j) := G_2(j)$ ;
15.       $j_3 := J_2$ ;
16.      for  $j := 2$  until  $J_1$  do
          begin
17.           $j_3 := j_3 + 1$ ;  $F_3(j_3) := G_1(j)$ 
          end
18.      EDGTRI( $T_3, F_3, j_3, G_3, J_3$ )
          end
19.   else  $T_3 := \emptyset$ ;
20.    $T := T_1 \cup T_2 \cup T_3$ 
      end

```

In order to simplify the justification of CMTTRI, we assume that  $[P, Q]$  is parallel to the x-axis of the two-dimensional Cartesian coordinate system and that it partitions  $R$  into two regions of non-empty interior. Under these assumptions, let  $R_L$  be the polygon which is the portion of  $R$  on or below  $[P, Q]$ , and let  $R_U$  be the polygon which is the portion of  $R$  on or above  $[P, Q]$ .

Lines 1 through 8 of CMTTRI essentially partition the vertices of  $R$  into two sets.  $F_1(j)$ ,  $j = 1, \dots, j_1$ , are the vertices of  $R$  that lie on or below  $[P, Q]$  in the order in which they appear in  $B(R_L)$  in a counterclockwise direction around  $R_L$  with  $F_1(1)$  equal to  $P$ .  $F_2(j)$ ,  $j = 1, \dots, j_2$ , are the vertices of  $R$  that lie on or above  $[P, Q]$  in the order in which they appear in  $B(R_U)$  in a counterclockwise direction around  $R_U$  with  $F_2(j_2)$  equal to  $P$ .

Clearly,  $R_L$  is comet-shaped relative to  $[P, Q]$  and  $[P, Q]$  is a linear component of its boundary. Thus, if  $j_1 \geq 3$  then algorithm EDGTRI can be used in line 9 of CMTTRI to compute  $T_1$ , a collection of triangles that is the union of triangulations for the components of non-empty

interior of the convex realization in  $R_L$  of  $F_1(j)$ ,  $j = 1, \dots, j_1$ ; and  $G_1(j)$ ,  $j = 1, \dots, J_1$ , those points among  $F_1(j)$ ,  $j = 1, \dots, j_1$ , that lie in the convex envelope in  $R_L$  of  $F_1(j)$ ,  $j = 1, \dots, j_1$ , in the order in which they appear in  $B(R_L)$  in a counterclockwise direction around  $R_L$  with  $G_1(1)$  equal to  $F_1(1)$ .

Similarly, if  $j_2 \geq 3$  then algorithm EDGTRI can also be used in line 11 of CMTTRI to compute  $T_2$ , a collection of triangles that is the union of triangulations for the components of non-empty interior of the convex realization in  $R_U$  of  $F_2(j)$ ,  $j = 1, \dots, j_2$ ; and  $G_2(j)$ ,  $j = 1, \dots, J_2$ , those points among  $F_2(j)$ ,  $j = 1, \dots, j_2$ , that lie in the convex envelope in  $R_U$  of  $F_2(j)$ ,  $j = 1, \dots, j_2$ , in the order in which they appear in  $B(R_U)$  in a counterclockwise direction around  $R_U$  with  $G_2(1)$  equal to  $F_2(1)$ .

If  $Q$  is not a vertex of  $R$ , lines 14 through 17 of CMTTRI identify  $F_3(j)$ ,  $j = 1, \dots, j_3$ , the vertices of the polygon whose boundary is the line segment  $[F_1(j_1), F_2(1)]$ , the convex envelope in  $R_L$  of  $F_1(j)$ ,  $j = 1, \dots, j_1$ , and the convex envelope in  $R_U$  of  $F_2(j)$ ,  $j = 1, \dots, j_2$ .  $F_3(j)$ ,  $j = 1, \dots, j_3$ , are in the order in which they appear in the boundary of the aforementioned polygon in a counterclockwise direction around the polygon with  $F_3(1)$  equal to  $F_2(1)$ . Since the aforementioned polygon is comet-shaped relative to  $[F_1(j_1), F_2(1)]$ , algorithm EDGTRI can be used in line 18 of CMTTRI to compute  $T_3$ , a triangulation for this polygon.

Finally, since merging  $T_1$ ,  $T_2$ , and  $T_3$  produces a triangulation for  $R$ , this is done in line 20 of CMTTRI.

As for the complexity of CMTTRI, it depends essentially on the complexity of the executions of EDGTRI. Since the latter depends linearly on the largest of  $j_1$ ,  $j_2$ , and  $j_3$ , it follows that the complexity of CMTTRI depends linearly on  $r$ .

#### 4. The OPTTRI algorithm

Let  $R$  be a simple polygon, and let  $T$  be any triangulation for  $R$ . In what follows, we present algorithm OPTTRI which computes from  $T$  a Delaunay triangulation for  $R$ . The input for OPTTRI must consist of  $T$ , the known triangulation for  $R$ , tacitly in the form of a data structure that describes the triangles and their interrelations; and  $t$ , any triangle in  $T$ , in the form of a variable that locates it in  $T$ . The output from OPTTRI will consist of  $T^*$ , a Delaunay triangulation for  $R$ . Here, given a triangle  $\hat{t}$  with vertices  $P_1$ ,  $P_2$ ,  $P_3$ , in one of the three orders in which they appear in  $B(\hat{t})$  in a counterclockwise direction around  $\hat{t}$ , we denote  $\hat{t}$  by either  $\Delta P_1 P_2 P_3$  or  $\Delta P_2 P_3 P_1$  or  $\Delta P_3 P_1 P_2$ , and say that each one of the three ways of denoting  $\hat{t}$  identifies  $\hat{t}$ . The outline of OPTTRI follows.

```

procedure OPTTRI( $T^*, T, t$ )
  begin

```

$P_1, P_2, P_3 :=$  points such that  $\Delta P_1 P_2 P_3$  identifies the triangle located by  $t$  in  $T$ ;  
 $P_4 := P_1$ ;  $T^* := \{\Delta P_1 P_2 P_3\}$ ;  $j := 0$ ;  
**for**  $i := 1$  **until** 3 **do**  
    **begin**  
    **if** (there exists  $\hat{P}$  such that  $\Delta \hat{P} P_{i+1} P_i$  identifies a triangle in  $T$ ) **then**  
        **begin**  
         $\hat{P} :=$  point such that  $\Delta \hat{P} P_{i+1} P_i$  identifies a triangle in  $T$ ;  
         $j := j + 1$ ;  $H(j) := \Delta \hat{P} P_{i+1} P_i$ ;  
        **end**  
    **end**  
**while** ( $j \neq 0$ ) **do**  
    **begin**  
     $P^*, P', P'' :=$  points such that  $\Delta P^* P' P'' = H(j)$ ;  
     $j := j - 1$ ;  
    **if** (there exists  $\hat{P}$  such that  $\Delta \hat{P} P^* P''$  identifies a triangle in  $T$ ) **then**  
        **begin**  
         $\hat{P} :=$  point such that  $\Delta \hat{P} P^* P''$  identifies a triangle in  $T$ ;  
         $j := j + 1$ ;  $H(j) := \Delta \hat{P} P^* P''$ ;  
        **end**  
    **if** (there exists  $\hat{P}$  such that  $\Delta \hat{P} P' P^*$  identifies a triangle in  $T$ ) **then**  
        **begin**  
         $\hat{P} :=$  point such that  $\Delta \hat{P} P' P^*$  identifies a triangle in  $T$ ;  
         $j := j + 1$ ;  $H(j) := \Delta \hat{P} P' P^*$ ;  
        **end**  
     $P_{adj} := P'$ ;  $P_{cur} := P''$ ;  
     $flag := 1$ ;  
    **while** ( $flag = 1$ ) **do**  
        **begin**  
        **if** (there does not exist  $\hat{P}$  such that  $\Delta \hat{P} P_{cur} P_{adj}$   
        identifies a triangle in  $T^*$  or (there exists  $\hat{P}$  such that  
         $\Delta \hat{P} P_{cur} P_{adj}$  identifies a triangle in  $T^*$  and  
         $\hat{P}$  is not inside the circumcircle of  $\Delta P^* P_{adj} P_{cur}$ )) **then**  
            **if** ( $P_{adj} \neq P'$ ) **then**  
                **begin**  
                 $\hat{P} :=$  point such that  $\Delta P^* \hat{P} P_{adj}$  identifies a triangle in  $T^*$ ;  
                **end**



```

         $P_{cur} := P_{adj}; P_{adj} := \hat{P}$ 
    end
else  $flag := 0$ 
else
    begin
         $\hat{P} :=$  point such that  $\Delta\hat{P}P_{cur}P_{adj}$  identifies a triangle in  $T^*$ ;
         $T^* := (T^* \setminus \{\Delta P^*P_{adj}P_{cur}, \Delta\hat{P}P_{cur}P_{adj}\}) \cup \{\Delta P^*P_{adj}\hat{P}, \Delta P^*\hat{P}P_{cur}\}$ ;
         $P_{adj} := \hat{P}$ 
    end
end
end
end

```

The iterative edge-swapping procedure based on the circle criterion that is used in OPTTRI, has been discussed, among others, in [2], [3], [4], for incrementally computing Delaunay triangulations for point sets, polygons, etc. In each case, during the incremental step, an existing triangulation, say  $T_1$ , in which each triangle satisfies the circle criterion, is merged with a new triangle, say  $\hat{t}$ , and a new triangulation is formed, say  $T_2$ , equal to  $T_1 \cup \{\hat{t}\}$ . The iterative edge-swapping procedure based on the circle criterion is then used on  $T_2$ , starting with  $\hat{t}$ , and a third triangulation is obtained, say  $T_3$ , the union of the triangles of which is the union of the triangles in  $T_2$ , and in which each triangle satisfies the circle criterion. In each case, the procedure works because the triangles in  $T_1$ , if any, that are adjacent to  $\hat{t}$  and that are unaffected by the procedure continue to satisfy the circle criterion in  $T_3$ .

The same is true for OPTTRI. During the incremental step, an existing triangulation in which each triangle satisfies the circle criterion, is merged with a new triangle, and the iterative edge-swapping procedure based on the circle criterion is then used on the resulting triangulation. Since the new triangle is adjacent to exactly one triangle in the initial triangulation and this triangle is affected by the procedure, the procedure works due to the absence of triangles in the initial triangulation that are adjacent to the new triangle and that are unaffected by the procedure.

## 5. Summary

We have presented two triangulation algorithms which combined produce an algorithm for computing a Delaunay triangulation for a comet-shaped polygon. The first algorithm, called CMTTRI, computes in linear time a triangulation for a comet-shaped polygon. The second

algorithm, called OPTTRI, constructs a Delaunay triangulation for a polygon from any triangulation for the polygon.

A specialized combination of the two algorithms has been implemented at the National Institute of Standards and Technology for the purpose of deleting a vertex anywhere in a Delaunay triangulation and obtaining a Delaunay triangulation for the remaining vertices. With this implementation, only the triangles having the vertex in common are affected, and each computed triangle is contained in their union.

Finally, we remark that algorithm CMTTRI can be used to construct in linear time a triangulation for a polygon  $R$  if points  $P$  and  $Q$  exist in  $B(R)$  such that  $R$  would be comet-shaped relative to  $[P, Q]$  if only  $P$  were in  $V(R)$ . To do this, we first obtain a triangulation  $T$  by executing CMTTRI for  $R$ ,  $P$  and  $Q$  as if  $P$  were in  $V(R)$ . Next, we let  $R'$  be the polygon which is the union of the triangles in  $T$  that have  $P$  as a vertex, and let  $P'$  and  $Q'$  be the vertices of  $R$  for which  $[P', Q']$  contains  $P$ . Clearly,  $[P', Q']$  is a linear component of  $B(R')$ , and  $R'$  is comet-shaped relative to  $[P', Q']$ . Finally, we eliminate from  $T$  each triangle that has  $P$  as a vertex, and obtain a triangulation  $T'$  by executing CMTTRI, or for that matter EDGTRI, for  $R'$ ,  $P'$  and  $Q'$  as if each vertex of  $R$  in  $R'$  were in  $V(R')$ . Clearly,  $T \cup T'$  is a triangulation for  $R$ .

## References

- [1] J. L. Bentley, B. W. Weide and A. C. Yao, Optimal expected-time algorithms for closest point problems, *ACM Trans. Math. Software* **6**, No. 4, 1980, 563-580.
- [2] J. Bernal, On constructing Delaunay triangulations for sets constrained by line segments, NIST Technical Note 1252, National Institute of Standards and Technology, Gaithersburg, Md., 1988.
- [3] L. De Floriani, B. Falcidieno and C. Pienovi, Delaunay-based representation of surfaces defined over arbitrarily shaped domains, *Comput. Vision Graphics Image Process.* **32**, 1985, 127-140.
- [4] C. L. Lawson, Software for  $C^1$  surface interpolation, *Math. Software III*, (J. R. Rice, Ed.), pp. 161-194, Academic Press, New York, 1977.
- [5] D. T. Lee and B. J. Schachter, Two algorithms for constructing a Delaunay triangulation, *Int. J. Comput. Inform. Sci.* **9**, No. 3, 1980, 219-242.

- [6] T. Ohya, M. Iri and K. Murota, A fast Voronoi-diagram algorithm with quaternary tree bucketing, *Inform. Proc. Letters* **18**, No. 4, 1984, 227-231.
- [7] M. I. Shamos, *Computational geometry*, Ph.D. dissertation, Yale University, 1978.
- [8] M. I. Shamos and D. Hoey, Closest-point problems, in *Proc. 16<sup>th</sup> Ann. IEEE Symp. on the Foundations of Comp. Sci.*, 1975, pp. 151-162.



### BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER

NISTIR 4716

2. PERFORMING ORGANIZATION REPORT NUMBER

NOVEMBER 1991

3. PUBLICATION DATE

4. TITLE AND SUBTITLE

Computing Delaunay Triangulations for Comet-Shaped Polygons

5. AUTHOR(S)

Javier Bernal

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE  
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY  
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

In this paper, we present two triangulation algorithms which combined produce an algorithm for computing Delaunay triangulations for comet-shaped polygons. The first algorithm constructs in linear time a triangulation for a comet-shaped polygon. The second algorithm constructs a Delaunay triangulation for a polygon from any triangulation for the polygon. The algorithms can be used for deleting vertices in a Delaunay triangulation and for computing constrained Delaunay triangulations.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

algorithm, comet-shaped polygon, computational complexity, computational geometry, constrained Delaunay triangulation, Voronoi diagram

13. AVAILABILITY

UNLIMITED

FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES

18

15. PRICE

A02





