Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2004)
O. Deussen, C. Hansen, D.A. Keim, D. Saupe (Editors)

Visualization of Anatomic Tree Structures with Convolution Surfaces

S. Oeltze[1] and B. Preim[1]

[1] Department of Computer Science, The Otto-von-Guericke-University of Magdeburg, 39106 MD, PO Box 4120, Germany
steffen.oeltze@student.uni-magdeburg.de | preim@isg.cs.uni-magdeburg.de

Abstract

We present a method for visualizing anatomic tree structures, such as vasculature and bronchial trees based on clinical CT- or MR data. The vessel skeleton as well as the diameter information per voxel serve as input. Our method adheres to these data, while producing smooth transitions at branchings and closed, rounded ends by means of convolution surfaces. We discuss the filter design with respect to irritating bulges, unwanted blending and the correct visualization of the vessel diameter. Similar to related work our method is based on the assumption of a circular cross-section of vasculature. In contrast to other authors we employ implicit surfaces to achieve high quality visualization. The method has been applied to a large variety of anatomic trees and produces good results. The time to construct a geometric model is reduced by means of different bounding volumes and careful choice of parameters for polygonization.

1. Introduction

In medical education as well as in therapy planning, the visualization of tree-like structures is crucial. It is desirable that spatial relations, in particular the topology of nerves or vascular trees, can be correctly inferred from the visualization. Moreover, the curvature, the depth relations, and the diminution of the diameter towards the periphery should be depicted correctly. Traditional methods of medical volume visualization, such as direct volume rendering, threshold-based isosurface rendering, or maximum intensity projection are not well-suited for the above-mentioned goals. Due to image noise and the limited resolution of computed tomography (CT) or magnetic resonance imaging (MRI) conventional methods produce noisy 3d visualizations. Also, the visual separation of contrast-enhanced vascular structures and other high-intensity structures, such as bones, might be very difficult. Artifacts and discontinuities in the visualization distract the viewer. For educational purposes and therapy planning, vascular structures should be reconstructed based on the radiological data of that patient and some model assumptions as to the shape of vasculature (Gerig et al. [GKS*93]).
We describe a new method for vessel visualization which originates from implicit modeling. This method uniformly handles anatomic trees with different types of branchings and produces smooth surfaces even at endpoints and branchings. The vessel skeleton serves as the input for the visualization process. It is represented as a directed graph [SPSP02]. Edges of this graph are approximated by line segments connecting adjacent voxels. Each segment is described by its two endpoints and one associated radius (the radius of the largest circle completely enclosed by the segmentation result), respectively. Usually, these graphs represent tree structures in a mathematical sense. However, our method does not rely on this property. The visualization method applied here is based on *Convolution Surfaces* introduced by Bloomenthal and Shoemake [BS91]. As a difference to previous applications of convolution surfaces we shall faithfully represent the radius distribution given by the vessel analysis and we shall employ large datasets (>10K line segments).

2. Prior and Related Work

Following the pioneering work of Gerig et al. (recall [GKS*93]) several attempts have been made to develop special visualization techniques for anatomic tree structures in general and vascular trees in particular. Our visualization technique is based on the model assumption that cross sections of non-pathologic vessels have a circular shape, as discussed in [MMD96]. Masutani et al. fitted cylinders along the skeleton to visualize vascular structures. The drawback of this method is that discontinuities at branchings arise where cylinders with different diameters coincide. A special problem, the visualization of cerebral structures, has been tackled by Puig et al. [PTN97]. She modeled typical bifuractions and pathologic situations, such as aneurysms and stenosis, and fitted these models to the patient specific data at hand. The focus of her work is on the geometric continuity and on realistic shading.

In [HPSP01] a vessel visualization pipeline has been introduced. The skeleton and the local vessel diameter determined in the image analysis stage are smoothed with a [121] binomial filter. Special care is taken at branchings to weight the incident branches. In the mapping step truncated cones are used to represent the diminutions of the vessel diameter faithfully. However, the fitting of primitives to one another does not generate smooth transitions between vessel segments at branchings. In [FFKW02], branchings are treated separately to avoid rendering discontinuities and the construction of objectionable structures inside the geometric model. Along straight parts, a vessel segment is approximated by four quadrilateral patches. The method is based on the iterative refinement of a rough, initially constructed basic mesh by means of subdivision surfaces. The smoothness of the final model is dependent on the iteration depth of the refinement step. A comprehensive survey of methods for vessel visualization is given in [BFLC].

3. Modeling of Tree-like Structures with Implicit Surfaces

Implicit surfaces offer an alternative to explicitly constructing the surface of an object by a set of polygons or parametric patches. They describe the surface by an equation which is often more compact than its parametric counterpart. Especially in modeling smooth, deformable objects, implicit surfaces unfold their full strength. James F. Blinn introduced implicit surfaces in computer graphics. He developed *Blobby Molecules* to visualize electron density fields [Bli82].

3.1. Implicit Surfaces: a Brief Introduction

A classical example for an implicit equation is the description of a sphere with radius r: $x^2+y^2+z^2-r^2=0$. This formula may be used to represent all points ***p****(x,y,z)* in space which are on the surface of a sphere with radius r. Furthermore, it facilitates point classification by simply checking the sign of the resulting scalar value. According to [OM95], the above equation can be rewritten as Eq. 1:

$$F(p) - Iso = 0 \tag{1}$$

$F(p)$ is called the *scalar field function* because a scalar value may be computed for each point p. *Iso* denotes the *isovalue* for generating an *isosurface* which represents the surface where the implicit equation is zero. Naturally, the surface also depends on F whose variation results in diverse scalar fields around a point. In order to create more complex objects, several points might be specified whose scalar fields overlap if close enough.

The scalar field function employed for modeling 'blobs' [Bli82] is given in Eq. 2:

$$F(p) = be^{-\sigma d^2} \tag{2}$$

where $d = \sqrt{x^2+y^2+z^2}$ is the Euclidean distance between point p and the center of the source of energy. Eq. 2 describes a Gaussian bump centered at d, having height b and standard deviation σ. For several energy sources, the scalar value at p can be calculated as Eq. 3:

$$F(p) = \sum_i b_i e^{-\sigma_i d_i^2} \tag{3}$$

Other popular field functions were presented in [NHK*85] (*Metaballs*) and [WMW86a] (*Soft Objects*). Note that only scalar fields around point primitives were considered so far.

3.2. Convolution Surfaces

Bloomenthal and Shoemake extended the concept to skeletal primitives of theoretically any kind, e.g. line segments, polygons, or planar curves [BS91]. This enhancement overcomes a major drawback of point primitives namely their deficiency in describing flat surfaces and smooth generalized cylinders. They propose *Distance Surfaces* and *Convolution Surfaces* to model the surface of an object around its skeleton; these are described below. In the following, S denotes a skeleton and s refers to a single point on the skeleton.

Distance Surfaces demonstrate one way to generate an implicit surface. Unfortunately, depending on the blending function distance surfaces exhibit bulges or creases where two skeletons form an entity. This effect is undesirable for the faithful visualization of anatomic trees.

Convolution Surfaces avoid bulges and creases for non-branching skeletal structures. The scalar value is calculated according to Eq. 4:

$$F(p) = f(S,p) = \int_S \left(e^{\left(\frac{-\|s-p\|^2}{2} \right)} \right) ds \tag{4}$$

where $f(S,p)$ is the convolution of a skeleton with a three-dimensional Gaussian filter. In contrast to distance surfaces, the value is computed considering *all* points of the skeleton by integration. The resulting surface does not show creases and is bulge-free for non-branching skeletons. Since our method is based on convolution surfaces we present more detail in the following.

Convolution surfaces utilize a concept which is well known from signal processing namely the modification of a signal by a filter. For a Gaussian filter function with height 1 and standard deviation $1/2$ Eq. 4 may be rewritten as

$$F(p) = f(S,p) = (h \otimes S)(p) \tag{5}$$

where S is the signal, h is the filter function and $\otimes$ denotes the convolution operator. For the visualization of vasculature, the vessel skeleton corresponds to the signal. The constitutive idea is to smooth this signal and thereby letting high frequencies gently drop off in the neighborhood. The resulting field around the skeleton corresponds to the scalar field mentioned earlier. By constructing an isosurface through this field the convolution surface is formed.

For the understanding of bulge-free blending it is necessary to elaborate on the superposition property of convolution:

$$h \otimes (S_1 + S_2) = (h \otimes S_1) + (h \otimes S_2) \tag{6}$$

> This guarantees, for example, that two abutting, collinear segments [...] produce the same convolution as does the single segment that is their union. [Blo95]

Furthermore, superposition has an impact on implementation issues and the modeling process. It permits the convolution of a complex object primitive by primitive in an arbitrary order instead of considering the skeleton as a whole.

3.3. Filter Selection

According to [She98], an appropriate filter function should be continuous and monotonic. Furthermore, it should have finite support (or be negligible beyond a certain distance), and exhibit zero or near zero gradient at this distance. These requirements restrict the filter selection to low-pass filters. The Gaussian is a prime example for such filter functions. [She98] includes an analysis of the most widespread functions with respect to computational complexity and the types of modeling primitive for which a closed-form integral solution may be obtained. For convolving a line segment each of the surveyed kernels is applicable. However, the scope of eligible filter functions is strongly reduced when a convolution surface should faithfully represent a given radius distribution. As elucidated below, only three kernels were examined with respect to this problem.

Initially, the convolution surface along a line segment has a fixed radius. In [She98], the creation of a surface that resembles a tapered cylinder by using linear profiling functions is described. The use of Bezier curves with two control points for representing linear radius distributions is introduced in [JTFP01]. Both methods are convenient when interactive radius modification is required. However, they are inadequate for modeling surfaces that exactly converge against a given radius. As far as we know, solely in [Blo95] and [HAC03] approaches have been published that concentrate on this problem.

In [Blo95] a Gaussian function is utilized for convolution (see Fig. 1(a)).

$$h(p) = e^{-d^2\omega}, \ \omega = \ln 2, \ d > 0 \tag{7}$$

ω is known as the *width coefficient* and equals $1/(2\sigma^2)$, where σ is the standard deviation.

In [HAC03] two other methods are published to correctly represent a linear radius distribution. The first approach utilizes the kernel (see Fig. 1(b))

$$h(p) = \frac{1}{d(p,H)^3} \tag{8}$$

where $d(p,H)$ is the minimal distance between point p and the line which proceeds through the considered line segment (if p is near the end of the segment, H might be beyond the segment).

This kernel delivers a good and computationally fast approximate solution. A mathematically more exact approach is presented using the new filter function (see Fig. 1(c), where $r(H) = 1$)

$$h(p) = \frac{r(H)^2}{d(p,H)^2} \tag{9}$$

Here, the radius is integrated into the kernel before convolution which maintains the good properties of convolution surfaces, e.g. their invariance under skeleton subdivision [HAC03].

3.3.1. Discussion

Computational speed determined our choice of a filter function. By definition of the convolution surface (Eq. 4), the entire skeleton needs to be considered when calculating the scalar value at a point p. For the visualization of vascular structures this means a prohibitively high computational effort. To improve the performance, we will restrict the computation of the scalar field using bounding volumes along line segments. The tightness of a bounding volume is heavily dependent on the filter function, in particular on the distance from the center where the function value is neglible. We choose the Gaussian because it drops much faster to zero than the other two kernels (Fig. 1).

3.4. Correct Representation of the Radius Distribution

According to [Blo95], the computation of the convolution integral in Eq. 4 may be simplified by separating it into the product of an *integration filter* and a *distance filter*. Whereas the first term requires solving a one-dimensional integral the second is simply a single evaluation of the kernel:

$$h(p) = e^{-(d(p,H))^2\omega} \tag{10}$$

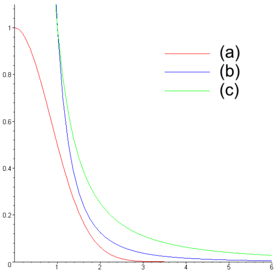

Figure 1: *Filter functions for convolution surfaces, (a): Gaussian [Blo95], (b)* $1/d(p,H)^3$ *and (c)* $r^2/d(p,H)^2, r = 1$ *as used in [HAC03]*

The separation of the computation into two filters is crucial for the evaluation of the implicit function. Instead of the three-dimensional integration (Eq. 4) we have to solve a one-dimensional integral which can be precomputed and stored in a lookup table. We use the same lookup table as [Blo95] with 10000 entries.

For adapting the radius of the resulting convolution surface, $d(p,H)$ is divided by radius $r(H)$. $r(H)$ is determined with linear interpolation between the radii at the segment endpoints.

In order to let the convolution surface converge against a desired radius, appropriate iso-values and width coefficients ω must be selected. Bloomenthal employed an isovalue of $1/2$ so as to let the convolution surface pass through the segment endpoints. Now, let us consider the convolution surface of a sufficiently long cylinder and a point p which is located exactly on the surface and in the middle of it. Here, the integration filter equals 1 since the kernel is fully subtended by the segment. With the constraint that $d(p,H) = r(H)$ for point p on the convolution surface it follows:

$$F(p) = e^{-(r(H)/r(H))^2\omega} - 1/2 = e^{-\omega} - 1/2 = 0 \qquad (11)$$

Thus, $\omega = \ln 2 \approx 0.6931$.

4. Blending

The ability to create smooth transitions between simple objects intending to form a complex organic shape is a strength of implicit surfaces. Here, so-called *blends* are used instead of parametric free-form surfaces.

For convolution surfaces, blending corresponds to an integration of the filter along the entire skeleton. At the skeleton joints, the scalar fields of adjacent primitives overlap. The convolution surface constructed through the resulting field forms a smooth envelope of the underlying joint. In an implementation, each primitive may be convolved separately due to the superposition property of convolution. Blending may also have negative effects for a faithful visualization of anatomic tree structures which are discussed in the following.

4.1. Blending Strength at Branchings

With the initial filter design in [Blo95], the transitions at branchings were very smooth but deviated strongly from the skeleton (see Fig. 2 (middle)). Discussions with an experienced radiologist showed that this is undesirable and in some cases the interpretation of the topology is even hampered. We propose a narrower filter kernel which results in a surface which tracks the skeleton more faithfully.

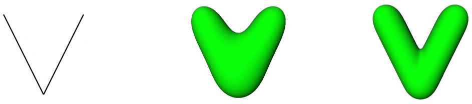

Figure 2: *Blending strength, left: skeleton, middle: high blending strength, right: reduced blending strength*

4.2. Unwanted Blending

For precise modeling of complex shapes it is essential to control the blending between certain parts of the object. Concerning vascular structures, vessel segments whose skeleton is not connected should not blend with each other. In [OM93] the use of a *restricted blending graph* was suggested to overcome this problem. Based on the topology of the given skeleton, primitives are classified into blending-groups of the following types: all primitives blendable, all primitives unblendable, blendable and unblendable primitives. Unfortunately, this solution does not ensure C^1 continuity of the shape. A more recent approach [AJC02] introduces *local convolution*. This concept is based on defining a restricted skeleton range for computing the scalar value at a certain point in space. It prevents blending even between small folds of the skeleton.

4.3. Bulging

Convolution surfaces are bulge-free for non-branching line segment skeletons due to the superposition property of convolution. However, as shown in [Blo95], they do exhibit bulges at branchings (see Fig. 3). This effect is disturbing for the visualization of vascular structures since a bulge might be easily mistaken for a pathological variation, e.g. an aneurysm. Even though our visualization method is not targeted at supporting vascular diagnostics, we shall minimize this drawback. In [Blo97], two methods are described that address the bulging problem.

- *combination surfaces* which are an interpolation between *union surface* (distance surface with the maximum-operator as a blending function) and convolution surface.

Combination surfaces combine the advantages of union surface (bulge-free along convex parts) and convolution surfaces (crease-free along concave parts).

- The use of two dimensional skeletal elements, e.g. polygons is suggested.

Combination surfaces tend to produce artefacts, however, if branchings are very close to each other. Problems also occur if the normals of coinciding branches differ strongly. The use of two-dimensional skeletal elements eliminates bulges. However, the convolution of a polygon is computationally much more expensive.

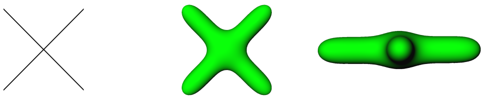

Figure 3: *Bulging problem, left: skeleton, middle: convolution surface, right: Bulge at the branching in horizontal view*

4.4. Consequences

As a consequence of our analysis of different filters we choose the Gaussian [Blo95] for the convolution. However, we suggest a modification of the original kernel to reduce the blending strength at branchings. Employing the altered filter function has an impact not only on the blending strength but also on the unwanted blending problem as well as the bulging problem. Furthermore, the modified filter function the polygonization of scalar fields might be accelerated by means of tighter bounding volumes. These aspects are discussed in Sect. 5.

5. Visualization of Anatomic Tree Structures with Convolution Surfaces

In this section we describe how a modified width coefficient allows to solve the blending problems. We go on and discuss a preprocessing of the data in order to reduce the effort to compute the convolution surface. The same vessel analysis results same which have been used in [HPSP01] are the input for our method. We also employ the binomial filtering to smooth the vessel skeleton and diameter.

5.1. Filter Modification

We carefully evaluated different width coefficients using a variety of datasets and found that a value of $5ln(2)$ is suitable to prevent the undesired effects such as unwanted blending and bulging. Note, that the width coefficient has been increased with the effect that the filter function is narrower (see Fig. 4). In order to correctly represent the radius distribution along a line segment, a recalculation of the isovalue is required. Under consideration of our new width coefficient the isovalue (Iso) is evaluated as follows:

$$F(p) = e^{-(r(H)/r(H))^2 5\ln 2} - Iso = e^{-5\ln 2} - Iso = 0 \quad (12)$$

Hence, $Iso = 1/32 = 0.03125$.

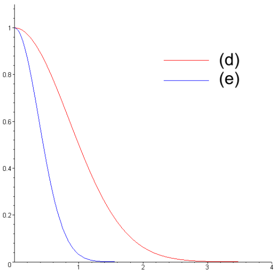

Figure 4: *Original Gaussian function (d) from Bloomenthal compared to the modified version of our method (e).*

To evaluate our filter function, we employ a simple skeleton with a trifurcation (4 coinciding branches). In anatomic tree structures, we never found more branches coinciding. In Fig. 5 it is illustrated how the blending strength is reduced for the simple skeleton. Also, the bulging problem is avoided with the modified filter function (see Fig. 6). To study unwanted blending, we use an S-shaped skeleton with 3 mm distance between the horizontal lines. The radius was increased until the problem occurred. With the original filter, unwanted blending appeared for a radius of 1.07 mm, whereas with the modified width coefficient, it appeared at 1.37 mm. The ratio between the distance of the convolution surfaces and the distance of the centerlines determines the occurrence of unwanted blending. With the modified width coefficient, we could reduce the ratio from 29% (see Fig. 7 (left)) to 9% (see Fig. 7 (right)).

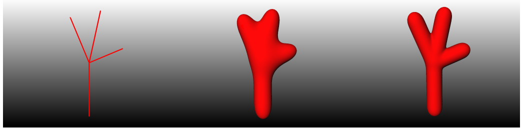

Figure 5: *Transition at branching, left: skeleton, middle: convolved with original filter, right: convolved with modified filter*

5.2. Computational Complexity

For the construction of the convolution surface it is necessary that for every point in space the scalar value can be evaluated. Per definition of the convolution surface this permanently requires considering the whole tree which is very unsatisfying with respect to the computation time. By contrast to previous applications of convolution surfaces with small

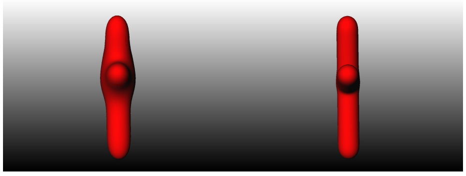

Figure 6: *Side view to skeleton used in Fig. 5. The bulging problem (left) is avoided with the modified filter (right).*

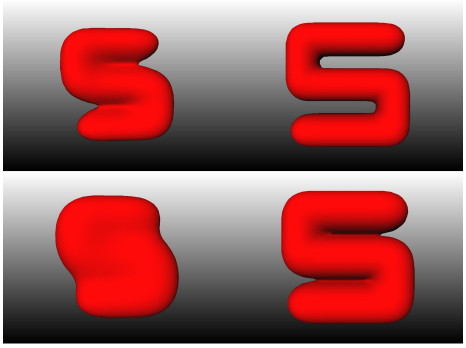

Figure 7: *Unwanted blending, Distance between the branches of the S-shapes is 3 mm. The radius of all branches is 1.07 mm in the upper row and 1.37 in the lower row. Unwanted blending (left) is considerably reduced with the narrower filter (right).*

to moderate-sized structures we want to display anatomic trees with more than 1000 branchings.

In order to accelerate the computation, we determine which line segments have a significant influence on a point and neglect the small influence of the remaining segments. For this purpose, a partitioning of space is necessary. We employ the voxel grid from the underlying data as a basis for the partitioning. This strategy has two advantages:

- We partition the space with the same resolution as the underlying data.
- All computations are simple to carry out because the skeletonization result is represented in voxel coordinates.

Our preprocessing proceeds as follows: We iterate over all line segments and construct a cylindrical bounding volume (CBV). The cylindrical shape of the bounding volume allows to closely approximate the shape of the convolution surfaces. The radius of the CBV is twice the maximum radius along the line segment.

The CBV is later employed to check for a voxel the inclusion in a CBV. Only those voxels which pass the test are influenced by the current line segment (see Fig. 9). For the realization of the CBV-test we have to solve two problems:

- to efficiently check whether a point (the center of a voxel) is inside a cylinder, and
- to restrict the voxels which undergo the inclusion test.

Concerning the first problem, an implicit description of a cylinder is appropriate for a fast and easy test of inclusion. With respect to the second problem, for each line segment an axis-aligned bounding box (AABB) is computed in voxel coordinates (see Fig. 8). The test of inclusion in the CBV is performed (only) for all voxels inside the AABB of the current line segment (see Fig. 9). The preprocessing results in a list with entries of the form $< voxel\_id, line\_segments >$ representing which line segments have an influence on the voxel uniquely specified by *voxel_id*. The use of the CBV accelerates the whole computation by two orders of magnitude (factor 100-200).

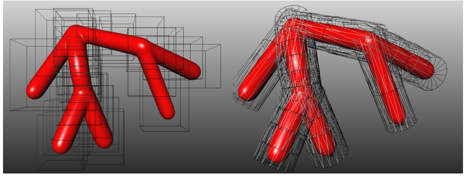

Figure 8: *Axes-aligned bounding boxes (left) and cylindrical bounding volumes (right) for an artificial tree (right)*

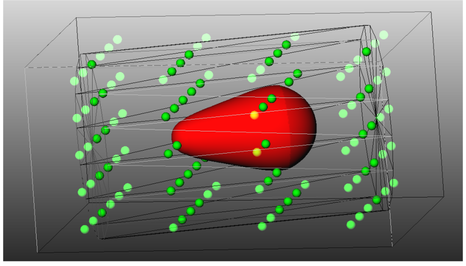

Figure 9: *Voxel selection for a single line segment. Green spheres mark midpoints of voxels which are within the AABB of this segment. Dark green spheres mark midpoints of voxels which passed the cylindrical bounding volume test.*

5.3. Construction of a Geometric Model

For the visualization of the vascular structures the convolution surface is transformed into a triangle mesh. We developed an object-oriented version of the *Implicit Polygonizer* [Blo94].

The implicit polygonizer partitions the space surrounding the surface based on a continuation scheme presented in [WMW86b]. An initial cube is centered at an arbitrary point on the convolution surface. We use the root in our tree data structure as seed point for computing the position of the initial cube. The size of the cube is derived from the voxel size in the underlying data to reflect the resolution of the data. As a typical example, for a voxel size of 0.7*x*0.7*x*3 mm, we choose 0.7 mm as size of the cube. Thus we can ensure that all parts of the anatomic tree structure are considered.

Initially, we have to determine two points p_1 and p_2 with $f(p_1) > 0$ and $f(p_2) < 0$. The first point is simply our seed point which belongs to the skeleton and is therefore inside the surface. To find p_2 (outside and with negative value of f), we evaluate the implicit function f in the neighborhood of p_1. According to [Blo94] random numbers within a growing neighborhood are employed for this search. The zero-crossing of f represents a point p_surf on our surface which is determined by binary subdivision between p_1 and p_2 f. The initial cube is translated so that it is centered around p_surf. The stop criterion for the binary subdivision has a critical influence on the overall performance. Bloomenthal suggests to terminate after 10 iterations. For our purpose, it turns out that 3 iterations are sufficient. With this criterion and regarding that f is monotonic we can ensure that the error is at most 1/8 of the size of the cube (usually about 0.1 mm).

For the evaluation of f we employ the data structure generated in the preprocessing step (recall Sect. 5.2). We determine which voxel the currently evaluated point belongs to and then look up which line segments influence it. The evaluation of f is accelerated using a lookup table for the integration filter (recall Sect. 3.4).

In the next stage, f is evaluated for all vertices of a cube and those faces consisting of vertices with opposite polarity are determined. Continuation proceeds by generating new cubes (with the same size) across any such face; this process is repeated until the surface is closed (for each cube an adjacent cube with respect to the 6-neighborhood is generated).

In order to avoid ambiguity cases, we employ the strategy described in [Blo94]. The polygonization process usually results in a global list of coordinates, normals and indices. We modified the data structures so that a separate list of indices for coordinates and normals is maintained for each edge. With this edge-related information interaction facilities based on the branching graph are feasible.

6. Exploration of Vasculature

In educational settings as well as in therapy planning systems it is often desirable to restrict the visualization of anatomic tree structures or to focus it on certain subtrees. For example, in tumor surgery, vessel segments around the tumor are more important than more distant parts. In general, surgeons would like to know which vessel segments must be reconstructed, if they have to be cut (those with a diameter above 5 mm). The vessel analysis results allow for interactions to support such visualization goals. Based on the branching graph, vessel segments which depend on a user-selected branch might be highlighted or removed from the visualization. Using a region-selection (called lasso selection), an arbitrary subset of vessels might be selected based on their position in the viewport. Instead of completely removing the selected edges it is often useful to show them less focused using gray colors and/transparency (see Fig. 10).

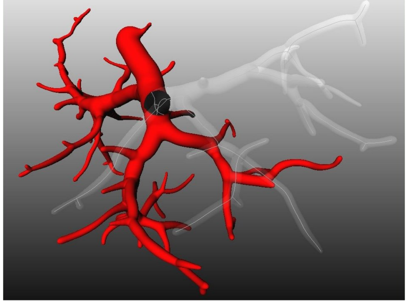

Figure 10: *Lasso selection and defocused visualization of the selected vessel segments. Inside the semitransparent vessels the centerline is presented.*

7. Validation

A crucial aspect for any visualization technique which is intended for clinical use is its validation. We carefully validated the segmentation and skeletonization approach which produces the underlying tree data (skeleton and vessel diameter, see [SPSP02]). With respect to the visualization of these data the validation should answer the following questions:

- Are (small) branches of the tree structure suppressed in the visualization (false negatives)?
- Are there occurrences of small branches which are not represented in the data (false positives)?
- Are there occurrences where separate branches appear as one branch?
- Is the constriction of the vessel diameter represented correctly?

These questions are relevant since the implicit modeling approach is more prone to such visualization errors than explicit methods where graphics primitives are explicitly fitted along the skeleton. We choose two different methods to answer the above questions: first we analyzed the visualization of artificial data with different branching types (e.g. with a trifurcation) and second we compared the implicit method with the method previously developed by Hahn et al. [HPSP01]. The tests with artificial data showed that unwanted effects could be strongly reduced. The unwanted blending problem may occur in rare cases.

The comparison of both methods (Fig. 11 and Fig. 12) reveals a good correspondence. The method by Hahn et al. produces longer branches at the leafs of the vessel tree. This is due to the construction where a half sphere is generated at the end of segments.

In order to further study the accuracy of our method, we compare it with an isosurface visualization of the vessel seg-

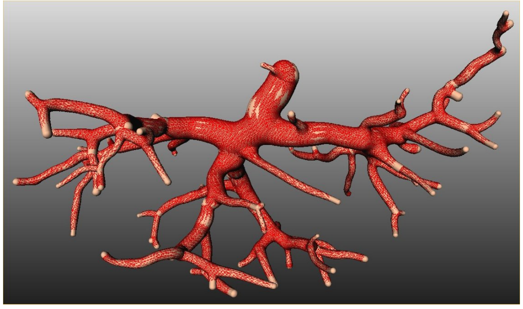

Figure 11: *The visualization method described by Hahn et al. [HPSP01] (concatenated truncated cones) and our method are combined. The convolution surfaces are rendered as wire-frame while the results of the other method are shaded.*

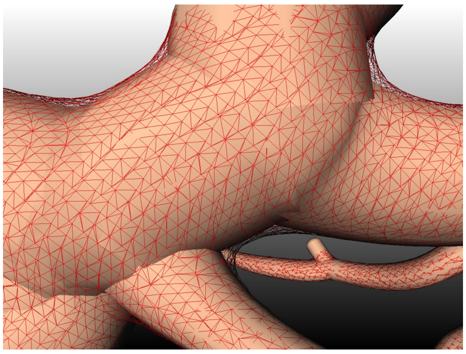

Figure 12: *A close-up of the visualization from Fig:11*

mentation result (the validation of the segmentation is beyond the scope of this paper). Differences between the segmentation result and the implicit surface visualization can be expected because:

- the vessel diameter which guides the convolution surface is the diameter of the incircle - therefore the convolution surface underestimates the segmentation result and
- the skeleton and the vessel diameter have been smoothed.

In Fig. **??** a comparison is shown. With respect to the four questions at the beginning of this section, we have evidence that none of the problems discussed actually occurs.

Over and above the visual inspection, a quantitative analysis is desirable. For this purpose, the convolution surface is transformed to a voxel representation V_conv. Based on this representation, measures can be derived which characterize the overlap of both sets of voxels or the distance between the contours. As a first attempt, we determined the similarity index [ZDMP94] between the segmentation result and convolution surface according to Eq. 13 and yielded a value of 0.88. This measure defines the volume overlap and is 1.0 if both sets are identical. The difference between both sets can be explained again with the use of the incircle - all voxels from V_conv are contained in V_seg. The quantitative validation, however, has to be extended to more datasets and other measures, in particular the Hausdorf-metric.

$$\frac{2 \mid V_i \cap V_j \mid}{\mid V_i \mid + \mid V_j \mid} \tag{13}$$

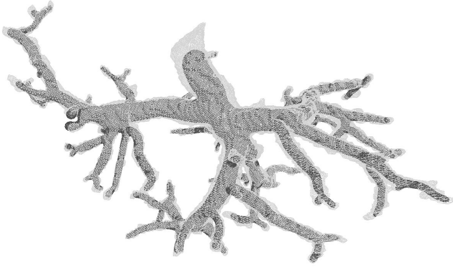

Figure 13: *The convolution surface is completely enclosed by the visualization of the vessel segmentation result.*

8. Results

Our method has been applied to 10 clinical datasets so far. In Fig. 14 we compare the visual quality of our method with another state-of-the-art technique (recall [HPSP01]).

Other examples for the visual quality of the visualization can be seen in Fig. 15-17. The visualizations do not exhibit any of the unwanted effects. We carefully examined the surfaces near branchings and noted that geometric continuity was achieved for all kinds of branchings and branching angles. The surfaces are terminated at leafs of the anatomic trees with a rounded appearance which is a consequence of the construction method. An example of a realistic scenario for our method is shown in Fig. 18 where the vessel visualization is embedded in the visualization of surrounding structures. We used a Pentium 4 CPU 3.06GHz, 1024MB RAM, ATI Radeon 9600.

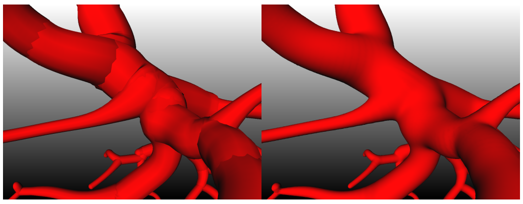

Figure 14: *Close-up of a visualization with truncated cones with artifacts along the seams. Smooth visualization with implicit functions of the same dataset.*

To give an idea of the complexity of the resulting geometry and the timings involved in the computation, we present

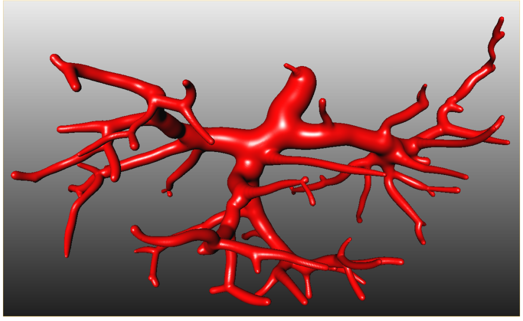

Figure 15: *Visualization of the portal vein derived from a clinical CT data set with 136 edges.*

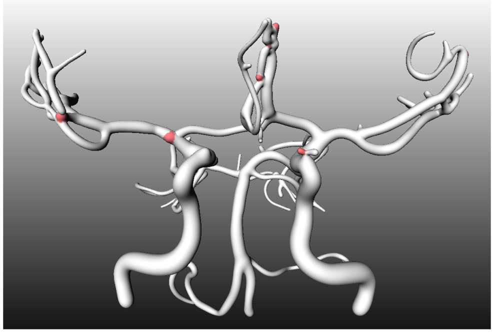

Figure 16: *Visualization of cerebral blood vessels derived from a clinical MR angiography with 149 edges.*

some results in Table 1. The setup time includes the preprocessing step. The first three lines represent the datasets shown in Fig. 15-17. The last line in Table 1 represents a complex anatomic tree from a corrosion cast. Due to the size of this model, the Open Inventor optimization failed which explains the low frame rate in the last row.

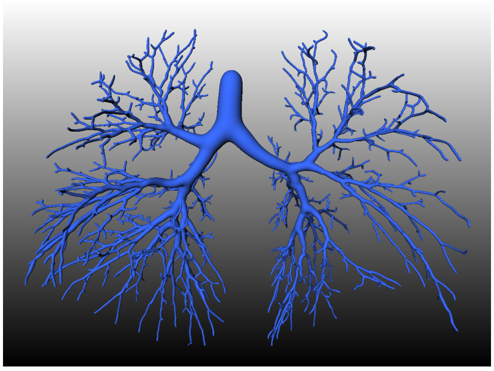

Figure 17: *Visualization of a bronchial tree derived from a clinical CT dataset with 1504 edges.*

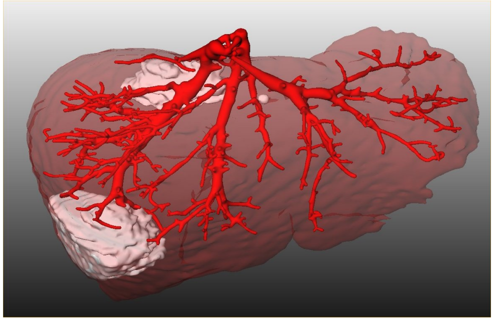

Figure 18: *Visualization of vascular structures with other intrahepatic structures for liver surgery planning.*

edges	line segments	triangles	setup time (s)	fps
136	1,652	124,884	7.5	64.0
149	2,623	252,836	9.8	42.8
1504	18,333	1,125,540	65.4	14.6
3461	23,973	2,366,008	65.0	1.8

Table 1: *Performance measurements for anatomic tree structures*

9. Conclusions

We have presented a method for visualizing anatomic tree structures, such as vasculature and bronchial trees, which adheres to the underlying data (skeleton and local vessel diameter), while producing smooth transitions at branchings. The method is based on convolution surfaces, a variant of implicit surfaces. The filter design has been fine-tuned to prevent irritating bulges and to represent the course of the vessel diameter faithfully. The width of the Gauss filter turned out to be the essential parameter to accomplish our visualization goals. With a preprocessing step which computes bounding volumes we can efficiently compute the convolution surface even for large vascular and bronchial trees.

We applied the method to a large number of vessel trees. The unwanted blending problem is not completely solved; however, it did not occur in clinical data. We compared our new method with another state of the art technique (recall [HPSP01]) and could show the superior visual quality. We analyzed the differences between the methods with respect to the precision and come to the conclusion that the new method is at least as close to the underlying data than the method by Hahn et al. There is room for improvement concerning the efficiency of the method. With an adaptive refinement approach it should be possible to reduce the time to polygonize the scalar field.

Acknowledgements

We want to thank M. Hindennach, H. Hahn and O. Konrad-Verse for providing the image analysis results on which this work is based and for fruitful discussions. Furthermore, we are indebted to J. Bloomenthal for his support on theoretical and implementation aspects. We are grateful to A. Sherstyuk, A. Angelidis and S. Hornus for many enlightening conversations. We also want to acknowledge the substantial comments provided by the anonymous reviewers.

References

[AJC02] Angelidis A., Jepp P., Cani M.-P.: Implicit modeling with skeleton curves: Controlled blending in contact situations. In *Shape Modeling International* (2002), ACM, IEEE Computer Society Press.

[BFLC] Bühler K., Felkel P., La Cruz A.: *Geometric Methods for Vessel Visualization and Quantification - A Survey*. Kluwer Academic Publishers.

[Bli82] Blinn J.: A generalization of algebraic surface drawing. *ACM Transactions on Graphics (TOG) 1*, 3 (1982), 235–256.

[Blo94] Bloomenthal J.: An implicit surface polygonizer. In *Graphics Gems IV*. Academic Press, Boston, 1994, pp. 324–349.

[Blo95] Bloomenthal J.: *Skeletal Design of Natural Forms*. PhD thesis, Univerity of Calgary, Calgary, Alberta, Canada, 1995.

[Blo97] Bloomenthal J.: Bulge elimination in convolution surfaces. In *Computer Graphics Forum* (1997), vol. 16, pp. 31–42.

[BS91] Bloomenthal J., Shoemake K.: Convolution surfaces. *Computer Graphics 25*, 4 (1991), 251–256.

[FFKW02] Felkel P., Fuhrman A.-L., Kanitsar A., Wegenkittl R.: Surface reconstruction of the branching vessels for augmented reality aided surgery. *BIOSIGNAL 2002 16* (June 2002), 252–254.

[GKS*93] Gerig G., Koller T., Szkely G., Brechbühler C., Kübler O.: Symbolic description of 3d structures applied to cerebral vessel tree obtained from mr angiography volume data. In *Proc. of Information Processing in Medical Imaging* (1993), vol. 687, Springer, LNCS, pp. 94–111.

[HAC03] Hornus S., Angelidis A., Cani M.-P.: Implicit modeling using subdivision curves. *The Visual Computer 19*, 2-3 (2003), 94–104.

[HPSP01] Hahn H., Preim B., Selle D., Peitgen H.-O.: Visualization and interaction techniques for the exploration of vascular structures. *IEEE Visualization* (2001), 395–402.

[JTFP01] Jin X., Tai C.-L., Feng J., Peng Q.: Convolution surfaces for line skeletons with polynomial weight distributions. *Journal of Graphics Tools 6*, 3 (2001), 17–28.

[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), vol. 1131, Springer, LNCS, pp. 161–171.

[NHK*85] Nishimura H., Hirai M., Kawai T., Kawata T., Shirakawa I., Omura K.: Object modelling by distribution function and a method of image generation. *The Transactions of the Institute of Electronics and Communication Engineers of Japan J68-D*, 4 (1985), 718–725.

[OM93] Opalach A., Maddock S. C.: Implicit surfaces: Appearance, blending and consistency. In *Proc. of Eurographics Workshop on Animation and Simulation* (1993), pp. 233–245.

[OM95] Opalach A., Maddock S. C.: An overview of implicit surfaces. In *Introduction to Modelling and Animation Using Implicit Surfaces* (1995), pp. 1.1–1.13.

[PTN97] Puig A., Tost D., Navazo I.: An interactive cerebral blood vessel exploration system. In *IEEE Visualization* (1997), IEEE Computer Society, pp. 443–446.

[She98] Sherstyuk A.: *Convolution Surfaces in Computer Graphics*. PhD thesis, Monash University, Melbourne, Victoria, Australia, 1998.

[SPSP02] Selle D., Preim B., Schenk A., Peitgen H.-O.: Analysis of vasculature for liver surgical planning. *IEEE Transactions on Medical Imaging 21*, 11 (Nov. 2002), 1344–1357.

[WMW86a] Wyvill G., McPheeters C., Wyvill B.: Animating soft objects. *The Visual Computer 2*, 4 (Aug. 1986), 235–242.

[WMW86b] Wyvill G., McPheeters C., Wyvill B.: Data structure for soft objects. *The Visual Computer 2*, 4 (Aug. 1986), 227–234.

[ZDMP94] Zijdenbos A., Dawant B., Margolin R., Palmer A.: Morphometric analysis of white matter lesions in mr images: Method and validation. *IEEE Transactions on Medical Imaging 13*, 4 (1994), 716–724.