Reconstruction and Representation of Tubular Structures using Simplex Meshes

Alexander Bornik Bernhard Reitinger Reinhard Beichel

Institute for Computer Graphics and Vision
Graz University of Technology
Inffeldgasse 16/II, A–8010 Graz, Austria
contact: bornik@icg.tu-graz.ac.at

ABSTRACT

Modelling and reconstruction of tubular objects is a known problem in computer graphics. For computer aided surgical planning the constructed geometrical models need to be consistent and compact at the same time, which known approaches cannot guarantee. In this paper we present a new method for generating compact, topologically consistent, 2-manifold surfaces of branching tubular objects using a two-stage approach. The proposed method is based on connection of polygonal cross-sections along the medial axis and subsequent refinement. Higher order furcations can be handled correctly.

Keywords

mesh generation, branched tubular objects, simplex meshes, medical context

1 INTRODUCTION

Modelling surface representations of tubular objects has been addressed by many researchers in the field of computer graphics leading to numerous approaches. Since many anatomical structures like vessels are of tubular type, medical applications are one important application field for such techniques. Arrangements of such tubular tubular structures like the portal vein tree inside the liver or airway trees are highly complex and heavily branched. New algorithms are able to detect even very thin tubular structures [Beich04] in e.g. CT images, leading to medial axis representations organized in graphs, which are not suitable for direct rendering. Therefore surface models, which can be efficiently rendered need to be set up first for applications like computer aided surgical planning.

Medical application require consistent surface models without gaps or self-intersections. In case the model is a polygonal or triangular mesh, it should be a 2-manifold. The mesh must be adaptive to account for changing diameter. The overall primitive count should be small, while still accurately representing small geometrical details. The presented work was done in the context of developing a virtual liver surgery planning system [Borni03] supporting physicians with tools for planning and staging of liver tumor resections using a virtual reality (VR) environment. This application involves visualization of the portal vein inside the liver. Since VR techniques are used, the graphical representation of the portal vein needs to be compact in order to meet the real-time requirement, while still providing the physicians with the correct topological information needed for planning a surgical intervention.

WSCG SHORT papers, ISBN 80-903100-9-5
WSCG'2005, January 31–February 4, 2005
Plzen, Czech Republic.
Copyright UNION Agency Science Press

2 RELATED WORK

The probably most popular surface reconstruction algorithm is named Marching Cubes (MC) [Loren87]. It can be used for direct reconstruction of surface models from 3D data organized in a grid and is often used in the medical context. However, one would have to convert a centerline representation to a voxel dataset first, in order to employ MC. Besides, MC is not adaptive and produces complex geometry. For VR applications mesh compression has to be employed, introducing processing overhead. Staircase artifacts produced by MC must be removed using filtering techniques. Connectivity problems for 26-connected structures, which are not acceptable in surgical planning, require a modified algorithm [Lewin03]. Besides the MC there are generic algorithms involving deformable models directly [Delin99], which is time consuming for tubular objects. A second class of algorithms involves surface tiling from cross-sections and algorithms connecting tubular objects to a single model [Oliva96, Meyer92, Lluch04]. Most of these algorithms have problems representing furcations of higher order correctly, some do not support changing radii along branches. An algorithm related to ours can be found in [Felke04], however the technical solution is different. Since the approach is not based on a deformable model, mesh refinement is not possible. An approach mainly focussing on rendering performance, not mesh consistency, can be found in [Hahn01].

3 METHODS

The proposed algorithm generates compact surfaces representations of branched tubular objects. It can handle bifurcations, trifurcations and higher order furcations. The output is a topologically consistent 2-manifold mesh. Geometrically our approach is based on a simplex mesh structure [Delin99]. The mesh for the tubular object is constructed in two stages:

1. **An initial mesh is constructed** connecting cross-section polygons along the medial axis.
2. **Mesh Refinement** based on simplex meshes as deformable model.

3.1 Preprocessing

We use a directed acyclic graph data structure as input data for mesh initialization. The nodes of the graph are either branching points or endpoints. Each node stores its location, local radius, cross-section normal and connected links. Links (branches) store their endpoints, direction and a list of intermediate centerline points with cross-section radii and normals. This data is provided by a vessel-mining approach [Beich04]. Since a deformable model is employed in the second stage of our algorithm, limiting to circular cross-sections during the initialization is possible without loss of generality. Spheres (with cross-section radius) spanned by neighboring centerline points and radii can intersect. Sometimes a whole child branch is located within the sphere at its parent's branching point, which can lead to artifacts. Preprocessing by thinning leads to initial meshes almost free of visual artifacts, since possibly overlapping cross-sections are removed. Thinning also makes initial meshes compact, because the number of remaining cross-sections directly influences the number of mesh polygons. Figure 1 shows details of the thinning process.

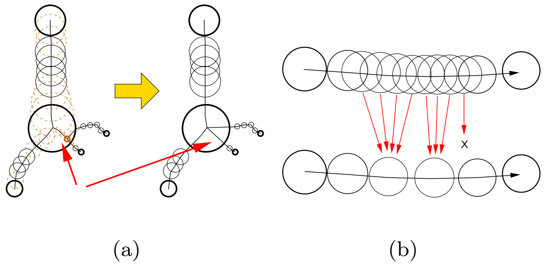


Figure 1: (a) Thinning near branching points: Cross-sections intersecting the influence area of a branching point (sphere) are removed. Sometimes this requires graph restructuring (see arrows). (b) Removal of centerline points within branches.

3.2 Initial Mesh Setup

The initial mesh is set up by traversing the preprocessed input data in a breadth-first-search manner starting with a simplex mesh cylinder at the root node.

Connection of cross-sections

Further cross-sections of the root branch are successively connected to the existing geometry at the top polygon. This is achieved by connecting each vertex of the cross-section polygon to the existing geometry as shown in Figure 2. According to this schema, each inserted cross-section of the tubular object has to have the same number of vertices as the polygon it connects to. This limitation can be overcome by either not splitting each edge to connect to or inserting intermediate polygons. For maximum visual quality, polygons are oriented so that the overall connection edge length is minimized. The visual quality can be further improved by placing additional vertices at the barycenter of the triangles formed by the base edge of the polygon to connect to and the connecting cross-section

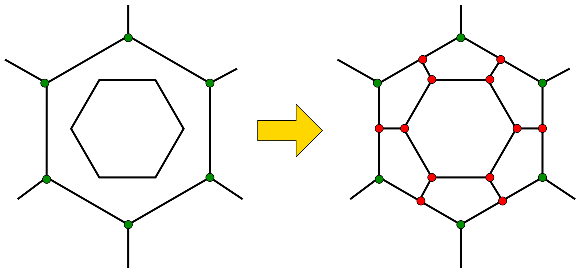


Figure 2: Polygon insertion schema.

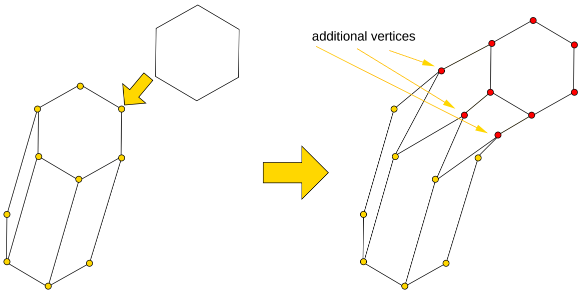


Figure 3: Polygon insertion operation in 3D.

vertex. Figure 3 shows a 3D example.

Connection of branches

New branches are added by connecting their first medial axis cross-section to the existing geometry. The best-suited polygon is found in the vicinity of the branching point, the so-called *hot spot region*, which is maintained for all branching points. When it comes to connecting the first cross-section i with center $\vec{c}_i$ and normal $\vec{n}_i$ of a new branch, the following criterion is calculated for each hot spot region polygon p with $\vec{c}_p$:

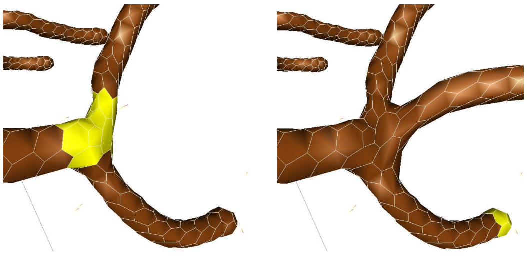
$$\zeta_i = \frac{1}{\|\vec{c}_p - \vec{c}_i\|}\vec{n}_p \cdot \vec{n}_i$$

The polygon that maximizes ζ_i is connected using the same technique as for intermediate cross-sections. Figure 4 shows the hot spot region of a tubular mesh before connecting a new branch as well as how the same region looks like in the final mesh.

3.3 Mesh Refinement

The overall visual quality of the mesh and particularly the quality near high order furcations is improved using the simplex mesh as a deformable model based on a Newtonian law of motion, which is iteratively solved:

$$\vec{P}_i^{t+1} = \vec{P}_i^t + (1 - \gamma)(\vec{P}_i^t - P_i^{t-1}) + \alpha_i \vec{F}_{int} + \beta_i \vec{F}_{ext}$$



(a)

(b)

Figure 4: (a) Simplex mesh of a tubular objec before connecting a new branch. The hot spot region is marked bright. (b) Mesh after the new branch has been added.

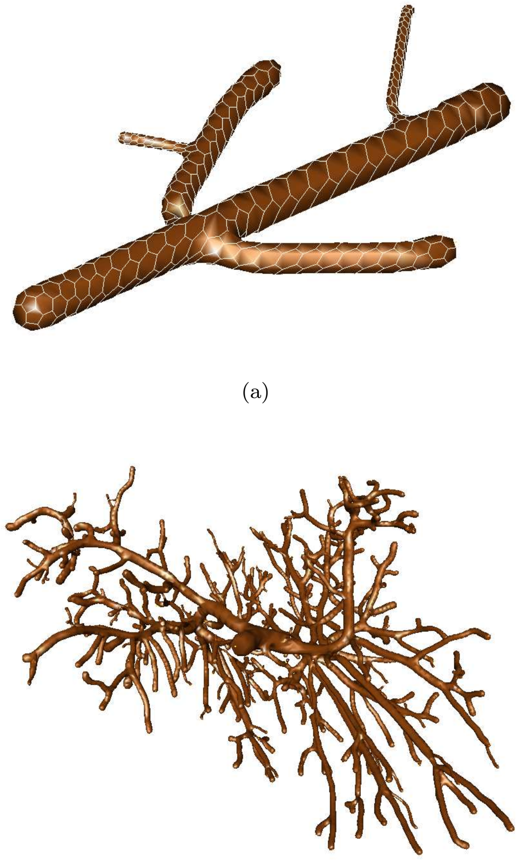
Each vertex $\vec{P}_i$ is displaced based on internal regularizing forces $\vec{F}_{int}$ and external forces F_{ext}. α_i and β_i are control parameters, while γ is a damping factor. If only a medial axis representation is available external $\vec{F}_{ext}$are calculated towards sampling points defined by the cross-sections and radii. Performing several iterations having centerline/radius input mainly results in geometrical changes near branching points. In case of a present binary segmentation in form of a 3D volumetric dataset, external forces are calculated by searching the closest boundary voxel in the direction of each vertex' normal. In this case mesh iterations also affect regions, with other than circular cross-section shape, leading to more accurate results.

4 RESULTS

We have tested our algorithm with different types of medical datasets, including the vasculature of the liver, sheep and human airway trees and data from high-resolution scans of portal vein corrosion casts. Our approach produced consistent and visually appealing meshes in all cases. The number of polygons produced by our approach is the same order of magnitude as output produced by a simplified and filtered MC approach. Figure 5 shows results for the datasets described in Table 1. Table 2 gives an overview of the mesh complexity and timings of the proposed algorithm.

5 CONCLUSION

This paper presented an algorithm for generating compact consistent 2-manifold meshes of branched



(a)

(b)

Figure 5: Results: (a) DS 1: Small artificial tubular dataset. (b) DS 2: Human portal vein tree.

DS	Description	#b	#cs
1	Artificial	8	64
2	Human portal vein	693	2680

Table 1: Test dataset overview: $\#b$: number of branches. $\#cs$: number of input cross-sections.

DS	#vtx	#pol	t_i	t_r	n
1	950	477	28	25	40
2	24199	48394	527	1738	10

Table 2: Complexity/timings: $\#vtx$, $\#pol$: number of vertices/polygons. t_i: mesh initialization time in milliseconds. t_r time for one iteration in the refinement stage. n: number of iterations performed for the images in Figure 5.

tubular objects based on contour-connection using simplex meshes. The visual quality of the output mesh is improved in an optional refinement step.

ACKNOWLEDGMENTS

This work was supported by the Austrian Science Foundation (FWF) under grant **P17066-N04**.

References

[Beich04] R. Beichel, T. Pock, Ch. Janko, and et al. Liver segment approximation in CT data for surgical resection planning. In J. Michael Fitzpatrick and Milan Sonka, editors, *In SPIE Medical Imaging '04*, volume 5370, pages 1435–1446, San Diego, 2004. SPIE Press.

[Borni03] A. Bornik, R. Beichel, B. Reitinger, and et al. Computer aided liver surgery planning: An augmented reality approach. In R.L. Galloway Jr., editor, *In SPIE Medical Imaging '04*, volume 5029, pages 395–406. SPIE Press, May 2003.

[Delin99] Herve Delingette. General object reconstruction based on simplex meshes. *International Journal on Computer Vision*, 32(2):111–146, 1999.

[Felke04] Petr Felkel, Rainer Wegenkittl, and Katja Bühler. Surface models of tube trees. In *Proceeding of the Computer Graphics International CGI'04*, 2004.

[Hahn01] Horst K. Hahn, Bernhard Preim, Dirk Selle, and Heinz Otto Peitgen. Visualization and interaction techniques for the exploration of vascular structures. In *Proceedings of the conference on Visualization '01*, pages 395–402. IEEE Computer Society, 2001.

[Lewin03] T. Lewiner, H. Lopes, A. Wilson Vieira, and G. Tavares. Efficient implementation of marching cubes: Cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003.

[Lluch04] J. Lluch, R. Vivó, and C. Monserrat. Modelling tree structures using a single polygonal mesh. *Graphical Models*, 66(2):89–101, March 2004.

[Loren87] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[Meyer92] David Meyers, Shelley Skinner, and Kenneth Sloan. Surfaces from contours. *ACM Trans. Graph.*, 11(3):228–258, 1992.

[Oliva96] J.-M. Oliva, M. Perrin, and S. Coquillart. 3D reconstruction of complex polyhedral shapes from contours using a simplified generalized voronoi diagram. *Computer Graphics Forum*, 15(3):397–408, 1996.