COMPARISON OF TWO METHODS FOR RECONSTRUCTION OF 3D OBJECTS FROM CONTOURS

Petr Felkel **Jiří Janáček***

Department of Computer Science and Engineering, Faculty of Electrical Engineering
Czech Technical University, Karlovo nám. 13, 121 35 Prague, Czech Republic
felkel@fel.cvut.cz

*Institute of Physiology, Academy of Sciences, Vídeňská 1083, 142 20 Prague 4, Czech Republic
janacek@biomed.cas.cz

ABSTRACT

Two approaches for reconstruction of 3D objects from contours in serial sections (cross-sections) are compared. The first method is based on thresholding and 3D volume reconstruction, the second on direct reconstruction from parallel contours. The accuracy of surface and volume measurement of reconstructed structures was estimated. Both methods construct a triangulated surface of the object and its inner structures.

Keywords: 3D reconstruction from contours, surface measurement, volume measurement

1 INTRODUCTION

Imaging methods in biology usually yield 2D projections or cross-sections of objects under study, containing only partial information on shape and dimensions. Computer based 3D reconstruction of such objects enables better understanding of shape and topological properties via visualization and enables direct measurements of objects geometrical and topological characteristics. Perfectly registered optical sections obtained by confocal microscopy represent almost ideal source of data for 3D reconstruction.

We have implemented and tested two approaches for 3D reconstruction of objects in serial sections (cross-sections). As the automatic segmentation of the data subjected to study (human placenta villi) is not available until yet we performed the reconstruction from interactively drawn contours.

Both methods construct a triangulated surface of the object and its holes.

2 DATA

The optical sections of human placenta villi, 1μm apart, were obtained as images with interslice resolution 0.25μm captured by confocal laser scanning microscope BioRad MRC-600. Contours of capillaries and vessels visible in the sections were drawn interactively using images of sections in A4 format, digitization tablet Genius 1812D (Kye Systems, Taiwan), and custom made program `CutView` under MS Windows [Jan98].

The input data consist of a set of contour lines outlining the shape of the villus (we call them *contours*) and the shape of the capillaries inside (we call them *holes*) — see an example in Fig. 1. The data acquisition method does not check for possible contour self or mutual intersections, or overlapping caused by multiple outlining of one shape and such errors can occur in the input.

The first method described is resistant to the intersections and overlapping; the second one is not and a contour preprocessing step has to precede the reconstruction.

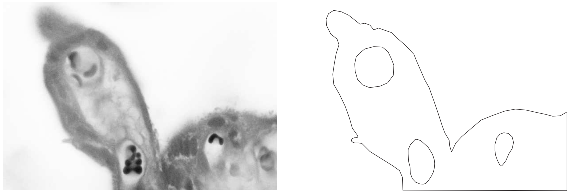

Figure 1: Object and appropriate contour and holes

3 3D VOLUME BASED METHOD

In this method further processing was performed in the graphical programming environment IRIS Explorer running on Indy workstation using custom made modules for data processing and the original Explorer modules for displaying 2D images, for isosurface detection and for rendering the 3D scenes (For details see [Jan98, Felke97a, Felke97b]).

The algorithm works as follows:

- Read the contour coordinates,
- convert coordinates into image with resolution $0.25\mu\text{m}\times0.25\mu\text{m}\times1\mu\text{m}$ by drawing Bresenham lines,
- fill in the rasterized contours (4-neighborhood 2D fill),
- resample the 3D grey volume by averaging (to get the resolution $4\mu\text{m}\times4\mu\text{m}\times4\mu\text{m}$),
- detect the isosurface (marching cubes algorithm [Loren87]),
- store the reconstructed surface (VRML file),
- measure the volume and surface (separate VRML filter).

The above described steps do not need long comment. The resampling step was performed to get regular grid of samples, which simplifies the implementation of 3D filtration. Resampling was done by simple averaging.

4 DIRECT RECONSTRUCTION FROM CONTOURS

This method was implemented partially in JAVA and C++ and tested on SGI workstation Ox and Indigo running IRIX and on PC based Linux systems.

The processing is performed in the following steps:

- Correction of the "typing errors" such as single or double point contours,
- manual correction of overlapping parts,
- reduction of the dataset,
- surface computation,
- volume and surface measurement,
- writing the reconstructed 3D object into VRML file.

The necessity of the first two steps is implied by the unreliability of the input contour data caused by the manual method for outlining contours and holes. The contour based reconstruction algorithm is extremely sensitive to the cases, when one object part is outlined twice with a slight different shape, or when the holes intersect their contours and leak out of the reconstructed surface.

The data reduction step does not affect the reconstruction method itself, but affects the number of triangles generated. And tens of thousands triangles are hard to display and explore in the conventional VRML browsers.

The reduction works on the principle of geometrical averaging. The points (pixels) of the input contour or hole are sequentially processed and their center of gravity (CG) is adaptively computed, i.e. the new points modify (slightly) the position of the CG. The point-to-CG distance is then tested and the points below the given threshold distance are removed. The first point which gets over this threshold distance survives and creates a new CG.

We have used the threshold value 4.24 (the diagonal of a 3x3 pixel square).

The surface computation step is described in Section 4.1 below, the principle of volume and surface measurement in Eq. (1) and Eq. (2) in Appendix A. Writing a reconstructed 3D object into the VRML file is similar and the structure of this file can be found elsewhere, e.g [vrm].

4.1 Surface Based Reconstruction Algorithm

We have implemented our version of the general surface reconstruction algorithm proposed by Oliva [Oliva95, Oliva96]. This method handles severe topology and morphology variations between neighboring cross-sections, such as holes and multiple-branching in a fully automatic and systematic way – The algorithm transforms the 3D tiling problem of adjacent contours into a

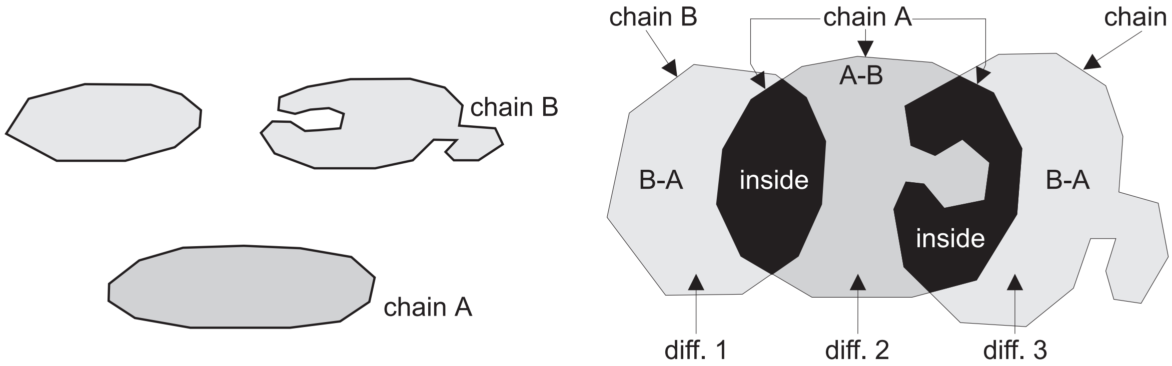

Figure 2: Contours in two cross sections (left), and their projection into the plane and computation of the areas of overlap and difference (right) (from [Oliv96])

2D task. The algorithm constructs a surface with a valid topology (non-degenerated) by *adaptive addition of intermediate contours*. New contours have such a shape to create a smooth interstage between the pairs of given contours. The ability to construct a non-degenerated surface, i.e. the surface consisting of self-non-intersecting triangles which fit closely without cracks, and the ability of handling a general branching are the main advantages of this algorithm.

The algorithm at first detects the contour correspondence (topology) between each pair of adjacent cross-sections by the existence of the contour overlap in perpendicular projection into one plane. The overlapping parts are taken as the interior of the object and are not triangulated. The surroundings of the union of the contours is taken as the surroundings of the object and is also ignored. Only the polygonal *areas of difference* are subject to the processing (see Fig. 2).

The border of each area of difference (see Fig. 2) is divided into sections — The parts, which originally belong to the contours in one section are marked as chain A, the parts which originally belong to the contours in the second section are marked as chain B.

The areas of difference are treated one after the other and the number of chains controls the next steps. The minimal number of chains in one area of difference is two - one chain A and one B, except for the top and bottom of the reconstructed structure.

If the area of difference consists of two chains (one chain A and one B), the triangulation uses the minimize span length criterion (see Fig. 3).

Complex polygonal areas of difference are in the original algorithm recursively split into more simple ones; In our case, we have applied only one split and a non-convex polygon triangulation. The split step is based on the computation of the *Angular Bisector Network* (ABN, by other authors called *straight skeleton* [Aichh95]) of the area of difference. ABN is defined as a network of straight line segments lying on the angular axes of the neighboring or opposite polygonal edges (The precise definition can be found in [Aichh95, Oliva96, Felke98]). The principle why angular axes are applied is based on the the presumption that the intermediate contour lays in the middle plane between the sections. Perpendicular projection projects such constructed contours exactly into the angular axes.

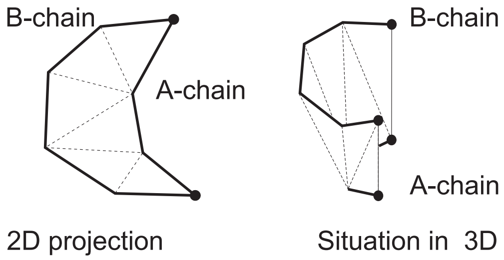

Figure 3: Principle of triangulation of the two-chain area of difference in 2D and the same situation in 3D

Each split part of the area of difference is handled separately. The split curve goes through the *middle bisector*, i.e. a part of ABN where the bisector is common to chain A and B. The ABN divides the area of difference into cells, which are triangulated separately. The resulting triangulation of the second area of difference from the example is in Fig. 4.

This algorithm solves the correspondence problem (by application of local criterion of adjacent contour overlap), the triangulation and general branching problems (also with holes), and generates the topologically correct surface without cracks and triangle intersections. It also correctly

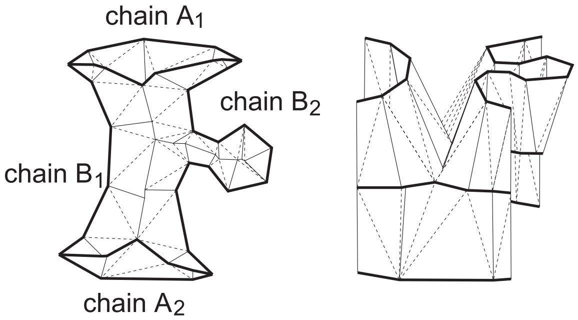

Figure 4: Tiling of the second area of difference from the Fig. 2 (from [Oliva96])

handles objects with cycles, such as the torus.

On the other hand, it can not recognize the correspondence of contours if they do not overlap and creates wrong connections if the non-related contours overlap (e.g. two skewed cones with an overlap). In the case of two contours of equal shape the whole overlap is understood as being inside of the object and is not triangulated. To avoid such a case little translation or scaling of contours is necessary.

5 EXPERIMENT

5.1 Data

We have tested both methods described here on four datasets of the human placenta villi captured by confocal microscopy (see Chapter 2 for details).

Villus No.		Volume V_M $[\mu m^3]$	Volume V_S $[\mu m^3]$	$\frac{(V_S - V_M)}{V_M}$ [%]	Surface S_M $[\mu m^2]$	Surface S_S $[\mu m^2]$	$\frac{(S_S - S_M)}{S_M}$ [%]
vil1	c	169970	165123	-2.79 %	20042	19739	-1.51%
	h	49849	43876	-11.9 %	30174	28285	-6.26 %
vil2	c	301698	296020	-1.88 %	37207	36180	-2.76 %
	h	90312	84745	-6.16 %	47584	45058	-5.31 %
vil5	c	868076	852093	-1.84 %	70742	69789	-1.35 %
	h	207835	195259	-6.05 %	88288	83762	-5.13 %
vil8	c	292323	286644	-1.94 %	42297	40827	-3.48 %
	h	69919	63759	-8.81 %	43925	41081	-6.47 %

Table 1: The comparison of results obtained by 3D volume reconstruction (M) and direct surface reconstruction (S) methods. c depicts the outer surface – contours, h the holes.

5.2 Results

The results of volume and surface measurements are collected in Tab. 1. First column describes the name of the dataset, the second and third the computed volume, the fifth and sixth the computed surface. The columns four and seven show the difference of the volume and surface area related to the values obtained by 3D volume reconstruction method method (M).

6 DISCUSSION

The results of surface and volume computations are similar. The tolerance less then 10% can be caused by different resolution the algorithms work in and is not significant for practical use. The 12% difference (holes of vil1) is most presumably caused by multiply drawn overlapping contours, i.e. errors of the manual outlining.

The simple 3D volume reconstruction method decreases spatial resolution of the data drastically. The advantage of decreasing resolution is that it lowers the memory requirements and computation time for reconstruction and rendering and yields nice-looking smooth surface due to filtering of small artifacts. The disadvantages of lowered resolution are the loss of fine details and changes of objects connectivity e.g. by breaking of thin structures or by fusion of boundaries of neighboring ones.

Direct surface reconstruction was done to achieve better topology preservation, as the details are not lost by the volume resampling step. On the other hand, the surface generated by this method is less smooth (jagged), as it respects the detailed shape of original contours, drawn by hand. (Compare the quality in Figures 5 and 6).

We see the improvement of the results and future work in better input data preparation without irregularities – in better segmentation method for contour outlining and in finishing the creation of a system for contour editing.

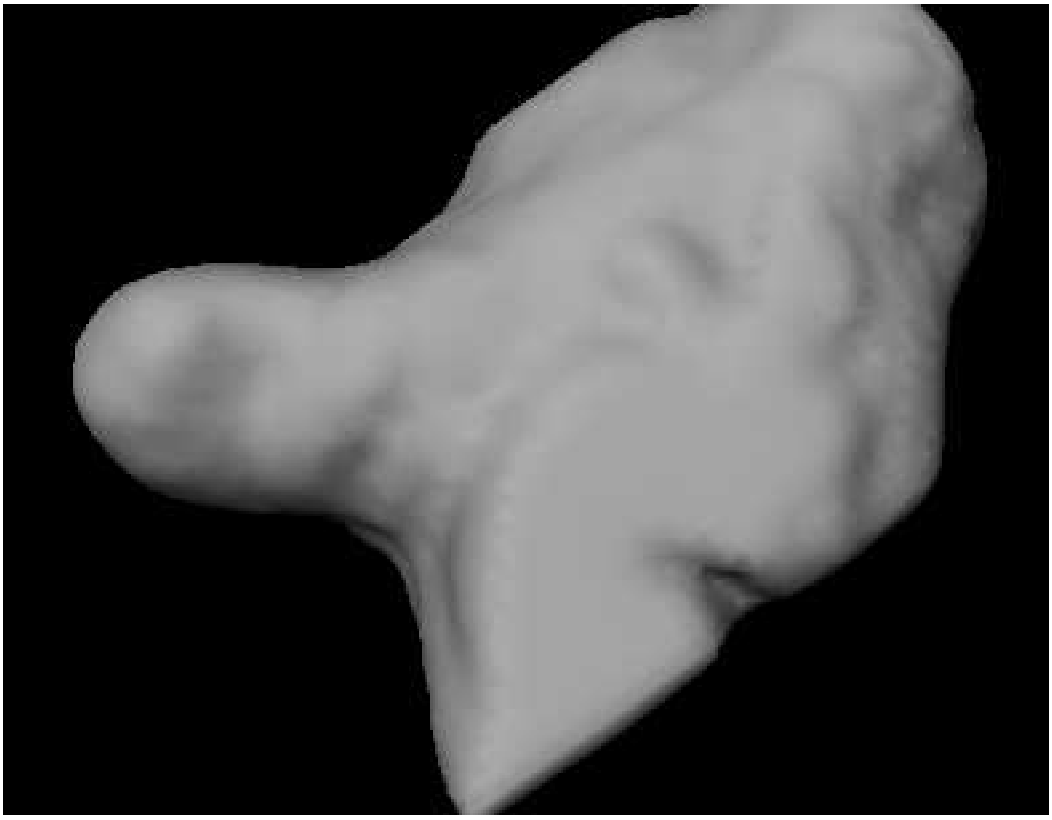

Figure 5: Example of 3D reconstruction performed by the volume reconstruction method

7 CONCLUSION

We have implemented two approaches for reconstruction of 3D objects from contours in serial sections. At first the method based on thresholding and 3D volume reconstruction, secondly direct reconstruction from parallel contours. We have tested them on four datasets of human placenta villi captured by confocal microscopy and compared them from two points of view – the quality of surface reconstruction and their applicability for surface and volume measurement.

8 ACKNOWLEDGEMENS

We wish to thank Dr. Jirkovská (Faculty of Medicine, Charles University) for providing contour datasets and specimen of human placenta and to our student Marek Šikl, who implemented the major part of surface reconstruction algorithm and performed the significant part of the measurements.

REFERENCES

[Aichh95] Oswin Aichholzer, Franz Aurenhammer, David Alberts, and Bernd Gärtner. A Novel Type of Skeleton for Polygons. *Journal of Universal Computer Science, http://www.iicm.edu/jucs_1_12, Institute for Image Processing and Computer Supported New Media*, 1(12):752–761, 1995.

[Felke97a] P. Felkel and J. Janáček. Biomedical imaging application of Explorer. In T. Plachetka and A. Ferko, editors, *SCCG 1997, Proceedings of Spring Conference on Computer Graphics – Company Presentions Posters*, volume 2, pages 19–20, Budmerice, Slovakia, June 5–8 1997. Comenius University Bratislava.

[Felke97b] P. Felkel and J. Janáček. Custom-Made Modules of the Explorer Programme Package for Image/Volume Processing. In V. Eck, editor, *Information Technology Applications in Biomedicine – ITAB'97*, pages 90–91, Piscataway, NJ, September7–9 1997. IEEE.

[Felke98] P. Felkel and Š. Obdržálek. Straight skeleton computation. In Lázló Szirmay-Kalos, editor, *Spring Conference on Computer Graphics*, pages 210–218, Budmerice, Slovakia, April 23–25 1998.

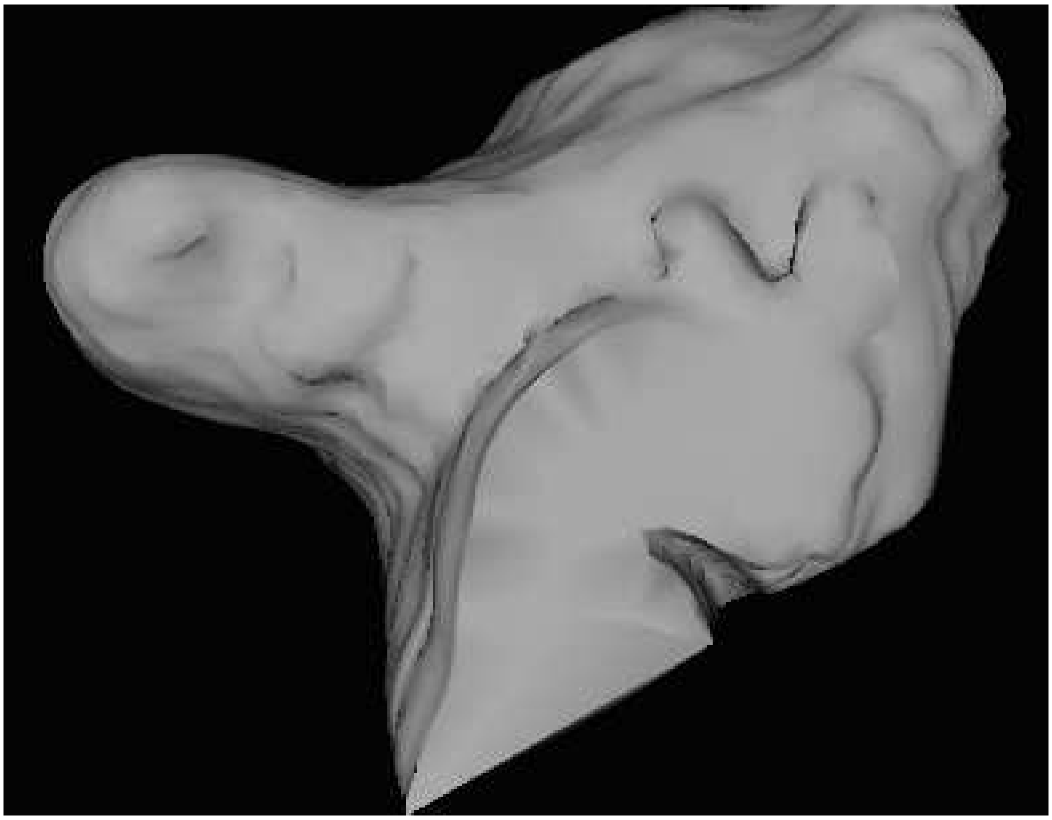

Figure 6: Example of 3D reconstruction performed by direct reconstruction from contours

[Jan98] J. Janáček and L. Kubínová. 3D reconstruction of surfaces captured by confocal microscopy. To appear in *Acta Stereologica*, 1998.

[Loren87] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics (SIGGRAPH '87 Proceedings)*, 21(4):163–169, July 1987.

[Oliva95] J.-M. Oliva. *Reconstruction Tridimentionnelle d'Objets Complexes à l'aide de Diagrammes de Voronoï Simplifiés.* No 130 gd, Thése de l'Université et de l'École des Mines de Saint-Etienne, October 1995.

[Oliva96] J.-M. Oliva, M. Perrin, and S. Coquillart. 3D Reconstruction of Complex Polyhedral Shapes from Contours using a Simplified Generalized Voronoï Diagram. *Computer Graphics Forum*, 15(3):C–397–C–408, 1996.

[vrm] http://www.cgg.cvut.cz/VRML.

APPENDIX A: SURFACE AREA AND VOLUME COMPUTATION

The **surface of a triangulated object** S is computed simply as a sum of areas of N single triangle patches S_t defined by points $A(a_x, a_y, a_z)$, $B(b_x, b_y, b_z)$, $C(c_x, c_y, c_z)$. In 3D space, it can be done as follows:

$$S = \frac{1}{2}\sum_N S_t, \text{where :} \tag{1}$$

$$S_t = \sqrt{(S1^2 + S2^2 + S3^2)} \text{ and}$$

$$S1 = \begin{vmatrix} a_x, a_y, 1 \\ b_x, b_y, 1 \\ c_x, c_y, 1 \end{vmatrix}, \quad S2 = \begin{vmatrix} a_x, a_z, 1 \\ b_x, b_z, 1 \\ c_x, c_z, 1 \end{vmatrix},$$

$$S3 = \begin{vmatrix} a_y, a_z, 1 \\ b_y, b_z, 1 \\ c_y, c_z, 1 \end{vmatrix}$$

The signed **volume of a closed triangulated object** V is computed as the sum of mixed products $[\vec{a}\vec{b}\vec{c}]$ of radiusvectors of triangle vertices $A(a_x, a_y, a_z)$, $B(b_x, b_y, b_z)$, $C(c_x, c_y, c_z)$. N denotes the number of triangle patches.

$$V = \frac{1}{6}\sum_N \begin{array}{c} (a_x b_y c_z \;+\; a_y b_z c_x \;+\; a_z b_x c_y \\ -\, a_z b_y c_x \;-\; a_x b_z c_y \;-\; a_y b_x c_z\;) \end{array} \tag{2}$$