

Qualitative analysis of two reconstruction techniques and their application in dentistry

Luis Gustavo Nonato
Rosane Minghim

University of São Paulo

ICMC-Instituto de Ciências Matemáticas e de Computação

C. P. 668, 13560-970, São Carlos-SP, Brazil

E-mail: gnonato@icmc.sc.usp.br

Milton Hirokazu Shimabukuro

UNESP—Pres. Prudente-SP, Brazil

Abstract. *Although visualization in the field of dentistry has some of the same requirements as the medicine field, the differences in goal demand specific approaches. This paper reports on the implementation of two fundamentally different approaches to reconstruction of structures from planar cross sections and their application to dentistry data. One of the approaches was an implementation of a distance-based sampling technique, and the other is a new algorithm, based on the Delaunay triangulation. Both were tested using contour data of teeth and the results are compared here in the light of the target applications, which are teaching and training dentistry, as well as simulation of dental procedures and illnesses. Widely mentioned problems encountered in local reconstruction methods such as marching cubes for these cases are clearly illustrated in this paper, and a very satisfactory alternative is given. © 2000 SPIE and IS&T. [S1017-9909(00)01504-X]*

1 Introduction

Visualization has been improving the way people work in many fields of expertise in the last two decades. In medicine this is especially true. In the related field of dentistry, however, this trend has been less noticeable, partly due to its different approaches to diagnosis. Images do not play the same role in diagnosis for dentistry as they do for other fields of health sciences. Reconstruction of three dimensional (3D) objects, that is, the development of models to represent three-dimensional structures from two-dimensional (2D) data, is a major concern in all of these applications, and its goals vary from educational systems to technologically advanced surgery. Potential applications for computer visualization employing 3D models of teeth and mouth are many. In dentistry, initial reconstruction tasks using general-purpose packages have been performed,¹ and some other activities in the field have been reported, including mandible reconstruction² and chewing simulation,³ but

these developments are still modest compared to what can be achieved employing currently available software and hardware technology.

This paper describes part of the work performed in the context of a project for virtual dentistry, which intends to develop research and to design systems that implement visualization techniques for support to several activities in the dentistry field. The main targets of the project are education, training and assistance to dentistry professionals, as well as simulation of dental procedures, as suggested in many parts of this paper.

In the field of visualization,⁴ most of the graphical techniques to present data fall into two categories: surface fitting (SF) and direct volume rendering (DVR). In surface fitting methods, lately expanded to encompass volume definition, geometrical structures are created from data sets that represent the phenomenon under study. These geometrical structures are then presented to the user by means of conventional computer graphics rendering. The second category of visualization methods, DVR, generates images straight from the data sets, without creating intermediate structures. Both types of technique are being developed, implemented and tested within the virtual dentistry project. However, due to some of the goals of the project, and particularly to our need to explore numerical simulation and interaction in the context of dentistry, there is the need to have a geometrical and a topological definition of the reconstructed objects. These requirements led to the exploration of reconstruction surface and volume. This was done by studying the effects that the several available techniques for reconstruction would have on teeth data, and selecting two of those for further analysis.

The main intent of this paper is to show the results of this task, presenting actual differences between two typical approaches to reconstructing data for medicine and related applications, taking the reconstruction of teeth as a test case. Both reconstruction techniques were implemented in the course of this work. They reconstruct objects from contours of interest, taken in consecutive parallel cross sections. The first technique is a distance-based implicit sam-

Paper VDA-13 received Dec. 1, 1999; revised manuscript received Apr. 26, 2000; accepted Apr. 26, 2000.
1017-9909/2000/\$15.00 © 2000 SPIE and IS&T.

pling technique followed by a local method of surface extraction. The second technique is a novel algorithm based on Delaunay triangulation (DT). We conclude by showing the effectiveness of this new algorithm in reconstructing models with configurations that are suitable to various kinds of tasks, including real-time interaction, simulation, and visualization.

The work in virtual dentistry is part of a project called *PowerVis*,⁵ which intends to review visualization techniques in the light of human perception, user tasks and application specific issues, and expand multimodal presentation of objects.^{6,7} The dentistry case is, besides a rich and open application area, a platform to apply, test, and develop new interaction, simulation, and visualization tools.

In Sec. 2, we review some of the relevant literature regarding the reconstruction of 3D models. Section 3 presents the two techniques employed in this work, and Sec. 4 shows the results and their comparative analysis.

2 3D Reconstruction in Dentistry and Related Fields

A reconstruction algorithm for an object builds a 3D surface that matches the topology presented in individual slices, in an attempt to reflect its original spatial form. Data for reconstruction in dentistry and medical fields usually come from images, detected in a variety of ways (scans or photographs are the most common). Reconstruction from these data can be done directly from the image, and also through the definition of 2D contours (for individual elements of the object undergoing reconstruction). These contours are then piled up to form a 3D set of slices. This is the setup for the algorithms presented here.

There are several strategies to solve the three-dimensional reconstruction problem, such as: implicit, voxel, optimal, deformable, and heuristic models. Some of these techniques build the two-dimensional surface that bound the object, and others can generate some representation of the volume within the reconstructed objects.

Some of the classical problems in the field reconstruction include:

- i. branching, that is, what to do when a contour splits or several contours unite between consecutive slices;
- ii. correspondence, that is, how to decide, among the contours in neighboring slices, which to connect;
- iii. resampling, that is, how to obtain the original cross sections from the reconstructed object;
- iv. mesh generation, that is, what is a good mesh and how to obtain it from the data available.

In order to handle the problem of creating 3D models from 2D data, implicit surface reconstruction techniques aim to create a smooth implicit function so that the contour points are in the zero set of that function. In Jones and Chen,⁸ a potential field function is defined from a signed distance for each of the contours in the set. Then, by interpolating the function in two consecutive planar sections, the overall implicit function is obtained. With implicit techniques it is possible to ensure that the function created contains the original contours in the zero set, but to guarantee that the contours in the surfaces generated are the same as the original ones may be difficult. One of the algorithms employed here was implemented based on this approach⁹ and it is described in Sec. 3.1.

Voxel based models¹⁰ visualize the volume of the object directly without building a surface or volumetric mesh. Though this method does not present the correspondence, tiling, and branching problems, it is difficult to interact with the generated model or to use it for applications like simulation. It can, however, be an interesting alternative to visualizing volume information.

Attempts to produce an optimal solution to reconstruction usually employ graph theory. The first work in reconstruction from planar sections was developed by Keppel.¹¹ In this work, the surface bounding the maximal volume of the object under reconstruction is obtained from an optimal search in a graph. Shinagawa *et al.*¹² expand the discrete toroidal graph to form a continuous graph, and homotopy is used for reconstructing parametric surfaces. Meyers *et al.*¹³ assemble contours into cylinders and a minimum spanning tree determines the correspondence between them. The optimal approach solves the mesh generation problem but some heuristic is necessary in order to find a solution for the branching problem.

Deformable models combine geometry, physics, and approximation theory to generate a model of the original object. A good overview of deformable models can be found in “*Deformable Models in Medical Image Analysis*.”¹⁴ An advantage of deformable models is that segmentation may be incorporated into the reconstruction process, instead of working as a preprocessing step, as do other methods. However, objects with complex topology are difficult to handle with deformable models. Delingette¹⁵ introduces simplex meshes that are deformable models where the topology can be locally altered.

Heuristic techniques employ some criteria to affect the connections between consecutive contours, and are generally faster than optimal or deformable model methods. The work by Ekoule *et al.*¹⁶ maps contours into their convex hulls and the shortest edge is used to tile the convex hulls of the contours placed in adjacent levels. The branching problem is solved by creating an intermediate contour between two consecutive slices. Bajaj *et al.*¹⁷ define three criteria that the reconstructed surface must satisfy and an algorithm is built based on these criteria. In the regions where the conditions cannot be satisfied the surface is completed through a triangulation that takes into account the Voronoi skeleton. Boissonnat¹⁸ uses the three-dimensional Delaunay triangulation for connecting the contours in consecutive slices. This work builds a graph using the projection of the 2D Voronoi diagrams of every two consecutive slices. From that, a volumetric model is created through tetrahedron elimination. Geiger¹⁹ improves Boissonnat’s approach by projecting the two-dimensional Voronoi skeletons from one slice to the adjacent slice in order to handle complicated branching and dissimilar contours.

The new algorithm discussed here was developed based on this last type of strategy, based on computational geometry. This algorithm, first introduced by Nonato^{20,21} was motivated by Boissonnat’s ideas, that is, it also uses the three-dimensional Delaunay triangulation to make a volumetric model of the object. The differences are on the definition of the connected components of the reconstructed object and in the fact that it always generates a 3D piecewise linear (PL) manifold between two adjacent planar sections. This algorithm, described in Sec. 3.2, proved good in

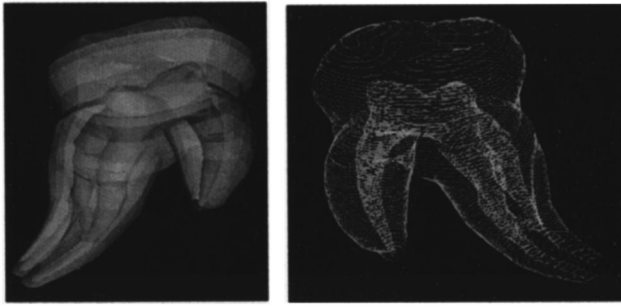


Fig. 1 Tooth model and corresponding set of contours.

all four aspects of the reconstruction problem mentioned earlier.

The two algorithms in Sec. 3 were chosen to investigate the dentistry problem due to their apparent suitability to reproduce the shape of teeth and other mouth structures. We implemented and tested both (Jones *et al.*⁸ and Nonato^{20,21}), and here we compare the results obtained in the light of suitability to the virtual dentistry application and related fields. As well as showing the better suitability of this computational geometry approach, the remainder of this paper serves the purpose of confirming the good qualities of the algorithm presented by Nonato,^{20,21} and to illustrate in practice some of the claims against the implicit approach for this application, such as difficulty in orientation and creation of either too blocked or too large meshes.

3 Three-Dimensional Reconstruction using the Distance-to-Geometry and DT-Based Approaches

Two algorithms for reconstruction were implemented and a selection process was started, to define what to use and why to use it in the remainder of the project. One of the algorithms employs a distance-based (DB) implicit function to sample the contours into a regular grid.⁸ This grid is then processed for reconstruction under a voxel by voxel intersection algorithm such as marching cubes.²²

The other algorithm employs Delaunay triangulations (DT) to perform connections between corresponding contours in neighboring slices.^{20,21}

Both algorithms assume that the input is a set of contours that are closed and simple curves. Both produce models whose mesh connects contours in adjacent slices and treat multiple contours in the same slice.

These algorithms are presented in the following text. Figure 1 shows one of the models reconstructed by the DT algorithm, and its corresponding set of contours. There is an internal and another external surface, that can be reconstructed together or independently. We employ this set of contours throughout the remainder of this paper due to its suitability to show most cases to be encountered in reconstruction of tooth data, and also because they present most important characteristics that we wish to discuss.

3.1 Distance-Based Approach

This algorithm follows the implicit approach mentioned earlier.²² This type of approach does not need the contour matching stage employed by other algorithms. While this saves time, it makes it difficult to change connections.

The algorithm employs a two-stage strategy. In the first stage, a regular 2D grid is imposed on each slice, with resolution chosen by the user. Field functions are then applied on each grid point calculating its distance to the closest contour in the set, as explained in the following. As a result, a regular 3D grid is obtained, with scalar values stored at the vertices of grid cells.

In the second stage, surfaces of interest are extracted from the volume grid using a local reconstruction method such as the marching cubes algorithm.²² In our case, we employed a predefined filter belonging to the software VTK.⁴

The following two field functions were used to calculate the distance of each grid point to the contour:

$$f(x,y) \begin{cases} -1 \\ 0 \\ 1 \end{cases} \quad \text{and} \quad f(x,y) \begin{cases} -\text{dist}(x,y) \\ 0 \\ \text{dist}(x,y) \end{cases}$$

where $\text{dist}(x,y)$ is the distance between a point (x,y) on the grid and the closest edge of the contour. The top value in each function is the result for points inside a contour, the bottom value gives values for points outside a contour. 0 is the value for grid points lying on a contour.

The second set of functions above smooth the surfaces obtained, compared to an approach using only the first set, but its use over the whole domain is time consuming. To improve performance, a combination of both sets is employed, according to the criterion of interception between surfaces and cells of the grid. The interception test between surface and voxel is given by the following rule: if all the voxel vertices are inside or outside the surface, then it is not intercepted by the surface. Otherwise, it is intercepted, and the distance has to be calculated for all vertices of the voxel. In our case, Euclidean distance was used. A value 1 is stored for all vertices in voxels inside and -1 for voxels outside the surface.

In order to close the surface at the ends of the contour set, the implemented algorithm adds an extra ‘‘slice’’ to the grid. In the additional voxels formed, values are placed on the grid so that the intersection decision criterion turns false.

The algorithm was implemented as a filter of the software VTK. This is a library of visualization functions, available with source code that can be modified and expanded to suit particular applications. VTK⁴ has been the basis of most of our implementations. The input to the implemented filter is a polygonal structure (*vtkPolyData*) representing the set of slices defined on regularly spaced parallel planes. The output of the implemented filter is a regularly spaced 3D grid (*vtkStructuredPoints*) with scalar values in its vertices. Scalar values represent the distances to contours in all the voxels intersected by a particular tooth surface, and -1 's and 1 's in the remaining voxels. From the resulting grid in the filter output, tooth surfaces are traced using VTK's marching cubes filter, as mentioned above. Results of the application are given in Sec. 4.

The next section details the other algorithm employed in this work.

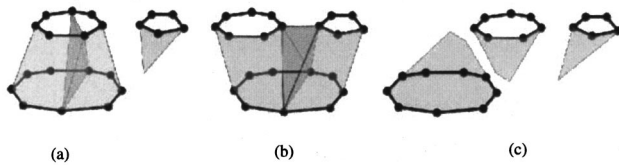


Fig. 2 (a) Reverse tetrahedron connecting two well positioned contours; (b) reverse tetrahedron connecting three well positioned contours; (c) contours are not well positioned and the reverse tetrahedron does not exist.

3.2 DT-Based Algorithm

Computational geometry reconstruction techniques make use of Delaunay triangulations and Voronoi diagrams to generate a volumetric model from planar sections of an object.¹⁸ By volumetric model we mean the reconstruction of both boundary and interior elements of structures in a data set.

Delaunay triangulation (DT) and Voronoi diagram (VD) can be defined as follows:

Given a discrete set S of sample points in \mathbf{R}^3 , the Voronoi diagram of S is a cell decomposition of \mathbf{R}^3 where each cell of the diagram contains a point of S and the part of \mathbf{R}^3 closer to it than to any other sample point. That way, if the points in S are in general position, each cell in VD is a convex polytope where each vertex is equidistant from exactly four points of S . A set of points are in general position if no four points are coplanar and no five points are cospherical. These four points form a Delaunay simplex and the set of Delaunay simplices are the Delaunay triangulation. An important property of the Delaunay triangulation, consequent upon definition, is that all three-dimensional simplices have empty circumspheres, i.e., the circumsphere of a tetrahedron does not contain any points of S in its interior.

A good source of information on DT and VD is given in the work by Fortune.²³

The reconstruction process presented here is a new algorithm based on DT.^{20,21} It can be subdivided in three basic steps:

- (1) Delaunay triangulation and edge classification;
- (2) Definition of connected components;
- (3) Model generation.

Step 1: Delaunay triangulation and edge classification.

Given a set of contours in two adjacent planes, the reconstruction begins by generating the three-dimensional DT from the vertices of these contours. Assuming that the contours have a coherent orientation, the edges of DT that are contained in adjacent planes can be classified as internal, external, or contour edges, according to whether they are located internally, externally, or on any contour. If there are contour edges that do not belong to the triangulation, they must be subdivided and the triangulation locally rebuilt in order to include them.¹⁸

Step 2: Definition of connected components.

After the DT triangulation has been built and the edges classified, we must define the connected components in order to resolve the correspondence problem. In DT, the geometrical position of the contours in adjacent slices is related to a kind of tetrahedron called reverse tetrahedron. Reverse tetrahedrons have only one edge in each adjacent plane,

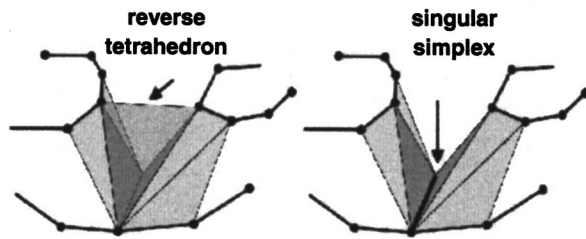


Fig. 3 Singular edge generated with the elimination of a reverse tetrahedron.

where these edges are not contour edges. If two contours are geometrically well positioned, they have a reverse tetrahedron connecting them. Otherwise, this tetrahedron does not exist.^{20,21} Figure 2 illustrates the situation described above.

After the connected components have been defined, we must eliminate the tetrahedrons with external edges that are among the components in order to disconnect them.

Step 3: Model generation

Within each connected component, the tetrahedrons with external edges must also be eliminated because we want to guarantee that the original contours are preserved under intersection with the original planar sections. However, the elimination of these tetrahedrons can generate singular edges in the triangulation and singularities must be avoided in simulations. An edge is singular if its link is not homeomorph to a sphere or semisphere. Figure 3 shows a singular edge generated when a reverse tetrahedron is removed.

In order to solve the singularity problem we remove only those external tetrahedrons (tetrahedrons with external edges) whose elimination does not generate singular edges. The remaining external tetrahedrons in the component are subdivided and the new vertices created by the subdivision are translated to an intermediate position between the slices. Figure 4 illustrates this situation for a reverse tetrahedron.

The volumetric model generated with the process above has an important property: it is a volumetric PL manifold, making it adequate to be used in numerical simulations like finite element and finite volume analysis.

Although the algorithm above automatically finds the connected components for the model, it can be adjusted so that the user decides how the connections (or connected components) must be realized. The user decides which contours must stay in each component so the DT triangulation and the model generation steps are separately executed for each one of them.

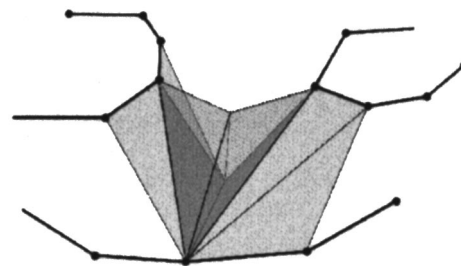


Fig. 4 Tetrahedron subdivision eliminating singular edges.

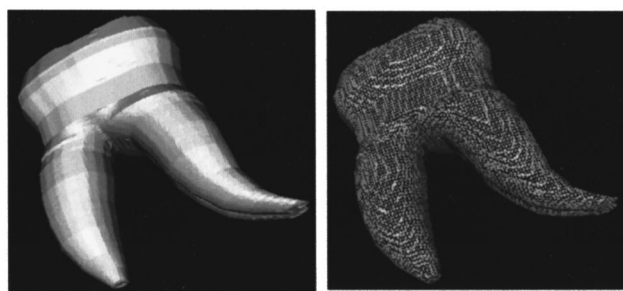


Fig. 5 Application of DB algorithm for the tooth contours, and corresponding triangular mesh.

Another advantage of this reconstruction process is that geometrical calculations are executed only in the Delaunay triangulation generation. The tests to find the connected components and singularities are topological tests, and if we make use of an appropriate data structure these tests become robust and efficient.²⁴ Additionally, this technique is capable of reproducing the exact same original contours under the section by the same planes as the slices.

4 Results and Discussion

In view of the goals of the virtual dentistry project, particularly those of real time interaction and simulations, the characteristics of the models and their computer representations are primary concerns. This section presents the results obtained when applying both approaches discussed in the previous section, thus showing the overall best conformation of the models produced by the DT-based algorithm.

The distance-based approach (DB) was implemented by a member of our team, and added to VTK structure as a filter derived from its class hierarchy. This filter takes as input a polygonal definition of a series of contours, and applies the sampling-by-distance routine to them, in order to build a 3D regular grid. Following that, VTK's version of the marching-cubes algorithm is employed to generate the individual surfaces. After reconstruction, the resulting tooth model is visualized by means of conventional surface rendering methods.

The DT-based (DT) approach was implemented using C in Unix environment, and the resulting meshes were translated into VTK's file format. Using these files, VTK was used to visualize the models and generate the images presented here. All images presented in this section were generated using this framework.

In addition to the visualizations, VRML models of the reconstructed teeth were created using exporters of VTK.

The models produced by both techniques are compared here under several criteria, specifically: configuration of the surface triangulation, flexibility of the algorithm, adaptability to the application, treatment of the connectivity problem, and capability of real time postprocessing of the models (particularly in respect to the number of cells generated). We also analyze the behavior of the methods when faced with special types of contours, as those that will represent singularities in the final model.

Figures 5 and 6 show the reconstruction of the external surface of the tooth of Fig. 1 for both algorithms, and their respective external triangular meshes. From these images, two important differences are noted between the different

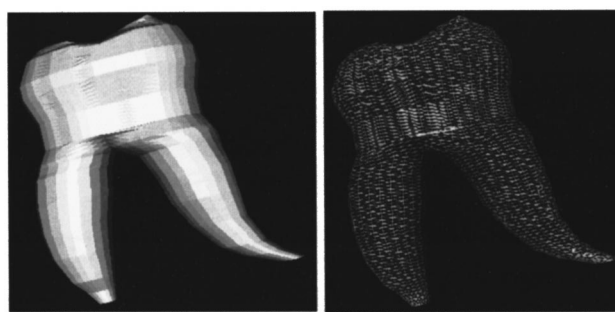


Fig. 6 Application of DT-based algorithm for the tooth contours, and corresponding triangular mesh.

processes of reconstruction, all of which will be discussed in the text that follows. First, there are certain ridges at the bottom of the tooth in the DB reconstruction (Fig. 5). Those, which are not actually present in the data, do not appear when the same set of contours is input into the DT program (Fig. 6). Besides being smaller, the DT mesh presents a set of triangles with much more regular behavior than that of the DB mesh.

Figures 7–12 illustrate the behavior of DT and DB towards the correspondence problem in reconstruction. Regarding contour correspondence, the DT algorithm implemented can behave in two different manners. Left to itself, it will separate contours that do not have proper overlap against orthogonal projection in one of the slice planes, as illustrated in Fig. 9. Only contours with high degree of overlapping will be defined as corresponding contours [see Figs. 8(a) and 9(a)]. Under the user's intervention, however, DT can connect any two contours in neighboring slices. For that, the user only has to choose which contours

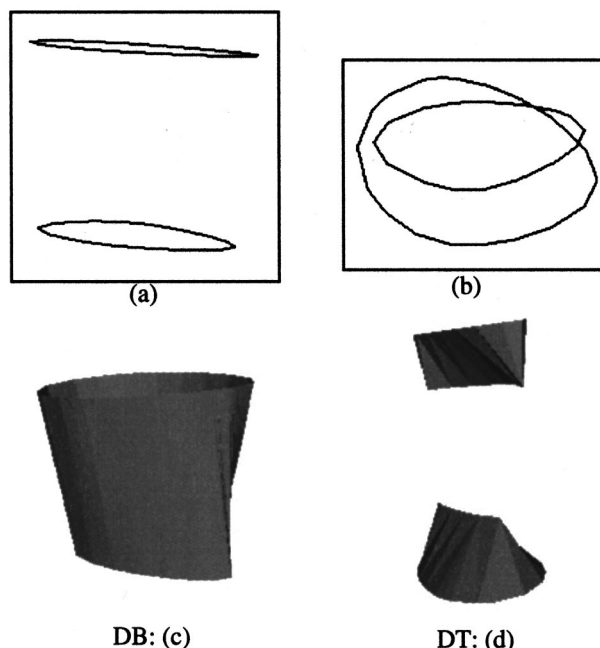


Fig. 7 Consecutive contours reconstructed by both DB (c) and DT algorithms (d).

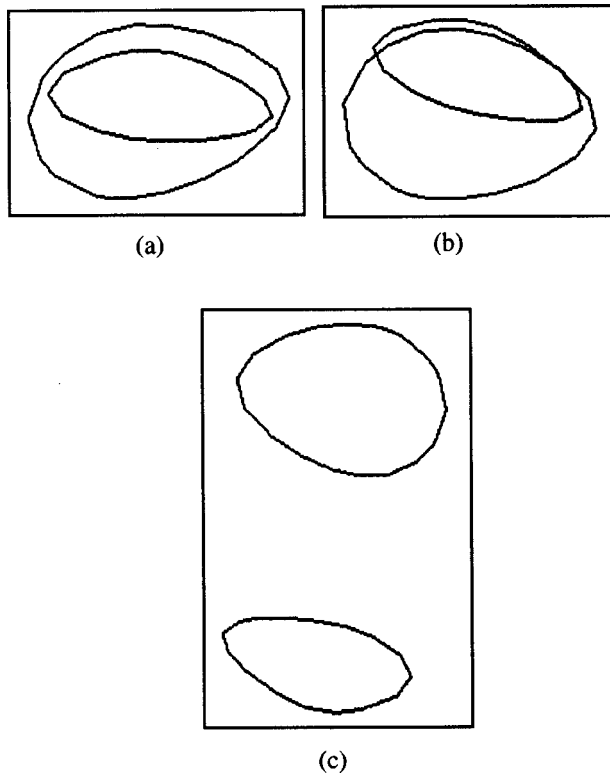


Fig. 8 Sequence of consecutive contours with overlapping patterns.

to connect. In those cases, that is, when contours are forced to connect, the algorithm also performs well, producing a triangulation that behaves quite adequately in position, size, and number of triangles (see Fig. 10). For the same set of contours, however, the DB algorithm does not behave as well. It tends to connect even slightly overlapped contours, and the interpolation chosen to do the intermediate layers of triangles is not always satisfactory, as Figs. 11(b) and 12 indicate. This is the reason for the ridges in Fig. 5. It is possible to force the DB algorithm to split connections if necessary, but it demands tampering with the original grid, which, in most of cases, is not desirable. Splitting overlapping contours would mean closing one and the other segment of the structure. This is done in a satisfactory and smooth way by the DB filter, as Fig. 11(c) illustrates. However, to force the closing of a surface it is necessary to insert an “empty” layer below or above the last slice before closure. The capability of controlling correspondence

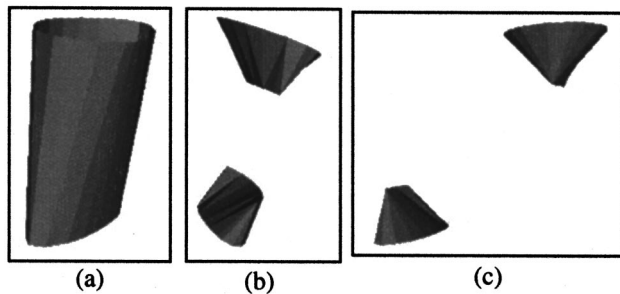


Fig. 9 Reconstruction from contours in Fig. 8 using DT approach.

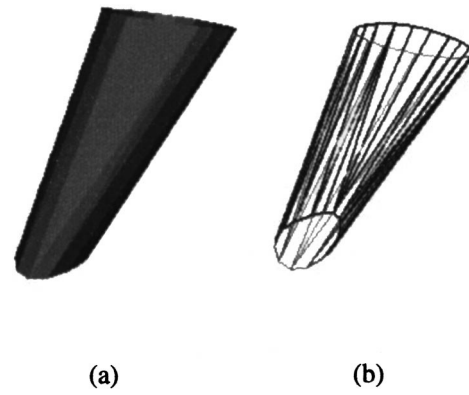


Fig. 10 Reconstruction DT from the same data as DB in Fig. 12(c), and corresponding cell mesh.

with very satisfactory results confers flexibility to the DT based algorithm, allowing for development of effective, application specific, automatic connection strategies.

One main concern in reconstruction of surfaces and volumes is how the technique behaves when special cases of contouring appear, as in the case of singularities. Figure 13 shows the comparison between DT and DB techniques in regard to this aspect of the reconstruction problem. Both techniques produced satisfactory results when presented with typical singularities in the tooth case. However, the quality of the triangulation differed sensibly. The DT technique produces much smaller and much more manageable cell configurations, as illustrated in Figs. 10(b) and 14(b). It is possible to note that the size of the individual triangles are very adequate in all cases. The DB technique, however, presents originally too many triangles, with many of them showing small areas with distant vertices, an aspect that impairs simulation, and may produce visual aliasing (see Fig. 14). Even when decimation is performed to reduce the number of cells [Fig. 14(d)], the configuration obtained maintains the disadvantage of uneven triangulation. Figure 15 presents those same arguments for a part of the bottom

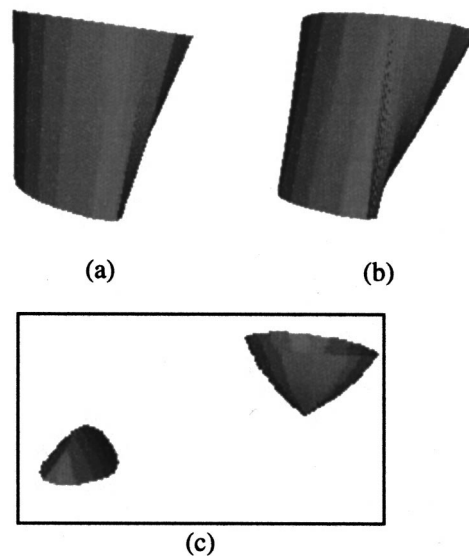


Fig. 11 Reconstruction from contours in Fig. 8 using DB approach.

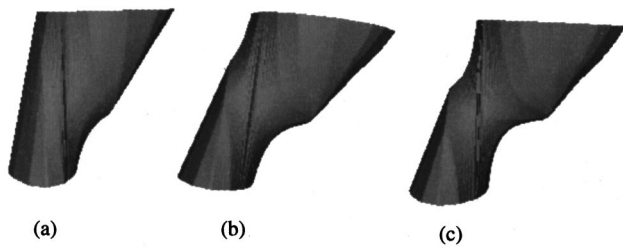


Fig. 12 DB reconstruction from contours in intermediate position between those in Figs. 8(b) and 8(c).

of the tooth model of Fig. 1. Figure 15(a) is the standard DT mesh which is regular with even triangle aspect ratios. Figure 15(b) is the standard DB mesh. It is well known that marching cubes produces a large amount of elements in its triangulation, and most of the time decimation is necessary. Figure 15(c) shows the maximum decimation VTK produces for the DB mesh. In this image, however, the smaller the mesh becomes, we can still observe its irregular aspect. The external mesh of the tooth has 94 627 triangles for the standard DB case, and 3817 in its maximum decimation. The DT standard triangulation has 7995 triangles, a quite manageable number. Even when DB mesh decimation matches the number of triangles of the DT mesh the appearance is quite blocked and the triangulation uneven. Figure 16 presents the shaded display of the exterior surface of the DB model after maximum decimation.

One basic concern of the final mesh configuration is orientation of the cells. Typically, approaches based on marching cubes make it difficult to define inside and outside faces. As for the DT based algorithm presented here, once the original contours are oriented (and so they must be), the final reconstructed structure is oriented as well. This makes many processing tasks much easier. Many algorithms, and particularly rendering algorithms only work well if the element inside and outside orientations are well defined.

Concerning triangulation errors, so far in all tests executed, not once did we detect wrong triangulations produced by the DT algorithm. VRML models generated from the surfaces were robust (see Fig. 17) and fast to explore. DB reconstructions, however, generated triangulations that seemed consistent inside VTK renderer, but when translated to VRML produced holes during exploration of the models, due to the orientation problems typical of local reconstruction algorithms such as marching cubes. Errors were also detected in models previously reconstructed from the same data using another algorithm also based on com-

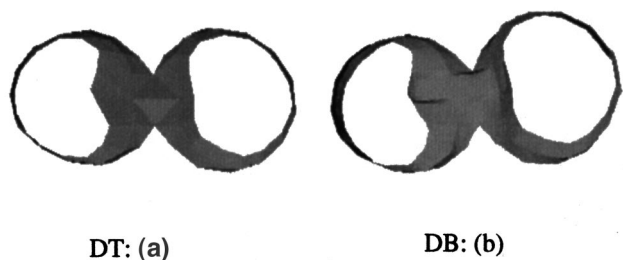


Fig. 13 Singularities reconstructed by both DT (a) and DB (b).

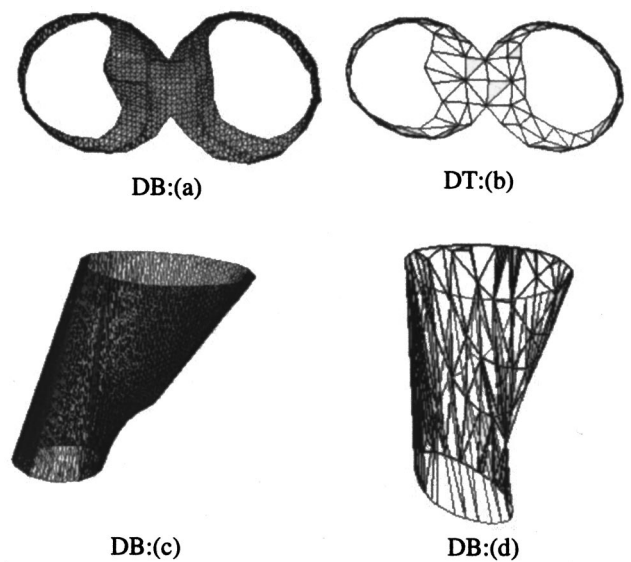


Fig. 14 Comparison between resulting meshes by both DB and DT approaches.

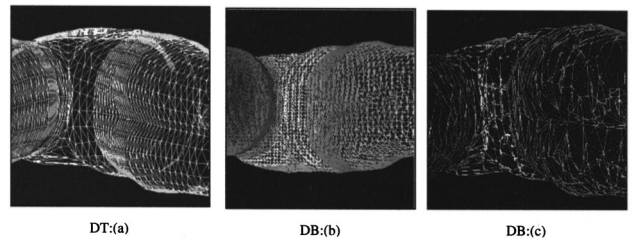


Fig. 15 Detail of tooth meshes; (a) is a DT mesh, (b) is DB, and (c) is DB with decimation.

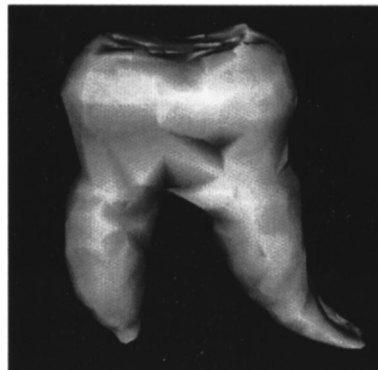


Fig. 16 Shaded mesh after decimation.

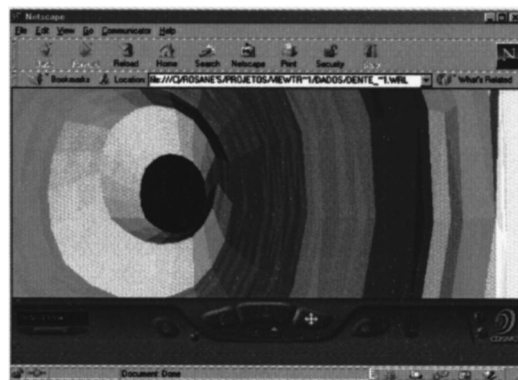


Fig. 17 VRML Internal exploration of DT-based tooth models.

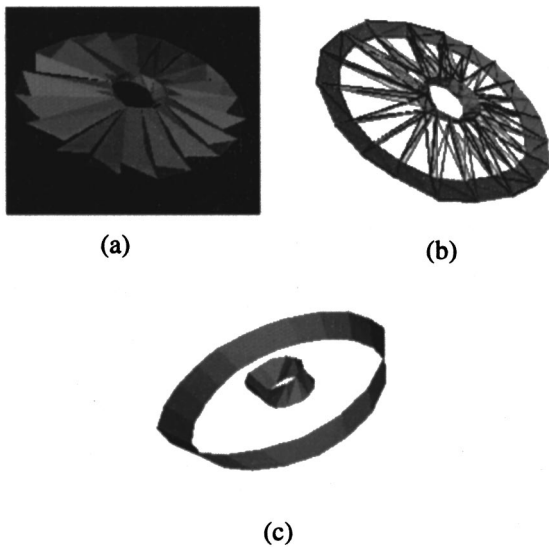


Fig. 18 Volume reconstruction, using the DT approach, between two neighboring slices: (a) internal elements; (b) wire-frame model of (a); (c) visualization of the boundary.

putational geometry, but using a graph constructed from the projection of the 2D Voronoi diagrams of every two consecutive slices.

One aspect that was left off of the above discussion was the fact that the DT-based reconstruction algorithm is capable of reconstructing the interior of contoured structures, while the distance-based local reconstruction is not. For simulation purposes, this characteristic is always desirable. Figure 18 presents a part of one such reconstruction realized with the algorithm implemented. As far as performance is concerned, the reconstruction stage of the DT strategy tends to be slightly slower than the DB technique. However, it is our main concern to have an efficient topology as a result, for the other processes we are targeting, rather than have a fast reconstruction process that produces less than satisfactory models. We are working to improve speed of reconstruction, since it is also important to allow fast iteration during the reconstruction stage of the visualization process.

The above discussion demonstrated some of the problems of the techniques that are based on local, voxel by voxel intersections, such as lack of orientation in the final mesh, creation of irregular, uneven, or too large meshes, and spurious shape effects in the reconstructed objects.

In all aspects of the reconstruction problem, the DT based algorithm behaved extremely well. It is expected that the mesh produced will be perfectly suitable for the purposes of the virtual dentistry project.

The data set used in the tests that were presented here were generated from real tooth data by a contour editor developed by our research partners.¹ Apart from those, we have three sets of images of sliced teeth for further developments.

Due to the evidence of the appropriateness of the DT algorithm to the dentistry case, and to the expectations that such performance will be maintained in other structures still to be reconstructed, it is our intention to concentrate effort in employing this method of reconstruction in the

remainder of the project. Appropriate improvements to the algorithms are being designed, as well as data structures that will help manipulation of the models during simulation and interaction.

The next steps in the project include: implementation of the DT algorithm in the context of VTK; implementation of a data structure designed to contain and handle the tetrahedralizations; implementation of multimodal interaction procedures with the tooth models; and implementation of direct volume rendering techniques to teeth data to investigate suitability of the technique for visualization in dentistry.

Acknowledgments

The authors wish to acknowledge Mike Goetz and Andy Day, from the School of Information Systems, UEA, UK, for the data used to generate most of the models presented here. This work is supported by FAPESP, Brazil. We also want to acknowledge the work of our students Carlos Frederico Rocha and Luis Fernando Chavier of ICMC-USP.

References

1. M. R. Goetz and A. M. Day, "Surface reconstruction for teeth," in *Proceedings of EUROGRAPHICS UK, 16th Annual Conference*, pp. 25–27, March, Leeds, UK (1998).
2. S. Seipel, I. Wagner, S. Koch, and W. Schneider, "Three-dimensional visualization of the mandible: A new method for presenting the periodontal status and diseases," *Comput. Methods Programs Biomed.* **46**, 51–57 (1995).
3. K. Myszkowski, G. Okuneva, J. Herder, T. L. Kunii, and T. L. M. Ibusuki, "Visual Simulation of the Chewing Process for Dentistry," in *Visualization and Modelling*, pp. 419–438, Academic, New York (1997).
4. W. J. Schröder, K. Martin, and W. Lorensen, *The Visualization Toolkit, An Object-Oriented Approach to 3D Graphics*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ (1998).
5. R. Minghim, M. C. F. de Oliveira, "PowerVis: Empowering the User with a Multi-Modal Visualization System," in *Proc. II IEEE Workshop on Cybernetic Vision*, pp. 106–111, Dec. 96, Brazil, IEEE CS (1997).
6. R. Minghim and A. R. Forrest, "An illustrated analysis of sonification for scientific visualization," in *Proc. IEEE Visualization'95*, IEEE CS, pp. 110–117 (1995).
7. A. M. Day, R. Minghim, S. M. Turner, and D. B. Arnold, "Visualization and Sonification of Marine Survey Data," in *Visualization and Modelling*, pp. 87–98, Academic, New York (1997).
8. M. W. Jones and M. Chen, "A new approach to the construction of surfaces from contour data," *Proc. EUROGRAPHICS'94*, **13**, pp. 75–84 (1994).
9. M. H. Shimabukuro, R. Minghim, and P. Licciardi, "Visualisation and Reconstruction in Dentistry," in *Proc. Int. Conf. on Information Visualisation IV'98*, London, UK, pp. 25–31, IEEE CS (1998).
10. A. Kaufman, "Advances in Volume Visualization," *SIGGRAPH'98*, Course Notes No. 24, Orlando, FL (1998).
11. E. Keppel, "Approximating complex surface by triangulation of contour lines," *IBM J. Res. Dev.* **19**, 2–11 (1975).
12. Y. Shinagawa and T. L. Kunii, "The homotopy model: a generalized model for smooth surface generation from cross sectional data," *Visual Comput.* **7**, 72–86 (1991).
13. D. Meyers, S. Skinner, and K. Sloan, "Surface from Contours," *ACM Trans. Graphics* **11**(3), 228–258 (1992).
14. A. Singh, D. Goldgof, and D. Terzopoulos, *Deformable Models in Medical Image Analysis*, IEEE CS, Los Alamito, CA (1998).
15. H. Delingette, "Simplex meshes: A general representation for 3D shape reconstruction," Technical Report No. 2214, INRIA (1994).
16. A. B. Ekoule, F. C. Peyrin, and C. L. Odet, "A triangulation algorithm from arbitrary shaped multiple planar contours," *ACM Trans. Graphics* **10**(2), 182–199 (1991).
17. C. L. Bajaj, E. J. Coyle, and K. Lin, "Arbitrary topology shape reconstruction from planar cross sections," *Graphic. Models Image Process.* **58**(6), 524–543 (1996).
18. J.-D. Boissonnat, "Shape reconstruction from planar cross sections," *Comput. Vis. Graph. Image Process.* **44**, 1–29 (1988).
19. B. Geiger, "Three dimensional modeling of human organs and its application to diagnosis and surgical planning," Technical Report, No. 2105, INRIA Sophia-Antipolis France (1993).
20. L. G. Nonato, "Volumetric Manifold Reconstruction from Planar

- Sections," PhD Thesis (in Portuguese), Mathematics Dep., Pontifical Catholic University, Rio de Janeiro, Brazil (1998).
21. L. G. Nonato, G. Tavares, R. Minghim, and M. C. F. de Oliveira, "Volumetric manifold reconstruction from planar sections and its applications," (unpublished).
 22. W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithms," *ACM SIG. Comp. Graph.* **21**, 163–169 (1987).
 23. S. Fortune, "Voronoi Diagrams and Delaunay Triangulation," D.-Z. Du and F. K. Hwang, Eds., pp. 193–233, *Computing in Euclidean Geometry*, World Scientific, Singapore (1994).
 24. H. Lopes, L. G. Nonato, S. Pesco, and G. Tavares, "Dealing with topological singularities on volumetric reconstruction," *Curves and Surfaces Design*, J-P. Laurent, P. Sablonniere, and L. L. Schumaker, Eds., pp. 229–238, Vanderbilt University Press, Nashville, Tennessee (1999).



Luis Gustavo Nonato received his mathematics degree in 1988 from the São Paulo State University (UNESP), Brasil and his PhD in Mathematics from the Catholic University at Rio de Janeiro, in 1998. In 1999 he joined the Computer Science Department of the University of São Paulo (ICMC-USP/São Carlos), where he works in three-dimensional reconstruction, computational geometry, scientific visualization, and simulation.



Rosane Minghim received her Computing Sciences degree from the University of São Paulo (USP) in 1985, her MsC in Electrical Engineering from the University of Campinas in 1990, and her PhD in Computer Studies from the University of East Anglia, U.K., in 1995. She has been a lecturer at the University of São Paulo in São Carlos (ICMC-USP/São Carlos) since 1986, and does research on visualization and sonification.



Milton Hirokazu Shimabukuro works as a software engineer at the School of Science and Technology of the São Paulo State University (UNESP) at Presidente Prudente, where he is involved in several research projects and acts as a local area computer network administration and as database administrator. He received his Computing Sciences degree from the University of São Paulo (USP) at São Carlos in 1986 and his MSc and degree in the same department in 1998. He is currently engaged in a doctorate program at ICMC-USP.