Interner Bericht

Polyhedral Reconstruction of 3D Objects by Tetrahedra Removal

Frank Isselhard, Guido Brunnett, Thomas Schreiber

Interner Bericht Nr. 288/97

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-67653 Kaiserslautern

Fachbereich Informatik

Universität Kaiserslautern · Postfach 3049 · D-67653 Kaiserslautern

Polyhedral Reconstruction of 3D Objects by Tetrahedra Removal

Frank Isselhard, Guido Brunnett, Thomas Schreiber

Interner Bericht Nr. 288/97

Fachbereich Informatik
Universität Kaiserslautern
Postfach 3049
D-67653 Kaiserslautern

Februar, 1997

Herausgeber: AG CAD und Algorithmische Geometrie
Leiter Professor Dr. Guido Brunnett

Polyhedral Reconstruction of 3D Objects by Tetrahedra Removal*

Frank Isselhard, Guido Brunnett and Thomas Schreiber
University of Kaiserslautern
Department of Computer Science
D-67663 Kaiserslautern

Abstract

The problem of constructing a geometric model of an existing object from a set of boundary points arises in many areas of industry. In this paper we present a new solution to this problem which is an extension of Boissonnat's method [2]. Our approach uses the well known Delaunay triangulation of the data points as an intermediate step. Starting with this structure, we eliminate tetrahedra until we get an appropriate approximation of the desired shape.

The method proposed in this paper is capable of reconstructing objects with arbitrary genus and can cope with different point densities in different regions of the object. The problems which arise during the elimination process, i.e. which tetrahedra can be eliminated, which order has to be used to control the process and finally, how to stop the elimination procedure at the right time, are discussed in detail. Several examples are given to show the validity of the method.

1 Introduction

The reconstruction of an object from surface points irregularly located in 3D-space is one of the most interesting open problems in the field of computer aided engineering. In recent years, this problem is of increasing importance since efficient scanning techniques become available. A survey on different methods of data acquisition was given by Jarvis [9]. These methods often produce a large amount of data that can not be handled by current CAD-systems. The surface reconstruction modules of such systems are until now restricted by limiting the number of points which they are able to manage. Furthermore these systems are usually based on either curve fitting or edge detection and therefore only provide a restricted reconstruction tool. So the process of surface reconstruction, which is also referred to as reverse engineering, still remains a time consuming task.

This paper presents a new method for the reconstruction of an object from a set of boundary points. This method is focused on the generation of a polyhedral approximation of the object

*This work was supported by Deutsche Forschungsgemeinschaft (DFG) under contract Br 1185/3-2

which is an important step in the process of reverse engineering as stated by Várady [14]. The proposed method is an extension of Boissonnat's work [2]. He suggests a volume based surface reconstruction method which uses the well known Delaunay triangulation of the data points as an intermediate step. The basic strategy of his algorithm is to eliminate tetrahedra form the triangulation to extract an object representation and to approximate the desired shape.

In contrast to Boissonnat's method, which is restricted to the reconstruction of objects with genus 0, the algorithm presented here is able to reconstruct objects with genus > 0. This is an important advantage since real world objects are usually not restricted to genus 0. In order to reconstruct an object in a reasonable way, it is necessary to get the data points from different samples using multiple views. Hence we can get different densities of the sample points in different regions of the object. A nonuniform distribution of the data points is also desired if there are different levels of details within the object. Our method is also able to cope with this phenomenon, provided that the object is totally sampled. To do this, we developed a strategy for automatically terminating the elimination process. Thus the method avoids undesired holes in the polyhedral approximation.

In the remaining part of this section the problem which we want to solve is described in detail. After that some assumptions are made which are required for the method to produce acceptable results. These assumption do not really restrict the method, they rather claim some facts that can be stated as common sense. At the end of the section a short overview of some related work that has been done in the past on the same topic is given.

The volume based reconstruction method is described in particular in the second section. Details like defining criteria for removable tetrahedra, defining a cost function to control the elimination process and the automatic termination of the removal are elaborately explained. An algorithm is given which recapitulates the different items of the method.

A few examples demonstrate the validity of our approach and the quality of the resulting approximation. We conclude our work with some remarks on further steps which have to be done to make reverse engineering a powerful and versatile engineering tool.

1.1 Statement of the Problem

The problem that we want to solve in this paper can be stated as follows:

> Given a set $\mathcal{P} = \{p_1, \ldots, p_n\}$ of unstructured data points sampled from the surface $\mathcal{S}$ of an unknown object, construct a closed polyhedron of triangular faces through all data points.

The solution to this problem is not unique and it is difficult to characterize a polyhedron that approximates the original object in a convenient way. Among the possible solutions, polyhedra with special properties may be preferred. Polyhedra of minimal area as suggested by O'Rourke [12] may yield strange results [2]. Another criteria was given in [1] where the curvature of the original surface is used to generate an object model.

Since there is a large number of possible solutions, it is unsuitable to generate all possible polyhedra to find an optimal one. So we must find a way to reduce the complexity of the problem. The basic idea of the method presented here is to use a global structure on the data points, from which an optimal representation can be obtained.

1.2 Assumptions

In contrast to previous methods, the algorithm presented here makes relatively few assumptions about the set of data points and the object from which they are sampled. The most important advantage of the new method is that the object to be reconstructed may have arbitrary topological type. Nevertheless some assumptions are necessary to guarantee an appropriate approximation of the desired shape.

It is obviously clear that we can't expect an adequate reconstruction of the intended object if the data points are inaccurately sampled from the object. Since the only knowledge of the object the algorithm is given is the set of unstructured sample points, it is impossible to recover details in regions where the point density is insufficient. So we assume that the density of the sample points is higher than the dimension of the smallest detail.

To assure that the surface $\mathcal{S}$ of the unknown object is properly reconstructed we have to claim another property of the data points. Since the resulting surface consists of triangles which are faces of tetrahedra of the Delaunay triangulation, we should suppose that these triangles are in some way part of the original surface $\mathcal{S}$ which we want to reconstruct. This is the case when all three edges of the triangle are on $\mathcal{S}$. To satisfy this condition, we demand the following property which we call **density constraint**:

> For each point on $\mathcal{S}$, the nearest data point p_i according to the euclidean distance in $\mathbb{R}^3$ is also the nearest data point with respect to the distance measured on $\mathcal{S}$.

However, this constraint is a necessary but not sufficient prerequisite for our method. It should be noted that the method presented here does neither require nor exploit structural information in the data points, e.g. points lying on scan lines or additional information such as normal vectors in each data points.

1.3 Related Work

In recent years a lot of work was published which is concerned with the problem of object reconstruction. A great deal of this methods assume that the data points do have some structural information, e.g. lying on cross sections or scan lines, or additional information such as normal vectors are available in each data point. Since we do assume that we only have the 3D-coordinates of the data points and no additional information, we restrict our attention to methods that have the same prerequisites.

As mentioned before, a volume based object reconstruction method was introduced by Boissonnat [2]. Another method which uses the Delaunay triangulation as an intermediate step was proposed by Edelsbrunner and Mücke [6]. They extend the concept of α-shapes which was introduced in [5] to the three-dimensional case. This geometric structure can be generalized to weighted α-shapes [4] to cope with different point densities. However, the determination of appropriate weights in the data points is still an open problem.

A reconstruction method that construct a boundary representation out of an initial graph is due to Veltkamp [16]. O'Rourke [13] presented an algorithm that reconstructs a simple polygon out of a set of boundary points in the plane. Since this method make use of the concept of Voronoi diagrams, it can be easily extended to the three-dimensional case. Another method which is also due to O'Rourke [12] construct polyhedra of minimal area as 3D object models. However, these method may produce strange results [2]. An approximative surface reconstruction method was suggested in [8]. In this method, the resulting approximation do not contain the sampled data points. An detailed overview of additional surface reconstruction methods can be found in [15].

2 Volume based method

To obtain an approximation of the desired shape we use a volume based reconstruction method typically consisting of the following steps:

1. Create a volumetric description of the convex hull of the given points. This model yields a tessellation of the hull into small volume elements.
2. Eliminate volume elements from the initial object representation in order to improve the approximation of the desired 3D-shape. This step make use of a heuristic which guarantees that the remaining structure is still a polyhedron.
3. Extract the outer surfaces of the volume elements in order to obtain a boundary representation of the generated object.

Following Boissonnat's approach we use the Delaunay triangulation of the points as volume representation. This geometric structure suggests itself because of its nice properties (see Lawson [11]) and the fact that there are well known algorithms to compute this triangulation efficiently even in the three-dimensional case. Therefore we have to define criteria to decide whether a tetrahedron can be eliminated or not.

In addition the order the order in which the tetrahedra are removed has to be specified. It is obvious that the quality of the resulting surface depends on this order. We use a cost function c to assign a value $c(t)$ to each tetrahedron t. This value is used to determine the elimination sequence.

In general it turns out that the elimination process has to be stopped even if there are still tetrahedra that can be removed according to the defined criteria. Figure 3 shows an example where removing all removable tetrahedra results in a clearly not acceptable approximation. This difficulty raises the problem of finding an appropriate stopping point for the tetrahedra removal.

These problems, i.e. defining criteria for removable tetrahedra, determining an appropriate elimination sequence and finding the best stopping point, are discussed in detail in the next sections.

2.1 Criteria for the elimination of tetrahedra

If we eliminate a tetrahedron from the triangulation, we must guarantee that the remaining structure is still a polyhedron. Thus we can only eliminate tetrahedra if the elimination does not result in a isolated point, i.e. a point which does not belong to a tetrahedron any more. This rule is obvious because all the data points are sampled from the surface of the object and therefore our goal is to reconstruct a coherent surface without isolated points.

As we want to sculpture our object from the triangulation, it is clear that we want to eliminate only these tetrahedra which have at least one face on the surface of the current approximation.

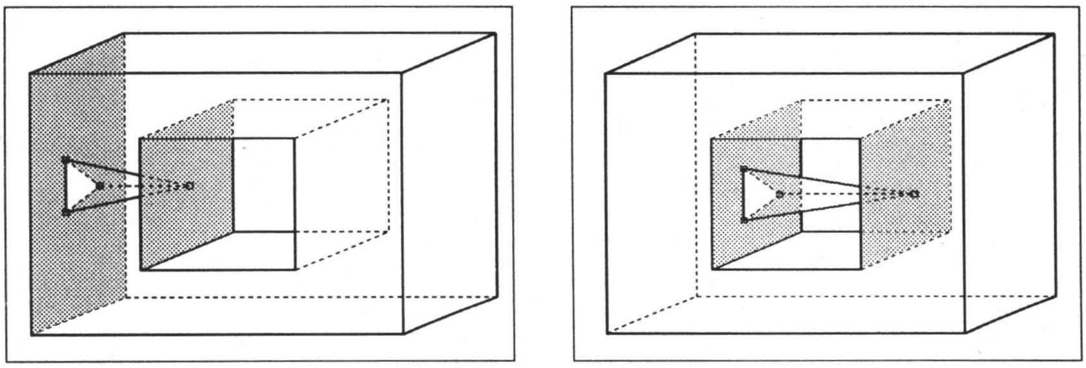
In his work, Boissonnat [2] gave the following two criteria for the elimination of tetrahedra:

A tetrahedron can be eliminated, if

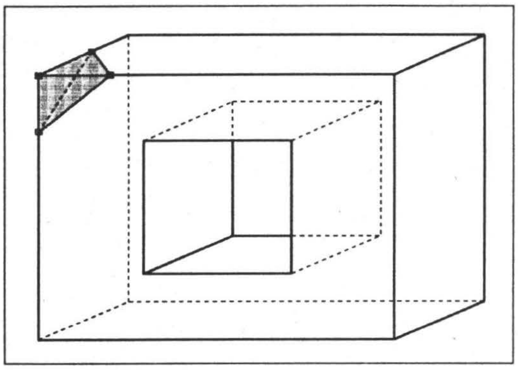
- exactly one face, three edges and three points are on $\mathcal{S}$ or
- exactly two faces, five edges and four points are on $\mathcal{S}$

where $\mathcal{S}$ is the surface of the current approximation.

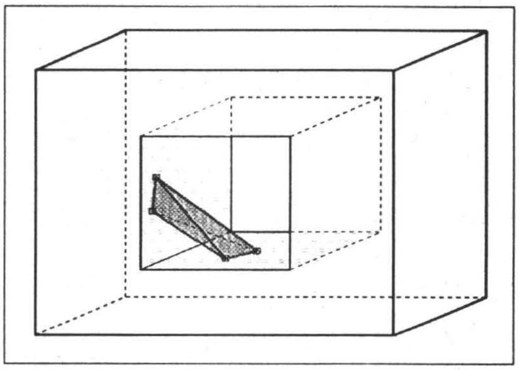
Since Boissonnat's method is restricted to the reconstruction of objects with genus 0, these criteria are sufficient for his purpose. The tetrahedra specified by this two criteria are not the only ones that can be eliminated without violating the rule mentioned above. As we want to reconstruct objects with genus > 0, we must extend Boissonnat's criteria.



a) *b)*



c)



d)

Figure 1: Examples of removable (*a*), *b)* and *d*)) and not removable (*c*)) tetrahedra

Figure 1 shows three types of tetrahedra that can be removed in addition to the tetrahedra specified up to now. Figure 1a) shows a tetrahedron with one face and four points on $\mathcal{S}$. This one càn be eliminated because the point not belonging to the face does not become isolated after the elimination. The tetrahedron in Figure 1b) has three faces and four points on the surface $\mathcal{S}$. The tetrahedron in figure 1c) has also three faces and four points on $\mathcal{S}$. But in contrast to the one in figure 1b) this one can not be eliminated because doing so the point P gets isolated which is a violation of our elimination rule. The tetrahedra in figure 1d) has four faces and four points on $\mathcal{S}$. This one can be eliminated because each point is a vertex of at least one other tetrahedron.

Therefore we get the following five criteria for the elimination of tetrahedra:

A tetrahedron can be eliminated if

- one face and tree vertices are on $\mathcal{S}$
- two faces and four vertices are on $\mathcal{S}$
- one face and four vertices are on $\mathcal{S}$ and the vertex not belonging to the face does not become isolated (figure 1a))
- three faces are on $\mathcal{S}$ and the common vertex of the faces does not become isolated (figure 1b))
- all four faces are on $\mathcal{S}$ and no vertex becomes isolated (figure 1d))

2.2 Definition of the cost function

It is obviously clear that the quality of the resulting approximation depends on the order in which the tetrahedra are eliminated from the triangulation. The required density constraint implies that all triangles of the desired approximation are faces of the tetrahedra of the Delaunay triangulation. This observation implies that we want to keep small and regular tetrahedra and we must remove irregular tetrahedra with long edges and large faces. We do so because it is unlikely that such a tetrahedron belong to the desired approximation if the point density is sufficiently high.

To determine the order in which the tetrahedra are removed, we associate a value $c(t)$ to each tetrahedron t. Since the tetrahedron with the maximum value should be removed, we must choose this cost function in way that undesired tetrahedra are removed first. This can be done in various ways.

Possible cost functions may be for example the value of the longest edge of the tetrahedron, the area of the largest face or the radius of the circumscribed sphere. In our algorithm, we use the following cost function:

$$c(t) = \max\{d(m, p_i) + (-1)^\alpha \cdot d(m, m_{f_i})\}$$

where

$$\alpha = \operatorname{sgn}(\boldsymbol{n} \cdot (\boldsymbol{m} - \boldsymbol{m}_{\boldsymbol{f}_i}))$$

and

m	center of the circumscribed sphere of t
p_i	vertex of t
d	euclidean distance
m_{f_i}	circumcenter of face f_i which is on $\mathcal{S}$
n	surface normal of f_i

This cost function, which is due to Boissonnat, is the maximum distance between the faces of t on $\mathcal{S}$ and the associated parts of the circumscribed sphere. $c(t)$ is illustrated in figure 2.

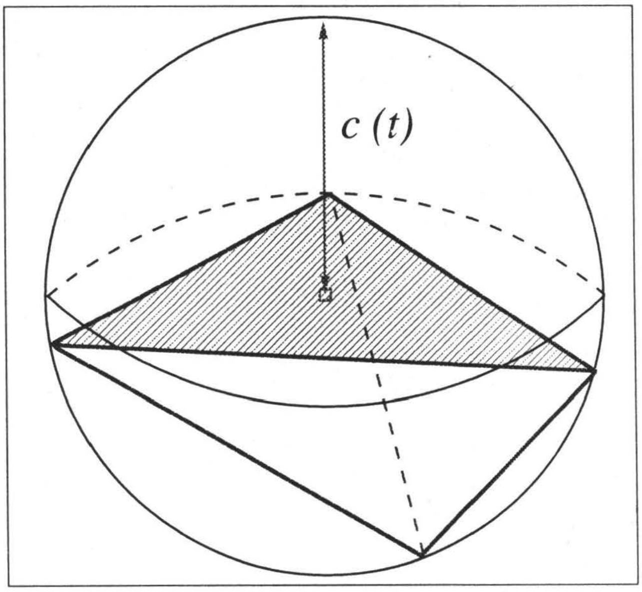


Figure 2: cost function $c(t)$

The cost function is chosen in a way that, if the density of the data points is sufficiently large, the value $c(t)$ of a removable tetrahedra t belonging to the interior of the object is smaller than the value of any tetrahedron belonging to the exterior of the object. So this cost function guarantees that we do only eliminate tetrahedra which do not belong to the interior of the object.

2.3 Automatic detection of the stopping point

In the last two sections, we defined criteria for the elimination of tetrahedra and by defining a cost function c, we determine the order in which the tetrahedra are removed since we eliminate the tetrahedron with the largest value first.

The question now is, how long do we eliminate tetrahedra? We can of course eliminate all tetrahedra which fullfill our elimination criteria. But in this case, a lot of tetrahedra are removed even after we got an appropriate approximation of the object. The fact that all data points come from the surface of the object leads to a first naive solution: Eliminate tetrahedra until all data point lie on the surface $\mathcal{S}$ of the current approximation. However, this choice has two major drawbacks:

1. To reconstruct reflex edges, it may be necessary to eliminate tetrahedra even if all data points are on the surface $\mathcal{S}$.

2. Different point densities in different regions of the object lead to undesirable holes in the polyhedral approximation.

The first problem was already mentioned by Boissonnat. His algorithm stops if all points are on the surface. Afterwards tetrahedra are removed until no decrease in the cost function is achieved. But his method is not able to cope with the second problem. It is important to notice, that this problem is not caused by our modified elimination rules, it also occurs when we only use the restricted criteria given by Boissonnat.

If the density of the data points is not uniform, the cost function does not guarantee any more that tetrahedra which do not belong to the interior of the object are eliminated first. In this case we remove tetrahedra which belong to the interior of the object and therefore we get undesirable holes in the approximation. Hence we must stop the process before we eliminate the 'wrong' tetrahedra.

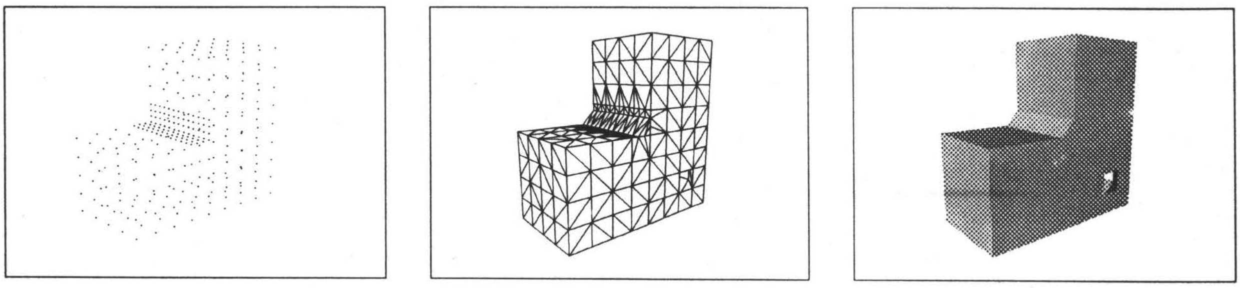


Figure 3: Example of an undesired hole in the polyhedral approximation

Figure 3 shows an example of an undesirable hole in the approximation we get if we do not stop the process at the right time. Near the reflex edge, the density of the data points is three times higher than at the rest of the object. So we get an hole at the lower right corner of the object while the data points belonging to the edge are not on the surface.

In the following we present a strategy for the termination of the elimination process. This strategy solves the two problems mentioned above. First, we automatically detect the appropriate stopping point. This solves the second problem. If all data points are now on the surface $\mathcal{S}$, we are done. If not, we must perform some more elimination steps to complete the surface.

The detection of the stopping point is done by investigating the elimination function

$$e(t) = \max_{t \in T} c(t).$$

It can be observed that this function has several significant jumps. One of these jumps corresponds to the optimal stopping point. This particular jump is characterized by a stabilization of the cost on a higher level. Figure 4 shows an example of a characteristic elimination function.

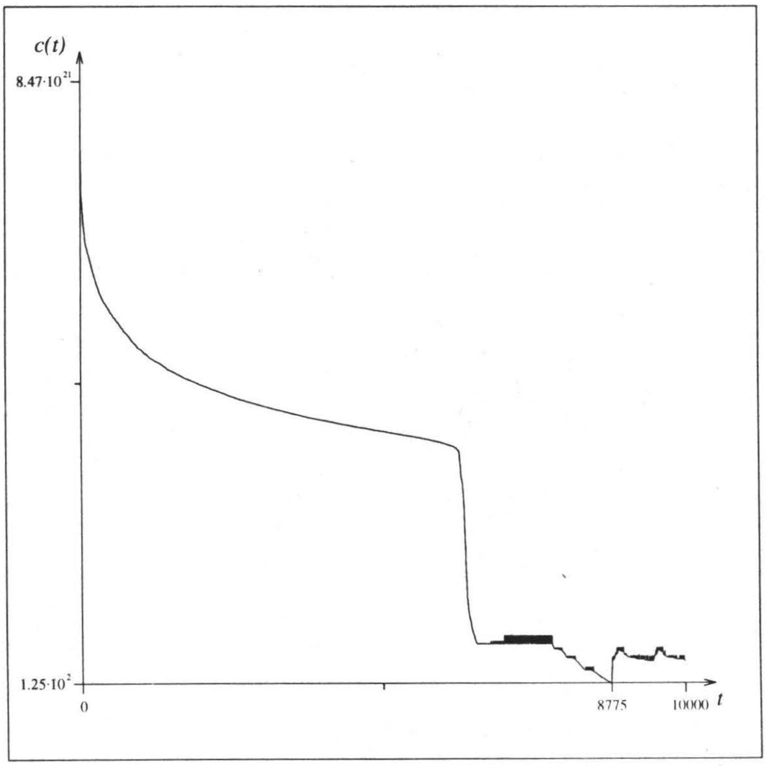


Figure 4: Characteristic elimination function

To realize that the optimal stopping point corresponds to the jump where the elimination function shows a stabilization of the cost on a higher level, we consider the following argument: At the desired stopping point a tetrahedron is eliminated which belongs to the interior of the object and has at least one face on the surface of the current tetrahedral approximation. After eliminating this tetrahedron, its neighbours have at least one face on the surface. So this neighbours can be eliminated if the elimination does not result in a isolated point. In the general case, such a tetrahedron has higher costs than a tetrahedron which do not belong to the interior of the object. This holds because the face of the tetrahedron which is on the surface is large since the vertices are on the surface of the object but the face is an interior one.

By the elimination of such a tetrahedron, more and more tetrahedra of this kind become removable and since they all have higher costs than the tetrahedron removed before the stopping point, we get a stabilization of the cost at a higher level.

If we want to detect the optimal stopping point, we must make some elimination steps even after this point. So we must undo this steps afterwards.

After detecting the stopping point, we perform the second part of our strategy. If all data points now are on the surface, we are done. At this point, if our object is of genus 0 and the data points are uniformly distributed, we get the same result as Boissonnat get with his method. If not all data points are on the surface, we must furthermore eliminate tetrahedra.

Since the elimination process is stopped to avoid undesired holes, the elimination rules or the cost function have to be changed before the algorithm proceeds. If the data points do not violate the assumption that the density of the data points is higher than the dimension of the smallest detail, we can expect that actually existing holes in the object are broken out before the stopping point. Therefore we can use the restricted elimination rules after this point. Since, as mentioned before, we also get undesired holes if we only use these restricted criteria, the cost function has to be changed too. This is done by scaling the cost function $c(t)$ by the radius of the circumscribed sphere, which is a measure for the size of the tetrahedron. So the cost of a small tetrahedron increases in comparison to the cost of a big tetrahedron. After the stopping point the modified cost function

$$c'(t) = \frac{c(t)}{d(m, p_i)}$$

is used where m and p_i are defined as so far. To complete the surface, tetrahedra are eliminated which fullfill the restricted elimination criteria. The order in which the tetrahedra are eliminated is determined by the modified cost function $c'(t)$ tetrahedra are eliminated until all data points are on the surface and no more decrease in the cost function is achieved.

2.4 Reconstruction algorithm

Using the preparatory work which is done in the last three sections, we are now able to describe our reconstruction algorithm. The algorithm starts with a set $\mathcal{P} = \{p_1, \ldots, p_n\}$ of data points sampled from the surface of the object. At the end the algorithm provides a tessellation of the object into tetrahedra; the faces of this tetrahedra which do only belong to one tetrahedron form the surface of the object.

The algorithm consists of five major steps:

(1) Compute the Delaunay triangulation $\mathcal{T}$ of $\mathcal{P}$

(2) Mark all tetrahedra $t \in \mathcal{T}$ that can be eliminated according to the modified elimination rules.

(3) For all marked tetrahedra t compute $c(t)$.

(4) Until the stopping point is not detected:

- **(a)** Eliminate tetrahedron $t_{max} = max\{t_i \in \mathcal{T} \mid c(t_i) \leq c(t_j) \ \forall t_i \neq t_j\}$.
- **(b)** Investigate the neighbours n_i of t_{max}, if they can be eliminated, compute $c(n_i)$.

(5) Complete the surface:

- **(a)** Mark all tetrahedra that can be eliminated according to the restricted elimination rules.
- **(b)** For all marked tetrahedra t compute the new costs $c'(t)$.
- **(c)** Repeat the steps (4)(a) and (4)(b) until all points of $\mathcal{P}$ are on the surface $\mathcal{S}$ and no more decrease in the costs is obtained.

The construction of the Delaunay triangulation even in three dimensions is a well known problem in computational geometry. Therefore many algorithms for its computation have been developed (see e.g. [3], [7], [10], [17]). In our method we us an implementation of an incremental algorithm given by Watson [17].

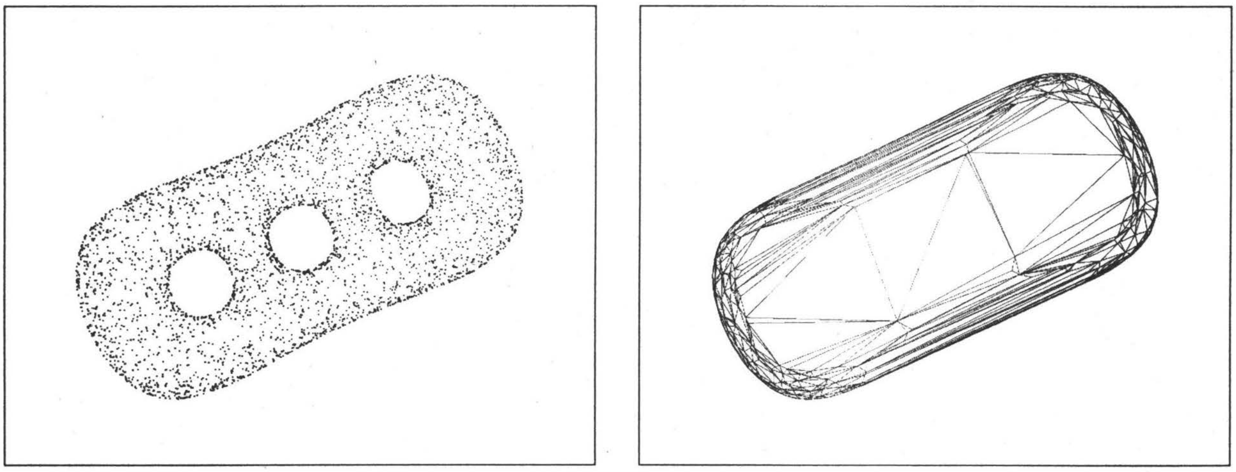
To mark a tetrahedron that can be removed, it may be necessary to check whether a vertex p_i becomes isolated after the elimination. This is done by computing recursively a list of all tetrahedra that have p_i as a vertex. If this list only contains the tetrahedron to be eliminated, the vertex becomes isolated and the tetrahedron can not be eliminated. If the list contains at least two tetrahedra, p_i does not become isolated after the elimination.

During the determination of the cost function $c(t)$ in step (3), the computational effort can be reduced, if we store the radius of the circumscribed sphere in each tetrahedron. This can be done during the computation of the Delaunay triangulation since this radius is already needed at this time.

The most important step of our algorithm is the elimination of the tetrahedra t_{max}. To do this in an efficient way, we must sort the tetrahedra by the value $c(t)$. If we store the tetrahedra sequentially in an array, we get into problems performing step (4)(b) because we must remove the tetrahedron t_{max} from the array and, if necessary, add some new tetrahedra to it. So it is necessary to resort the array afterwards. In our algorithm, we avoid this problem by using a balanced binary tree to store the tetrahedra. So we only have logarithmic cost if we remove or add one tetrahedron. This tree has to be rebuild in step (5)(b) since the elimination criteria are restricted after the stopping point and the value of the cost function changes for all removable tetrahedra.

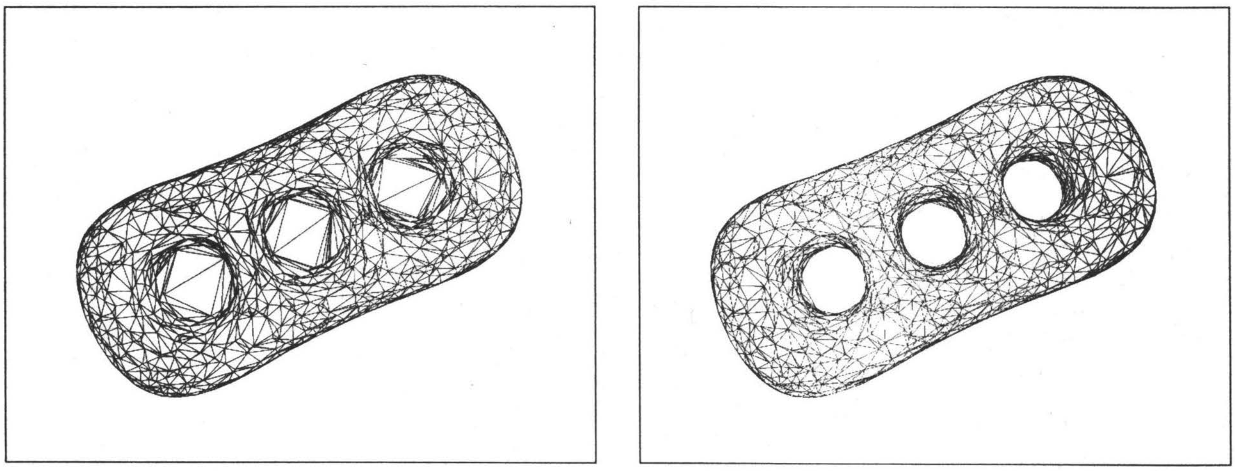
3 Examples

In this section we present two examples to demonstrate the validity of our method. Figure 5 shows an torus like object with three holds. Beside the data points and the final approximation, the triangulation of the convex hull and an intermediate step in the reconstruction process is shown.



a) data points

b) convex hull

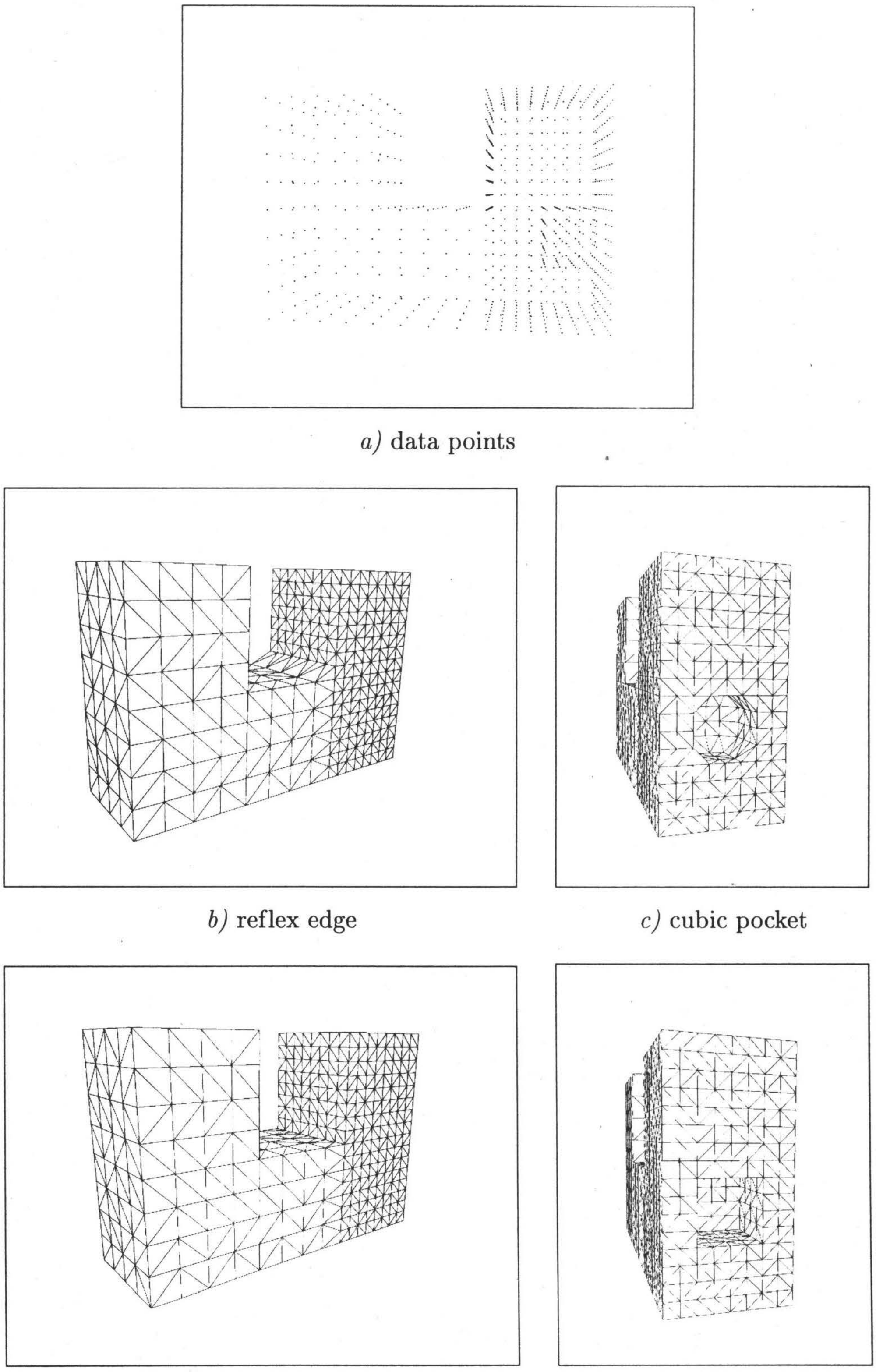


c) intermediate step

d) final approximation

Figure 5: torus like object with three holes

In this example, all data points are on the surface of the approximation when the stopping point is reached. In this case, no completion of the surface is necessary. The next example shows an object for which we need the completion of the surface. The data points shown in figure 6a) are artificially sampled from a small object which has a cubic pocket. Since the density of the data points is higher in one part of the object, not all data points are on the surface when the stopping point is reached.



a) data points

b) reflex edge

c) cubic pocket

d) reflex edge

e) cubic pocket

Figure 6: small object with a cubic pocket

Figure 6b) and 6c) show the approximation of the surface at this point. After the completion of the surface, we get the final approximation of the object which is depicted in figure 6d) and 6e), respectively. The reflex edge where the density of the data points is higher is fairly reconstructed as well as the cubic pocket.

4 Summary and Future Work

In this paper we described a new method for the construction of a polyhedral representation of an arbitrary object from a set of unstructured surface points. Our method is a volume based one which means that we start with a volumetric representation of the convex hull of the data points and then extract our object by sculpturing this representation. We use the Delaunay triangulation of the data points because this geometric structure is very natural, has nice properties and is easy to compute. So we sculpture our object by eliminating tetrahedra from the triangulation. To do this, criteria are defined which are used to decide whether a tetrahedron can be removed or not. In contrast to previous methods, by defining these criteria we are able to reconstruct objects with arbitrary genus. By assigning a cost function $c(t)$ to each tetrahedron t that fullfill one of our elimination criteria, we fix the sequence in which tetrahedra are eliminated since the tetrahedron with the largest value is eliminated first.

In addition to the benefit that the genus of the object is not restricted, another important advantage of our approach over already existing methods is the ability to cope with different point densities. We do this by an automatic termination of the elimination process an by completing the surface afterwards. Therefore we do not get undesired holes in our approximation and we assure that all data points are on the surface of the object, which is an important point for applications which are attached to our algorithm.

The construction of a polyhedral object representation out of the data points and generating a surface approximation at the same time is only a first step in the process of reverse engineering. The final goal of this task is to create a CAD-model of the object. To do this, several steps are necessary (see [14] for an overview). Based on the triangulation of the surface we must find a segmentation of the surface of the object and after that we perform a modeling step in which we represent the the different segments by different CAD-primitives (planes, cones, cylinders, free-form surfaces, etc.).

Segmentation means grouping points belonging to the same region. This can be done by identifying points that have similar properties, for instance normal vectors or curvature. This additional information can be computed from the surface triangulation. To improve the quality of these additional information, we can optimize the surface triangulation using data dependent triangulations.

During the modeling step we must assure that the created CAD-surfaces meet some quality requirements and that the surfaces are constructed in a way that we can guarantee a certain continuity along the boundaries of adjacent patches. This satisfies an important requirement in the reverse engineering process: we must be able to manufacture the reconstructed object out of the CAD-representation.

In these areas, a lot of future work is required before reverse engineering becomes a powerful and profitable tool in the design and manufacturing process.

Acknowledegment

The authors would like to thank Hugues Hoppe for the courtesy of providing the data set of the object depicted in figure 5.

References

[1] J.-D. Boissonnat. Representation of objects by triangulating points in 3-D space. In *Proceedings of the International Conference on Pattern Recognition*, pages 830–832, 1982.

[2] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transaction on Graphics*, 3(4):266–286, October 1984.

[3] A. Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24:162–166, 1981.

[4] H. Edelsbrunner. Weighted alpha shapes. Technical report, Department of Computer Science, University of Illinois, 1992.

[5] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, July 1983.

[6] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, January 1994.

[7] T.-P. Fang and L. A. Piegl. Delaunay triangulation in three dimensions. *IEEE Computer Graphics and Applications*, 15:62–69, 1995.

[8] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *ACM Computer Graphics*, 26:71–78, 1992.

[9] R. A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):122–139, March 1983.

[10] B. Joe. Three-dimensional triangulations from local transformations. *SIAM Journal for Scientific Statistical Computing*, 10(4):718–741, 1989.

[11] C. L. Lawson. Properties of n-dimensional triangulations. *Computer Aided Geometric Design*, 3:231–246, 1986.

[12] J. O'Rourke. Polyhedra of minimal area as 3D object models. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 664–666, Vancouver, Canada, 1981.

[13] J. O'Rourke. Connect-the-dots: A new heuristic. *Computer Vision, Graphics, and Image Processing*, 39:258–266, 1987.

[14] T. Várady, R. R. Martin, and J. Cox. Reverse engineering of geometric models - an introductrion. To be published in Computer Aided Design, 1997.

[15] R. C. Veltkamp. *Closed Object Boundaries from scattered Points.* LNCS 885. Springer, 1994.

[16] R. C. Veltkamp. Boundaries through scattered points of unknown density. *Graphical Models and Image Processing*, 57(6):441–452, November 1995.

[17] D. F. Watson. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, 24(2):167–172, 1981.