

Survey of Cardiovascular Visualization Techniques

Christian Brändle & Niko Leopold

TU Wien

Abstract

The cardiovascular system and its components have been vastly studied and a lot of visualization techniques to study the heart or the arteries of the human body have been proposed. In this STAR report, you need to conduct a survey on important medical visualization applications that cover the topic of cardiovascular visualization, to propose a taxonomy of the existing applications, and to provide an overview of yet-unaddressed or challenging topics.

Categories and Subject Descriptors (according to ACM CCS): 1.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism—Color, shading, shadowing and texture; 1.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—;

Keywords: Cerebral arterial system, geometric modeling, vascular system modeling, vessel visualization, convolution surfaces, implicit modeling, vessel reconstruction method, Level Sets, Quad dominant meshes

1. Introduction

2. Surface visualization techniques

In this section we will focus on different surface visualization techniques. Roughly we can divide surface mesh generation into *model-based* and *model-free*.

The model-based mesh generation makes some assumptions about the model, in our case about the vascular system. In particular it makes assumptions about the topological structure, the observed junctions and the profiles of vessels. The topological structure is mostly represented by a graph and limited to a tree like representation. Also the observed junctions often are limited to certain kinds of junctions like bifurcations. Finally the profiles of the vessels are simplified to circular or elliptical structures. All this simplifications and assumptions can help to speed up the process of surface generation, to make it more reliable or more visually pleasing or even transform the underlying data into more usable data for further processing. The downside is that model-based meshes do not resemble the underlying data as accurate as it could be done and also sometimes simplify the underlying data too much such that essential features are lost. Especially when diagnoses of vascular degenerations or similar detail based inspections need to be done model-based approaches may not provide enough vivid information of the underlying data.

Here model-free mesh generation comes into play. The goal here is to retain as much of the underlying data as possible. But without an underlying model the extraction and creation of the mesh is more time consuming and cumbersome. Moreover unwanted results can arise if the underlying data suffers from noise or other failures in capturing. So care must be taken at the decision what for data is extracted and finally represented in the resulting mesh.

From the literature observed we determined that most of the methods require at some points a skeletonization of the underlying data. For now we assume that the underlying data consists of a voxel model that represents a density field of captured values. As model-based methods require a model, all of the observed methods require an extraction of a skeleton prior to the actual mesh generation. In

regards of skeletonization the interested reader is referred to Ebert et al. [EBN02] or Strzodka et al. [ST04].

As seen model-based mesh generation is based more or less on geometrical assumptions and reasoning on the model. Volkau et al. [VZB*05] for instance models a cerebral arterial model from segmented and skeletonized angiographic data. He constructs a vascular structure consisting of tube segments and bifurcations, so this model-based approach limits the domain of representable data quite hard in regards of branching possibilities. The centerline is smoothed with a sliding average filter to prevent problems with outliers and the two kinds of topologies, namely tubular and B-Subdivision based ones, are combined to form the complete vessel structure.

A connected graph is established that captures also the blood flow direction. This is done by splitting it up into two connected tree structures where one is designated as inflow structure. The model shapes the tube segments with decreasing radius with respect to the bloodflow as well as it captures radii change at bifurcations. This is based on statistical data. Radii at bifurcations are approximated with an exponential function whereas radii evolution at tubular parts are done via linear regression. At connection points C1 continuity is established via further constraints. So no abnormal cross section shape or diameter will show up easily in the final result.

The bifurcations themselves are constructed based on B-subdivision. The generation of quadrangles of a bifurcation till the overlapping internal part is shown in figure 1. Internal parts are subdivided to fit the outer subdivision. Problems with this scheme arise when all three vessels at the bifurcation form a close to orthogonal connection, where the inner face will be undefined. Also the B-subdivision of tubular segments show problems, namely self intersections when the radius of curvature is smaller than the ra-

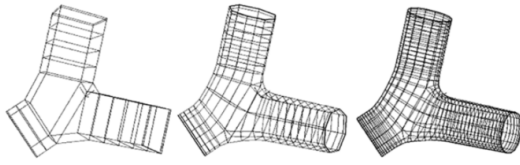


Figure 1: Example of refinement with B-subdivision.

[VZB*05]

dius of the tube, shown in figure 2. Further problems mentioned are 'oversampling' or unrealistic sampling of bifurcations according to a too coarse sample pattern or problems with centerline smoothing in very jaggy areas as an acceptable smoothing also will remove essential features of the underlying vascular structure.

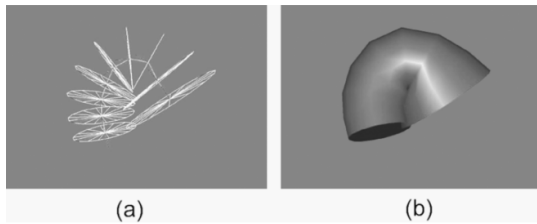


Figure 2: Fold of vessel when curvature is higher than tube radius.

[VZB*05]

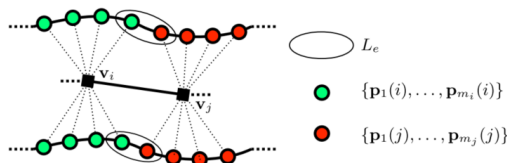


Figure 3: Estimation of thickness of graph edges.

[SEEK12]

Another geometry based modelling approach is provided by Sibbing et al. [SEEK12]. His work is focusing on quad meshes as they are visually more pleasant, require fewer elements and can produce more stable results in further processing, like FEM. Here the topologically correct extraction of the branching structure is done from a post-processed triangle mesh that is generated from an adapted *Marching Cubes* algorithm based on the interior voxels which are identified via a variant of the *Level Set* method. The extracted mesh is shrunk to a close to one dimensional structure that serves as basis for a tree based graph as well as for radius estimation along its edges, see figure 3. Further the graph is used to subdivide the triangle mesh into tubular parts and parts with one or more furcations with appropriate placed cutting planes. The so called *interfaces* are 2D polygons between adjacent segments. They are defined by the intersection of cutting planes with the triangular mesh.

Quad meshes for the junctions are created with the *Mixed Integer Quadrangulation* by Bommers et al. [BZK09]. The quads are

just required to align with the defined boundary to generate radially arranged quads.

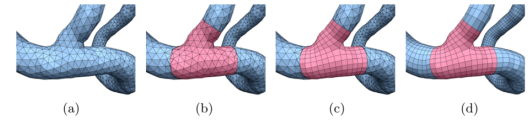


Figure 4: (a) The input to our remeshing stage is a smoothed and resampled version of the mesh extracted using the *Marching Cubes* Algorithm. (b) The mesh is partitioned into junction and tube components. The cuts insert additional vertices at the respective interfaces. (c) We remesh the junctions yielding a fair quad layout. (d) Tubes are remeshed, such that the final mesh is a quad dominant mesh.

[SEEK12]

Tubes now only have to provide a smooth transition between two interfaces with a possibly different number of vertices. The rings of the tubes are generated out of equidistant iso-contours on the harmonic scalar field on the tube. Triangles are introduced between so called transition rings to compensate for the difference in vertices. The distribution of the transition rings as well as the distribution of the triangles around the tube in a transition case is done via adaption of the Bresenham line rasterization algorithm. Stretching of quads along the tube axis as well as rotations of the rings to get configurations with almost 90 degree angles in the quads, yields close to optimal quad dominant meshes. The advantage of this reconstruction is the smoothing of unwanted artifacts on the original triangle mesh as well as the already shown properties of quad meshes, see figure 4.



Figure 5: Transition at branching.

[OP05]

A technique that uses the vessel skeleton as input data but only respects the diameter information at the provided graph is called *convolution surfaces*. The main idea is that *implicit surfaces* describe the original surface by equations to model smooth, deformable objects. Oeltze et al. [OP05] used a combination of so called 'blobs' to model an *iso-surface* out of this scalar field function. The resulting convolution surface is a surface of an object around its skeleton. The convolution itself is done on an integral of 'blob-functions' along the skeleton. It is actually the modification of a signal by a filter, in this case a Gaussian filter. To speed up the process filter functions with finite support are used and the scalar field

is only computed within certain bounding volumes around the line segments. Unwanted blending at branchings is avoided by more narrow filter kernels, see figure 5. Blending between segments that are close but not connected is avoided by restricted support on the skeleton, see figure 6.



Figure 6: Unwanted blending.

[OP05]

Still the problems of convolution surfaces are that they only support a circular base shape at tubes, that unwanted bulging can be suppressed but not completely removed and that unwanted blending of close parts is still possible. Finally the end segments have always the shape of a half sphere which neither represents the normally open ends of vascular trees nor has low deviation from the original surface in that areas.

In contrast model-free mesh generation mostly rely on some implicit functions that should resemble the input data instead of explicit constructive reasoning in geometric elements like tubes and junctions. Also here there are different techniques that respect the original shape more or less. Convolution surfaces therefore form a bridge between model-based and model-free mesh generation as the technique already use an implicit function but limit the shape in a model-based fashion.

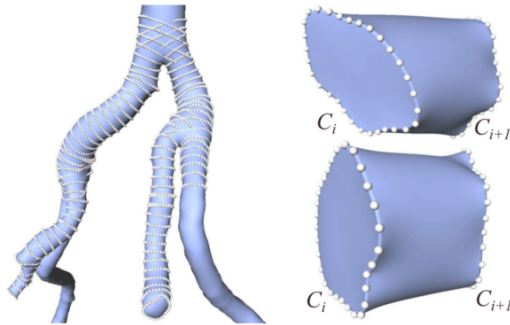


Figure 7: Reconstruction of an arterial tree from free-form contours.

[KBT*12]

To capture vascular profiles and open endings of vascular structures Kretschmer et al. [KBT*12] uses an adaptive modelling with non circular cross sections. He generates intersection-free surfaces from centerlines with complex outlines, see figure 7. He decomposes the centerline description into segments which are locally described by implicit functions and combines them using boolean

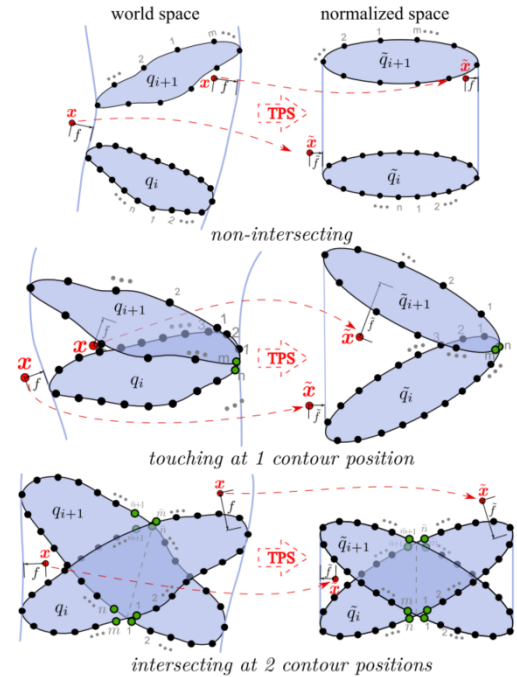


Figure 8: Mapping of neighboring free-form contour pairs to normalized shapes.

[KBT*12]

operations. In this fashion all intersection- and furcation-related issues are solved in similar fashion to convolution surfaces with the difference that the capturing of the vascular outline is not so limited. A watertight scale-adaptive mesh is generated out of an adaptive octree in the end. The key difference to CS is that the so called *admissible distance functions* (ADFs) just form a class of functions with the property to detect valid iso-surfaces. To find a topology-preserving mapping that forms a intersection free surface part between two adjacent contours on the vascular structure found by the ADFs they follow the idea of *thin plate splines*. The assignment of contour pairs to a limited set of target shapes guarantee a valid surface topology when the contours do not touch, touch or penetrate each other as shown in figure 8.

The octree that should capture also the smallest vessels is recursively subdivided at surface intersections with the grid. To speed up queries on the octree a localized approximation of the implicit function with a certain threshold to safely prune cells is used. To improve the quality of the extracted mesh a bisection method is used to interpolate between values when marching cubes is applied, advantages of this scheme is shown in figure 9.

Finally Wu et al. [WWL*10] also generates scale-adaptive surfaces from vascular structures with similar properties. He extracts the vascular boundary voxels from the segmentation and use them to build a 3D point cloud where normals are estimated via covariance analysis. Then a 3D implicit indicator function is computed from the point cloud by solving a Poisson equation. Finally the surface is generated by adaptive polygonization. The surface

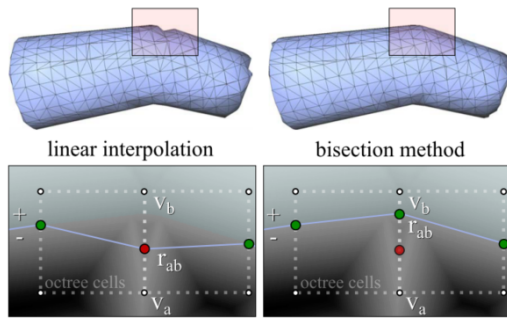


Figure 9: Improved root-finding of the iso-surface using bisection method.

[KBT*12]

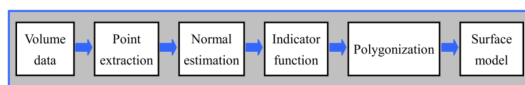


Figure 10: Pipeline of geometric modeling of vascular tree structures.

[WWL*10]

is a smooth, morphologically topology correct two-manifold that is scale-adaptive to the local curvature and has fewer and better shaped triangles compared to earlier approaches. The pipeline used is shown in figure 10.

To reconstruct very thin vessel structures an adaptive point extraction is used that relies on the constellation of adjacent object voxels. Normals are estimated on k -nearest neighbours of a point in the point cloud and the smallest eigenvalue of their covariance matrix. The implicit function that approximates the surface of the point samples is a Poisson surface reconstruction. It utilizes the vector field and generates a function gradient best approximates the normal field of the point cloud.

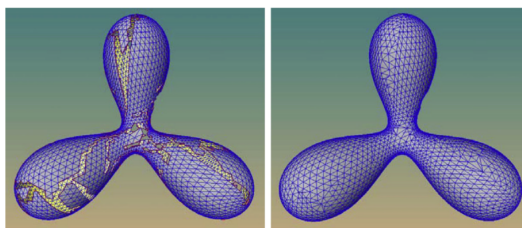


Figure 11: Polygonization of a trifurcate model. A long gap is produced upon the termination of mesh expanding stage (left), and is sewn in the subsequent gap-stitching stage (right).

[WWL*10]

The polygonization is done by an advancing front algorithm utilizing Newton step method. Care has been taken to ensure that the minimal angle of triangles is maximized in the triangulation step. The resulting gap on the surface is then closed via a stitching opera-

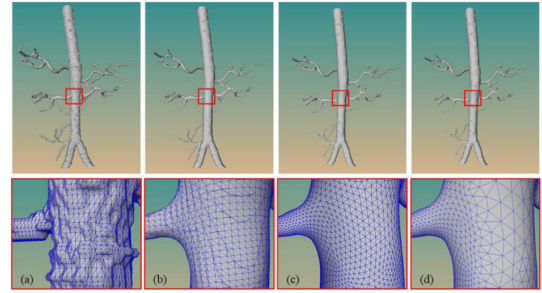


Figure 12: Comparison of triangle quality for an aorta tree. Surface model generated by the marching cubes (a), multi-level partition unity based method (b), subdivision surface based method (c) and scale-adaptive surface method (d). The bottom row is a zoomed region corresponding to the rectangle region of the top row.

[WWL*10]

tion, see figure 11. Also here the patching triangles are subdivided to a similar size than the surrounding ones. To further refine and smooth the mesh a Loop based subdivision is performed. A comparison of several methods is shown in figure 12. It can be seen that the scale-adaptive nature of the reconstructed mesh together with the smooth and but still good approximation of the underlying shape yields very good results on vascular structures.

2.1. Conclusions

References

- [BZK09] BOMMES D., ZIMMER H., KOBELT L.: Mixed-integer quadrangulation. *ACM Transactions On Graphics (TOG)* 28, 3 (2009), 77. 2
- [EBN02] EBERT D., BRUNET P., NAVAJO I.: An augmented fast marching method for computing skeletons and centerlines. 1
- [KBT*12] KRETSCHMER J., BECK T., TIETJEN C., PREIM B., STAMMINGER M.: Reliable adaptive modelling of vascular structures with non-circular cross-sections. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1055–1064. 3, 4
- [OP05] OELTZE S., PREIM B.: Visualization of vasculature with convolution surfaces: Method, validation and evaluation. *IEEE transactions on medical imaging* 24, 4 (2005), 540–548. 2, 3
- [SEEK12] SIBBING D., EBKE H.-C., ESSER K. I., KOBELT L.: Topology aware quad dominant meshing for vascular structures. In *Workshop on Mesh Processing in Medical Image Analysis* (2012), Springer, pp. 147–158. 2
- [ST04] STRZODKA R., TELEA A.: Generalized distance transforms and skeletons in graphics hardware. In *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization* (2004), Eurographics Association, pp. 221–230. 1
- [VZB*05] VOLKAU I., ZHENG W., BAIMOURATOV R., AZIZ A., NOWINSKI W. L.: Geometric modeling of the human normal cerebral arterial system. *IEEE transactions on medical imaging* 24, 4 (2005), 529–539. 1, 2
- [WWL*10] WU J., WEI M., LI Y., MA X., JIA F., HU Q.: Scale-adaptive surface modeling of vascular structures. *Biomedical engineering online* 9, 1 (2010), 75. 3, 4