Eurographics/ IEEE-VGTC Symposium on Visualization (2007)
Ken Museth, Torsten Möller, and Anders Ynnerman (Editors)

Model-free Surface Visualization of Vascular Trees

C. Schumann[1,2] and S. Oeltze[2] and R. Bade[2] and B. Preim[2] and H.-O. Peitgen[1]

[1]MeVis Research, Bremen, Germany
[2]Department of Simulation and Graphics, Otto-von-Guericke University of Magdeburg, Germany

Abstract

Expressive and efficient visualizations of complex vascular structures are essential for medical applications, such as diagnosis and therapy planning. A variety of techniques has been developed which provide smooth high-quality visualizations of vascular structures based on rather simple model assumptions. For diagnostic applications, these model assumptions and the resulting deviations from the actual vessel surface are not acceptable. We present a model-free approach which employs the binary result of a prior vessel segmentation as input. Instead of directly converting the segmentation result into a surface, we compute a point cloud which is adaptively refined at thin structures, where aliasing effects are particularly obvious and artifacts may occur. The point cloud is transformed into a surface representation by means of MPU Implicits, which provide a smooth piecewise quadratic approximation. Our method has been applied to a variety of datasets including pathologic cases. The generated visualizations are considerably more accurate than model-based approaches. Compared to other model-free approaches, our method produces smoother results.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling J.3 [Life and Medical Sciences]: Health

1. Introduction

The visualization of vascular structures deserves special attention since vascular trees are among the most complex structures of the human body. In case of the diagnosis of vascular diseases, the local evaluation of the vessel cross section is essential to detect and characterize narrowings, such as a stenosis as well as widenings, e.g. an aneurysm. For intervention planning, however, it is more important to understand the topology of a vascular tree in order to predict the consequences of an intervention strategy (Which portion of a vascular tree is affected by a resection? Which area is supplied by this portion?). As a consequence of these different scenarios, the requirements for an appropriate visualization are also different. Whereas accuracy is of crucial importance for diagnostic tasks, it is less important for therapy planning where more idealized visualizations better serve the purpose. A variety of dedicated methods for intervention planning and medical education has been developed [MMD96, HPSP01, OP04]. These methods have in common that the visualization is based on model assumptions, in particular a circular cross-section is often assumed. Conversely, primarily direct volume rendering or projection methods are employed for diagnostic tasks. These methods have their benefits, however they are often not able to clearly distinguish vascular structures from adjacent structures with similar intensity values. The depth perception is limited in particular for projection methods. According to our experience smooth visualization of vascular structures are essential for intervention planning [SPSP02]. Since the human visual system is very sensitive to discontinuities, which normally represent important features for object detection and classification, our goal is to develop a visualization method for diagnosis which provides smooth, easy-to-interpret and accurate images of a 3D vascular tree. We chose surface visualizations based on an explicit prior segmentation and attempt to visualize the segmentation result as smoothly and accurately as possible. To emphasize the principal difference to existing approaches, we refer to our method as model-free visualization. Model-free approaches provide two adavantages over model-based techniques: they require less preprocessing effort (e.g. no centerline determination is required) and they offer an accuracy that cannot be achieved by methods which adhere to model assumptions. The basic principle of our method is to convert the segmentation result into a point cloud to which a smooth quadratic surface is fitted.

2. Related Work

We first describe volume rendering approaches which do not require image segmentation. Later, model-free and model-based surface visualizations are discussed. Model-free surface visualizations employ the segmentation result as input, whereas model-based approaches commonly rely on the centerline and the local vessel diameter determined after segmentation.

2.1. Direct Volume Rendering

The most common volume rendering technique is the maximum intensity projection (MIP) which displays for every ray the voxel with the highest image intensity. In order to display small vessels in front of larger one's, the MIP method was modified by a threshold which specifies that the first local maximum above the threshold should be depicted. This method is known as either local MIP [SSN*98] or closest vessel projection (CVP) [Zui95].

In contrast to projection methods, direct volume rendering (e.g. raycasting, splatting) provides realistic depth-cues by blending data considering the "in-front-of" relation. The ability of 1D transfer functions to discriminate vascular structures from its surrounding is limited.Therefore, 2D transfer functions have been explored [VST*03], where gradient magnitude has been used as second dimension in addition to image intensity. However, the adjustment of 2D transfer functions is difficult to accomplish and can not be completely standardized since gradient values differ strongly for different acquisition devices, modalities and protocols.

2.2. Model-free Surface Visualization

In principle, surface visualizations can be generated based on a threshold that is used as input for isosurface rendering. Due to image noise and inhomogeneities in the contrast agent distribution this does not lead to accurate results, in particular in MRI data where these problems are more severe than in CT. In many cases vascular structures need to be explicitly segmented, although segmentation may be tedious and is also a source of error. The most common surface reconstruction technique is Marching Cubes (MC) [LC87] with an appropriate threshold. Unfortunately, the quality of the resulting visualizations is relatively low due to the usage of linear interpolation and due to the fact that the segmentation results are binary volume data. Aliasing artifacts may be reduced after surface reconstruction in a smoothing step. It must be noted, however, that appropriate smoothing of vascular structures is challenging and in general does not lead to the desired results [BHP06]. Simple smoothing procedures such as Laplacian smoothing should not be used at all because small branches may collapse. Better results are achieved with smoothing approaches that try to prevent shrinking, such as LowPass-filtering [Tau95] and the improved Laplacian smoothing, introduced in [VMM99]. Advanced feature-preserving smoothing approaches often fail completely since they recognize staircases that occur due to boundary discretization as features. Only few smoothing techniques are dedicated to the typical terracing artifacts resulting from visualizing binary volume data. The best general method to provide smooth visualizations of binary segmentation results is Constrained Elastic Surface Nets (CESN) [Gib98]. CESN yields a good trade-off between accuracy and smoothness by constraining the smoothing of an initial mesh such that the vertices stay inside their cell. However, for small vascular structures, the constraint to a particular cell is not sufficient, since these are often only one voxel wide. Such structures degenerate after smoothing to a line (Fig. 1).

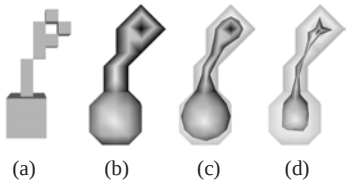


Figure 1: *Reconstruction of a thin structure (a) with CESN: result without any smoothing (b), with one iteration (c) and with 5 iterations (d).*

2.3. Model-based Surface Visualization

Model-based approaches are less relevant here and therefore not discussed in detail. The assumption of a circular cross-section gave rise to techniques which construct a geometry explicitly with graphics primitives which exhibit a circular cross-section. Cylinders [MMD96] and truncated cones [HPSP01] have been employed. The latter allows to faithfully represent how the diameter diminishes towards the periphery. To achieve smooth transitions at branchings, a variety of methods has been investigated. B-spline surfaces have been proposed in [HPea00] to approximate small vascular structures and nerves which could not be completely segmented. Simplex Meshes can also be used for high-quality visualization of vascular structures, as [BRB05] recently showed. Finally, variants of implicit surfaces have also been employed. Convolution Surfaces (CS), developed by Bloomenthal [BS91], allow to construct a scalar field along skeletal structures. With an appropriate filter for the convolution, the scalar field can be polygonized [Blo94] in such a way that the vessel diameter is precisely represented. The use of this approach for vessel visualization requires sophisticated acceleration strategies [OP04].

3. Reconstruction of Surfaces based on Point Clouds

Surfaces can be described by sets of points lying on the border. Those sets are refered to as point clouds and may be acquired by laser range image scanners. Numerous algorithms for surface reconstruction based on point clouds have

been developed. Many of those algorithms reconstruct an implicit function based on the point cloud. One class of state-of-the-art methods is based on globally or compactly supported Radial Basis Functions (RBFs). Recent algorithms use Level Sets [ZO02], Multi-level Partition of Unity Implicits [OBA*03] or Regularized Membrane Potentials [JR06]. We choose MPU Implicits for surface reconstruction because it is one of the fastest and most robust methods.

3.1. Multi-level Partition of Unity Implicits

Ohtakte et al. use an adaptive subdivision of the point cloud as the basis for their Multi-level Partition of Unity Implicits (MPUI) [OBA*03]. The input point cloud consitis of points and associated normal vectors. In a first step, it is scaled such that its bounding box equals a cube with a diagonal of unit length. An octree is used to subdivide the rescaled point cloud. Every octree cell defines a spherical point aggregation domain. The initial radius R of this domain equals αd where d is the length of the diagonal of the octree cell. If the domain includes less than n_{min} points, the size of the domain is increased, because at least n_{min} points are to be used for a local approximation. Therefore, R is iteratively increased by λR until at least n_{min} points are in the domain. α, λ and n_{min} are parameters that have to be specified by the user. Once the points have been gathered, they are approximated by quadratic functions. Ohtake et al. present several fitting strategies for that, each suited for a different constellation of points and normal vectors in the domain. When a local approximation has been fitted, its deviation from the points in the domain is calculated. If the deviation surpasses 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 or the maximum recursion depth $level_{max}$ is reached. 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.

Bloomenthal's implicit polygonizer [Blo94] is used to generate a polygonal representation of the surface based on the approximated implicit function. This surface tracking algorithm uses a cubical space partitioning element which is moved along the surface. The intersections of the edges of the cube with the isosurface determine the positions of vertices of the polygonal mesh. The user has to specify the size of the space partitioning element and the used isovalue. The polygonization result is scaled back to world coordinates.

The use of two groups of functions, the local shape functions and the associated weighting functions, is the main difference between MPUI and RBF-based methods. The computational cost does not depend on the number of points but on the complexity of the described object. Hence MPUIs are more efficient than RBFs for large datasets. The accuracy and smoothness of the reconstruction can be controlled by several parameters.

3.2. Reconstruction of Medical Surfaces with MPU Implicits

Braude [BMM*07] utilized MPUIs to reconstruct medical surfaces from segmented images. Midpoints of contour voxels are used to define a point cloud. Normal vectors are computed based on the gradient at those voxels. The method produces smooth surfaces of the described objects. However, the method fails for very thin structures like small vessels because those structures are represented by too few points. Spherical artefacts are the result (Fig. 2). Another drawback of the method is the need to specify numerous parameters (recall section 3.1). Braude does not explain how to choose "good" values for those parameters.

Figure 2: *Reconstruction of a liver tree with MPUIs: spherical artefacts occur at thin structures that are represented by too few points (left). Our method adapts point density to prevent those artefacts (right).*

3.3. Reconstruction of Vascular Structures with MPU Implicits

Our method is similar to the one described in [BMM*07] but targeted at the appropriate representation of vascular structures. The segmentation results are obtained using a refined region-growing variant for vasculature [SPSP02]. We propose a new algorithm for the point cloud generation that allows the reconstruction of small vessels and strategies to determine adequate reconstruction parameter values automatically. The proposed reconstruction pipeline is given below:

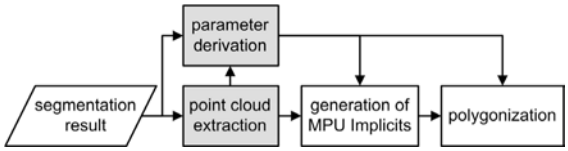


Figure 3: *Surface generation for vascular structures with MPU implicits.*

In principle, this pipeline may be used to reconstruct any surface model. However, the point cloud extraction is carefully dedicated to typical geometric features of vascular trees (branchings, thin elongated structures, etc.).

4. Point Cloud Extraction

As shown in Figure 2, MPUIs may produce artefacts if details are represented by too few points. In this case, the fitting of local approximations might fail. To allow the reconstruction of vascular structures with very thin branches, we propose an adaptive point cloud extraction technique that increases the point density in the vicinity of thin structures.

4.1. Placing of Points and Computation of Normal Vectors

Our point extraction algorithm is driven by the voxel grid of the segmentation result but does not strictly adhere to it. In [BMM*07] boundary voxels are used as the basis of the algorithm. However, that way very thin branches might be represented as lines of points that can not describe a volume. A reconstruction with MPUIs would not be possible. Instead we use those background voxels that are closest to given object voxels. We refer to those voxels as *outer boundary voxels*. Based on the 3D-6-neighbourhood 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. There is only one object voxel in the neighbourhood of *v* (Fig. 4 a). In this case, one point is placed in the center of the *boundary face* (face of *v* that is adjacent to the object neighbour voxel).
2. There are two object voxels on two opposite sides of *v* belonging to two different structures (Fig. 4 b). Two points are placed in the centers of the boundary faces.
3. There are four object voxels in the neighbourhood of v, placed in a plane (Fig. 4 c). Four points are placed in the centers of the boundary faces.
4. There are five object voxels in the neighbourhood, *v* is in a small cavity (Fig. 4 d). We place one point on the center of the boundary face that could be considered the ground of the cavity.
5. There are six objext voxels in the neighbourhood, *v* is a hole. We ignore this hole and place no points.
6. In all other cases the objectvoxels in the neighbourhood are placed in a staircase and one point is placed in the center of *v* (Fig. 4 e, f).

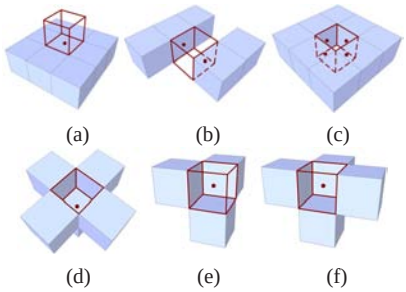


Figure 4: *Placing of points (red) for different constellations of object voxels (blue) in the neighbourhood of an outer boundary voxel (red).*

Once the points for an outer boundary voxel are placed, the related normal vectors are computed. If only one point has been placed (cases 1,4 and 6), the normal vector can be computed based on the gradient of the outer boundary voxel in the segmentation result. If more points were generated (cases 2 and 3) we use the normal vectors of the boundary faces.

4.2. Adaptive Subsampling of the Data Set

The method that has been described so far does not generate enough points to describe thin structures (structures with a cross section smaller than 3×3 voxels). To increase the density of the point cloud, we subdivide the voxels into eight subvoxels. Since this might lead to very large point clouds we restrict the subsampling to those object voxels that represent thin structures. First we identify thin structure voxels using a top-hat-transformation with a $3 \times 3 \times 3$ structure element. All outer boundary voxels that are next to thin structure voxels are subdivided into eight subvoxels. Points and normal vectors are computed for those subvoxels in the same manner as described before for voxels. However, this strategy leads to a constellation of points that adheres very strictly to the original voxel grid (Fig. 5 b).

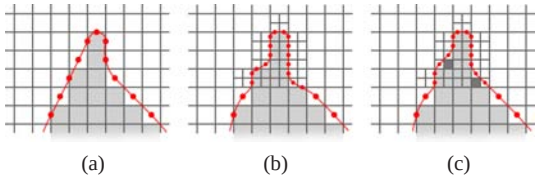


Figure 5: *Due to subsampling, the constellation of points adheres very strictly to the original voxel grid (b). Additional subvoxels are labeled as object-subvoxels to prevent staircase artefacts (c).*

To prevent staircase artefacts in the reconstruction with MPUIs, we propose the labeling of additional background-subvoxels as object-subvoxels as depicted in Figure 5 (c) prior to the actual positioning of points. Outer boundary voxels are candidates for the addition of object-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 is gathered. The subvoxel is labeled as object-subvoxel if there is at least one pair of object-subvoxels in the neighbourhood that fullfills *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.
- The other subvoxel shares one face or edge with the considered subvoxel.

Examples for the addition of object-subvoxels are depicted in Figure 6. The resulting point cloud allows a smoother reconstruction of thin branches (Fig. 7). The point cloud is transformed into world coordinates after default parameter values have been derived.

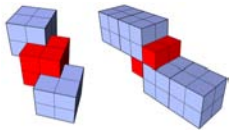


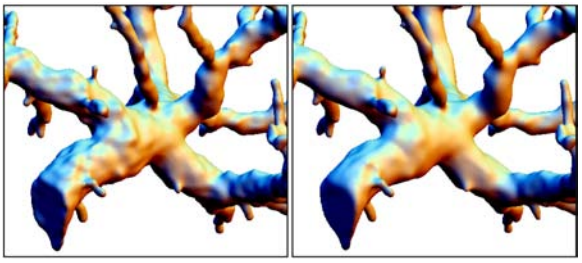
Figure 6: *Examples for the labeling of additional subvoxels (red) as object-subvoxels.*

(a) (b) (c)

Figure 7: *Reduction of stair case artefacts by the addition of object-subvoxels demonstrated for a single thin branch (a): Staircase artefacts occur due to subsampling of the voxel grid (b). The addition of object-subvoxels yields a smoother reconstruction (c).*

5. Determination of Default Parameter Values

The generation and polygonization of the MPUIs has to be adjusted by several parameters (recall section 3.1). Our goal was to find appropriate default values. Default values for n_{min}, α and λ can be set once for all data sets. We use higher values than those suggested by Ohtake et al. [OBA*03] to achieve a smoother reconstruction. Figure 8 shows a comparison of results generated with our values and the ones proposed in [OBA*03].



α	=	0.75	α	=	0.8
λ	=	0.1	λ	=	0.2
n_{min}	=	15	n_{min}	=	200

Figure 8: *Effect of different values for n_{min}, α and λ: The left image shows a reconstruction using the values proposed in [OBA*03]. A smoother reconstruction of the vessel surface is possible using higher values (right).*

We want to limit the deviation of the surface to the point cloud to half a voxel diagonal. ε_0 is thus related to a rescaled point cloud with a bounding box which has a diagonal of unit length. Therefore, ε_0 can be computed by dividing the desired maximum deviation by the diagonal length $diag$ of the bounding box of the unscaled point cloud:

$$\varepsilon_0 = \frac{\sqrt{3}}{2 \cdot diag} \tag{1}$$

However, the maximum deviation is not the only parameter to control the octree subdivision. The subdivision depth has to be limited to prevent too small cells that do not contain enough points for a local approximation. We limit $level_{max}$ in a way that the minimum cell size roughly corresponds to the size of one voxel. A proposal for $level_{max}$ can be given based on the bounding box extensions ext_x, ext_y and ext_z of the point cloud:

$$level_{max} = \lceil \log_2(\min(ext_x, ext_y, ext_z)) \rceil \tag{2}$$

The polygonization grid size is also computed based on the bounding box extensions of the point cloud:

$$grid\_size = \frac{0.7}{\max(ext_x, ext_y, ext_z)} \tag{3}$$

That way, the space partitioning element has a size slightly smaller than the voxels of the segmentation result. We use an isovalue that is slightly above 0 because the surface tracking polygonization procedure might miss very thin branches. This leads to a small increase of volume, which we consider as acceptable to guarantee the reconstruction of thin branches. We found 10% of ε_0 to be a good isovalue that corresponds to a surface offset of 5% of one voxel diagonal.

6. Results and Discussion

The proposed point cloud extraction algorithm and the parameter estimation have been prototyped in MeVisLab (www.mevislab.de). For the generation and polygonizaion of the MPUIs, we used the MPUI software by Ohtake et al. (www.mpi-sb.mpg.de/∼ohtake/mpu_implicits). Our method has been applied to a bronchial tree (BT), a liver tree (LT), a cerebral tree (CT) and an aneurysm (AN). All reconstructions were generated using parameter values returned by the above parameter determination rules (Eq. 1, 2 and values of table 1). The generated surfaces are smooth and do not exhibit spherical artifacts as can be seen in Figures 2, 9, 10 and 11. A careful visual examination of the surfaces near branchings showed that geometric continuity was achieved for all kinds of branchings and branching angles. We compared the results of our algorithm with the results of MC, CS and CSEN when applied to the same segmentation result. A visual comparison of the MPUI results with the respective MC results gave first evidence that morphology and topology are represented correctly applying MPUIs. No tree structures are suppressed and no branches occur that are not represented in the data. The visualization with MPUIs

Dataset	resolution	voxel size	ε_0	$level_{max}$	cell size
BT	$343 \times 193 \times 259$	$0,57 \times 0,57 \times 1$	0,00184	8	0,0020
LT	$342 \times 256 \times 81$	$0,633 \times 0,633 \times 2$	0,00265	6	0,0028
CT	$322 \times 233 \times 180$	$1 \times 1 \times 1$	0,00199	8	0,0022
AN	$129 \times 107 \times 45$	$1 \times 1 \times 1$	0,00503	6	0,0055

Table 1: *Summary of characteristics of the data sets. Voxel sizes are given in millimeters. The three rightmost columns show the automatically derived parameter values for the octree subdivision (ε_0 and $level_{max}$) and the polygonization of the implict function (cell size).*

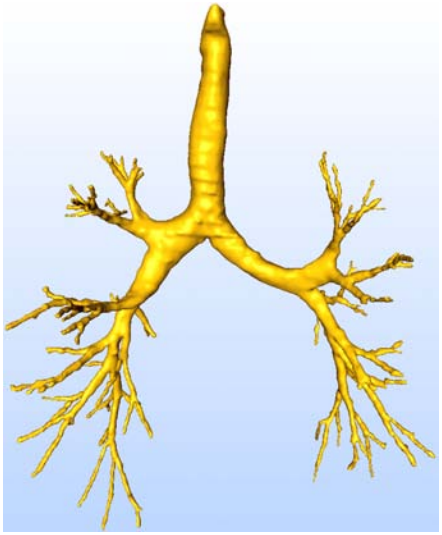


Figure 9: *Reconstruction of a bronchial tree with MPUIs.*

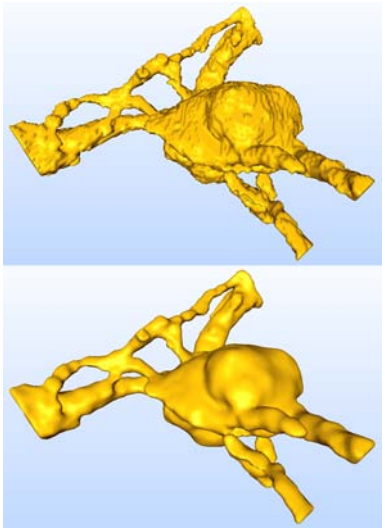


Figure 10: *Reconstruction of an aneurysm with MC (top) and MPUs (bottom).*

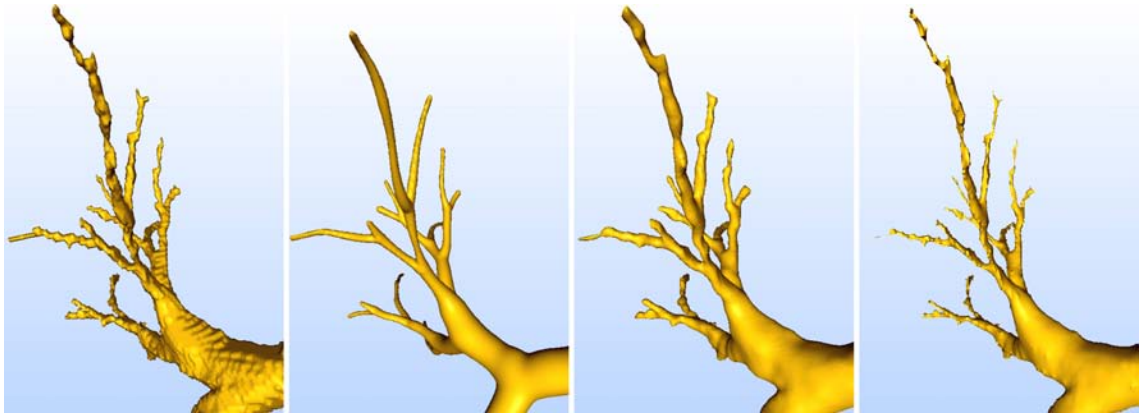
avoids staircase artifacts which hamper the visual quality of surfaces reconstructed using MC (see Fig. 10 and Fig. 11 (a) and (c)). Subtle surface features are represented correctly in contrast to the reconstruction with CS. The latter guarantees a high geometric continuity, however, due to the simplifying model assumption of circular cross-sections the morphology is only approximated (compare Fig. 11 (b) and (c)). Very thin elongated structures are reconstructed faithfully based on the subsampling described in section 4.2. Here, our approach outperforms CESN, which generate a surface that collapses to a line with an increasing number of smoothing steps (see Fig. 1 and 11 (d)).

6.1. Reconstruction Accuracy

We compared our method with CS and CSEN with respect to accuracy by measuring the respective deviations to the MC results using AMIRA (www.amiravis.com). Although it is not the most accurate method to visualize a segmentation result, MC is considered as the "'gold standard"' to which a validation should relate since it is widely used in radiological workstations. Table 2 lists the median and the maximum of the surface distances. The median of the deviation is very similar to the one of CESN. However, the maximum is larger for our method and exceeds the specified maximum deviation of half a voxel diagonal. Those outliers occur due to small concave features which are not represented by enough points. 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. Although our method suffers from minor inaccuracies, it is allmost as accurate as CESN while representing the segmentation result more faithfully than CS. The latter exhibits strong deviations due to the underlying model assumptions.

Data	CESN		CS		MPUI	
-set	med	max	med	max	med	max
BT	0,17	0,44	0,31	5,53	0,16	1,40
LT	0,14	0,47	0,30	6,87	0,15	0,84
CT	0,21	0,39	0,32	2,70	0,20	1,68
AN	0,14	0,40	0,72	6,25	0,19	1,91

Table 2: *Accuracy of CESN, CS and MPUI. The table lists the median (med) and the maximum (max) of the deviations to the corresponding MC results. All values are given in relation to the length of one voxel diagonal of the respective volume data set.*



(a) (b) (c) (d)

Figure 11: *A detailed look at the brochial tree: Recontruction using MC (a), CS (b), our method (c) and CESN (d).*

6.2. Efficiency of the Reconstruction

In Table 3, we summarize information concerning the geometric complexity of the resulting models and the performance. The setup time includes the point cloud generation which makes up approximately 30% of the entire time. The reconstruction times with MPUIs are in the range of 5 sec. for AN and 38 sec. for BT which represents the largest and most complex vessel tree that has been investigated. CS perform equally except for the CT. Since this tree only consists of less ramified and long straight parts, the reconstruction strongly benefits from bounding volumes described in [OP04]. Reconstruction times with MC and CESN are generally shorter. Concerning the latter approach, time depends on the number of smoothing steps. This has been set to 8, since smaller values result in a decreased smoothness whereas larger values tend to produce artificial edges and plateaus. With MPUI, a speed-up may be achieved at the cost of smoothness as indicated in section 5. In general, the reconstruction time depends on the resolution of the original data, the complexity of the vessel tree, and the size of the partitioning element. The number of polygons generated with MPUI is still in a range that modern graphics hardware may process at interactive frame rates. It is mostly larger than the one generated with CS since a slightly smaller partitioning element has been chosen. Compared to MC and CESN, it is up to 1.2 times larger (see Table 3) due to the computed size of the partitioning element (see Eq.3). This computation leads to an element smaller than the original voxel which is precisely the partitioning element for MC and CESN. The stronger polygon count deviation for LT (see Table 3, row 2) follows from the anisotropic voxel size of this data set (see Table 1). Hence, more than two partitioning elements fit into one voxel.

Data	MC		CESN		CS		MPUI	
-set	p	t	p	t	p	t	p	t
BT	166k	3	166k	13	201k	36	180k	38
LT	80k	2	81k	5	125k	15	167k	14
CT	115k	3	115k	8	93k	9	142k	28
AN	54k	1	50k	2	39k	11	61k	5

Table 3: *Efficiency of MPUI compared to MC, CESN and CS. The table lists the number of generated triangles (p) and the reconstruction time (t) in seconds.*

6.3. Surface Smoothness

Visual inspection indicated a smooth reconstruction. In addition, we analyzed the distribution of curvature values on the reconstructed surfaces. Figure 12 shows the histograms of curvature values for the reconstruction of the BT with MC and our method. The number of vertices in the different classes varies strongly for MC. Vertices with different curvature values are equally distributed all over the surface. There is no correspondence to the vessel radius. The surface that has been reconstructed with our method exhibits small curvature values only on thick branches and high curvature values only on thin branches.

7. Conclusion

We have presented a method for visualizing vascular tree structures, which adheres to the binary result of the vessel segmentation, while producing smooth transitions at branch-

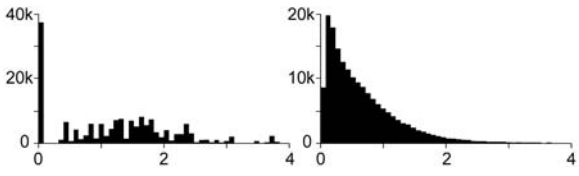


Figure 12: *Histograms of curvature values for the MC reconstruction (left) and the MPUI reconstruction (right).*

ings. The 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, we are able to accurately represent even very thin vessels. The patient-individual automatic determination of appropriate default parameter values for accurate and smooth reconstructions may facilitate the acceptance of our reconstruction technique in clinical routine. We applied our method to a variety of vessel trees and compared the results with other state-of-the-art techniques (MC, CESN, CS) regarding reconstruction accuracy and efficiency. For that purpose, we analyzed surface differences between the isosurface and CESN, CS, and MPUIs. With an average median deviation of 0.18 voxel diagonals, our approach is almost as accurate as CESN (0.17). It represents the segmentation result more accurately than CS (0.41), especially for pathologic cases (see Fig. 10). The low surface differences indicate that our method is suited for vessel diagnosis. The accuracy could be further increased by adapting our subsampling strategy for small concave features. To be employed in therapy planning and for exploration purposes, a mapping should be established between the reconstructed surface and a graph representation of the vessel tree as applied in [HPSP01]. An incorporation of Convolution Surfaces for the reconstruction of very thin vessels could further improve the results since an exact cross section for such vessels is not given by the segmentation result and is not of big diagnostic relevance. Although reconstruction times of our method are in the range of seconds, there is room for improvement. The use of an adaptive polygonization method for example could drastically reduce the reconstruction time and the number of polygons.

References

[BHP06] BADE R., HAASE J., PREIM B.: Comparison of Fundamental Mesh Smoothing Algorithms for Medical Surface Models. In *Simulation und Visualisierung* (2006), pp. 289–304.

[Blo94] BLOOMENTHAL J.: An Implicit Surface Polygonizer. In *Graphics Gems IV*. 1994, pp. 324–349.

[BMM*07] BRAUDE I., MARKER J., MUSETH K., NISSANOV J., BREEN D.: Contour-Based Surface Reconstruction using MPU Implicit Models. *Graphical Models 69*, 2 (2007), 139–157.

[BRB05] BORNIK A., REITINGER B., BEICHEL R.: Reconstruction and Representation of Tubular Structures using Simplex Meshes. In *Proc. of Winter School of Computer Graphics (WSCG)* (2005), pp. 61–65.

[BS91] BLOOMENTHAL J., SHOEMAKE K.: Convolution Surfaces. *Computer Graphics 25*, 4 (1991), 251–256.

[Gib98] GIBSON S. F. F.: Constrained elastic surface nets: Generating smooth surfaces from binary segmented data. In *Proc. of MICCAI* (1998), pp. 888–898.

[HPea00] HÖHNE K.-H., PFLESSER B., ET AL. A. P.: A Realistic Model of the Inner Organs from the Visible Human Data. In *Proc. of MICCAI* (2000), pp. 776–785.

[HPSP01] HAHN H. K., PREIM B., SELLE D., PEITGEN H.-O.: Visualization and Interaction Techniques for the Exploration of Vascular Structures. In *IEEE Visualization* (2001), pp. 395–402.

[JR06] JALBA A. C., ROERDINK J. B. T. M.: Efficient surface reconstruction from noisy data using regularized membrane potentials. In *IEEE / Eurographics Symposium on Visualization* (2006), pp. 83–90.

[LC87] LORENSEN W. E., CLINE H. E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In *Proc. of ACM SIGGRAPH* (1987), pp. 163–169.

[MMD96] MASUTANI Y., MASAMUNE K., DOHI T.: Region-Growing-Based Feature Extraction Algorithm for Tree-Like Objects. In *Proc. of Visualization in Biomedical Computing* (1996), pp. 161–171.

[OBA*03] OHTAKE Y., BELYAEV A., ALEXA M., TURK G., SEIDEL H.-P.: Multilevel Partition of Unity Implicits. *ACM Transactions on Graphics*, 22 (2003), 463–470.

[OP04] OELTZE S., PREIM B.: Visualization of Vascular Structures with Convolution Surfaces. In *IEEE / Eurographics Symposium on Visualization* (2004), pp. 311–320.

[SPSP02] SELLE D., PREIM B., SCHENK A., PEITGEN H.: Analysis of Vasculature for Liver Surgical Planning. *IEEE Transactions on Medical Imaging 21*, 11 (2002), 1344–1357.

[SSN*98] SATO Y., SHIRAGA N., NAKAJIMA S., TAMURA S., KIKINIS R.: Local Maximum Intensity Projection (LMIP): A new Rendering Method for Vascular Visualization. *Journal of Computer Assisted Tomography 22*, 6 (1998), 912–917.

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *Proc. of ACM SIGGRAPH* (1995), pp. 351–358.

[VMM99] VOLLMER J., MENCEL R., MUELLER H.: Improved laplacian smoothing of noisy surface meshes. In *Proc. of EuroGraphics* (1999), pp. 131–138.

[VST*03] VEGA F., SAUBER N., TOMANDL B., NIMSKY C., GREINER G., HASTREITER P.: Enhanced 3D-Visualization of Intracranial Aneurysms Involving the Skull Base. In *Proc. of MICCAI* (2003), pp. 256–263.

[ZO02] ZHAO H., OSHER S.: Visualization, analysis and shape reconstruction of unorganized data sets. In *Geometric Level Set Methods in Imaging, Vision and Graphics*. 2002, pp. 1–25.

[Zui95] ZUIDERVELD K.: *Visualization of Multimodality Medical Volume Data using Object-Oriented Methods*. PhD thesis, Universiteit Utrecht, 1995.