FULLY-AUTOMATIC IMPROVEMENT OF THE GEOMETRY OF A VESSEL GRAPH

J. Bruijns, F. J. Peters, R. P. M. Berretty and B. Barenbrug
Philips Research, High Tech Campus 36, 5656 AE, Eindhoven, The Netherlands

Keywords: Medical Image Analysis, 3D Rotational Angiography, Computer Assisted Diagnosis, Vessel Analysis.

Abstract: Volume representations of blood vessels acquired by 3D rotational angiography are very suitable for diagnosing a stenosis or an aneurysm. For optimal treatment, physicians need to know the shape parameters of the diseased vessel parts. Therefore, we developed a method for semi-automatic extraction of these parameters from a surface model of the vessel boundaries. To facilitate fully-automatic shape extraction along the vessels, we developed a method to generate a vessel graph. This vessel graph represents the topology faithfully. However, the nodes and the branches are not always located close to the center lines of the vessels. Nodes and branches outside the center region decrease the accuracy of the extracted shape parameters. In this paper we present a method to improve the geometry of a vessel graph.

1 INTRODUCTION

Volume representations of blood vessels acquired by 3D rotational angiography after injection with a contrast agent (Moret et al., 1998) have a clear distinction in gray values between tissue and vessel voxels. Therefore, these volume representations are very suitable for diagnosing a stenosis, a local narrowing of a vessel caused for example by cholesterol, or an aneurysm, a local widening of a vessel caused by a weak vessel wall.

For optimal treatment of a stenosis or an aneurysm, physicians need to know the cross-sectional shape parameters in the neighborhood of the diseased vessel parts. We developed a method for semi-automatic extraction of these parameters from a surface model of the vessel boundaries (Bruijns, 2000). However, if two vessel branches are close together, it is possible that vertices of the neighbor vessel branch are included in the set of selected vertices which are used to estimate the local shape parameters of the vessel branch investigated. To exclude surface vertices of neighbor vessel branches, we developed a method for fully-automatic branch labelling (Bruijns, 2001) to give the vessel voxels (and from these the surface vertices) a unique label per vessel branch.

This method results also in a set of directed graphs (one for each component of the voxel vessel structures) with nodes (one for each vessel bifurcation and one for each vessel extremity) and edges (one for each vessel branch). An edge (called *"skeleton branch"*) consists of a set of face connected vessel voxels, called *"skeleton voxels"*, between the begin and end node of the skeleton branch. The first skeleton voxel is located at the begin node. The last skeleton voxel at the end node. Each node and each skeleton voxel has a radius derived from the local cross-section. Each skeleton voxel has also a plane. The plane position is initialized to the position of the skeleton voxel. These initial plane positions form a staircase curve (one for each skeleton branch). Since the plane normals are derived from the plane positions, each curve with plane positions is smoothed by a constrained relaxation algorithm (van Overveld, 1995). The resulting curve of plane positions is a piecewise cubic curve. The plane normal is equal to the normalized direction from the plane position of the lower neighbor to the plane position of the higher neighbor of the skeleton voxel.

The generated graphs facilitate fully-automatic vessel tracing along the skeleton branches of the same graph. After the user has selected a begin and end po-

Bruijns J., J. Peters F., P. M. Berretty R. and Barenbrug B. (2007).
FULLY-AUTOMATIC IMPROVEMENT OF THE GEOMETRY OF A VESSEL GRAPH.
In *Proceedings of the Second International Conference on Computer Vision Theory and Applications - IFP/IA*, pages 243-252

sition, a minimum path between the two corresponding skeleton voxels is generated. Next, for each skeleton voxel of this minimum path the shape parameters of the local cross-section (e.g. an orthogonal plane with an ellipse) are computed (Bruijns, 2000; Bruijns, 2001), using the plane position and the plane normal of the skeleton voxel as initial estimate for the ellipse center respectively the normal of the orthogonal plane.

The generated graphs represent the topology of the vessels faithfully. However, the nodes and the skeleton voxels are not always located close to the center lines of the vessels. A very clear example of an incorrectly located node is shown in Figure 9 in Section 6, namely the node with the square around the wire frame sphere. Nodes and skeleton voxels outside the center region of the vessel not only decrease the accuracy of the extracted shape parameters but also hamper the optimal functioning of other algorithms ((Bruijns et al., 2005b; Bruijns et al., 2005a)). In this paper we present a method to detect such incorrectly located nodes and skeleton voxels and to move them to the center lines as close as possible.

We use the Manhattan distance transform (Borgefors, 1984) (faster to compute than the Euclidian distance transform) with regard to the vessel boundaries to improve the geometry of a vessel graph. We call this distance transform the primary distance transform, abbreviated to PDT. We use the PDT because for each orthogonal cross-section the PDT is greater for vessel voxels closer to the vessel center.

In summary our method is as follows. First, the location of the nodes is improved (see Section 3). Next, the location of the skeleton voxels is checked (see Section 4). After all, only if the skeleton voxels of a branch are not close enough to the center line, the location of the skeleton branch can be improved. Finally, the location of the skeleton branches is improved (see Section 5). The branches are processed after the nodes because even if the original skeleton voxels of a branch are perfectly located, new skeleton voxels for a branch have to be generated if one or both nodes of this branch are moved. The results for 53 clinical volume datasets are presented in Section 6.

Remark:

The displaced skeleton voxels should be closer to (but do not represent) the center lines of the vessels! The new plane positions and the new plane normals of the skeleton voxels, used as initial estimates, should result in more accurate shape parameters so that the positions of the estimated ellipses represent the center lines better but that is not yet investigated.

2 RELATED WORK

The graph structure which represents the topology of the vessels, can be generated by various skeletonization algorithms. Skeletonization algorithms based on topological thinning are presented in (Bertrand and Malandain, 1994; Dokldal et al., 1999; Dokldal, 2000; Marion-Poty and Miguet, 1994). Skeletonization algorithms based on morphological thinning are presented in (Eiho and Qian, 1997; Maglaveras et al., 2001). Skeletonization algorithms based on distance transformations are described in (Chen et al., 2000; Bitter et al., 2001). Skeletonization algorithms based on thinning and distance maps are presented in (Selle et al., 2002; Boskamp et al., 2004). Skeletonization algorithms based on propagation are described in (Langs et al., 2004; Quek and Kirbas, 2001; Yim et al., 2000; Zahlten et al., 1994; Zahlten et al., 1995a; Zahlten et al., 1995b). A skeletonization algorithm based on path tracking is presented in (Kanitsar et al., 2001; Kanitsar et al., 2002; Kanitsar et al., 2003). A skeletonization algorithm based on ridge extraction is described in (Lobregt et al., 1980).

Algorithms for correction of the generated graph structure are presented in (Yim et al., 2000; Selle et al., 2002; Boskamp et al., 2004):

Yim et al. described a method for gray-scale skeletonization of small vessels in MRA. First, an acyclic connectivity graph is generated with the voxels as nodes and some of the pairs of neighbor voxels as edges. Starting at one or more seed voxels, the region and the boundary of the graph grows by including the neighbor voxels of the boundary voxel with the highest intensity. By storing also the parent voxel of the included neighbor voxels, the path from an endpoint voxel back to a seed voxel can be traced and highlighted. A more clearly visible skeleton can be created by removing short side branches. Erroneous junctions and occlusions are repaired using local shape properties.

Selle et al. reported a system for the analysis of the vasculature for liver surgical planning. First, the vessels are segmented. Next, the skeleton of the segmented vessels is computed and a graph of the branching structure augmented with local shape parameters (e.g. vessel radius) is generated. Using the shape information the graph is split in sub graphs for the different vessel systems. Finally, the liver is segmented and the vascular territories are computed.

Boskamp et al. described a interactive tool for morphometric quantification and visualization of vessels in CT and MR volumes. First, the vessels are segmented with user controlled thresholds. Next, a watershed transform of the gray value image is used

to create separate regions for connected bright structures (for example bones touching vessels). The user selects the vessel regions by placing markers in these regions. After that, the vessel skeleton is computed by thinning based on the distance transform. The skeleton is refined (i.e. small side branches removed) using the gradient of the distance transform. Next, cross-sectional MPR (mutliplanar reformation) images and a stretched MPR image of the vessel between two user selected vessel points are presented. A watershed transform of the gradient of a cross-sectional MPR image and the masks compute during vessel segmentation are used for a more accurate vessel segmentation of the 2D image. Finally, the cross-sectional area and diameters are extracted from each segmented cross-sectional MPR image.

3 IMPROVE LOCATION OF THE NODES

To prevent that extremity nodes (i.e. a node at the end of a vessel part) and nodes at the neck of an aneurysm, which have at most two branches, are moved, only nodes with more than two skeleton branches (i.e. the nodes at a bifurcation) are moved.

3.1 First Displacement

The location of the nodes is improved in two steps. First, the nodes with more than two skeleton branches are moved closer to the center of the bifurcation along a growing path of face connected vessel voxels. Two voxels v_1 and v_2 are face connected if and only if

$$\begin{aligned} &|ix(v_1) - ix(v_2)| + \\ &|iy(v_1) - iy(v_2)| + \\ &|iz(v_1) - iz(v_2)| = 1 \end{aligned} \tag{1}$$

with $ix(v_i)$ the first index, $iy(v_i)$ the second index and $iz(v_i)$ the third index of voxel v_i.

Given the current last voxel v_i of this growing path of face connected vessel voxels for node n_k, a face neighbor vessel voxel v_{i+1} of this last voxel is selected for movement of this node if this face neighbor vessel voxel fulfills the following conditions:

1. The PDT of the face neighbor vessel voxel v_{i+1} is greater than the PDT of the current last vessel voxel v_i.
2. The distance between the original position of node n_k and the face neighbor vessel voxel v_{i+1} is less than or equal to the original radius of node n_k.

   This condition prevents that the node moves away from the original bifurcation (for example in case of a tapering vessel as shown in Figure 1).

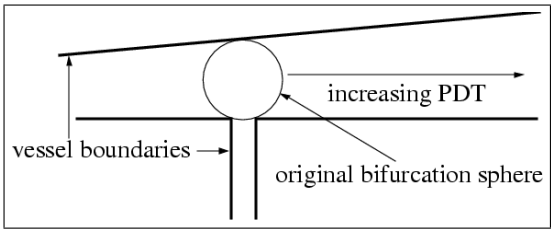

Figure 1: Bifurcation in a tapering vessel.

3. The distance between the face neighbor vessel voxel v_{i+1} and the position of all other nodes is greater than or equal to two times the size of a voxel.

   This condition prevents that the moved node coincides with one of these other nodes (for example in case of a number of closely spaced side branches).

If there are more face neighbor vessel voxels of the current last voxel v_i which fulfill these conditions, we select the face neighbor vessel voxel v_{i+1} closest to the original node position:

$$\| p(v_{i+1}) - p(v_0) \| \leq \| p(v_k) - p(v_0) \| \tag{2}$$

with v_k any other face neighbor vessel voxel of v_i fulfilling the three afore mentioned conditions.

If there are still more face neighbor vessel voxels which are now equally suitable, we select the first one discovered during testing (the order of the six face neighbors is implicitly defined in the software).

3.2 Second Displacement

If the original node is located at the beginning of a small side branch of a continuous main branch, the node may be still far away from the center line of the main branch. A schematic example is shown in Figure 2.

If in this case the original node is located in the small side branch below the lowest PDT 2 voxel, the first displacement algorithm (see Section 3.1) will move the node to this lowest PDT 2 voxel. A better, because closer to the center line of the continuous main branch, PDT 2 voxel for the node is given by the center PDT 2 voxel of the horizontal line of the T shape formed by the PDT 2 voxels. Next we use the observation that this voxel is also closest to the center position of the PDT 2 voxels face connected to the lowest PDT 2 voxel.

0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	2	2	1	1	1	1
0	0	0	0	1	**2**	1	0	0	0	0
				0	1	0				
				0	1	0				
				0	1	0				
				0	1	0				

Figure 2: Small side branch. Empty squares are tissue voxels. Squares with a PDT 0 are boundary vessel voxels. Squares with a PDT 1 are face neighbor vessel voxels of the boundary vessel voxels. And so on. The square with the bold digit 2 is the final voxel of the first displacement algorithm.

In general, if the bifurcation contains a cluster of maximum PDT voxels, a better voxel for the node is given by the maximum PDT voxel closest to the center of this cluster. So, after all nodes are possibly moved by the first displacement algorithm (as described in Section 3.1), the following algorithm is applied to find a better voxel for the nodes with more than two skeleton branches:

1. Compute the center position of the cluster of maximum PDT voxels:

$$\mathbf{c} = \frac{1}{N}\sum_{i=1}^{N} p(v_i) \tag{3}$$

   A vessel voxel v_i is member of this cluster if and only if:

   (a) The vessel voxel v_i has a PDT greater than or equal to the PDT of the vessel voxel v_k of the moved node n_k.

   (b) The distance between the vessel voxel v_i and the position of the moved node n_k is less than or equal to the original radius of the node.

2. Select the vessel voxel v_c closest to this center position **c**:

$$\| p(v_c) - \mathbf{c} \| \leq \| p(v_i) - \mathbf{c} \| \tag{4}$$

   with v_i any other vessel voxel.

If the distance between the selected vessel voxel v_c and the positions of all other nodes is greater than or equal to two times the size of a voxel, we move the node n_k to the selected vessel voxel v_c.

3.2.1 Replacement of a Lesser Center Voxel

In case of a local cavity at the bifurcation, the PDT of the selected center voxel v_c may be less than the PDT of the vessel voxel v_k of the moved node n_k. An example is shown in Figure 3 (the bold digit 1 in the center column of the third row is the selected center voxel).

0	0	0		0	0	0
1	1	1	0	1	1	1
2	2	2	**1**	2	2	2
1	1	1	2	1	1	1
0	0	0	1	0	0	0
			0			
			0			
			0			

Figure 3: Bifurcation with a local cavity. Empty squares are tissue voxels. Squares with a PDT 0 are boundary vessel voxels. Squares with a PDT 1 are face neighbor vessel voxels of the boundary vessel voxels. And so on. The square with the bold digit 1 is the lesser center voxel.

In such a case, we select the vessel voxel v_n closest to this center voxel v_c which fulfills the following conditions:

1. The vessel voxel v_n has a PDT greater than or equal to the PDT of the vessel voxel v_k of the moved node n_k.
2. The distance between the vessel voxel v_n and the position of the moved node n_k is less than or equal to the original radius of the node.

If there are more vessel voxels fulfilling these conditions, we select the vessel voxel closest to the position of the moved node:

$$\| p(v_n) - p(n_k) \| \leq \| p(v_i) - p(n_k) \| \tag{5}$$

with v_i any other vessel voxel fulfilling the two afore mentioned conditions.

If there are still more vessel voxels which are now equally suitable, we select the first one discovered during testing (the order is implicitly defined in the software).

4 CHECK LOCATION OF THE SKELETON VOXELS

Nodes are not connected to other nodes. Therefore, if one of the face neighbor vessel voxels of a node has a higher PDT, the node can be moved in principle closer to the center line. So, we have a very simple test to identify “incorrectly” located nodes.

Such a simple test is not possible for “incorrectly” located skeleton voxels. First, in case of a vessel with

a changing radius two face connected skeleton voxels located on the center line can have different PDT values (for example in case of a tapering vessel as shown in Figure 1). So, we do not compare skeleton voxels with face neighbor skeleton voxels. Secondly, the skeleton voxels form a face connected set of vessel voxels in 3D voxel space.

Taking the complexity of the 3D configuration of the vessel voxels into account, a skeleton voxel is marked as "incorrectly" located if the following conditions are fulfilled:

1. The skeleton voxel itself has at least one face neighbor vessel voxel (not a skeleton voxel) with a higher PDT.

   This kind of face neighbor vessel voxels are collected in set S1.

2. The next two skeleton voxels of the skeleton branch have each at least one face neighbor vessel voxel (not a skeleton voxel) with a higher PDT.

   This kind of face neighbor vessel voxels are collected in set S2 respectively set S3.

3. Three different face neighbor vessel voxels (one of set S1, one of set S2 and one of set S3) should form a face connected path.

   Note that a face neighbor vessel voxel with a higher PDT can be member of more than one set!

A simple example of an "incorrectly" located skeleton voxel is shown in the 2D configuration of Figure 4 with the skeleton voxels of the skeleton branch given by the bold digits 1 and the direction of the skeleton branch going from the bottom row to the top row. In this case, the skeleton voxel at the bottom row will be counted as an "incorrectly" located skeleton voxel.

	0	1	2	**1**	0	
	0	1	2	**1**	0	
	0	1	2	**1**	0	

Figure 4: "Incorrectly" located skeleton voxel. Empty squares are tissue voxels. Squares with a PDT 0 are boundary vessel voxels. Squares with a PDT 1 are face neighbor vessel voxels of the boundary vessel voxels. And so on. The squares with the bold digit 1 are the skeleton voxels.

Remark:

Since the begin node of a self loop branch is also the end node of this branch, and since our method to improve the location of a skeleton branch requires that the begin and end node are separated, the locations of the skeleton voxels of a self loop branch cannot be improved. Therefore, a self loop branch is skipped during this test. The special adjustment of a self loop branch for the possibly moved node is described in Section 5.3.

5 IMPROVE LOCATION OF THE SKELETON BRANCHES

If one or both nodes of a skeleton branch are moved (see Section 3), or if at least one skeleton voxel of this branch is "incorrectly" located (see Section 4), the location of the skeleton branches must be improved. Instead of moving the current skeleton branch (i.e. replacing skeleton voxels by neighbor vessel voxels) and adjusting the begin and/or end of this branch for a moved begin and/or end node (i.e. removing and/or adding skeleton voxels at the extremities of the branch), a new skeleton branch of face connected vessel voxels between the begin and end node is generated to replace the old skeleton branch.

Starting with a new skeleton voxel at the possibly moved begin node, a new skeleton branch is generated by a growing algorithm: given the current last skeleton voxel, the new skeleton branch is extended by adding the "best neighbor" vessel voxel (given further detail in Section 5.2) of the current last skeleton voxel. This "best neighbor" vessel voxel should be "closer" to the possibly moved end node and as close as possible to the center line (i.e. select a vessel voxel from the set of the neighbor vessel voxels with the highest PDT).

If the vessel branch between the begin and end node is approximately a rectilinear tube, the Euclidean distance can be used to select a vessel voxel "closer" the end node. But if the vessel branch is curved, "closer" means that the new skeleton branch should form an onwards path inside the curving vessel branch between the begin and end node. To guarantee that the skeleton voxels of the new skeleton branch form an onwards path, a guide moving along the old skeleton branch is used to indicate in which direction the new skeleton voxel should be selected. A snapshot of such a growing process is shown in Figure 5.

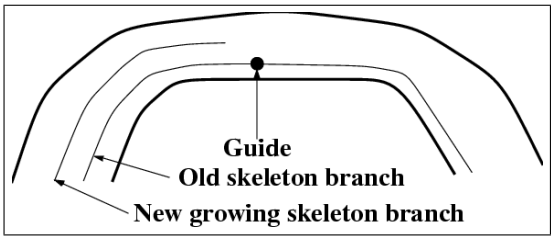

Figure 5: A snapshot of a growing process.

5.1 Moving the Guide

The guide can be followed only if the guide remains in front of the growing new skeleton branch, and if the guide is not hidden behind a vessel bend. Therefore, before a new skeleton voxel for the new skeleton branch is selected, the guide, starting at the first skeleton voxel of the old skeleton branch, is moved from its current position along the skeleton voxels of the old skeleton branch as long as one of the following conditions is fulfilled:

1. The old skeleton voxel is also member of the new growing skeleton branch.
2. The old skeleton voxel is located behind the plane of the new skeleton voxel.

   If a vessel is very small, the new growing skeleton branch, possibly following a piece of the old skeleton branch, may catch up with the moving guide. Without these first two conditions this situation may lead to a premature end or an infinite new skeleton branch generation.
3. The old skeleton voxel is located inside the sphere of the new skeleton voxel or the new skeleton voxel is located inside the sphere of the old skeleton voxel.

   As already mentioned, each skeleton voxel has a radius derived from the local cross-section (i.e. from the Manhattan distance to the closest vessel boundary). The radius defines together with the position of the skeleton voxel a sphere around each skeleton voxel.

   Since the guide stops moving along the skeleton voxels of the old skeleton branch when this condition is violated, the guide is never too far away from the new skeleton voxel as illustrated in Figure 6.

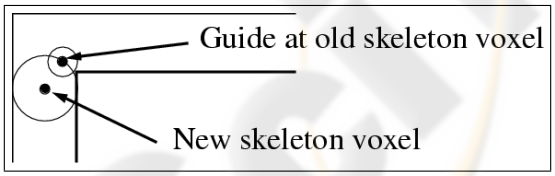

Figure 6: Guide at a vessel bend (right angle for easy sketching).

When the guide has reached the last skeleton voxel of the old skeleton branch, or when the new skeleton voxel is located inside the sphere of the possibly moved end node, the guide is set at the position of this end node.

5.2 Selecting the "Best Neighbor"

Given the current position of the guide and the current skeleton voxel of the new skeleton branch, it is tempting to extend the new skeleton branch by selecting the "best face neighbor" of the current skeleton voxel to guarantee that the new skeleton branch consists of face connected vessel voxels. But, because the "best face neighbor" should be as close to the guide as the current skeleton voxel to guarantee an onwards path, the new skeleton branch is not always located on the vessel voxels with the highest PDT value. Suppose we have the 2D configuration shown in Figure 4 with the skeleton voxels of the old skeleton branch given by the bold digits 1, the current skeleton voxel of the new skeleton branch given by the bold digit 1 in the bottom row and the guide given by the bold digit 1 in the top row.

In this case, the face neighbor given by the bold digit 1 in the middle row will be selected as the next skeleton voxel of the new skeleton branch because the face neighbor given by the digit 2 in the bottom row is farther away from the guide than the current skeleton voxel. So, the new skeleton branch will be located at the same position as the old skeleton branch instead of closer to the center line of the vessel.

Therefore the new skeleton branch is extended by selecting the "best corner neighbor" of the current skeleton voxel. Two voxels v_1 and v_2 are corner neighbors (see also Figure 7) if and only if

$$\begin{aligned} 1 \leq & |ix(v_1) - ix(v_2)| + \\ & |iy(v_1) - iy(v_2)| + \\ & |iz(v_1) - iz(v_2)| \leq 3 \end{aligned} \tag{6}$$

Since the new skeleton branch must consist of face connected vessel voxels, not only the "best corner neighbor" but also zero, one or two intermediate vessel voxels have to be selected so that the "best corner neighbor" is connected via a face connected path of vessel voxels to the current skeleton voxel.

Such an intermediate face path depends on the type of the "best corner neighbor": the 26 corner neighbors of the current skeleton voxel can be subdivided into 6 face neighbors (see Equation 1), 12 edge neighbors and 8 vertex neighbors. Two voxels v_1 and v_2 are edge neighbors if and only if

$$\begin{aligned} & |ix(v_1) - ix(v_2)| + \\ & |iy(v_1) - iy(v_2)| + \\ & |iz(v_1) - iz(v_2)| = 2 \end{aligned} \tag{7}$$

Two voxels v_1 and v_2 are vertex neighbors if and only if

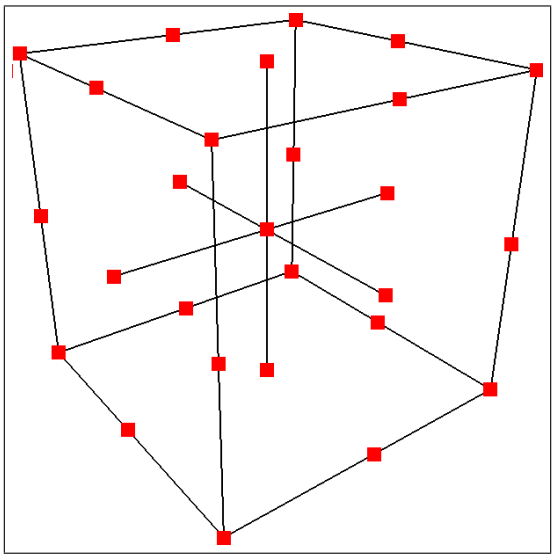

Figure 7: The corner neighbors. The current skeleton voxel is the square in the center of the cube. Its face neighbors are the squares at the centers of the cube faces. Its edge neighbors are the squares at the centers of the cube edges. Its vertex neighbors are the squares at the cube corners.

$$\begin{aligned} &|ix(v_1) - ix(v_2)| + \\ &|iy(v_1) - iy(v_2)| + \\ &|iz(v_1) - iz(v_2)| = 3 \end{aligned} \tag{8}$$

For face neighbors of the current skeleton voxel no intermediate voxels are needed. So, their intermediate face paths are empty.

For edge neighbors of the current skeleton voxel one intermediate voxel is needed. Each edge neighbor can be face connected via two different intermediate voxels with the current skeleton voxel. So each edge neighbor has two different intermediate face paths. Each intermediate face path consists of one voxel, face connected both to the edge neighbor and the current skeleton voxel.

For vertex neighbors of the current skeleton voxel two intermediate voxels are needed. Each vertex neighbor can be face connected to three different edge neighbors. So each vertex neighbor has six different intermediate face paths. Each intermediate face path consists of two face connected voxels. The first voxel of this intermediate face path is face connected to the vertex neighbor, the second voxel to the current skeleton voxel.

So, the total number of combinations (i.e. a corner neighbor and a possibly empty intermediate face path) is 6 + 12 * 2 + 8 * 6 = 78. The best combination of the set S_0 of all combinations is selected as follows:

First the valid combinations are selected (and thus the invalid combinations eliminated):

1. The set S_1 are those combinations of the set S_0 for which the distance between the guide g and the corner neighbor c_k is less than or equal to the distance between this guide and the current skeleton voxel v_i:

$$\| p(g) - p(c_k) \| \leq \| p(g) - p(v_i) \| \tag{9}$$

2. The set S_2 are those combinations of the set S_1 for which the corner neighbor and the intermediate voxels are inside the volume.
3. The set S_3 are those combinations of the set S_2 for which the corner neighbor and the intermediate voxels are neither a tissue voxel nor already a member of the new skeleton branch.

Next, the better combinations are selected (and thus the lesser combinations eliminated):

1. The set S_4 are those combinations of the set S_3 for which the PDT of the corner neighbor c_k is equal to the maximum PDT of the corner neighbors of the set S_3:

$$\begin{aligned} &PDT(c_k) = \\ &max(PDT(c_l) \mid c_l \in S_3) \end{aligned} \tag{10}$$

2. The set S_5 are those combinations of the set S_4 for which the distance between the guide and the corner neighbor c_k is equal to the shortest distance between the guide and the corner neighbors of the set S_4:

$$\begin{aligned} &\| p(g) - p(c_k) \| = \\ &min(\| p(g) - p(c_l) \| \mid c_l \in S_4) \end{aligned} \tag{11}$$

3. The set S_6 are those combinations of the set S_5 for which the number of intermediate voxels $nv(ifp_i)$ (e.g. 0, 1 or 2) of the intermediate face path ifp_i is equal to the minimum number of intermediate voxels of the intermediate face paths in the set S_5:

$$\begin{aligned} &nv(ifp_i) = \\ &min(nv(ifp_l) \mid ifp_l \in S_5) \end{aligned} \tag{12}$$

So, we prefer face neighbors. If there are no face neighbors anymore, we prefer edge neighbors. After this step we have either only face neighbors or only edge neighbors or only vertex neighbors.

4. If we have face neighbors, the set S_7 is a copy of the set S_6. Else, the set S_7 are those combinations of the set S_6 for which the sum $sp(ifp_i)$ of the PDT values of the intermediate voxels of the intermediate face path ifp_i is equal to the maximum sum of the PDT values of the intermediate voxels of the intermediate face paths in the set S_6:

$$sp(ifp_i) = max(sp(ifp_l) \mid ifp_l \in S_6) \tag{13}$$

with

$$sp(ifp_i) = \sum_{v_j \in ifp_i} PDT(v_j) \tag{14}$$

Give the 2D configuration example at the beginning of this section, the "best corner neighbor" is the vessel voxel with PDT 2 in the middle row and the best intermediate face path consists of the face neighbor vessel voxel with PDT 2 in the bottom row, not the face neighbor vessel voxel with PDT 1 in the middle row.

5. The set S_8 are those combinations of the set S_7 for which the corner neighbor c_k has a "best face neighbor". A best face neighbor f_b is a face neighbor (see equation 1) with the greatest PDT:

$$PDT(f_b) \geq PDT(f_i) \tag{15}$$

and in case of equal PDT values the shortest distance to the guide g:

$$\| p(g) - p(f_b) \| \leq \| p(g) - p(f_i) \| \tag{16}$$

with both f_b and f_i face neighbors of possibly different corner neighbors of the combinations of the set S_7.

If there are still more corner neighbors which are now equally suitable, we select the first one discovered during testing (the order of the corner neighbors is implicitly defined in the software).

If the "best corner neighbor" is an edge neighbor or a vertex neighbor, there may be still more than one "best intermediate face path" for the "best corner neighbor". In that case, we select the first one discovered during testing (the order of the intermediate face paths is implicitly defined in the software).

After the "best corner neighbor" with the "best intermediate face path" is selected, the intermediate voxels and the corner neighbor are used to extend the new skeleton branch. If one of these new skeleton voxels is located at the moved end node, the new skeleton branch is complete.

Remark:

Only the distance between the guide and the "best corner neighbor" needs to be less than or equal to the distance between the guide and the current skeleton voxel. The distance between the guide and an intermediate voxel may be greater than the distance between the guide and the current skeleton voxel!

5.3 A Self Loop

Since the begin node of a self loop branch is also the end node of this branch, a self loop branch is treated differently. If the node of the self loop branch is moved, the new skeleton branch of the self loop (see Figure 8) consists of a new skeleton branch part from the moved node to the begin skeleton voxel of the old skeleton branch, followed by a copy of the old skeleton branch, followed by a new skeleton branch part from the end skeleton voxel of the old skeleton branch to the moved node. Such a new skeleton branch part consists of the set of face connected vessel voxels which form an approximate straight-lined connection between the begin and end of this branch part.

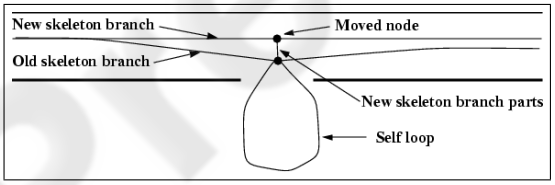

Figure 8: Moved self loop.

6 RESULTS AND CONCLUSIONS

We have applied the method to improve the geometry of a vessel graph to 53 clinical volume datasets (14 of them with a resolution of 256x256x256 voxels, the rest 128x128x128 voxels), acquired with the 3D Integris system (Philips-Medical-Systems-Nederland, 2001). The voxel size varies between 0.2 and 1.2 millimeter.

The effect of the method can be perceived by comparing Figure 9 with Figure 10.

The total number of nodes and skeleton voxels are given in the first column of Table 1. The relative number of "incorrectly" located nodes and skeleton voxels (see Section 4) before and after the method is applied, are given in the second respectively third column of Table 1.

The remaining "incorrectly" located nodes are due to nodes which cannot be moved to a better location because of nearby nodes. The remaining "in-

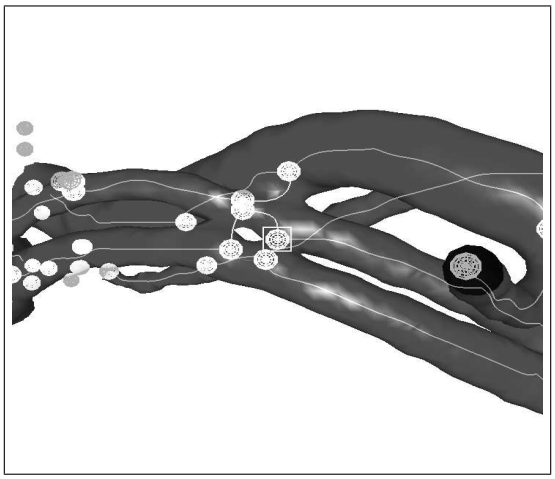

Figure 9: An original graph drawn on the vessel surface. The wire-frame spheres are the nodes. The polylines between the nodes are the piecewise cubic curves derived from the skeleton voxels. The square indicates the incorrectly located node.

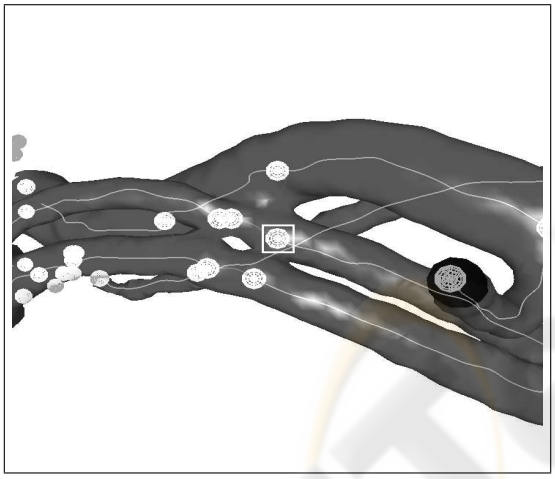

Figure 10: The improved graph drawn on the vessel surface. The square indicates the "incorrectly located" node at its new position.

correctly" located skeleton voxels are due to skeleton branches through very small and very short secondary vessel parts as shown in Figure 11. The skeleton voxels of this branch which are located in the main vessel part, have face neighbor vessel voxels with a higher PDT.

The relative number of face neighbors used as "best corner neighbor" is 37%. The relative number of edge neighbors is 47% and the relative number of vertex neighbors is 16%. So, trying to find the "best corner neighbor" instead of only the "best face neighbor" is indeed a useful approach.

Table 1: The number of nodes and skeleton voxels.

	Total	Before	After
Nodes	7267	19.2%	6.1%
Skeleton voxels	184318	1.1%	0.1%

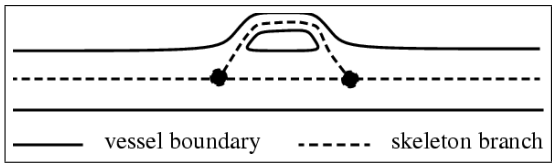

Figure 11: A secondary vessel part.

The time to improve the geometry of a vessel graph is a small fraction ($\leq$ 1 %) of the time for fully-automatic branch labelling. The average elapsed time for fully-automatic branch labelling of an 128x128x128 volume is 1.2 seconds on a Linux PC (2.8GHz Pentium 4). The average elapsed time for an 256x256x256 volume is 15.7 seconds on the Linux PC.

Concluding, our method to improve the geometry of a vessel graph makes fully-automatic extraction of the cross-sectional shape parameters in the neighborhood of the diseased vessel parts more reliable and the extracted shape parameters more accurate but a clinical validation has yet to be done.

REFERENCES

Bertrand, G. and Malandain, G. (1994). A new characterization of three-dimensional simple points. *Pattern Recognition Letters*, 2(15):169–175.

Bitter, I., Kaufman, A. E., and Sato, M. (2001). Penalized-distance volumetric skeleton algorithm. *IEEE Trans. Visual. Comput. Graphics*, 7(3):195–206.

Borgefors, G. (1984). Distance transformations in arbitrary dimensions. *Computer Vision, Graphics and Image Processing*, 27(3):321–345.

Boskamp, T., Rinck, D., Link, F., Kuemmerlen, B., Stamm, G., and Mildenberger, P. (2004). New vessel analysis tool for morphometric quantification and visualization of vessels in ct and mr imaging data sets. *Radiographics*, 24(1):287–297.

Bruijns, J. (2000). Semi-automatic shape extraction from tube-like geometry. In *Proc. VMV*, pages 347–355, Saarbruecken, Germany.

Bruijns, J. (2001). Fully-automatic branch labelling of voxel vessel structures. In *Proc. VMV*, pages 341–350, Stuttgart, Germany.

Bruijns, J., Peters, F., Berretty, R., van Overveld, C., and ter Haar Romeny, B. (2005a). Computer-aided treatment planning of an aneurysm: The connection tube and the

neck outline. In *Proc. VMV*, pages 265–272, Erlangen, Germany.

Bruijns, J., Peters, F., Berretty, R., van Overveld, C., and ter Haar Romeny, B. (2005b). Fully-automatic computation of the shape of a micro-catheter. In *Proc. CAR*, pages 401–406, Berlin, Germany.

Chen, D., Li, B., Liang, Z., Wan, M., Kaufman, A., and Wax, M. (2000). A tree-branch searching, multiresolution approach to skeletonization for virtual endoscopy. In *Proc. SPIE: Medical Imaging, volume 3979*, pages 726–734, San Diego, CA, USA.

Dokldal, P. (2000). *Grey-Scale Image Segmentation: A Topological Approach*. PhD thesis, University Marne La Vallee, France.

Dokldal, P., Lohou, C., Perroton, L., and Bertrand, G. (1999). A new thinning algorithm and its application to extraction of blood vessels. In *Proc. BioMedSim*, pages 32–37, ESIEE Noisy-le_Grand, France.

Eiho, S. and Qian, Y. (1997). Detection of coronary artery tree using morphological operator. In *Proc. IEEE Computers in Cardiology, Vol. 24*, pages 525–528, Lund, Sweden.

Kanitsar, A., Fleischmann, D., Wegenkittl, R., Felkel, P., and Groeller, E. (2002). Cpr - curved planar reformation. In *Proc. IEEE Visualization*, pages 37–44, Boston, MA, USA.

Kanitsar, A., Fleischmann, D., Wegenkittl, R., Felkel, P., and Groeller, E. (2003). Advanced curved planar reformation: Flattening of vascular structures. In *Proc. IEEE Visualization*, pages 43–50, Seattle, WA, USA.

Kanitsar, A., Wegenkittl, R., Felkel, P., Fleischmann, D., Sandner, D., and Groeller, E. (2001). Computed tomography angiography: A case study of peripheral vessel investigation. In *Proc. IEEE Visualization*, pages 477–480, San Diego, CA, USA.

Langs, G., Radeva, P., Rotger, D., and Carreras, F. (2004). Explorative building of 3d vessel tree models. In *Proc. of 28th annual workshop of the Austrian Association for Pattern Recognition (OAGM/AAPR): Digital Imaging in Media and Education*, pages 117–124, Hagenberg, Austria.

Lobregt, S., Verbeek, P., and Groen, F. (1980). Three-dimensional skeletonization: Principle and algorithm. *IEEE Trans. Pattern Anal. Machine Intell.*, 2(1):75–77.

Maglaveras, N., Haris, K., Efstratiadis, S., Gourassas, J., and Louridas, G. (2001). Artery skeleton extraction using topographic and connected component labeling. In *Proc. IEEE Computers in Cardiology, Vol. 28*, pages 265–268, Rotterdam, The Netherlands.

Marion-Poty, V. and Miguet, S. (1994). A new 2-d and 3-d thinning algorithm based on successive border generations. In *Proc. 4th Conference on Discrete Geometry in Computer Imagery*, pages 195–206, Grenoble, France.

Moret, J., Kemkers, R., de Beek, J. O., Koppe, R., Klotz, E., and Grass, M. (1998). 3d rotational angiography: Clinical value in endovascular treatment. *Medicamundi*, 42(3):8–14.

Philips-Medical-Systems-Nederland (2001). Integris 3d-ra. instructions for use. release 2.2. Technical Report 9896 001 32943, Philips Medical Systems Nederland, Best, The Netherlands.

Quek, F. K. H. and Kirbas, C. (2001). Vessel extraction in medical images by wave-propagation and traceback. *IEEE Trans. Med. Imag.*, 20(2):117–131.

Selle, D., Preim, B., Schenk, A., and Peitgen, H. (2002). Analysis of vasculature for liver surgery planning. *IEEE Trans. Med. Imag.*, 21(11):1344–1357.

van Overveld, C. (1995). Pondering on discrete smooth interpolation. *Computer-aided Design*, 27(5):377–384.

Yim, P., Choyke, P., and Summers, R. (2000). Gray-scale skeletonization of small vessels in magnetic resonance angiography. *IEEE Trans. Med. Imag.*, 19(6):568–576.

Zahlten, C., Juergens, H., Evertsz, C., Leppek, R., Peitgen, H., and Klose, K. (1995a). Portal vein reconstruction based on topology. *Eur J Radiol*, 19(2):96–100.

Zahlten, C., Juergens, H., and Peitgen, H. (1994). Reconstruction of branching blood vessels from ct-data. In *Proc. Visualization in Scientific Computing*, pages 41–53, Rostock, Germany.

Zahlten, C., Juergens, H., and Peitgen, H. (1995b). *Reconstruction of Branching Blood Vessels From CT-Data*, pages 41–52. Springer-Verlag, Wien. In Goebel, M., Mueller, H., Urban, B. (Hrsg): Visualization in Scientific Computing.