

Implicit Vessel Surface Reconstruction for Visualization and CFD Simulation

Christian Schumann¹, Mathias Neugebauer², Ragnar Bade², Bernhard Preim², Heinz-Otto Peitgen¹

¹ MeVis Research GmbH, Bremen, Germany e-mail: christian.schumann@mevis.de

² Department of Simulation and Graphics, Otto-von-Guericke University of Magdeburg, Germany

Received: date / Revised version: date

Abstract *Objective:* Accurate and high-quality reconstructions of vascular structures are essential for vascular disease diagnosis and blood flow simulations. These applications necessitate a trade-off between accuracy and smoothness. An additional requirement for the volume grid generation for Computational Fluid Dynamics (CFD) simulations is a high triangle quality. We propose a method that produces an accurate reconstruction of the vessel surface with satisfactory surface quality.

Methods: A point cloud representing the vascular boundary is generated based on a segmentation result. Thin vessels are subsampled to enable an accurate reconstruction. A signed distance field is generated using Multi-level Partition of Unity Implicits and subsequently polygonized using a surface tracking approach. To guarantee a high triangle quality, the surface is remeshed.

Results: Compared to other methods, our approach represents a good trade-off between accuracy and smoothness. For the tested data, the average surface deviation to the segmentation results is 0.19 voxel diagonals and the maximum equi-angle skewness values are below 0.75.

Conclusions: The generated surfaces are considerably more accurate than those obtained using model-based approaches. Compared to other model-free approaches, the proposed method produces smoother results and thus better supports the perception and interpretation of the vascular topology. Moreover, the triangle quality of the generated surfaces is suitable for CFD simulations.

Introduction

The reconstruction of vascular structures deserves special attention since vascular trees are among the most complex structures of the human body. An accurate and high-quality reconstruction is essential for applications in the field of diagnosis and blood flow simulation [1]. In diagnosis of vascular diseases, the local evaluation of the vessel cross section is essential to detect and characterize

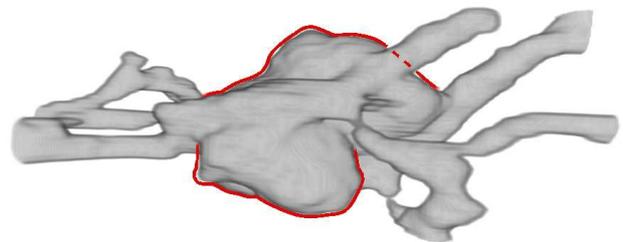


Fig. 1 Volume rendering of a cerebral aneurysm. The silhouette is emphasized to illustrate the highly irregular morphology of the aneurysm sack.

narrowings, such as a stenosis as well as widenings, e.g. an aneurysm. A high quality in terms of surface smoothness is important to avoid distractions caused by surface artifacts.

Furthermore, a high surface quality as well as high reconstruction accuracy is crucial for *Computational Fluid Dynamics (CFD)* to guarantee correct simulation results and to avoid numerical instabilities. Simulations of the blood flow enable the study of hemodynamic characteristics such as intra-aneurysmal flow patterns or the wall shear stress. The results could be used to decide if an aneurysm has to be treated by coiling or stenting. We do not discuss issues such as Non-Newtonian characteristics of blood flow, elastic behavior of the vessel wall or the appropriateness of laminar flow conditions. These issues are discussed e.g. in [1]. Instead we focus on general properties of grids in the preprocessing for blood flow simulations. An additional important prerequisite for a CFD simulation is a high triangle quality in terms of edge ratio. Thin and elongated triangles may cause numerical instabilities and need to be avoided. Moreover, the triangle size should not change abruptly. A higher resolution in areas with high curvature is desirable, however smooth transitions in triangle quality are required.

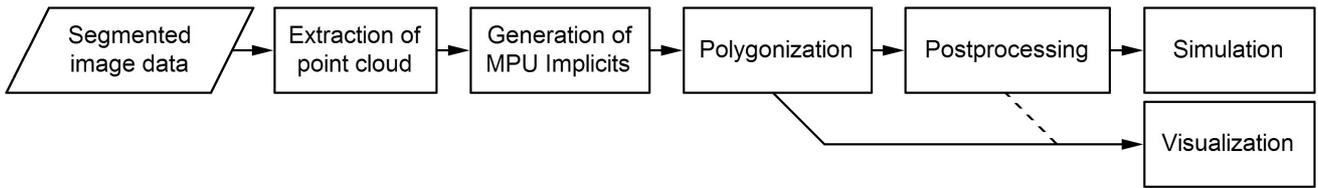


Fig. 2 Vessel surface reconstruction pipeline employing MPU Implicits. The results of the polygonization are suited for visualization tasks. The generation of volume grids for CFD simulations requires several post processing steps.

Model-based vessel visualization techniques are well established in the field of therapy planning. The reconstruction is based on model assumptions, in particular a circular cross-section is often assumed. A high surface quality can be achieved by explicitly fitting graphics primitives to the centerline, (see, e.g. [2] where truncated cones are employed) or by using implicit representations, (see, e.g. [3] where *Convolution Surfaces* are employed, see Figure 14 b). However, the accuracy of the reconstruction is not sufficient for diagnosis and blood flow simulation. Pathologic structures can not be captured correctly using circular cross sections since the morphology of such structures is highly irregular and does not exhibit a rotational symmetric shape (see Figure 1). In addition, the cross section of the vessel lumen might deviate from a circle (shaped like an ellipse or even like an "8") depending on the transmural blood pressure. A blood flow computation based on inaccurate surfaces generates misleading results.

A higher accuracy can be achieved using *model-free* approaches like *Marching Cubes* [4]. In principle, those algorithms can be applied directly to the image data and generate a surface based on a threshold. However, this is problematic in the case of vascular structures. Due to image noise and inhomogeneities in the contrast agent distribution this does not always lead to accurate results, in particular for MRI data. Furthermore, small structures might be suppressed depending on the chosen threshold. Hence vascular structures need to be explicitly segmented (e.g. with level sets or snakes) in most cases before a surface can be reconstructed. A subsequent application of a model-free method suffers from strong aliasing artifacts like staircases which might hamper the visual interpretation of the vessel surface and therefore complicate the diagnosis. Furthermore, staircase artifacts are very problematic for CFD simulations since they might lead to numerical problems. Common approaches to smooth these artifacts in the segmentation mask or in extracted surface meshes especially of filigree vascular structures mostly remove relevant detail and yield reduced accuracy [5]. Volume-preserving smoothing approaches like Constrained Elastic Surface Nets constrain the displacement of the vertices to prevent shrinkage [6]. However, even those methods fail to preserve small structures like thin vessels. An up to date overview of vessel visualization techniques can be found in [7].

Method

Pipeline Overview

The proposed reconstruction pipeline is summarized in Figure 2. The input for the pipeline is binary segmented image data that contains vascular structures. In the first step of our pipeline, a point cloud is generated that represents the boundary between the segmented vessel and the background. We developed an adaptive point cloud generation algorithm that allows the faithful reconstruction of small vessels. To generate a signed distance field based on this point cloud, we apply the *Multi-level Partition of Unity Implicits (MPU Implicits)* algorithm that was developed by Ohtake et al. [8]. The polygonization algorithm of Jules Bloomenthal is used to generate a polygonization of this function [9]. The result can directly be used for visualization.

To this point, the pipeline is similar to the method described by Braude et al. [10] but targeted at the appropriate representation of vascular structures. The major difference is the adaptive point cloud generation algorithm that strongly differs from the one proposed by Braude et al. In addition we derived rules for the estimation of appropriate parameter values for the generation and polygonization of the MPU Implicits to ease and speed up the application of our method.

To allow the utilization of the generated surfaces in the context of CFD simulation, we add additional post processing steps. For mesh quality improvement we apply edge collapsing and edge flipping as described in [11]. A reduction of triangles is yielded by using the Advancing Front remeshing algorithm of Schreiner et al. [12]. The result of the post processing can also be used for visualization. This is especially reasonable for the visualization of the simulation results on the surface.

Point Cloud Extraction

The point cloud extraction algorithm is aimed at the reconstruction of small structures like thin vessels. It is driven by the voxel grid of the segmentation result. However, to prevent aliasing artifacts, point positions are not strictly aligned to voxel centers but distributed in the volume of the voxels. We use those background voxels, that are closest to the given object voxels, as the basis of point placement. We refer to those voxels as *outer*

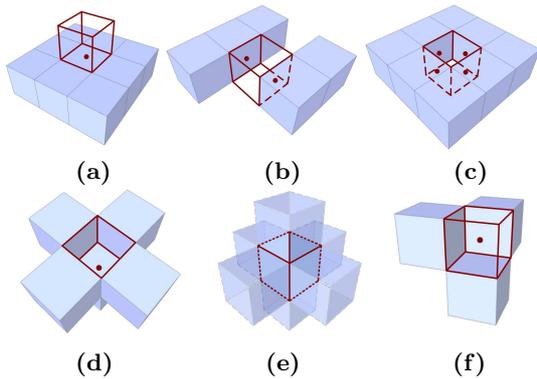


Fig. 3 Placing of points (red) for different constellations of object voxels (blue) in the neighbourhood of an outer boundary voxel (red).

boundary voxels. Based on the 3D-6-neighbourhood $n6(v)$ of an outer boundary voxel v , positions for one or more points within the volume of the voxel are derived along the following rules:

1. If there is only one object voxel in $n6(v)$, e.g. $|n6(v)| = 1$ (Figure 3 (a)), one point is placed in the center of the *boundary face* (face of v that is adjacent to the object neighbour voxel).
2. If $|n6(v)| = 2$ and the voxels are on two opposite sides of v (Figure 3 (b)), two points are placed in the centers of the boundary faces.
3. If $|n6(v)| = 4$ and the voxels are placed in a plane (Figure 3 (c)), four points are placed in the centers of the boundary faces.
4. If $|n6(v)| = 5$, v is in a small cavity (Figure 3 (d)). We place one point on the center of the boundary face that could be considered the ground of the cavity.
5. If $|n6(v)| = 6$, v is a hole (Figure 3 (e)). We ignore this hole and place no points.
6. In all other cases objectvoxels in $n6(v)$ are placed in a staircase formation. One point is placed in the center of v (Figure 3 (f)).

Normal Vector Computation: Once the points for an outer boundary voxel v are placed, the related normal vectors need to be computed for the implicit surface reconstruction. If only one point has been placed (cases 1,4 and 6), the normal vector is computed based on the gradient of v in the segmentation result. If more points

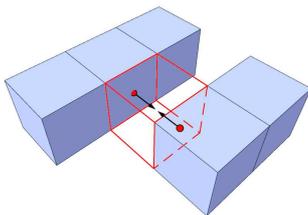


Fig. 4 If several points have been generated for one outer boundary voxel, the normals of the *boundary faces* are used as normal vectors.

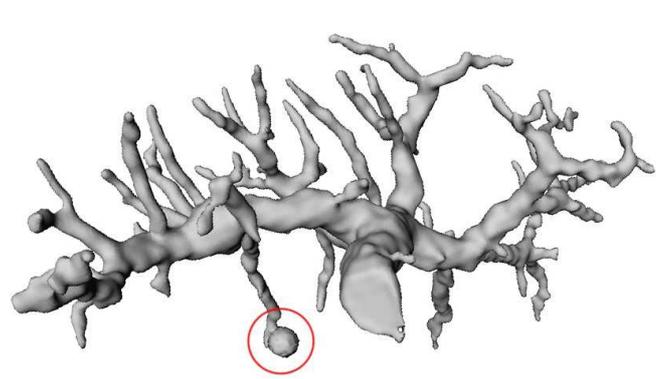


Fig. 5 Spherical artifacts occur at thin structures that are represented by too few points.

were generated (cases 2 and 3) the gradient can not be used, since every point would have the same normal vector. Furthermore, the gradient computation may not be valid under those circumstances because the neighborhood exhibits symmetry (see Figure 3 (c) for example). Since the points are placed in the centers of the boundary faces in those cases, the normal vectors of those faces are used as normal vectors instead (Figure 4).

Representation of Thin Structures: The method, that has been described so far, does not generate enough points to represent thin structures (structures with a cross section smaller than 3×3 voxels). This might lead to inaccuracy and the generation of spherical artifacts during the implicit reconstruction (Figure 5). To increase the density of the point cloud, we subdivide object voxels that represent thin structures into eight subvoxels. First we identify thin structure voxels using a *top-hat-transformation* with a $3 \times 3 \times 3$ structure element (Figure 8 (b)). All outer boundary voxels that are next to thin structure voxels are also subdivided into eight subvoxels. Points and normal vectors are computed for those subvoxels in the same manner as described before for voxels.

Aliasing-artifacts which are introduced by this subsampling step (see Figure 6 (b)) are reduced by labeling additional subvoxels as object-subvoxels before the point placement takes place (Figure 6 (c)). Outer boundary voxels are candidates for the addition of object-

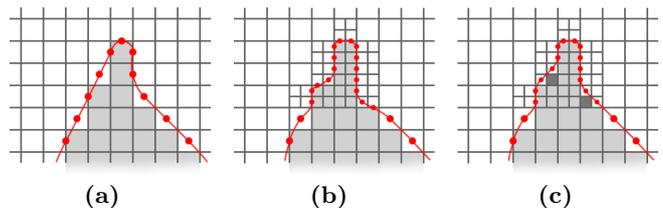


Fig. 6 Due to subsampling, the constellation of points adheres very strictly to the original voxel grid (b). Additional subvoxels are labeled as object-subvoxels to further reduce staircase artifacts (c).

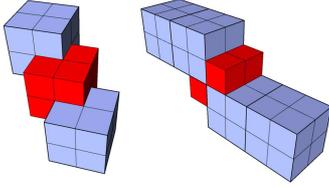


Fig. 7 Examples for the labeling of additional subvoxels (red) as object-subvoxels.

subvoxels. We have to exclude voxels that represent small cavities because the addition of subvoxels would fill those cavities. For all remaining outer boundary voxels, we analyze every of the eight subvoxels independently. For every subvoxel the 3D-18-neighbourhood on the subvoxel-level is gathered. The subvoxel is labeled as object-subvoxel if there is at least one pair of object-subvoxels in the neighbourhood that fulfill *all* of the following conditions:

- The two subvoxels do *not* share any face.
- One of the two subvoxels shares one face with the considered subvoxel.

Examples for the addition of object-subvoxels are depicted in Figures 7 and 8 b. Figure 8 c gives an example for an adaptively subsampled point cloud generated by the algorithm.

Implicit Surface Representation

An implicit surface representation is generated using *Multi-level Partition of Unity Implicits (MPU Implicits)* [8]. This algorithm utilizes an octree to subdivide the point cloud into smaller parts which are approximated by piecewise quadratic functions. A bivariate quadratic polynomial is fitted to points that exhibit small variations in normal direction. If the deviations among normal vectors are large, a general quadric is applied instead. After a local approximation has been determined, its deviation from the points in the cell is calculated. If the deviation surpasses a user defined ϵ_0 , the local approximation is not accurate enough and the cell is further subdivided. This recursive algorithm is terminated when a local approximation is assigned to every leaf cell of the octree. A weighting function is associated with every approximation. The influence of a local approximation at point X depends on the distance between X and the center of the octree cell to which the approximation belongs. The global implicit function is computed as the summation of all weighted local approximations.

The use of two groups of functions, the local shape functions and the associated weighting functions, is the main benefit of MPU Implicits. The computational cost does not depend on the number of points but on the complexity of the described object. Hence MPU Implicits are efficient even for large datasets.

The accuracy and smoothness of the reconstruction can be controlled by several parameters. The octree recursion depth and ϵ_0 have to be adjusted for every dataset. Inappropriate values might lead to inaccuracies, merging of neighboring structures or even the suppression of structures. Since a manual adjustment might be time consuming and tedious, we complement our method with an automatic parameter estimation based on properties of the input image and the generated point cloud (for details see [13]).

Polygonization

For the generation of a polygonal representation of the implicit function we use Bloomenthal’s implicit polygonizer [9]. This surface tracking approach utilizes a cubical space partitioning element which is moved across the surface. The intersections of the edges of the cube with the isosurface determine the vertex positions of the polygonal mesh. To capture even very thin vessels using this surface tracking approach, the size of the space partitioning element has to be chosen carefully. If the element is too large, vessels might penetrate the faces of the cube, but no edges. In that case, the vessel would not be reconstructed. We set the size to 70% of the size of one voxel and chose an isovalue which results in a surface offset of 5% of a voxel diagonal to guarantee the reconstruction of thin vessels. A detailed discussion of the parameter selection is given in [13].

Mesh Quality Improvement: The triangle quality of the results of Bloomenthal’s implicit polygonizer is suited for visualization tasks. However, it is not sufficient for the generation of volume grids for CFD computations due to the underlying space partitioning approach. Size, shape and number of the resulting triangles rather depend on orientation and size of the partition cells than on the actual implicit surface description. In addition to the smooth and artifact-free surface, that is provided by the MPU Implicits, well shaped triangles are crucial to obtain correct CFD results. Deformed triangles with strong deviating edge lengths lead to numerical discontinuities. In the worst case, the simulation process does not converge at all. The homogeneity of triangle sizes is also an important criterion for triangle quality. We discuss this issue in the section *Triangle Number Reduction*.

The equi-angle skewness S is used to describe the triangle quality:

$$S = \max\left(\frac{\alpha - 60^\circ}{120^\circ}, 1 - \frac{\beta}{60^\circ}\right)$$

The differences of the maximal interior angle (α) and the minimal interior angle (β) to the optimal angle of 60 degrees (equilateral triangle) are scaled to the interval

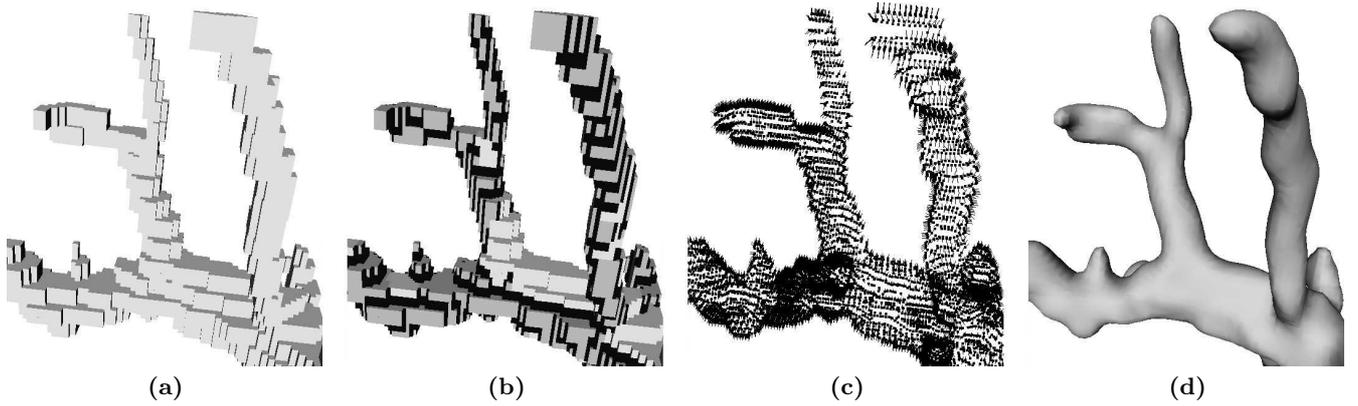


Fig. 8 A detailed look at the surface reconstruction: the segmentation result (a) is adaptively subsampled (b). Additional subvoxels (black) in the neighborhood of thin structures (dark gray) are labeled as object subvoxels. A point cloud with adaptive density (c) is generated and approximated using an implicit surface (d).

$[0, 1]$. The maximum difference characterizes the triangle quality. Well shaped, equilateral triangles provide equi-angle skewness values equal to zero, highly acute-angled and therefore bad shaped triangles yield values converging to one. Our experiments with the simulation software FLUENT (www.fluent.com) indicated that meshes yielding triangle quality values lower than 0.75 represent a good base for CFD simulations.

Triangles which do not comply with this requirement are removed by collapsing too small edges (Figure 9 (a)). Flipping edges leads to further mesh improvement [11]. To preserve important features during this optimization process, feature edges (the angle between the normal vectors of adjacent triangles surpasses a predefined threshold) are treated especially (see Figure 9 (b) and (c)).

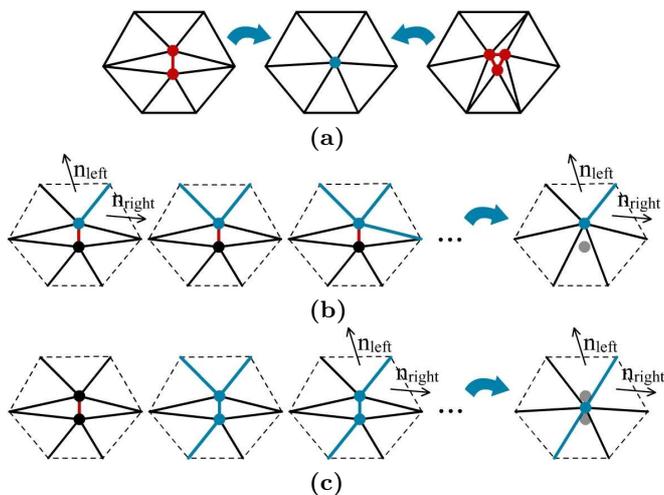


Fig. 9 Mesh improvement: Collapsing of too small edges and triangles (a); Collapsing onto a *feature vertex*, if the corresponding edge provides just one such vertex (b); Collapsing onto the edge center, if it provides no or two feature vertices (c). (From Bade et al. [14])

The resulting mesh of this efficient optimization step yields a good triangle quality, but normally still consists of too many, small triangles. This overrepresentation leads to slow visualization and also raises the computational effort during CFD simulations unnecessarily.

Triangle Number Reduction: To reduce the number of triangles and preserve important surface features, an *Advancing Front* algorithm is applied (see [15] for an introduction to the Advancing Front method). This class of surface meshing algorithms starts at a seed point and iteratively grows the triangulation across the surface. The seed point is projected onto the implicit surface using gradient based iterative approaches like Newton Step. An initial part of the mesh is created by projecting additional, adjacent points onto the surface. The outer boundary edges of the resulting mesh structure form the initial *front*. This front separates meshed and unmeshed regions. To create new mesh triangles, points are placed near to this front. The quality of the resulting mesh is influenced by the location of the new points. An issue during the front propagation process is to avoid self-intersection. A common solution is to define a heuristic edge length and check the distances between the new point and the existing points which are threatened by a possible intersection (see [16],[17]). If this test fails, the front merges with the existing mesh and therefore two new fronts are formed (see Figure 10).

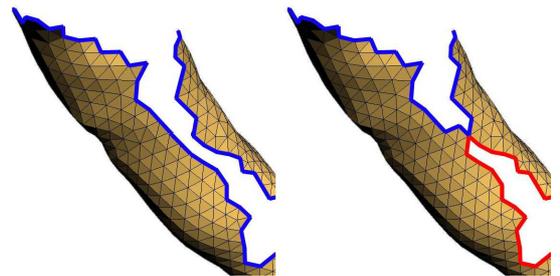


Fig. 10 Two new fronts are formed, if no further propagation of the initial front is possible.



Fig. 11 Due to a rapid change in size between small (green) and big (blue) triangles, degenerated triangles (red) are created.

The final step of the mesh creation is a full merge where no front remains. Using this surface based approach, it is possible, to associate the triangle size with the curvature of the implicit surface description. Features are preserved, since small triangles are created in regions with high curvature. The overall number of triangles is reduced, because bigger triangles are applied in regions with low curvature.

Several Advancing Front approaches are able to perform such curvature dependent polygonization (see [16]). Common problems concerning these methods are rapid local changes in triangle size due to changes in surface curvature. Such discontinuities are undesired since they cause numerical problems during CFD computation. Additionally, triangles that are generated between big and small triangles are often degenerated (see Figure 11).

To avoid these issues, the surface extraction approach of Schreiner et. al. is used [12]. This polygonizer applies a guidance field to provide smooth changes in triangle size and prevents the creation of degenerated triangles. The guidance field is a scalar function defined on the surface, providing information about the desired length of the edges incident to a point. For further details on the creation of the guidance field see [18].

Results

We applied our method to four datasets including a bronchial tree (BT), a liver tree (LT), a cerebral tree (CT) and a cerebral aneurysm (AN). A summary of the dataset properties is given in Table 1. The reconstructed surfaces offer a good trade-off between accuracy and smoothness. A careful visual examination of the surfaces near branchings showed that geometric continuity was achieved for all kinds of branchings and branching angles (see for example Figure 12 and Figure 14 (c)). The generated surfaces do not exhibit the spherical artifacts that are typical for reconstructions based on point clouds of low density. A visual comparison of the MPU Implicits results with the segmentation results gave first evidence that morphology and topology are represented correctly applying MPU Implicits. No tree structures are suppressed and no branches occur that are not represented in the data. Very thin elongated structures are

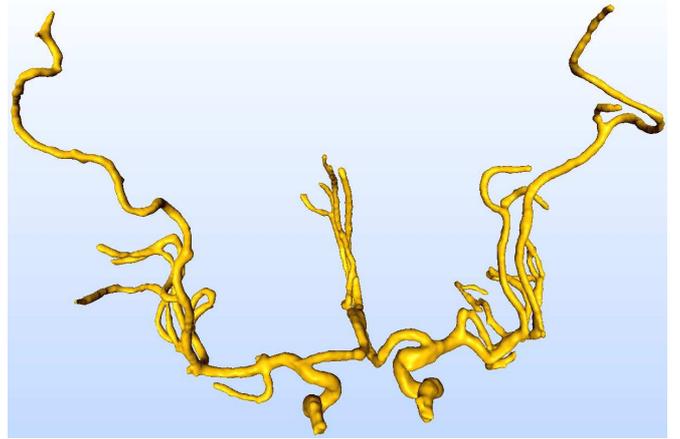


Fig. 12 Reconstruction of the thin elongated vessels of the cerebral tree.

reconstructed faithfully using MPU Implicits due to the proposed subsampling strategy (Figure 12). Using the automatically selected parameters, vessels that are in close proximity (running parallel or crossing each other) are represented separately by the implicit function. However, the structures may not be polygonized as separate structures and merge instead if the distance between them constitutes less than 3 voxels (compare Figure 13 (a) and (b)). This result represents a limitation of the Bloomenthal polygonizer and the proposed parameter selection for the polygonization (see section *Polygonization*). Using an isovalue of 0 and a sufficiently small par-

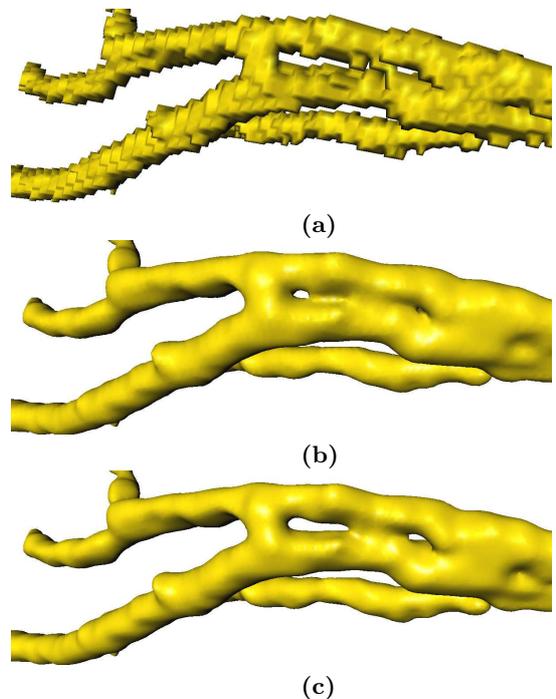


Fig. 13 Reconstruction of vessels, that are separated by only few voxels in the segmentation result (a): Using an isovalue larger than 0, vessels might merge (b). A correct reconstruction can be achieved using an isovalue of 0 (c).

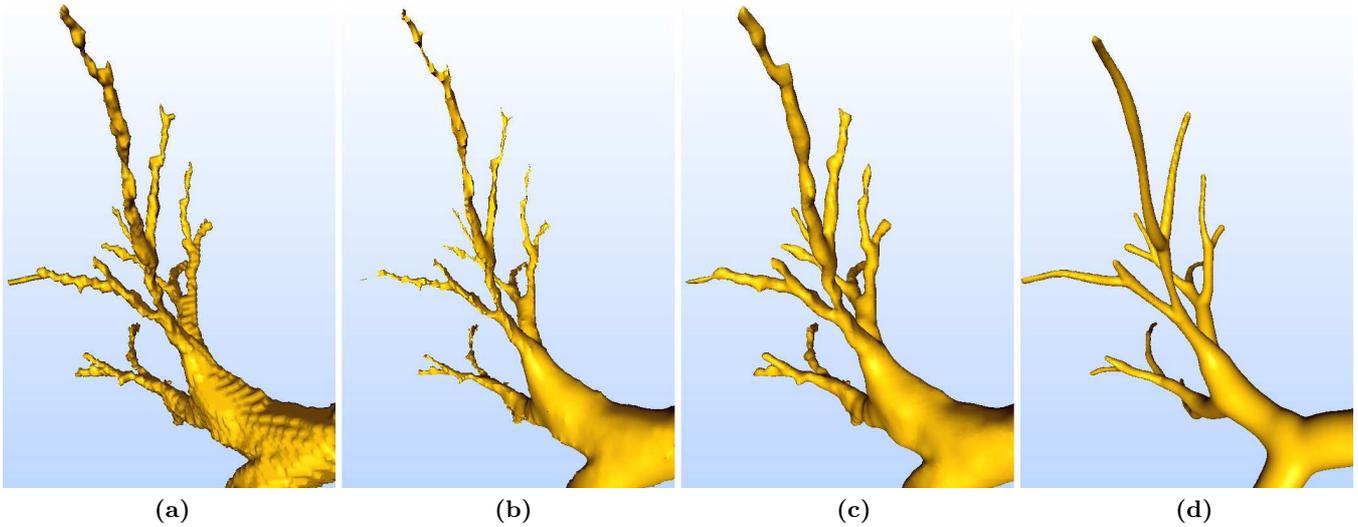


Fig. 14 A detailed look at the reconstruction of the bronchial tree using *Marching Cubes* (a), *Constrained Elastic Surface Nets* (b), *MPU Implicits* (c) and *Convolution Surfaces* (d).

Table 1 Summary of characteristics of the data sets. Voxel sizes are given in millimeters.

Dataset	Resolution	Voxel size
BT	$343 \times 193 \times 259$	$0,57 \times 0,57 \times 1$
LT	$342 \times 256 \times 81$	$0,633 \times 0,633 \times 2$
CT	$322 \times 233 \times 180$	$1 \times 1 \times 1$
AN	$129 \times 107 \times 45$	$1 \times 1 \times 1$

titioning element (e.g. 50% of the size of one voxel), the merging can be avoided (see Figure 13 (c)). However, this would lead to a strong increase of computation times and polygon count.

We compared the results of our algorithm with the results of Marching Cubes, Constrained Elastic Surface Nets [6] and Convolution Surfaces [3] when applied to the same segmentation result. In contrast to Marching Cubes, the reconstruction with MPU Implicits leads to smooth surfaces and avoids staircase artifacts (see Figure 14 (a) and (c)). Our method outperforms Constrained Elastic Surface Nets with respect to surface smoothness, especially for very thin vessels where the Constrained Elastic Surface Nets reconstructions exhibit volume shrinkage (see Figure 14 (b) and (c)) even if only very few smoothing iterations are used (we used 8 iterations in our experiments). The quality of the reconstruction with Constrained Elastic Surface Nets could be raised by using more smoothing iterations. However, this leads to further shrinkage or even the collapse of small vessels.

Using our method, subtle surface features are represented correctly in contrast to the reconstruction with Convolution Surfaces. The latter guarantees a high geometric continuity, however, due to the simplifying model assumption of circular cross-sections the morphology is only approximated (compare Figure 14 (a) and (d)).

Surface Smoothness

Visual inspection indicated a smooth reconstruction. This can be validated by visualizing curvature values directly on the surface. Figure 15 shows the Gaussian curvature for the reconstruction of the cerebral tree using Marching Cubes, Constrained Elastic Surface Nets, our method and Convolution Surfaces. The Marching Cubes based reconstruction exhibits strong positive and negative curvature values all over the surface (Figure 15 (a)). Those curvature values are associated to staircase artifacts rather than morphologic features. The benefit of the application of Constrained Elastic Surface Nets is limited since only very few iterations can be applied. For very thin vessels, the distribution of curvature values is similar to Marching Cubes (compare Figure 15 (a) and (b)). Other object parts exhibit less artifacts (see Figure 14 (b)). The MPU Implicits reconstructions show high curvature values only where morphology with high curvature is present (Figure 15 (c)). The smoothness of the surfaces is comparable to the one achieved by applying Convolution surfaces (compare Figure 15 (d)) due to the underlying implicit description.

Reconstruction Accuracy

We analyzed the reconstruction accuracy of our method with respect to the segmentation result that was the input of the pipeline. The validation of the segmentation procedure itself is beyond our scope. To measure the error that is introduced by our method with respect to the segmentation result, we first generate a polygonization of the binary segmentation result using the Marching Cubes Algorithm. We compute the deviation of our method with respect to the Marching Cubes results using AMIRA (www.amiravis.com). The goal is not to

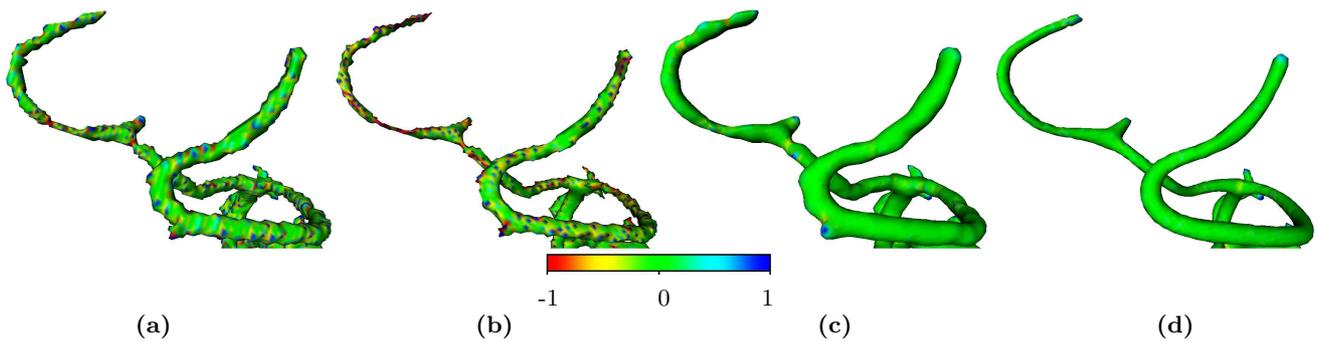


Fig. 15 Color coded visualization of gaussian curvature values for the reconstruction of the cerebral tree using *Marching Cubes* (a), *Constrained Elastic Surface Nets* (b), *MPU Implicits* (c) and *Convolution Surfaces* (d).

compare the accuracy of our method with the accuracy of the Marching Cubes algorithm itself. Instead we consider the result of the Marching Cubes algorithm as a valid representation of the binary segmentation result (although it is not the most accurate method to visualize a segmentation result, Marching Cubes is considered as the “gold standard” to which a validation should relate since it is widely used in radiological workstations).

Since absolute deviations are not expressive, we describe them in relation to the resolution of the image data. The median of the deviation is 0.19 times the length of the voxel diagonal ($voxDiag$) (this corresponds to 0.3 mm for the used datasets). The maximum deviation is up to 1.9 times $voxDiag$ (3.3 mm) for the aneurysm dataset. Those outliers occur due to small concave features which are not represented by enough points (Figure 16). Our subsampling strategy does not consider those structures because they are less relevant for the reconstruction of vascular surfaces. Other reasons are the limitation of the octree subdivision and the application of an isovalue larger than 0. A comparison of MPU Implicits with respect to accuracy with other state of the art visualization methods showed that it outperforms model-based methods like Convolution Surfaces with respect to accuracy (mean deviation for Convolution surfaces is 0.4 times $voxDiag$). Although our method suf-

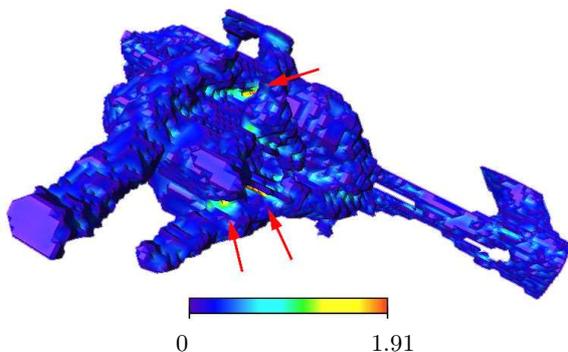


Fig. 16 Deviation of the MPU Implicits based reconstruction to the Marching Cubes result color coded on the Marching Cubes surface. Values are related to $voxDiag$. The red arrows highlight slight inaccuracies at small cavities.

fers from minor inaccuracies, it is almost as accurate as other model-free methods like Constrained Elastic Surface Nets (mean deviation: 0.17 times $voxDiag$; for details see [13]).

Reconstruction Efficiency

Vessel surface reconstruction using MPU Implicits is generally slower than Marching Cubes or Constrained Elastic Surface Nets. A reconstruction of the bronchial tree, the largest and most complex tested dataset, was achieved in 3 seconds using Marching Cubes and required 13 seconds using Constrained Elastic Surface Nets. The same dataset was processed with our approach in 38 seconds. The reconstruction time using Convolution Surfaces is 36 seconds. Using our approach, the cerebral aneurysm was reconstructed in 5 seconds (Marching Cubes: 1 seconds, Constrained Elastic Surface Nets: 2 seconds, Convolution Surfaces: 11 seconds). In general, the reconstruction times for the proposed method are very similar to the reconstruction times for Convolution Surfaces and depend on the resolution of the original data, the complexity of the vessel tree, and the size of the partitioning element (more details on reconstruction times can be found in [13]). The time needed for mesh refinement including the Advancing Front remeshing and edge collapse steps are in the range of several minutes.

In Table 2, we summarize information concerning the geometric complexity of the resulting models. The number of polygons generated using MPU Implicits in combination with the Bloomenthal polygonizer is still in a range that modern graphics hardware may process at interactive frame rates. Compared to Marching Cubes, it is approximately 1.2 times larger (see Table 2) due to the chosen size of the partitioning element. In the case of the liver tree the amount of polygons is doubled because of the strong anisotropy of the dataset (see Table 1).

The proposed mesh refinement leads to a reduction of the mesh complexity up to 50% (see Table 3, rows MPUI and AF). If the added computation time is acceptable, visualization tasks can also benefit from this reduction since it enables higher frame rates.

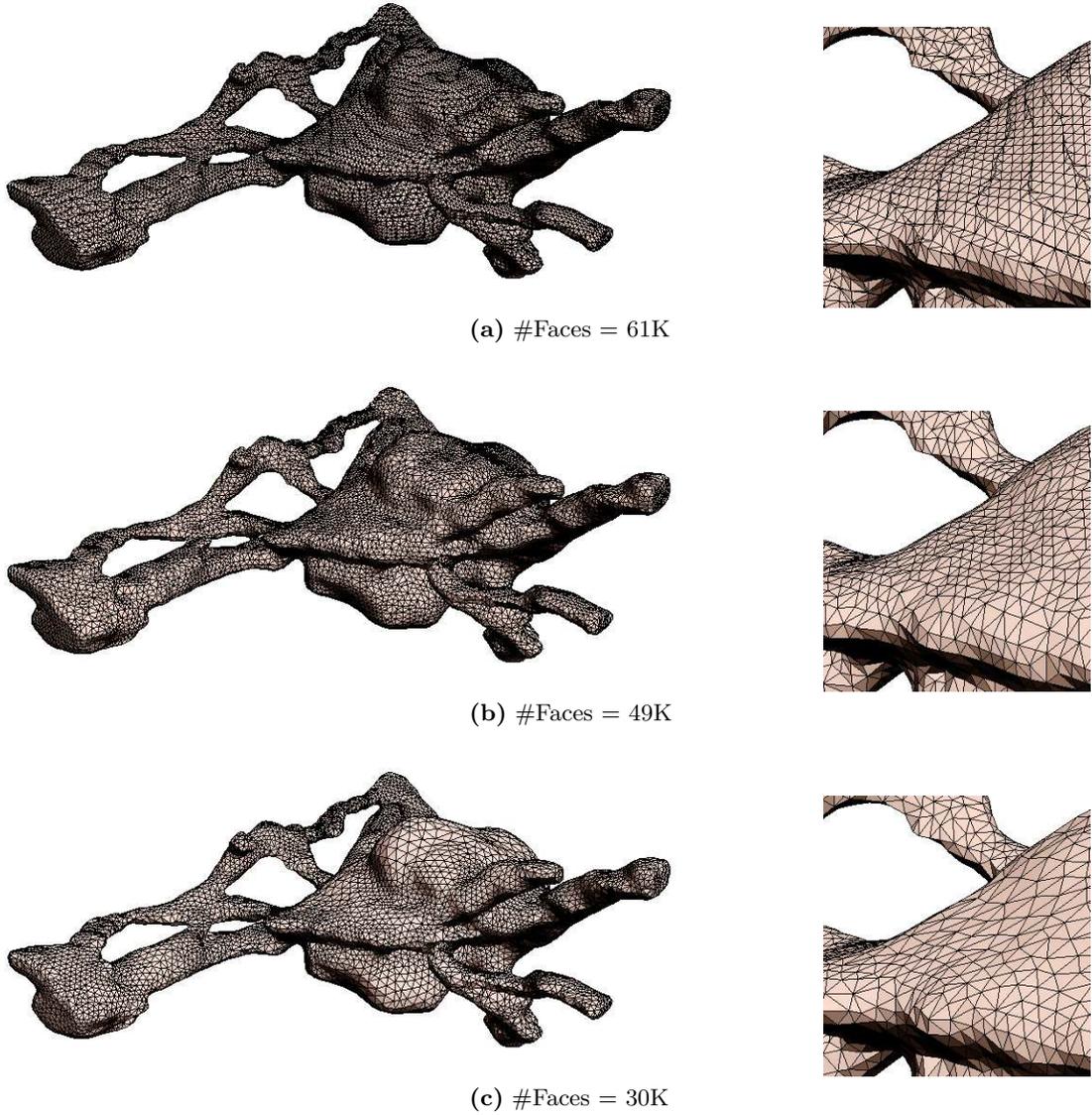


Fig. 17 The resulting meshes: MPU Implicits (MPUI) (a), MPUI after edge collapse (b) and MPUI after an Advancing Front remeshing step (c).

Table 2 Triangle count of MPU Implicits (MPUI) and MPUI combined with the proposed post processing steps (MPUI + PP) compared to Marching Cubes (MC).

Dataset	MC	MPUI	MPUI + PP
BT	166k	180k	122k
LT	80k	167k	106k
CT	115k	142k	102k
AN	54k	61k	30k

Table 3 Summary of the MPU Implicits (MPUI), Edge Collapse (EC) and Advancing Front meshes (AF) of an aneurysm model. The worst equi-angle skewness ($S(Max)$) is important for the simulation.

Meshtype	#Faces	$S(Max)$	$S(Min)$	$S(Mean)$
MPUI	61314	1.0	0.01	0.63
EC	49356	0.74	0.03	0.28
AF	30112	0.72	0.0004	0.23

Triangle Quality

To provide good convergence characteristics during the CFD simulation, the worst equi-angle skewness should be lower than 0.75 (see section *Mesh Quality Improvement*). To achieve this triangle quality, the guidance field driven Advancing Front polygonization of the remesh-

ing tool AFRONT (<http://afront.sourceforge.net>), created by Schreiner et. al., was used. Results for the cerebral aneurysm are presented in Table 3, comparing mesh complexity and triangle quality of the Bloomenthal-polygonization of the MPU Implicits with the optimized meshes and the Advancing Front meshes. It is obvious, that the optimized and Advancing Front meshes sat-

isfy the given requirements. All three resulting meshes are presented in Figure 17. The linear triangle artifacts which can be observed in the standard Bloomenthal-mesh are the result of the cell-based polygonization process which generates small elongated triangles. These degenerated triangles are removed by collapsing edges during the optimizing process. This leads to a reduced mesh complexity. Due to the curvature dependent triangle size, the mesh complexity is further reduced by applying the Advancing Front algorithm. The guidance field approach ensures a smooth transition of triangle size, which leads to a good quality of the resulting mesh.

Discussion

The smooth and accurate reconstruction of vascular surfaces based on a binary segmentation result is a demanding task. Methods such as Marching Cubes result in surfaces that directly correspond to the underlying segmentation but suffer from strong staircase artifacts. An application of simple smoothing algorithms to the result is problematic since shrinkage is introduced that might even result in the suppression of small features. Volume preserving methods like Constrained Elastic Surface Nets can reduce this effect in general but also fail in representing thin structures. Model-based methods such as Convolution Surfaces can produce very smooth surfaces. However, the underlying model assumption of circular cross sections leads to an inaccuracy that might be acceptable for certain visualization tasks, but not for many diagnostic applications or in particular for the generation of volume grids for CFD simulations. A combination of high accuracy and high surface smoothness is crucial for these applications. First results indicate that the proposed method fulfills both requirements. The properties of the underlying implicit representation of our method guarantee a high surface smoothness comparable to Convolution Surfaces. The accuracy of the results with respect to the binary segmentation result has been investigated and can be considered as very high. However, this does not allow to draw conclusions about the reconstruction accuracy with respect to the actual vessels since the segmentation method itself is not part of the proposed pipeline. The correctness of the segmentation result constitutes an essential precondition of the method. However, the combination with a dedicated segmentation method might be part of future work.

The investigation of the reconstruction accuracy revealed several limitations of the method in its current implementation. Inaccuracies, that might be too large for obtaining appropriate CFD grids have been observed at small concave features. The impact on the simulation still needs to be investigated. An adaption of the method to such features might be achieved by identifying those outer boundary voxels, that are next to those features. First tests indicate that this can be accomplished by

applying a top-hat-transformation to the inverted segmentation result. Once those voxels have been identified, a subsampling as described for small vessels can be performed. Additional inaccuracy has been introduced due to the usage of an isovalue larger than 0 which also results in the merging of neighboring vessels in few cases. This isovalue was selected because the Bloomenthal polygonizer may miss vessels that exhibit a diameter smaller than the size of the polygonization element. The usage of a smaller polygonization element would also facilitate the reconstruction of very thin vessels. That way, the general accuracy would also be raised and neighboring vessels would be reconstructed separately. First tests show that the application of an alternative parameter configuration (isovalue = 0; size of space partitioning element is 50% of the voxel size) results in higher accuracy at the cost of considerably higher computation times and polygon counts. This might be acceptable for simulation, but not for visualization tasks.

A common drawback of surface tracking approaches is that only one connected surface can be reconstructed. However, vascular systems may consist of several unconnected structures. To allow a complete reconstruction, independent structures may be identified using *Connected Component Labeling* [19]. In a next step the pipeline is applied to every structure individually.

In general, the space partitioning approach of the Bloomenthal polygonizer seems to be problematic because it generates more polygons than necessary. In addition, the generated polygons do not offer the quality that is needed for successful CFD simulations. The worst triangle quality of the generated meshes is equal to 1. Improving the Bloomenthal mesh by collapsing and flipping edges leads to better results. The achieved triangle quality satisfies the given requirement. Also the number of triangles is reduced, since small triangles are removed.

For guidance field driven Advancing Front remeshing the software AFRONT is used. The remeshed surfaces are used to create volume meshes with TGRID, a part of the FLUENT software package. These volume meshes represent the base for finite-volume CFD simulations. The equi-volume skewness S_V is a quality measure for volume elements and is, like the equi-angle skewness of surface meshes, defined in the interval $[0, 1]$. By applying the Advancing Front remeshing on the surface mesh, an improvement of the volume element quality could be observed. However, the triangle size control parameter in AFRONT has to be chosen carefully. If the chosen parameter value is too small, the resulting mesh consists of more triangles than the original Bloomenthal mesh, due to the creation of many, very small triangles in regions of high curvature. Very big triangles might be created in regions of low curvature if the parameter value is too large. This might lead to big volume elements and therefore potentially reduce the accuracy of the simulation result (see Figure 18).

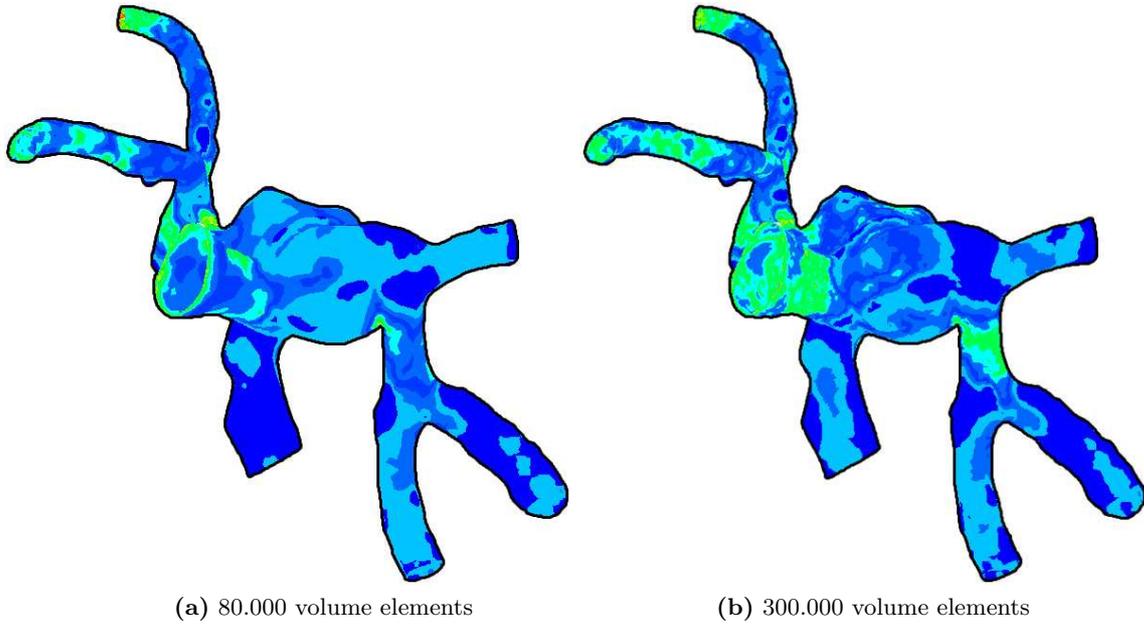


Fig. 18 Color coded wall shear stress of an aneurysm and the corresponding vessels. If the volume mesh resolution is too low (a), certain features, like the stress peaks on the aneurysm surface (b) are blurred or in worst case lost.

Using the Advancing Front approach degenerated triangles (equi-angle skewness $> .75$) are still created. However, the amount is very small ($< 0.1\%$). Those triangles are improved or removed using the mesh quality improvement steps that have been applied to the Bloomenthal results (see section *Mesh Quality Improvement*).

To perform a CFD simulation, inlets and outlets have to be defined. For numerical stability, these in- and outlets need to be sharp edged. They are created by plane-cutting the vessels at a predefined position. While the edge collapse based mesh improvement preserves these sharp edges, the Advancing Front remeshing using AFRONT introduces some artifacts (see Figure 19). Because of that, the in- and outlet-cuts are applied after the remeshing. Degenerated triangles that are created during the cutting process are remeshed locally by edge collapse.

The proposed refinement and remeshing steps were chosen with regard to requirements of the volume grid

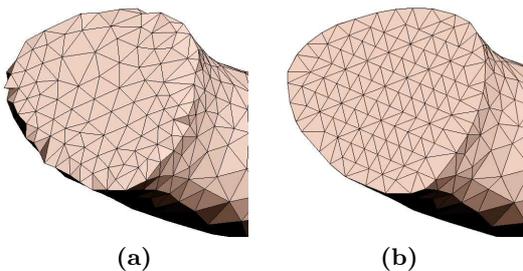


Fig. 19 An outlet cut: after Advancing Front reconstruction (a). To avoid these artifacts, the cut has to be performed after the Advancing Front remeshing (b). Degenerated triangles near the cutting edge are removed by edge collapse.

generation for CFD simulations. However, the obtained optimized surfaces might also be used for visualization. This might be of benefit for high quality visualization of flow features in future projects. If the same mesh is used for simulation and visualization, important surface-based flow features like the wall shear stress can be visualized in a direct way, because the surface mesh is directly linked to the computational mesh. That way, no further numerical interpolation artifacts would be introduced.

The parameter choice in the current remeshing pipeline is done manually. Based on the observation of the simulation result the mesh is redefined to find an optimal solution. In future work, these parameters should be linked directly to the computational accuracy of the CFD simulation, using information about the anatomic model and the simulation modalities.

In the current implementation, the AFRONT software is only used as a post processing step. The direct application of an adaptive polygonization procedure to the implicit function could lead to higher accuracy and triangle quality combined with lower reconstruction times. Future work should concentrate on the integration of an Advancing Front algorithm or a comparable adaptive polygonization method.

Conclusions

We have presented a method for the reconstruction of vascular tree structures based on the binary result of the vessel segmentation. The generated surfaces are suited for visualization and CFD simulation tasks. The required combination of high accuracy, smoothness and trian-

gle quality can not be achieved using existing standard methods. Model-free approaches such as Marching Cubes guarantee a high accuracy but suffer from aliasing artifacts. The application of smoothing methods leads to shrinkage or even the collapse of thin vessels. The model-based reconstruction with Convolution Surfaces achieves very high surface smoothness. However, the morphology of the vascular structures is only approximated. Our method represents a good trade-off between high accuracy and surface smoothness. In addition, a high triangle quality is guaranteed. Our method is based on MPU Implicits, a variant of implicit surfaces. With a preprocessing step which computes a point cloud based on the segmentation result and further refines it adaptively, this method is able to generate smooth reconstructions of the vessel surface and accurately represent even very thin vessels. We applied our method to a variety of vessel trees and compared the results with other state-of-the-art techniques regarding reconstruction accuracy, smoothness and efficiency. With an average median deviation of 0.19 voxel diagonals (0.3 mm for the tested datasets), our approach is very accurate. It represents the segmentation result more accurately than Convolution Surfaces [3], especially for pathologic cases. The accuracy could be further increased by adapting our subsampling strategy for small concave features. Several remeshing techniques including an Advancing Front [12] algorithm have been investigated to adapt the polygonization results to the requirements of CFD simulations. First experiments indicate that the proposed pipeline creates vessel surfaces that are appropriate for the generation of volume grids for CFD simulations of the blood flow. Although reconstruction times of our method are in the range of seconds, there is room for improvement. The computational times for subsequent mesh refinement steps are in the range of minutes, which is acceptable for CFD simulations but not for most visualization tasks. We identified several disadvantages of the used polygonization method. A gain in reconstruction accuracy, quality and speed might be achieved by direct application of an adaptive polygonization scheme to the implicit function.

Acknowledgements Many thanks go to Steffen Oeltze (University of Magdeburg) for input on the development of the reconstruction pipeline and to Horst Hahn (MeVis Research Bremen) for fruitful discussions. Stefan Zachow (Zuse Institut Berlin) supported our work concerning mesh enhancement. The simulation results have been generated by Dominique Thévenin, Gábor Janiga and Santhosh Seshadhri from the Laboratory of Fluid Dynamics and Technical Flows of the University of Magdeburg. Image data has been supplied by Özlem Kricschek and Martin Skalej (Universityhospital Magdeburg, Neuroradiology). Thanks to Claudio Silva from the University of Utah for access to the AFRONT software.

References

1. Cebra J, Castro M, Appanaboyina S, Putman C, Milan D, Frangi A (2005) Efficient pipeline for image-based patient-specific analysis of cerebral aneurysm hemodynamics: technique and sensitivity. *IEEE Trans. Med. Imaging*: 547-467
2. Hahn HK, Preim B, Selle D, Peitgen HO (2001) Visualization and Interaction Techniques for the Exploration of Vascular Structures. *IEEE Visualization*: 395-402
3. Oeltze S, Preim B (2004) Visualization of Vascular Structures with Convolution Surfaces. *IEEE / Eurographics Symposium on Visualization*: 311-320
4. Lorensen WE, Cline HE (1987) Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Proc. of ACM SIGGRAPH*: 163-169
5. Bade R, Haase J, Preim B (2006) Comparison of Fundamental Mesh Smoothing Algorithms for Medical Surface Models. *Simulation und Visualisierung*: 289-304
6. Gibson SFF (1998) Constrained Elastic Surface Nets: Generating Smooth Surfaces from Binary Segmented Data. *Proc. of MICCAI*: 888-898
7. Preim B, Oeltze S (2007) 3D Visualization of Vasculature: An Overview. In: *Visualization in Medicine and Life Sciences* (Springer, 2007): 19-39
8. Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel HP (2003) Multi-level Partition of Unity Implicits. *ACM Transactions on Graphics* **22(3)**: 463-470
9. Bloomenthal J (1994) An Implicit Surface Polygonizer. *Graphics Gems IV* (Academic Press, 1994): 324-349
10. Braude I, Marker J, Museth K, Nissanov J, Breen D (2007) Contour-Based Surface Reconstruction using MPU Implicit Models. *Graphical Models* **69(2)**: 139-157
11. Cebra JR, Lohner R (2001) From Medical Images to Anatomically Accurate Finite Element Grids. *International Journal for Numerical Methods in Engineering* **51**: 985-1008
12. Schreiner J, Scheidegger C, Silva C (2006) High-Quality Extraction of Isosurfaces from Regular and Irregular Grids. *IEEE Transactions on Visualization and Computer Graphics*: 1205-1212
13. Schumann C, Oeltze S, Bade R, Preim B, Peitgen HO (2007) Model-free Surface Visualization of Vascular Trees. *Eurographics/IEEE VGTC Symposium on Visualization EuroVis*: 283-290
14. Bade R, Schumann C, Seshadhri S, Janiga G, Bölke T, Kricschek Ö, Skalej M, Rose G, Thévenin D, Preim B (2007) High-quality Surface Generation for Flow Simulation in Cerebral Aneurysms. *CURAC*: 125-128
15. George PL, Seveno E (1994) The Advancing-Front Mesh Generation Method Revisited. *International Journal for Numerical Methods in Engineering* **37**: 3605-3619
16. Araújo B, Jorge J (2004) Curvature Dependent Polygonization of Implicit Surfaces. *XVII Brazilian Symposium on (SIBGRAPI'04)*: 266-273
17. Karkanis T, Stewart J (2001) Curvature-Dependent Triangulation of Implicit Surfaces. *IEEE Computer Graphics and Applications*: 60-69
18. Schreiner J, Scheidegger C, Fleishman S, Silva C (2006) Direct (Re)Meshing for Efficient Surface Processing. *Technical Report*, University of Utah
19. Ballard DH, Brown CM (1982) *Computer Vision* (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982): 408-429