Geometric Structures for Three-Dimensional Shape Representation

JEAN-DANIEL BOISSONNAT
INRIA, France

Different geometric structures are investigated in the context of discrete surface representation. It is shown that minimal representations (i.e., polyhedra) can be provided by a surface-based method using nearest neighbors structures or by a volume-based method using the Delaunay triangulation. Both approaches are compared with respect to various criteria, such as space requirements, computation time, constraints on the distribution of the points, facilities for further calculations, and agreement with the actual shape of the object.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*geometrical problems and computations*; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*representations, data structures, and transforms*; I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling—*curve, surface, solid, and object representations*

General Terms: Algorithms

Additional Key Words and Phrases: Delaunay triangulation, *k*-D tree, polyhedra

1. INTRODUCTION

This paper is concerned with computational structures between points lying on the boundary of a three-dimensional object. Our motivation comes from a number of problems in pattern recognition, computer vision, and graphics, where objects are known by the three coordinates of a set of points selected on the boundary of the object. The list of their coordinates is a poor representation of the object. A structure on the set of points is needed in order to make explicit the proximity relationships between points on the surface of the object. Such structures have been used to solve many problems, including definition of the shape of the object [18, 22], control of the automatic machining of surfaces [4], smooth interpolation between the points [6] or, contrariwise, reduction of the number of points without greatly damaging the actual shape of the object [5], and calculation of geometrical properties, such as area, volume, axes of inertia, definition of the normals of the surface at the points, and extraction of elementary shapes.

Author's address: INRIA, Domaine de Voluceau, Rocquencourt, B. P. 105-78150 Le Chesnay Cedex, France.

There are situations in which a partial structure between the points is explicitly given by the way the points are measured. Such is the case for an image where every pixel can be easily related to its neighbors. Such is also the case for three-dimensional data when the measured points lie on specific curves [8, 16]. In these cases, appropriate solutions can be found, but there are still some restrictions. For example, the approach used in [8] to solve the important case of points lying on parallel slices is restricted to slices composed only of a single connected contour. Furthermore, the a priori structure imposed by the way the points are measured does not correspond necessarily to the actual structure imposed by the metric of the surface. These criticisms become crucial when the points are measured in some more general way, perhaps from different points of view, using several sensors, or during a controlled motion of the object and/or the sensor. Each set of data gives us partial information concerning the structure, say a view of one face, a slice, or a more complex section. The problem remains to complete and integrate this information into a single description of the object as a whole. But since the means necessary to accomplish this task differ so much from one installation to the next, and because the task may, in fact, be an extremely complex one in many typical installations, we propose to take the point of view that *no a priori structure exists*. We feel this approach has the advantages of simplicity and general applicability.

The general problem can be formulated in the following way: We want to represent and make calculations on a three-dimensional shape whose boundary is a surface S on which a set M of N points $M_1 \cdots M_N$ are known by their three coordinates. To do so, we have to create some relations between the points, constituting what we call a structure that is precisely defined as a graph whose vertices are the given points M and whose edges join points that are related in some sense.

Among the different structures, the minimal ones (i.e., those with the fewest edges and that preserve the topology of the surface) play an important part; it is the first purpose of this paper to produce such minimal structures. A minimal structure is the graph of a 3-polyhedron with the measured points as vertices. In the general case no four points are coplanar; such a polyhedron is simplicial, that is, triangularly faceted. Of course such a polyhedron is not unique, and the characterization of a polyhedron that suitably approximates the initial surface is not easy. Among the possible solutions, O'Rourke [13] suggested polyhedra of minimum area, but this criterion may yield strange results, as is shown in Figure 1. Furthermore, there exists no good algorithm that computes polyhedra of minimum area. Other criteria related to the curvature of the surface may be preferred, as in [2]. In addition to the geometrical difficulty, there is a combinatorial one. In R^3 a general simplicial polyhedron is defined as a collection T of triangles (a triangulation) satisfying the following three conditions [9]:

(1) Two triangles are either disjoint, or have one vertex in common, or have two vertices and consequently the entire edge joining them in common.
(2) T is connected.
(3) For every vertex V of a triangle of T, the edges opposite V in the triangles of T having V as a vertex form a simple polygon (see Figure 2).

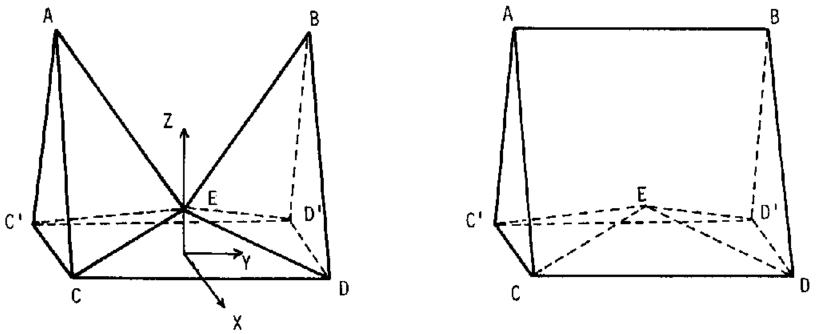


Fig. 1. $A(0, -1, 1)$, $B(0, 1, 1)$, $C(x, -1, 0)$, $C'(-x, -1, 0)$, $D(x, 1, 0)$, $D'(-x, 1, 0)$, $E(0, 0, h)$. If x and h are sufficiently small, the area of the polyhedron on the left is smaller than the area of the polyhedron on the right.

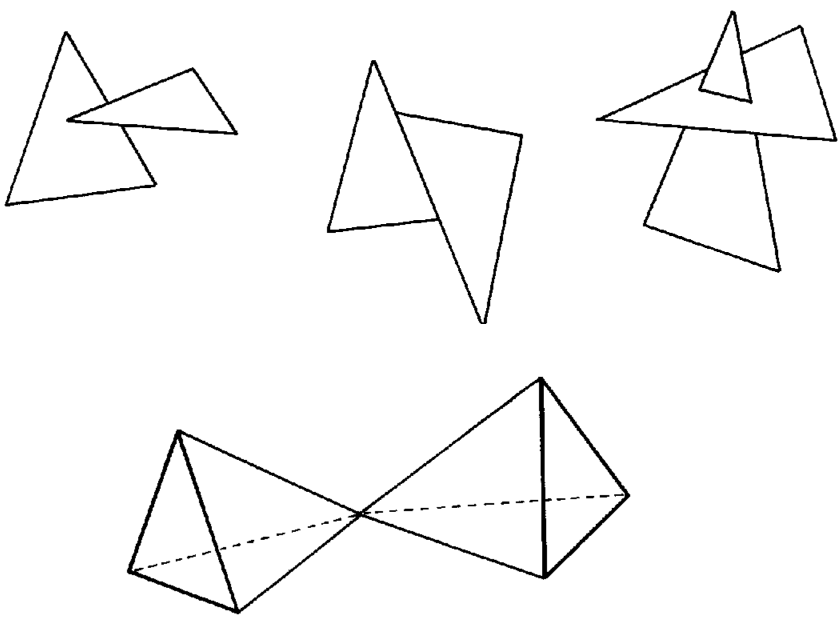


Fig. 2. Some configurations that are not allowed.

Clearly, making all the possible combinations is not feasible.This paper proposes two ways of reducing the complexity of the problem. The first idea is to use explicitly the fact that the points lie on a surface that is known to be, at least locally, diffeomorphic to R^2. If we can exhibit such a diffeomorphism, we unfold the surface and reduce the dimension of the space in which we are working. Such an idea is used by many authors, especially geographers, when they project their three-dimensional data onto a plane; but this has to be done injectively, which is impossible in many cases, as, for example, for closed surfaces. This idea has also been used for closed surfaces when we know, apart from the three coordinates of the points, some local properties of the surface at these points, namely, the normals and the sign of the Gaussian curvature [2]. We consider here the general case, in which only the three coordinates of the points are known. The above idea can be used again locally. This is shown in Section 2.

A second idea is to use a global structure on the points, from which minimal representations and other features can easily be obtained. In Section 3 the well-known Delaunay triangulation is considered in that respect.

Advantages and disadvantages of both methods are discussed in relation to different criteria, such as space requirement, computation time, constraints on the distribution of the points, facilities for further calculations, and correspondence with the actual shape of the object. The performances of both methods must be compared with the minimal requirements needed for such a problem. The number of vertices N, edges E, faces F, and handles H of a polyhedron are related by the Euler formula, which states that $N - E + P = 2 - 2H$ [9]. It follows that a simplicial polyhedron with H handles has $2N - 4 + 4H$ faces. A solution to the triangulation problem consists of a set of the faces, and thus the minimal storage requirement is $O(N)$. A lower bound to computing time is given by the convex case, which, in turn, can be reduced in $O(N)$ additional operations to the planar case by means of a stereographic projection (i.e., a projection of the convex body onto a plane containing one of its faces from the point of the body most distant from that plane). The complexity of a planar triangulation is $O(N \log N)$ [17], so the complexity of any algorithm constructing a triangulation is at least $O(N \log N)$.

2. SURFACE-BASED APPROACH

2.1 Theoretical Background

The aim of this section is to build a triangulation of the surface S by means of a local procedure. We proposed to make use, in the neighborhood of a point M, of the orthogonal projection p onto *the tangent plane* P of S at M. The proposition below gives the size of a domain in which this projection is a diffeomorphism; so, for that domain, triangulation in P provides a triangulation of S. The proofs in this section come from differential geometry. They are omitted here.

PROPOSITION 1. *Let S be a smooth surface in three-dimensional (3-D) space whose principal radii of curvature exceed R at every point. Let p denote the orthogonal projection on a tangent plane P of S at M. Then there exists an open set U of S such that p is a diffeomorphism from U onto any disk lying in P whose center is M and whose radius is smaller than R* (see Figure 3).

It must be noted that the hypothesis implies that $p(U)$ cannot fold over itself, and so a triangulation with straight edges in P will correspond to a triangulation with straight edges on S.

Moreover, we can control the validity of the method as is claimed by

PROPOSITION 2. *If every point of S is nearer than e from a point of the triangular mesh, the projection p allows the construction of a triangulated surface which approaches the region U of S in the following sense: p does not move the points of S more than e^2/R.*

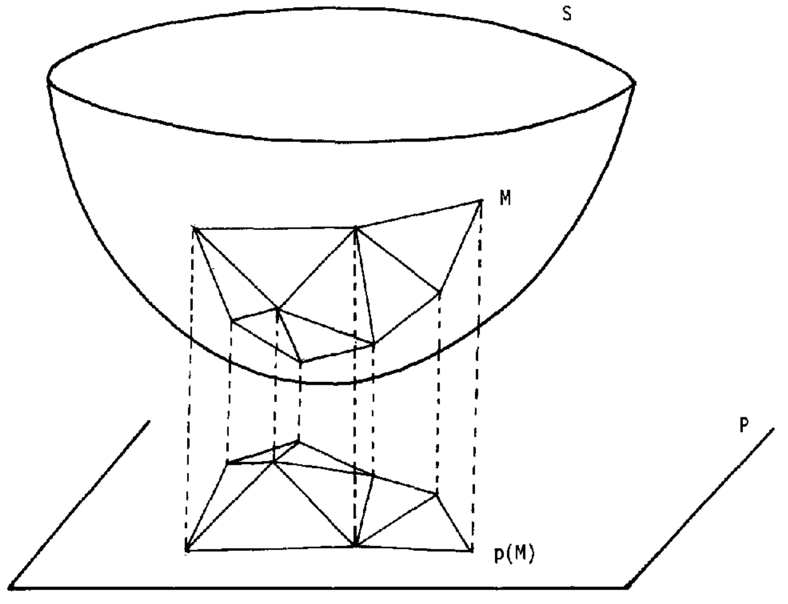


Fig. 3.

If we want to control this homeomorphism at the first order (for instance, if we want lengths and areas not to be greatly changed), we require that the constructed triangles not be too thin. In general, this can be achieved by choosing a sufficiently regular triangulation, such as the Delaunay triangulation, which builds the most equiangular triangulation [9].

Several remarks of importance for the sequel must be made.

(1) The propositions above guarantee a triangulation of the surface if the discretization is fine enough. More precisely, in the neighborhood of a point, the discretization must be finer than the smallest radius of curvature at this point. It must be noted that not even a rough approximation can be guaranteed if the number of points is not sufficiently large.

(2) The method is not canonical: In particular, when applied to two neighboring points, it does not necessarily give the same result. Some care will be required in the implementation.

(3) Although the method can approximate first-order quantities, it cannot preserve, in general, the second-order quantities. An example is given in Figure 4, where the local convexity of the surface (defined by the sign of the Gaussian curvature of the surface) is lost.

2.2 Implementation

An algorithm looking for a triangulation of the surface in the previous manner must successively achieve

(1) the definition, for each point, of its neighbors;
(2) the initialization of the triangulation process;

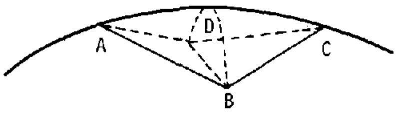


Fig. 4.

(3) the growth of the triangulated domain by successively adding new points to the mesh.

Before the triangulation process starts, the neighborhood of each point is defined as the set of its k nearest neighbors. For the neighborhood of a point such a definition allows nonuniform density of points and efficient computation. Finding the k nearest neighbors is a classical problem for which fast solutions have been proposed. One such solution makes use of a data structure called the k-D tree. In [7] it is shown that the time and space required to build a k-D tree are, respectively, $O(N \log N)$ and $O(N)$. The expected k-nearest-neighbors search time is shown to be $O(k \log N)$, *which is* $O(\log N)$ if k is independent of N.

Then the triangulation process can start. At each step of the process three entities are updated:

(1) the triangulated domain D, which is a set of the previously created triangles;
(2) the contour C of the triangulated domain, which is a doubly linked list of nodes—each node M_i of C is defined by its label M_i, the labels of the previous and the following nodes on the contour, pM and fM, and the label iM of the point inside the triangulated domain, which makes a triangle with M_i and fM; such a structure for the contour allows it to be modified very easily when creating a new triangle and allows us to calculate, for each edge M_i–fM of C, an approximation of the tangent plane around that edge, the plane of the triangle M_i–fM–iM;
(3) the set O of the points that are not yet inside the triangulated domain.

The initialization is performed by defining a first edge which joins a point I_0 and its nearest neighbor I_1. The corresponding contour C is composed of the two edges I_0–I_1 and I_1–I_0. The points iI_0 and iI_1 are taken in the approximate tangent plane, defined by a least square method applied in the neighborhood of I_0–I_1. Then the triangulation is developed by propagating C. This propagation is done by looking around an edge E of C so that point M_k of O in the neighborhood of E must be taken into account in creating a new triangle. The choice of that best point M_k is done in the approximate tangent plane by choosing the point such that $p(M_k)$ sees $p(E)$ under the largest angle. Then we add M_k to C and the triangle E–M_k to D. Eventually O is updated, and the process continues.

Owing to the Euler formula, the storage requirements for D, C, and O are of size $O(N)$, and so the total requirement is also of size $O(N)$. Because the procedure is a local one and because, at each step, we build a new triangle, the complexity of the method is $O(N)$ if the neighborhood of each point is known. If we use a k-D tree to define such neighborhoods, the total complexity is $O(N \log N)$.

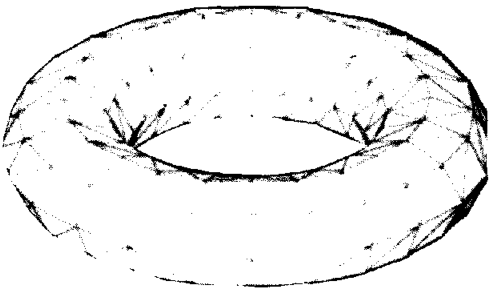


Fig. 5.

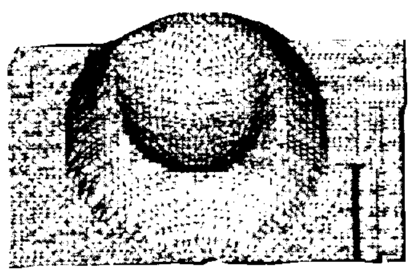


Fig. 6.

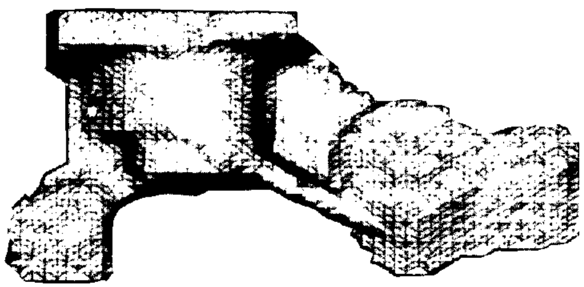


Fig. 7.

2.3 Validity of the Method

According to the theoretical results, the validity of the method is ensured, and so a polyhedral surface is obtained if the neighborhood of each edge is isotropic and does not fold over itself when projected onto the tangent plane. These two conditions may give rise to incompatible choices for the k number of neighbors. Indeed, the first condition requires k to be sufficiently large, whereas the second requires k to be sufficiently small. This difficulty comes, in part, from the fact that the Euclidean distance does not always suitably approximate the metric of the surface. A way of bypassing this difficulty, if a great number of points are available, is to take a rather large value of k and eliminate those neighbors that stand higher from the tangent plane than a given value.

Another difficulty comes from the fact that the method is not canonical. We have to ensure that the triangles do not cross one another. This is done by testing around each vertex of a new triangle to determine whether its edges actually lie outside D. This is readily performed owing to the doubly linked structure of C.

An implementation of this algorithm has been written in Pascal and has run on a number of well-sampled objects. A triangulation of a torus is shown in Figure 5. Figures 6 and 7 show results on real data provided by a laser range finder.

3. VOLUME-BASED APPROACH

3.1 Theoretical Background

In the previous section a polyhedral representation was obtained by pruning an initial graph, defining for each point its nearest neighbors using the Euclidean distance. The discretization might satisfy some constraints, giving rise, in the case of complex objects, to some of the difficulties discussed in Section 2. In order to avoid these difficulties, we use the Delaunay triangulation of the points M as an intermediate structure. It will be shown that such a geometrical structure defines neighborhoods that are symmetrical, isotropic, and closely related to the metric of the surface. Moreover, it is a global structure that will allow the extraction, in any case, of a minimal representation of the shape. The result will be a better and better approximation of the shape as the number of points increases.

Let us recall the definition of the Delaunay triangulation and some general results. More details can be found in [15], [17], and [19]. For a D-dimensional Euclidean space E and a set M of N points $M_1 \cdots N_N$, the associated Voronoi diagram is a sequence $(V_1 \cdots V_N)$ of convex polyhedra covering E, where V_i consists of all the points of E that have M_i as a nearest point in the set M. Thus

$$V_i = \{P \in E : \forall j, 1 \leq j \leq N, d(P, M_i) \leq d(P, M_j)\}$$

where d denotes the Euclidean distance.

The geometrical dual of the Voronoi diagram, obtained by linking the points M_i whose Voronoi polyhedra are adjacent, is called the Delaunay triangulation of M. Although the dual graph is usually considered an abstract graph expressing the topological relationships of the Voronoi polyhedra, our concern here is to take the joins to be straight-line segments and to use them as a framework for a simplicial subdivision of space. Figure 8 shows an example of a Voronoi diagram and its dual in a two-dimensional simple case. In the three-dimensional case, it can be shown that two elements are disjoint or have one vertex in common, or that they have two vertices and consequently the entire edge joining them, or that they have three vertices and consequently the entire face joining them. Moreover, the union of the elements of the Delaunay triangulation is equal to the interior of the convex hull of M. Owing to the definition, when no five points are cospherical, the elements of the Delaunay triangulation are tetrahedra, and the circumspheres (the Delaunay spheres) do not contain any point of M in their interior. In the case of more than five cospherical points, the elements can be decomposed into several tetrahedra so that, in every case, the Delaunay triangulation is composed of tetrahedra. Moreover, the Delaunay triangulation associates with each point M_i a set of at least three neighbors M_j, which are, roughly, the nearest neighbors of M_i in the different directions. Indeed, consider an inversion with M_i as center, which associates with a point M_i the point M_i', on the ray M_iM_j, whose distance from M_i satisfies the equation $M_iM_j \cdot M_iM'_j = k^2$. The reciprocal images of the half-spaces limited by the faces of the convex hull of the images $M_1' \cdots M_N'$ of $M_1 \cdots M_N$ are the interiors of the spheres passing through M_i and three other points of M (the corresponding two-dimensional case is shown in Figure 9). Because such half-spaces are empty, the interiors of the

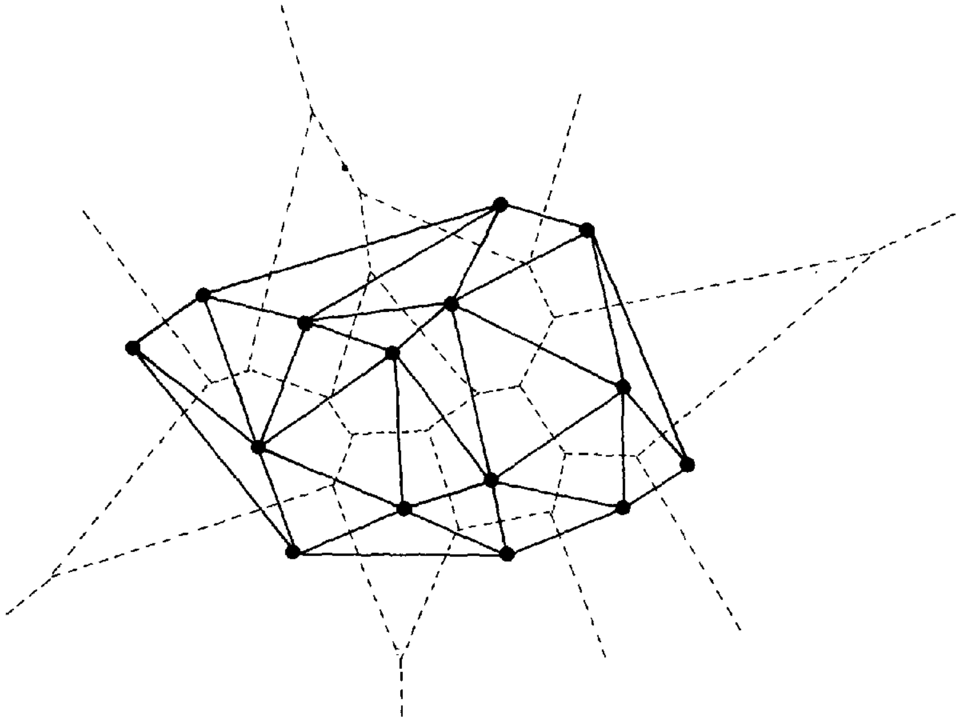


Fig. 8. ●, points M; ——, Delaunay triangulation of M; - - -, Voronoi diagram of M.

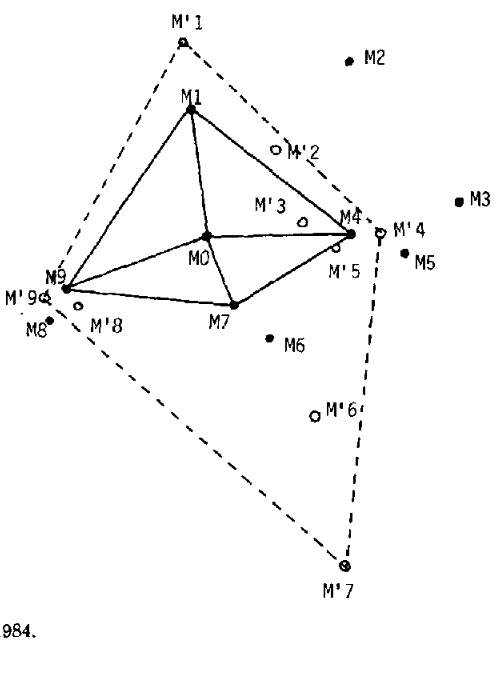


Fig. 9. ●, points M_i; O, images M'_i of the points M_i inverted with respect to a sphere whose center is M_0. - - -, convex hull of the M'_i; ——, triangles of the Delaunay triangulation containing M_0.

corresponding spheres are also empty; moreover, these half-spaces are the only empty half-spaces passing through three points of M, so the spheres are the Delaunay spheres passing through M_i. Thus, the neighbors of M_i are the points M_j, whose images by an inversion with M_i as center are the vertices of the convex hull of the images of $M_1 \cdots M_N$.

So the Delaunay triangulation is a 3-connected graph on M embedded in R^3, which defines symmetrical and isotropic neighborhood relationships between the points. It is to be noted that this is not necessarily the case of the k nearest neighbors or of other geometric structures included in the Delaunay triangulation, such as the minimum spanning tree, the Gabriel graph, or the neighborhood graph. Moreover, the Delaunay triangulation contains polyhedra and satisfies the first and second conditions of the definition of a polyhedron (cf. Section 1), which will appear to be a decided advantage.

One more advantage of the Delaunay triangulation is that it can be computed efficiently. Klee [11] placed a lower bound on the worst case complexity of any algorithm computing the Delaunay triangulation by proving that the number of tetrahedra in the Delaunay triangulation of N points is at most $O(N^2)$. No known algorithm reaches this bound, but an algorithm with an $O(N^2 \log N)$ lower bound can be easily deduced from the above. We have shown that finding the neighbors of a vertex in the Delaunay triangulation of N points is equivalent to computing a convex hull of N points. This can be done in $O(N \log N)$ time, and this is optimal [14]. Computing the entire Delaunay triangulation consists of finding the neighbors of the N vertices or, equivalently, of computing N convex hulls, which can be done in $O(N^2 \log N)$ time.

In addition to the general results mentioned above, we now give some specific properties that are due to the fact that the points M lie on a surface. Let us introduce these properties in the case in which the density of the points M increases indefinitely. In such case, the Delaunay spheres become tangent to the surface S of the object and are by the maximal spheres, whose centers constitute the skeleton of the object. Moreover, we can say something about the curvature: At a convex or parabolic point M, if one of the principal curvatures is sufficiently large, the Delaunay sphere cannot touch another region of S; since it does not contain any point in its interior, one of the radii of the Delaunay spheres passing through M is equal to the smallest radius of curvature. At a saddle point this is true on both sides of the tangent plane: The two radii of the two Delaunay spheres are equal, respectively, to the radii of curvature at M.

Thus we can control to some extent the local behavior of the Delaunay triangulation. Moreover, if we want to extract from the Delaunay triangulation a correct approximation of the surface, it is necessary that the Delaunay triangulation contain a polyhedron that respects the relative locations of the points on the surface. More precisely, this polyhedron must be diffeomorphic to a curved polyhedron tightly stretched on the surface passing through the points M. Owing to the definition, this condition is satisfied as soon as there exists a set of spheres, each of which passes through the vertices of one of the above curved triangles and does not contain any point of M in its interior. It is to be noticed that this condition is not very restrictive; in particular, very thin parts (with respect to the discretization) can be allowed, provided that there is enough free space around. See Figure 10.

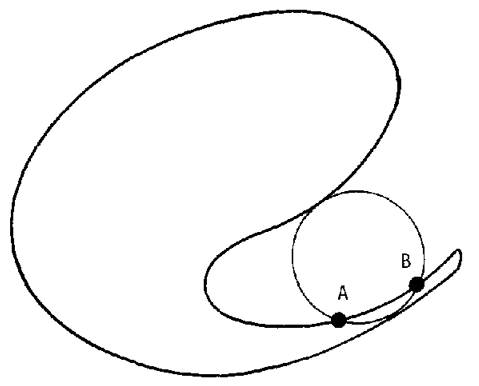


Fig. 10. The edge AB belongs to the Delaunay triangulation of the boundary of the object.

The preceding results show that, under weak assumptions, the Delaunay triangulation contains a polyhedron that suitably approximates the surface of the object. There remains the problem of the extraction of this polyhedron from the Delaunay triangulation. The method of Section 2 can, of course, be used, but the drawbacks of that local method can be avoided here owing to the properties of the Delaunay triangulation. Moreover, instead of purely surface information, the following method directly provides both surface and volumetric representations.

3.2 Sculpture

The Delaunay triangulation fills the interior of the convex hull of points M with tetrahedra. First, let us consider the case in which this convex hull contains all the points M: The Delaunay triangulation is a volumetric representation of the object. The object is represented by a set of tetradedra; the boundary of that set, the convex hull of M, is a polyhedral approximation of the surface of the object, the best possible one according to such criteria as minimum area, minimum variation of curvature, or distance between the polyhedral approximation and the object.

If not all the points are on the convex hull, we must eliminate tetrahedra until all the points M are on the boundary P of the polyhedral shape so obtained. This sculpture of the convex hull can be done sequentially, by eliminating one tetrahedron after another in such a way that, at each step, P satisfies the definition of a polyhedron. It can easily be proved, by looking at all the possible configurations (see Figure 11), that this is guaranteed when the following rule is respected.

Rule. The only tetrahedra that can be eliminated are those with exactly one face, three edges and three points on P, or those with exactly two faces, five edges and four points on P.

It can be proved that any polyhedron of genus O inside the Delaunay triangulation can be obtained by such a procedure.

At each step of the sculpture, the set S of the noneliminated tetrahedra is stored as a set of 4-tuples $T_i = (M_{1i}, M_{2i}, M_{3i}, M_{4i})$ with an adjacency graph that, for each tetrahedron, gives its adjacent tetrahedra in S $(T_{1i}, T_{2i}, T_{3i}, T_{4i})$. T_{ji} is

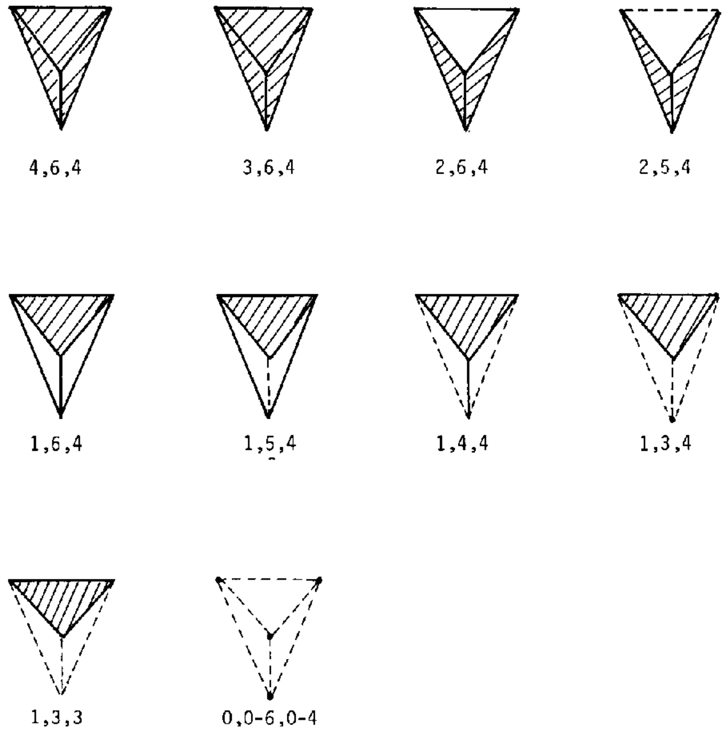


Fig. 11. The rule of Section 3.2 allows the elimination of tetrahedra of types (d) and (i) and forbids the elimination of types (b), (c), and (e)–(h); otherwise, a point will be isolated (b) or the surface will intersect itself at one point (h) or along an edge (c), (e)–(g). The tetrahedra of type (j) are not considered. The tetrahedra of type (a) cannot occur because the tetrahedra of type (b) cannot be eliminated. The numbers are, respectively, the number of faces, edges, and points on P.

the tetrahedron opposite M_{ji}; that is, with T_i it shares the face whose vertices are the M_{ki}, $k \neq j$. As a convention, $T_{ji} = 0$ if the face whose vertices are the M_{ki}, $k \neq j$, is a face of the boundary P of S. A value is associated with each tetrahedron of S having at least one face on P. This value is used to sort the tetrahedra. The tetrahedra with the largest values are eliminated first. The choice of a criterion defining these values may depend on the application. We propose to associate here with the tetrahedron T_i, the value $V(T_i)$ defined as the maximum distance between the faces of T_i on P, and the associated parts of the circumscribed sphere of T_i. When the density of points on the boundary of the object is sufficiently large, the value of any tetrahedron having a least one face on P and belonging to the interior of the object is smaller than the value of any tetrahedron belonging to the exterior of the object.

It has to be noted that the elimination of several tetrahedra may only add one point to the boundary P of the polyhedral shape. For example, in Figure 12, I is added to P after the elimination of the two tetrahedra $ABCI$ and $BCDI$. So the algorithm cannot simply stop when all the points lie on P. Let us call V the maximum value of $V(T_i)$ for all the tetrahedra inside P. V measures the goodness

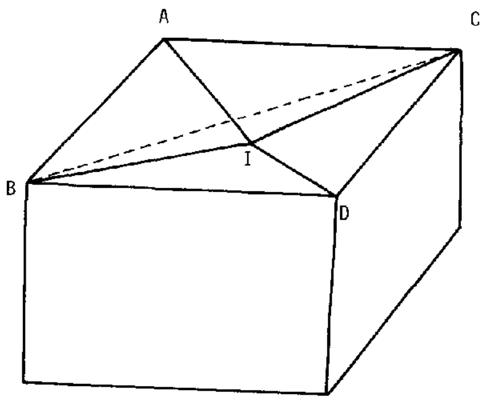


Fig. 12.

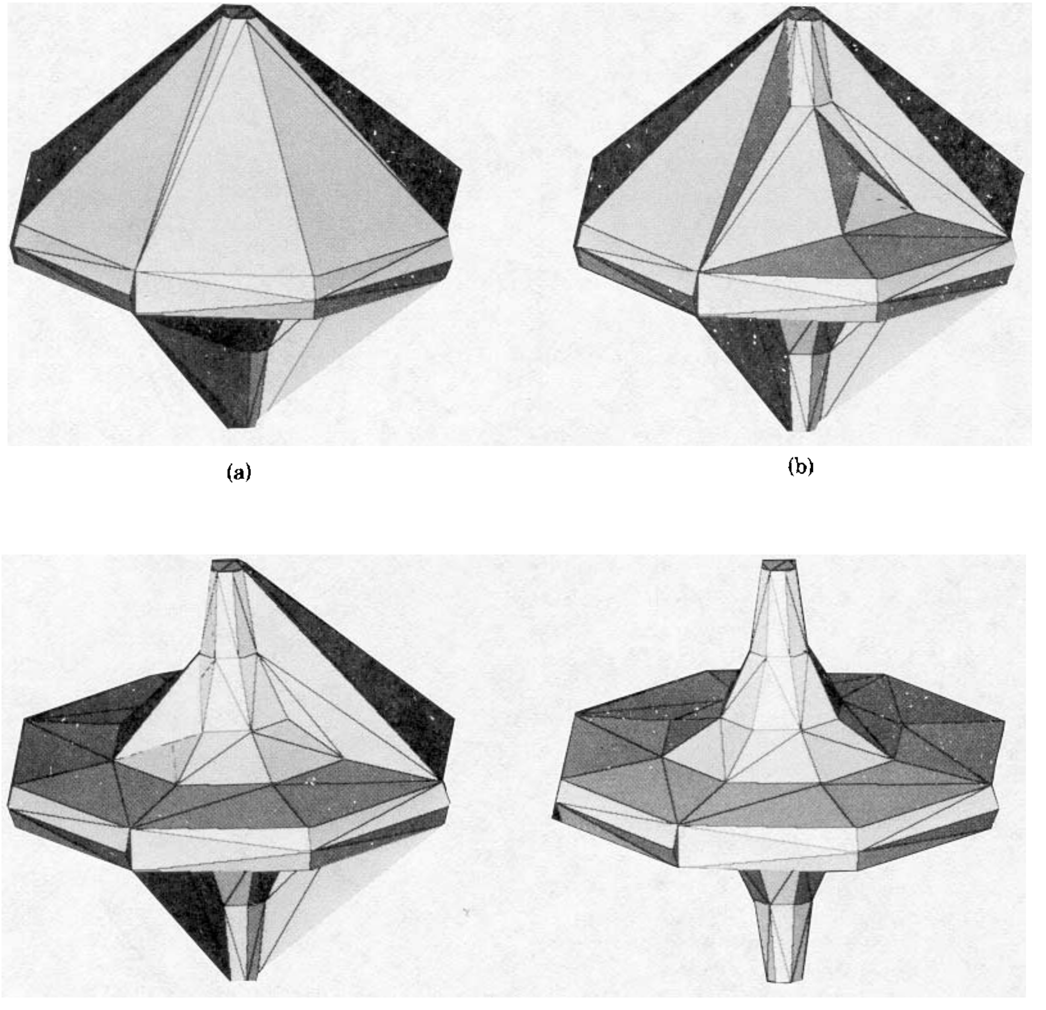
of the approximation of P by the set of the Delaunay spheres corresponding to the interior tetrahedra. In our implementation the algorithm stops when no point of M can be added to P and no tetrahedron can be eliminated without decreasing V. Owing to the remark above, this guarantees that the algorithm produces a correct polyhedral approximation when the density of points is sufficiently large.

At the end of the sculpture, the object is represented by the set S of the noneliminated tetrahedra. The boundary P of this set is the polyhedral approximation of the surface of the object.

The algorithm is roughly described as follows:

(1) $P \leftarrow$ convex hull; $S \leftarrow$ Delaunay triangulation; mark the points M_i and the edges M_iM_j of the convex hull.
(2) Build a heap containing the tetrahedra having a face on P, sorted according to the criterion.
(3) While the rule is not violated,
  (a) consider the tetrahedron T and S having the largest value—if all the measured points are not yet on P or if the removal of T causes V to decrease, then remove T;
  (b) evaluate the neighbors N_i of T, insert them into the heap, and replace T by 0 in the set of neighbors of the N_i;
  (c) update the sets of vertices and edges of P.

Let us now analyze the complexity of the algorithm. In Section 3.1 we have proposed an algorithm computing the Delaunay triangulation whose worst case complexity is $O(N^2 \log N)$. Other algorithms for computing the Delaunay triangulation of a set of N points in any dimension have been proposed in the literature [3, 10, 20]. Although in the three-dimensional case the number of tetrahedra in the Delaunay triangulation may be $O(N^2)$ [11], no such case has been experienced for fairly regular distributions of points on a closed surface. In the cases of practical interest, when fast nearest neighbor search techniques are used [1, 3], the algorithms appear to be almost linear—step (1). In order to update the set of the tetrahedra of S rapidly, we arrange them as a heap [21]. To arrange N elements as a heap takes $O(N \log N)$ time and to update the heap after each elimination of a tetrahedron takes $O(\log N)$ time. If Q is the number of points on the convex hull of M, the number of tetrahedra having a face on the



(a) (b)

(c) (d)

Fig. 13.

convex hull is $O(Q)$, and it takes $O(Q \log Q)$ time to build the initial heap, which is step (2). In almost every case, when adding a new point to P, the number of the eliminated tetrahedra is independent of N, so step (3) is scanned $O(N)$ times. If we use the data structure described above, testing whether the rules are violated and steps (3b) and (3c) require only constant time; so it takes $O(\log(Q+1)) + \cdots + O(\log N)$ time to take into account the $(N - Q)$ points not belonging to the convex hull, which is less than $O(N \log N)$. In the very exceptional case in which we may have to eliminate N tetrahedra when adding a new point to P, the total complexity remains lower than $O(N^2 \log N)$.

An implementation of this algorithm has been written in FORTRAN. Figures 13 and 14 show results on two synthetic objects. Figure 15 shows a result for real data provided by a laser range finder. In each case several steps of the sculpture

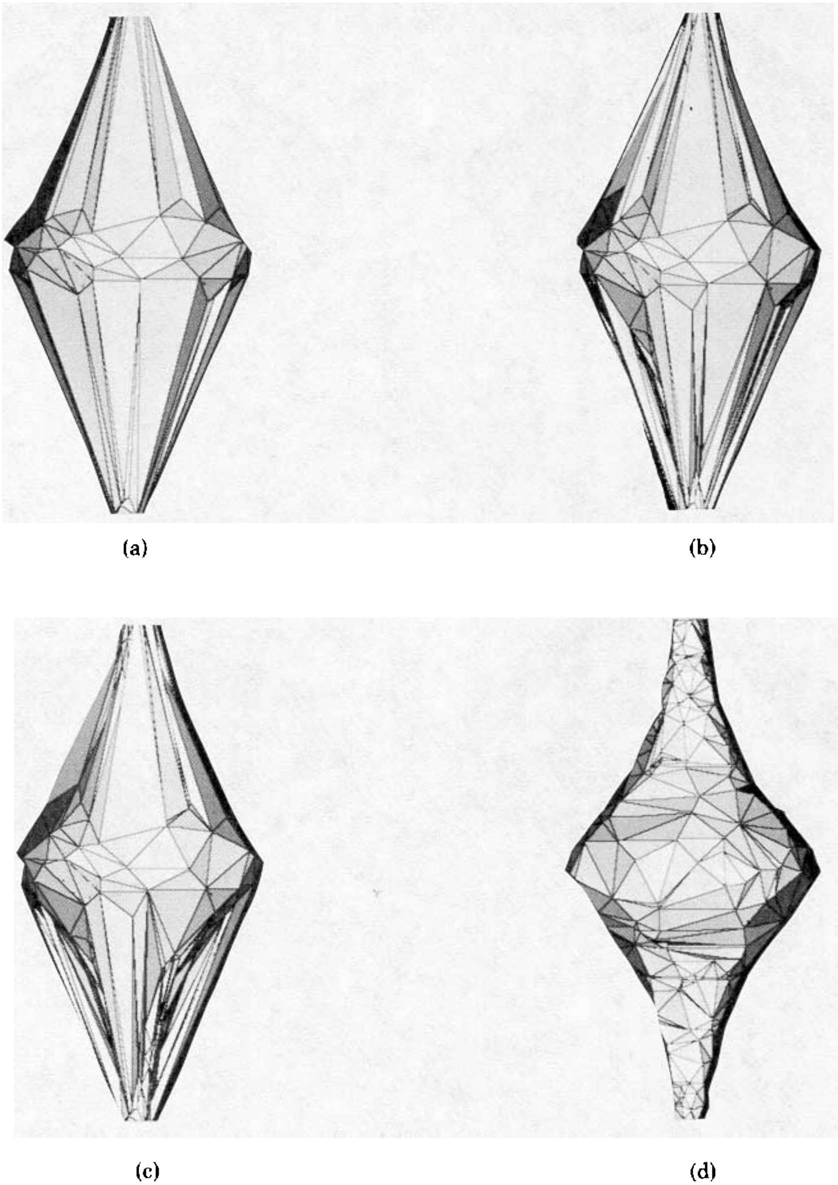


Fig. 14.

are shown. The hidden lines being removed, the points interior to the convex hull appear progressively on the boundary.

3.3 Applications and Extensions

3.3.1 *Automatic Modeling of Three-Dimensional Objects.* The above procedure requires only weak restrictions of the discretization and so is general and robust. It can be used as an automatic modeler able to handle complex objects. It gives a respresentaion of the volume of an object (a set of tetrahedra) and also a

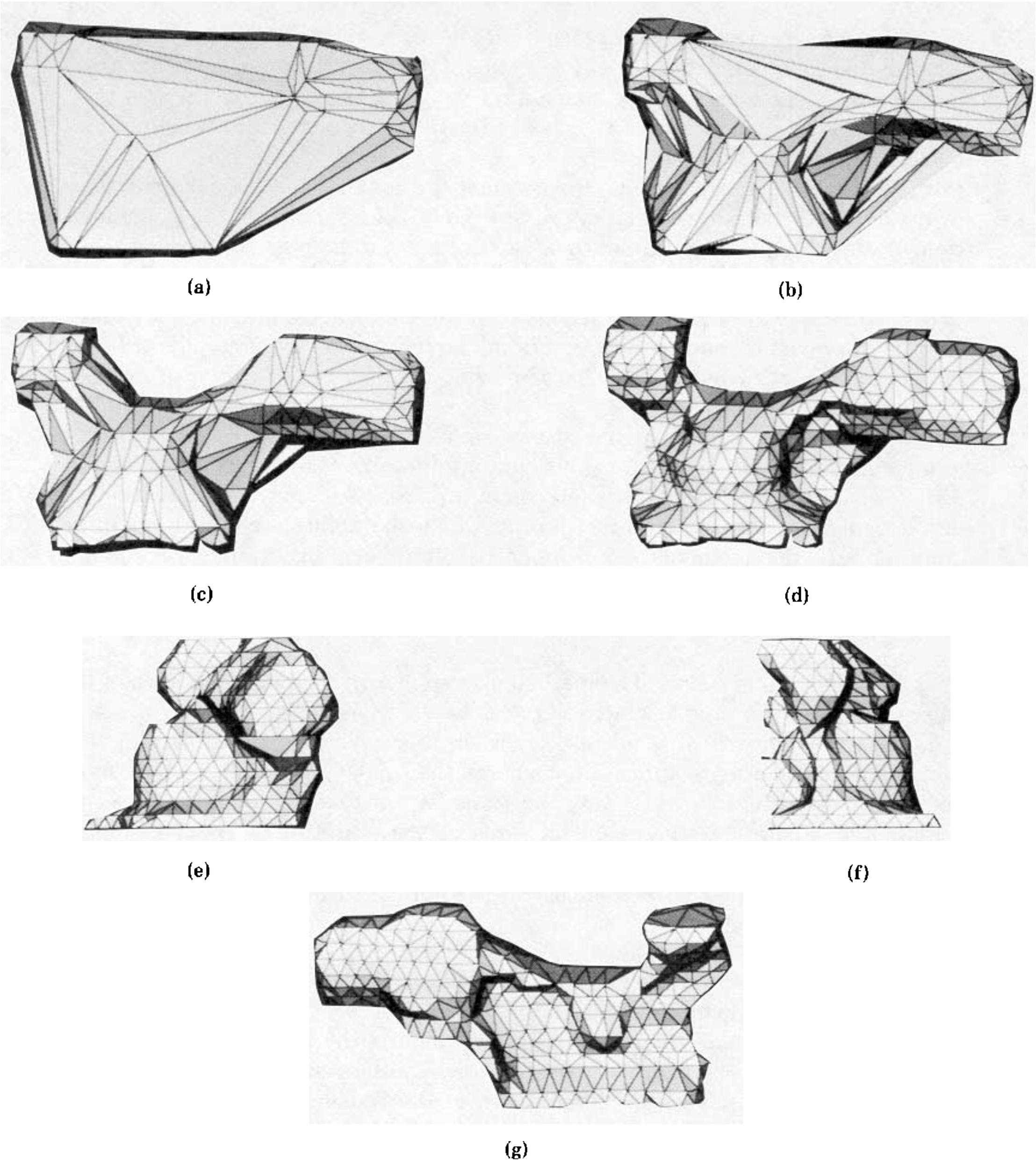


Fig. 15.

representation of the surface of this object (a polyhedron, which is the boundary of the set of tetrahedra). Moreover a by-product of the method is the convex hull of the approximated shape of the object. When we use the Delaunay triangulation and the so-called "sculpture," several applications become straightforward. We just mention some of them.

(1) The mass properties (volume, center of mass, moments of inertia, etc.) can be calculated by looking at the mass properties of the set of the interior tetrahedra. The volume is the sum of these elementary volumes, the center of mass is the center of gravity of the centers of mass of the different tetrahedra weighted by their volume, etc.

(2) The equilibrium positions are obtained by looking for the faces of the convex hull of the object, which is approximated by the boundary of the Delaunay triangulation; the faces contain in their interior the normal projection onto them of the center of mass.

(3) The interior tetrahedra constitute a mesh, which can be used to perform stress and thermal analyses by means of finite-element techniques. This mesh can be improved by adding new points in the interior of the object in order to obtain more regular tetrahedra. These new points may be the center of mass of the overelongated tetrahedra.

(4) Cruder polyhedral approximations can be performed by eliminating points producing elongated tetrahedra. This can be done very easily if, as is the case in [3], [10], and [20], the algorithm computing the Delaunay triangulation is implemented as an iterative procedure in which the points are added one after another and the triangulation is updated after each insertion. Only points significantly contributing to the shape of the object can be retained, thus reducing the storage requirements.

3.3.2 *Other Dimensions.* The method also applies in the planar case when the points belong to a simple closed curve. The shape is represented by a set of triangles (2-simplices) instead of tetrahedra (3-simplices) in the three-dimensional case. The only modification concerns the rule of Section 3.2, which must be replaced by the following: only the triangles with two vertices on P can be eliminated. Figure 16 shows several steps of the sculpture of a set of points belonging to a planar contour. Another example is shown in Figure 17. Although we have not tried to do so, there is no theoretical difficulty in applying the method to higher dimensions.

3.3.3 *Skeleton.* Even in the two-dimensional case, computing the skeleton of a polyhedron is a difficult task [12]; the above method provides a polyhedral approximation of the skeleton of an object which is the subset of the Voronoi diagram inside the polyhedral shape provided by the sculpture. This approximation, which is the skeleton of the union of the Delaunay spheres associated with the simplices interior to the shape, converges toward the actual skeleton when the density of points increases. The skeletons of the objects of Figures 16 and 13 are shown in Figures 18 and 19, respectively. Related applications include computation of the length of an object and decomposition of an object into convex or pseudoconvex parts.

3.3.4 *Shape Hull.* Finally, let us mention another extension: the definition of the shape hull of a dot pattern. This can be done by a procedure analogous to the "sculpture" procedure. The only difference consists of stopping the process when the criterion reaches a given value.

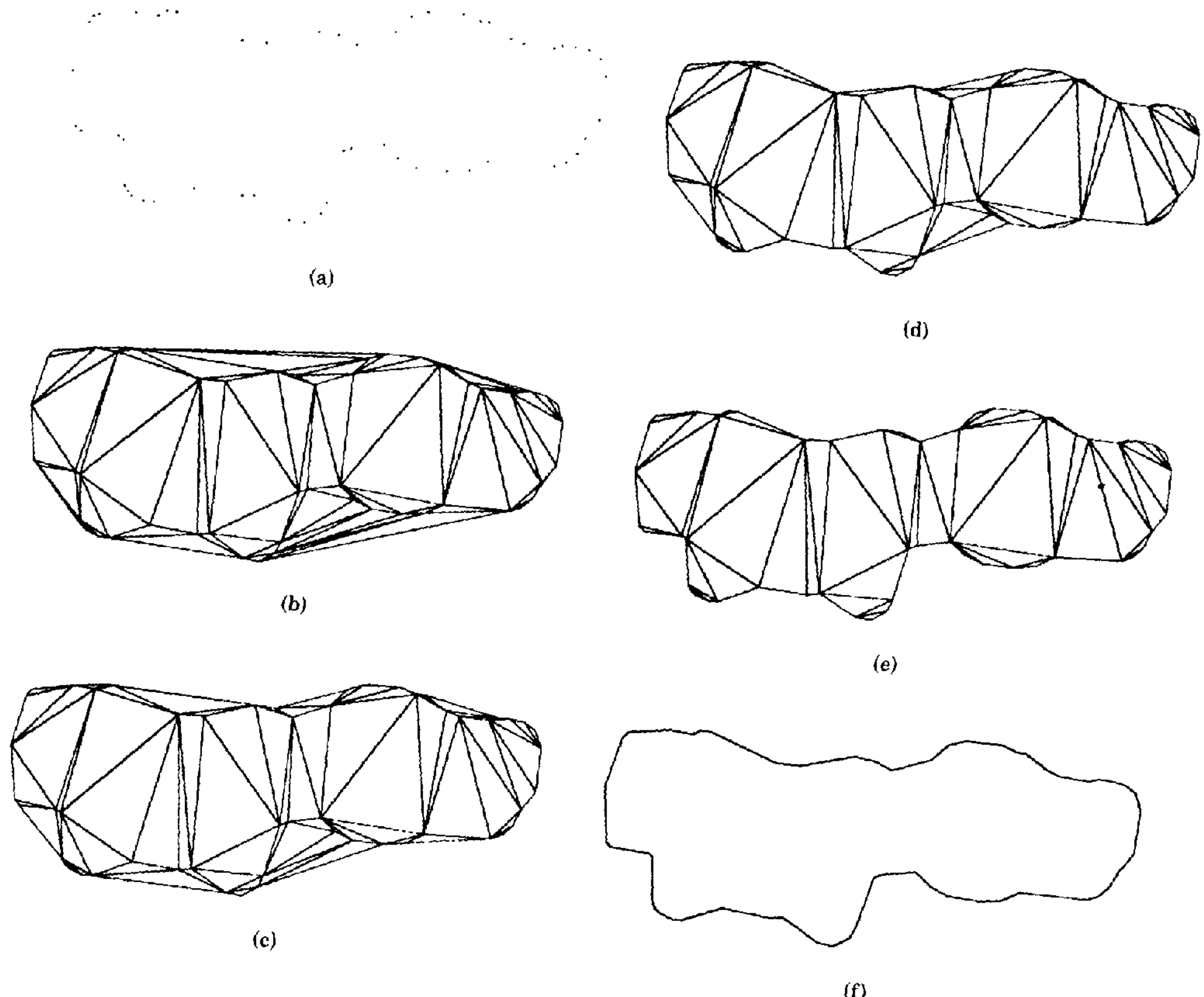


Fig. 16.

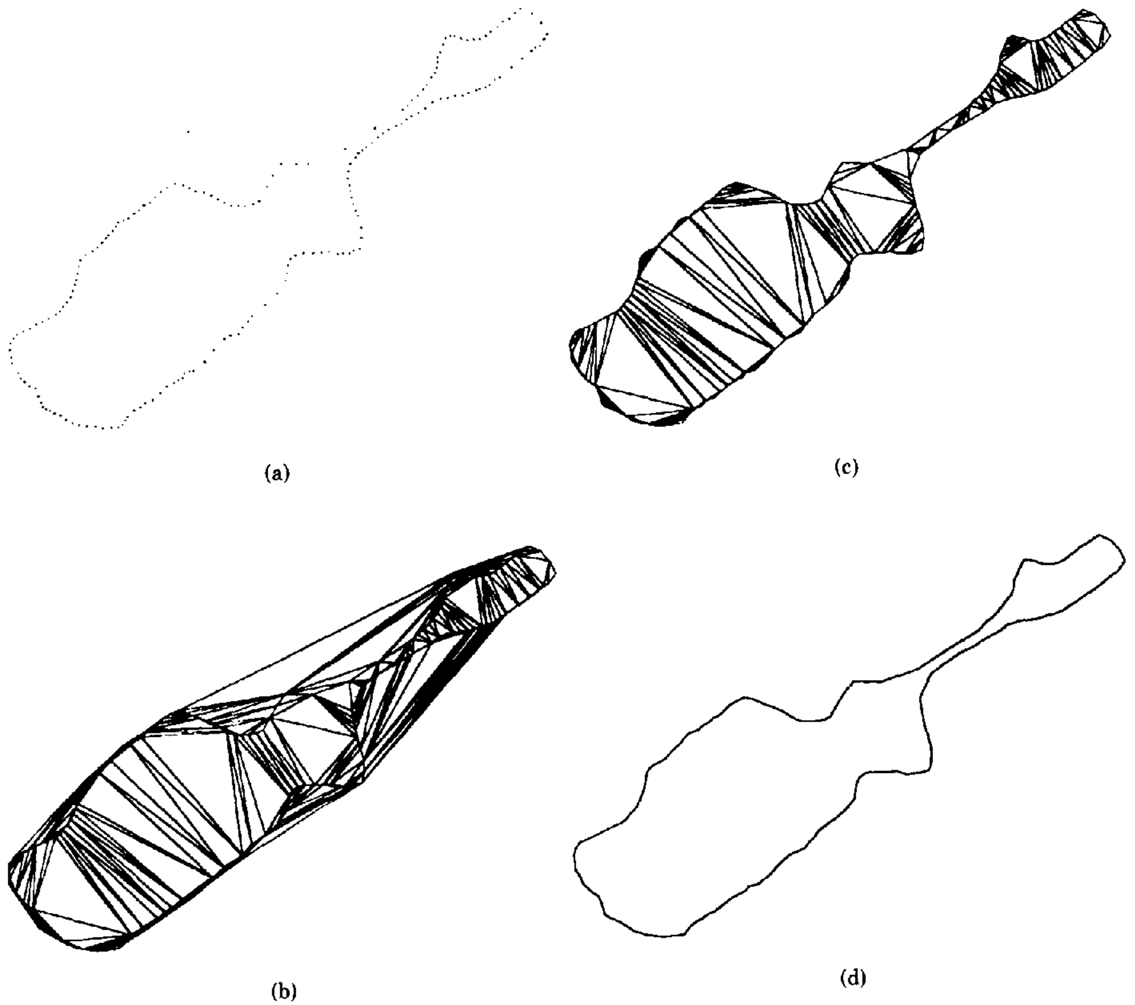


Fig. 17.

This paper is an initial exploration of the numerous applications of the Delaunay triangulation in the field of the computational geometry of surfaces. Of course, much work remains to be done and we just mention three unsolved problems:

(1) Given a set M of N points, does there exist a criterion for sorting the set of all the polyhedra having M as vertices such that relative to this criterion, the minimal polyhedron is sure to be one contained in the Delaunay triangulation?

(2) Is the number of tetrahedra smaller than $O(N^2)$ if the points are fairly regularly distributed on a closed surface?

(3) The procedure "sculpture" produces a simple polygon in the plane, a simple polyhedron in three-dimensional space passing through N points, and, therefore, can be considered a heuristic for the Traveling Salesman problem and its

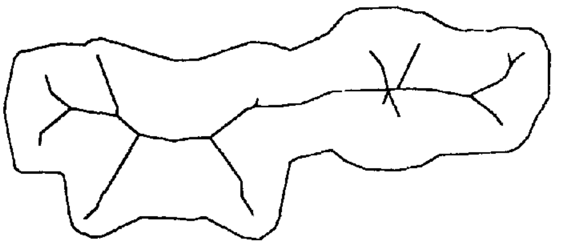


Fig. 18.

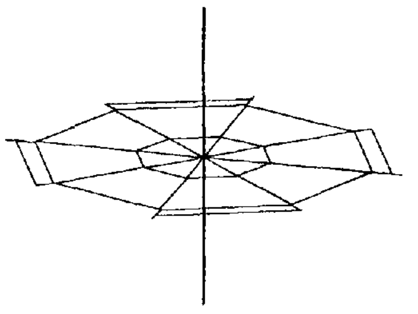


Fig. 19.

generalization in three-dimensional space. Do there exist bounds to the behavior of such a heuristic?

4. CONCLUSION

Two main approaches have been proposed for representing three-dimensional shapes defined by a set of N points on their boundary. The first uses some results in differential geometry and takes advantage of the fact that a surface in R^3 is essentially two dimensional. Such a method only requires storage proportional to $O(N)$ and, when using a k-D tree, time proportional to $O(N \log N)$. Moreover, theoretical results guarantee the quality of the approximation. But the approach is, by nature, local and can only be applied if the discretization is fine enough. The second approach takes into account the discrete nature of the problem and uses a global data structure, famous in the field of computational geometry: the Delaunay triangulation. This method is more costly in the worst case, $O(N^2 \log N)$, but is general and makes many applications straightforward.

ACKNOWLEDGMENT

The author would like to thank Dr. Henry Crapo for his thoughtful comments on this paper.

REFERENCES

1. BENTLEY, J. L., WEIDE, B. W., AND YAO, A. C. Optimal expected-time algorithms for closest point problems. *ACM Trans. Math. Softw. 6*, 4 (Dec. 1980), 563–580.
2. BOISSONNAT, J. D. Representation of objects by triangulating points in 3-D space. In *ICPR 82* (Munich). 1982, IEEE, Silver Springs, CA. pp. 830–832.

3. BOWYER, A. Computing Dirichlet tesselations. *Comput. J. 24*, 2 (May 1981), 162–166.
4. DUNCAN, J. P., AND MAIR, S. G. *Sculptured Surfaces in Engineering and Medicine*. Cambridge University Press, Cambridge, England, 1983.
5. FAUGERAS, O. D., HEBERT, M., MUSSI, P., AND BOISSONNAT, J. D. Polyhedral approximation of 3-D objects without holes. *Comput. Graph. Image Proc. 25* (Feb. 1984), 169–183.
6. FRANKE, R. Scattered data interpolation: Test of some methods. *Math. Comput. 38*, 157 (Jan. 1982), 181–200.
7. FRIEDMAN, J. H., BENTLEY, J. L., AND FINKEL, R. A. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw. 3*, 3 (Sept. 1977), 209–226.
8. FUCHS, H., KEDEM, Z., AND USELTON, S. P., Optimal surface reconstruction from planar contours. *Commun. ACM 20*, 10 (Oct. 1977), 693–702.
9. GIBLIN, P. J. *Graphs, Surface and Homology*. Chapman and Hall, London, England, 1977.
10. HERMELINE P. Triangulation automatique d'un polyedre en dimension *N. R.A.I.R.O. Anal. Numer. 16*, 3 (1982), 211–242.
11. KLEE, V. On the complexity of *d*-dimensional Voronoi diagrams. *Arch. Math. 34* (1980), 75–80.
12. LEE, D. T. Medial axis transformation of a planar shape. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-4*, 4 (July 1982), 363–369.
13. O'ROURKE, J. Triangulation of minimal area as 3-D object models. In *Proceedings of the International Joint Conference on Artificial Intelligence 81* (Vancouver, Canada). 1981, pp. 664–666.
14. PREPARATA, P. P., AND HONG, S. J. Convex hull of finite sets of points in two and three dimension. *Commun. ACM 20*, 2 (Feb. 1977), 87–93.
15. ROGERS, C. A. *Packing and Covering*. Cambridge University Press, Cambridge, England, 1964.
16. SATO, Y., KITAGAWAS, H., AND FUJITA, H. Shape measurement of curved objects using multiple slit-ray projections. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI 4*, 6 (Nov. 1982).
17. SHAMOS, M. I. Computational geometry. Ph.D. dissertation, Yale University, New Haven, Conn., 1978.
18. SHAPIRO, L. G., AND HARALICK, R. M. Decomposition of two-dimensional shapes by graph-theoretical clustering. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-3* (Jan. 1979), 10–20.
19. SIBSON, R. Locally equiangular triangulation. *Comput. J. 21* (1978), 243–245.
20. WATSON, P. P. Computing the *n*-dimensional Delaunay triangulation with application to Voronoi polytopes. *Comput. J. 24*, 2 (May 1981), 167–172.
21. WILLIAMS, J. W. J. Algorithm 232. Heapsort. *Commun. ACM 7*, 6 (June 1964), 347–348.
22. ZAHN, C. T. Graph-theoretical methods for detecting and describing gestalt cluster. *IEEE Trans. Comput. C-20* (Jan. 1971), 68–86.

Received November 1983