Fully-Automatic Branch Labelling of Voxel Vessel Structures

J. Bruijns

Philips Research Laboratories
Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
Email: `Jan.Bruijns@philips.com`

Abstract

Today, it is possible to acquire volume representations of the vessel structures in the brain. The self-adjusting probe, a new tool introduced in a previous paper, enables semi-automatic shape extraction.

The self-adjusting probe extracts shapes from a surface model. Yet, if two vessel branches are close together, it is possible that surface vertices of a neighbor vessel are included in the set of vertices used to extract the local shape of the vessel investigated. These erroneously included vertices deteriorate the accuracy of the estimated shape.

Therefore, a method has been developed to give the vessel vertices a unique label per vessel branch. Now, surface vertices of neighbor vessel branches can be excluded because their label is different.

Key Words: 3D Rotational Angiography; volume visualization; Virtual Angioscopy; Computer Assisted Diagnosis; Shape Extraction.

1 Introduction

Nowadays, it is possible to acquire volume representations of the brain which show a clear distinction in gray values between tissue and vessel voxels (see [4] for an example). Shape extraction, such as measuring a vessel's cross-sectional area, is done by interactively positioning and orienting a plane. The intersection of this plane with the volume gives a 2D image of gray values in which the vessel pixels have a different gray value than the tissue pixels. After selection of the proper object the cross-sectional area can be measured, for example by counting the number of non-black pixels.

The plane should be oriented so that it is orthogonal to the vessel whose shape has to be measured. An oblique plane would give the wrong cross-sectional area. Unfortunately, interactively orienting the plane orthogonally to the vessel is a time-consuming and error-prone task.

In a previous paper a new tool, the self-adjusting probe, was introduced [2]. A probe is a combination of a sphere and a plane through the centre of the sphere. After the probe has interactively been placed on a vessel in the neighborhood of the desired position, the probe automatically adjusts itself so that its plane is orthogonal to the vessel and the centre of its sphere is on the central axis of the vessel (see Figure 6). The ellipse radii of the vessel are also estimated. When the probe is aligned, it can be moved along the vessel in the direction of the plane normal. The probe aligns itself again after each step. It is therefore possible to let the probe follow the vessel automatically until the probe detects for example the end of a vessel or the beginning of an aneurysm.

Currently, the self-adjusting probe extracts shapes from a 3D triangle surface model, created for example by a marching cube algorithm [5]. However, if two vessel branches are close together (as in Figure 7 where the lower part shows the vertices of the vessel slice currently investigated including the estimated inner and outer circles, while the upper part shows the vertices of a neighbor vessel), it is possible that surface vertices of the neighbor vessel branch are included in the set of selected surface vertices which are used to extract the local shape of the vessel branch investigated. These erroneously included surface vertices deteriorate the accuracy of the estimated shape.

To improve the accuracy, a method has been developed to give the vessel voxels (and from these, the surface vertices) a unique label per vessel branch. Now, vertices of neighbor vessel branches can be excluded because their label is different.

This new method results also in a set of directed graphs (one for each component) which facilitates fully-automatic vessel tracing from one node (an extremity or bifurcation of the vessel structure) to another node of the same graph.

VMV 2001 Stuttgart, Germany, November 21–23, 2001

We introduce the fully-automatic branch labelling method in Section 2. In Section 3 and following, this method is given further detail. In Section 8, we present our results and conclusions.

2 Outline

The fully-automatic branch labelling method consists of five steps. Starting point is a segmented voxel volume **without tissue inclusions** (see Figure 5). The elements of this segmented voxel volume are signed bytes, with a 0 for a tissue and a 1 for a vessel voxel. This makes it possible to assign different labels to vessel voxels during these steps. Vessel voxels with a label 1 are called original vessel voxels. Tissue voxels are never changed. The final outcome is a segmented voxel volume in which almost all vessel voxels have a label indicating to which bifurcation or branch they belong (see Figure 13) and a set of directed graphs describing the topology of the vessels (see Figure 11).

In the first step, a wave propagation method similar to that of Zahlten et al. [7] is applied to find the extremities of the voxel vessel structures (the darker spots in Figure 9).

In the second step, the segmented voxel volume with the extremities indicated by a special label is peeled in a number of iterations. The resulting skeleton of branches and bifurcations (see Figure 10) is a good approximation of the centre structure of the vessels.

In the third step, the graphs are created. Starting point is the peeled segmented voxel volume with the bifurcation and extremity voxels indicated by special labels. A directed graph is generated for each connected set of vessel voxels in the peeled segmented voxel volume. This graph contains one node for each extremity and one node for each bifurcation voxel. The vessel voxel strings between adjacent nodes are stored in branches. Each branch gets a unique number. The voxels of a branch get this number as label (see Figure 11).

In the fourth step, node geometry is generated. Node geometry contains the position and the shape of the centre region of the skeleton bifurcation and the position and the shape of its branch regions (see Figure 3, Figure 4 and Figure 12).

In the fifth step the vessel voxels get their final label. First, the voxels in the branch regions (see Figure 4) get the label of their branch. Next, the voxels in the centre regions get the label of the centre region. Finally, the voxels of the branches between two adjacent branch regions get the label of their branch (see Figure 13).

After the vessel voxels are labelled, a modified marching cube algorithm is applied which generates the surface vertices including the label of their neighbor vessel voxel.

3 Wave Propagation Dissected

In the first step the wave propagation method is applied. Zahlten et al. [7] use wave propagation to label the gray value voxel volume and to generate the corresponding vessel graph. As is clear from Figure 8 where the voxels are depicted by gray squares on the basis of their label, wave propagation is not accurate enough at the bifurcations. The incoming waves continue too long at the bifurcations 1 and 2.

We use wave propagation to generate the extremities of the vessel voxel structures. If the current wave is at the far end of a branch, one of the voxels of the current wave is selected for inclusion in the set of extremities. The labels assigned during the wave propagation step are ignored.

We generate the starting points automatically (see Section 3.4). Selection by the user is too time-consuming because our segmented voxel volumes contain a lot of vessel graphs. These starting points are included in the set of extremities.

3.1 Basic Algorithm

A wave, with a branch number greater than 1, is a list of voxel descriptions. A voxel description contains a label with the value the voxel had when the description was created (except for trial waves discussed in Section 3.5), the memory address of the voxel, giving fast access to the current value of the voxel and its index (ix, iy, iz).

Starting with a single seed voxel as initial wave, a new wave with the same branch number is created by generating voxel descriptions for all corner neighbor vessel voxels[1] of the current wave which were not yet member of any wave. Whether a voxel was already member of a wave can be deduced from the current value of the voxel in the segmented voxel volume because voxels which are included in a wave get the branch number of the current wave. Next, the current wave is deleted and the new wave becomes the current wave.

---

[1] A corner neighbor is a voxel which shares at least one corner with the voxel examined.

If the voxels of the current wave are not corner connected[2] , the wave is split in two or more new waves with each a new unique branch number. These new branch numbers are also assigned to the voxels of the split waves. Splitting will occur when a wave travels through a bifurcation of the vessel voxel structures. The wave propagation method is applied to each of the new wave fragments. The process is finished when all new waves are empty.

The generation of three waves is shown in Figure 3 of [7] (that paper contains an extensive description of the wave propagation method).

3.2 Double Waves

As already mentioned, wave propagation is used to find the extremities of the voxel structures. Extremities are found when the new wave is empty. However, in case of a noisy surface the wave will be split in many sub-waves just before the end of a branch is reached. This yields many unwanted extremities.

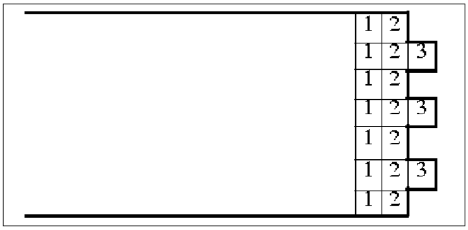

Figure 1: Splitting caused by bumpy surface

In the example of Figure 1 there are three waves shown. The first wave contains the voxels labelled 1, the second wave the voxels labelled 2 and the third wave the voxels labelled 3. The voxels of the third wave are not corner connected. So, this wave is split in three new waves. Because these three wave fragments do not have successors, three extremities are generated.

Double waves solve this problem. The waves described in the previous paragraphs are called single waves in the following discussion. A double wave contains old voxels, already present in the previous double wave, and new voxels, which are the corner neighbor vessel voxels of these old voxels. When from such a double wave a new double wave is created, the new voxels of the old double wave are copied to the new double wave as old voxels. Next, the corner neighbor vessel voxels of the old voxels of the new double wave are added as new voxels. The old voxels of a new double wave are also used to test whether the new voxels of this new double wave are corner connected. The new double wave is considered empty if it contains only old voxels.

In the example of Figure 1 there are three double waves. The first wave contains the voxels labelled 1 as old and the voxels labelled 2 as new voxels. The second wave contains the voxels labelled 2 as old and the voxels labelled 3 as new voxels. The third wave contains the voxels labelled 3 as old and no new voxels.

The new voxels of the second double wave are corner connected to each other via the old voxels of this wave. So, this second double wave is not split as was the case for the third single wave. The third double wave contains only old voxels. Hence, this wave is considered empty and the extremity is derived from all voxels of the second double wave, giving only one extremity for this branch end.

In the following sections, the waves for wave propagation (except the special waves[3]) are double waves.

3.3 Component Selection

Wave propagation is based on corner neighbors. But peeling, graph generation, generation of node geometry and final labelling are based on face neighbors[4]. Indeed, each voxel has 26 corner neighbors and 6 face neighbors. So, using face neighbors instead of corner neighbors saves a significant amount of computing time. Even more important, corner neighbors may yield much more unwanted connections between parallel neighbor vessels.

Therefore, before wave propagation is started for a new seed voxel, the original vessel voxels which are face connected, via vessel voxels, to this seed voxel, are labelled as component voxels. The seed voxel itself is also labelled as component voxel.

These face connected original vessel voxels are labelled by the following twin wave algorithm: Two empty waves are created. The first wave is filled with a voxel description of the seed voxel. The seed voxel is labelled as a component voxel. The second

---

[2] As explained in [7], splitting if the voxels are not face connected, generates too many bifurcations.

[3] Special waves are used to store small temporary subsets of the segmented voxel volume. Storing these subsets in a wave instead of an additional binary voxel volume saves main memory and computing time because scanning and checking a large binary voxel volume for a small subset takes much more time.

[4] A face neighbor is a voxel which shares a face with the voxel examined.

wave is filled with voxel descriptions of those original vessel voxels which are face neighbors of the voxels of the first wave. The corresponding voxels are labelled as component voxels. This process is repeated, changing each time the role of the first and second wave, until no original vessel voxels, face connected to the voxels of the current wave are found.

The normal wave propagation can now be applied on the component voxels. Vessel voxels which are corner but not face connected to a voxel of this component are now ignored because they do not have the component label.

3.4 Seed Voxels

Wave propagation yields the best results, if the waves travel from wider to narrower vessels. Because wave propagation is used to find the extremities of the vessel structures, seed voxels should be in the neighborhood of these extremities. Fortunately, very often the large dominant vessel structures start on the boundaries of the volume with their widest vessels. The centre voxels of the widest vessel can be found by means of the distance transform. The distance transform of a vessel voxel is an indicator for the length of the shortest path of face connected vessel voxels to a voxel of a vessel boundary.

It is possible that a local maximum in the distance transform is shared by a number of face neighbor vessel voxels in the same boundary volume slice. So, it is not safe too look for the vessel voxel with a distance transform greater than the distance transform of its face neighbors. Therefore, all voxels in the six boundary volume slices with a distance transform greater than or equal to the distance transform of its face neighbors are stored in a special wave, the so-called seed wave, by means of an insertion sort on the basis of their distance transforms. The insertion sort makes that the seed voxel with the highest distance transform is the first voxel description in the seed wave.

When a voxel is included in the seed wave, the distance transform of its face neighbor vessel voxels is decreased. This prevents that many vessel voxels with the same distance transform as their face neighbors are included in the seed wave.

Starting with (and removing) the first voxel description of the seed wave, the wave propagation algorithm including the component selection (see Section 3.3) is carried out. Many voxels of the seed wave will be altered because they belong to the same vessel component. So, skipping and removing the first voxel descriptions of the seed wave from which the corresponding voxels have already been changed, the wave propagation algorithm including the component selection is carried out for all first voxel descriptions of the seed wave from which the corresponding voxels are not yet processed, until the seed wave is empty.

3.5 Trial Waves

Many components do not start or end on a boundary of the volume. Scanning the segmented voxel volume for not yet processed voxels, will not always result in a seed voxel in the neighborhood of the extremities of a component. But, by sending trial waves from the start voxel found by scanning, the extremities will be found as shown in Figure 2. The extremities of the trial wave propagation pass are stored in the seed wave instead of in the set of extremities.

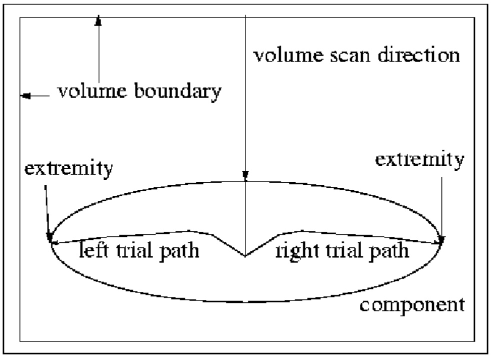

Figure 2: Trial waves

Waves do not only have a branch number but also a serial number. This serial number is increased every time a new wave is created from an old wave. The voxel descriptions of the extremity voxels of the trial waves get the serial number of the trial wave as label (not the current value of the extremity voxel as is normally the case). Because the seed wave is filled by means of an insertion sort, the extremity voxel of the trial wave with the highest serial number will be used as seed voxel for a normal wave propagation pass.

The trial waves get a special branch number, not used in case of a normal wave propagation pass. So, after the seed voxels are found, the voxels of the component can be reset to their original value.

The trial wave propagation pass is alternated with the normal wave propagation pass until all interior components have been processed.

4 Peeling Dissected

In the second step, the segmented voxel volume is peeled in a number of iterations by an algorithm similar to that of Dokldal et al. [3]. The resulting skeleton of branches and bifurcations (see Figure 10) is a faithful approximation of the centre structure of the vessels: the branches are close to the core lines of the vessel branches and the skeleton bifurcations are close to the centers of the vessel bifurcations.

There are many algorithms to create a skeleton (see [6] for an introduction in 2D and 3D thinning). We do not claim that the algorithm described in the sequel, is better than any other algorithm. It may be replaced by a better one as long as the resulting skeleton is a faithful approximation of the centre structure of the vessels.

Starting point is the original segmented voxel volume with the extremity voxels indicated by a special label (the darker spots in Figure 9). Each iteration creates first a skin layer by labelling the current boundary voxels except the extremity voxels. Each voxel has at most six face neighbors. An original vessel voxel (thus with label one) is a boundary voxel if and only if one of these neighbors is not a voxel with a positive label. The label of a boundary voxel is increased with the number of face neighbors with a zero or negative label (boundary voxels removed in a previous iteration).

Checking and removing (if the local topology does not change) all boundary voxels of the current skin layer before creating, checking and removing the boundary voxels of the next inner skin layer, guarantees that the remaining set of voxels approximates the core lines of the vessel graphs.

After peeling is finished, bifurcation voxels are marked by a special label. Bifurcation voxels are voxels with more than two positive neighbors in the peeled segmented voxel volume.

4.1 Processing of Boundary Voxels

The boundary voxels of the current skin layer are checked and possibly removed in order of their label. The volume is traversed from the lower left front corner to the upper right back corner, looking for boundary voxels with the current label.

Boundary voxels with exactly one vessel voxel as face neighbor are always removed. Boundary voxels with at least two and at most four vessel voxels as face neighbors are removed unless the local topology changes by removing this voxel (see Section 4.2). Boundary voxels with exactly five vessel voxels as face neighbor are never removed because removing such a boundary voxel causes a local concavity.

If a boundary voxel is removed, the labels of its face neighbor boundary voxels are adjusted. Face neighbors with a label greater than or equal to the label of the boundary voxel removed, are immediately processed in decreasing order of their new label before normal scanning from the lower left front corner to the upper right back corner continues.

4.2 Topology Checks

The topology checks are performed by looking at the corner neighbors of the boundary voxel tested. To this end a cube of 3 by 3 by 3 cells is filled with zero's. The centre cell corresponds to the boundary voxel checked. The other cells correspond to its corner neighbors. A cell is set to one if the corresponding corner neighbor has a positive label.

It is possible that some of these cells are equal to one although the corresponding corner neighbor vessel voxel is not face connected, via vessel voxels, to the boundary voxel examined. This can happen when two components are corner but not face connected to each other. These cells are reset to zero.

First, resetting the centre cell to zero, should not result in two disjunct face connected sets of positive cells. Secondly, resetting the centre cell to zero, should not decrease the number of face or the number of corner connected sets of zero cells. If one of these two complementary checks fails, the boundary voxel examined should not be removed.

These topology conditions differ slightly from those given by Bertrand et al. [1] but their conditions resulted in occasional small cycles not present in the segmented voxel volume.

5 Graph Generation Dissected

In the third step, the graphs are created. Starting point is the peeled segmented voxel volume with the bifurcation voxels indicated by a special label together with the extremities (stored in a special wave by the wave propagation step). A directed graph is generated for each face connected set of positive vessel voxels in the peeled segmented voxel volume. This graph contains one node for each extremity voxel and one node for each bifurcation voxel. A wave is created and stored in this graph for each branch between two nodes. This special wave consists of a list of voxel descriptions for the face connected positive vessel voxels between these two nodes. Each branch gets a unique number. The voxels of a branch wave get this number as label. An example of the resulting segmented voxel volume is shown in Figure 11.

The generated graphs facilitate not only fully-automatic vessel tracing from one node to another node, but are also required to label short branches correctly. In this last case, information about the bifurcation structure (especially its size) at one end of the branch is needed for labelling voxels at the other end of the branch (see Section 6.1). The generated graphs contain this neighbor information.

5.1 Data Structures

A graph contains amongst other things the number of branches in the graph, the number of nodes in the graph, a pointer to the list of its branch waves and two pointers, one to the first node and one to the last node of the list of nodes of the graph.

A node contains amongst other things a voxel description for the corresponding extremity or bifurcation voxel (called the centre voxel of the node from now on) and the number of branches in this node. The number of branches should be either equal to one or greater than two and less than or equal to six. If the number of branches is equal to one, the node corresponds to an extremity of the vessel structure. Two branches at a node is not possible because nodes are created either for an extremity or a bifurcation. A bifurcation voxel is a voxel with at least three and at most six positive face neighbors.

A node contains for each branch amongst other things a pointer to the corresponding branch wave, a direction number indicating whether the head or the tail of this branch wave is connected to this node and a pointer to the node at the other end of this branch wave. The pointers to other nodes represent the graph structure. The branch waves make it possible to travel from one node to another node going from one voxel to the next voxel of the branch wave.

Note that whether a branch of a node is incoming or outgoing does not imply direction of blood flow at that node.

5.2 Algorithms

The graphs are generated by scanning the list of extremities. It is easy to detect whether an extremity is already part of a graph because during the generation of the graph, the voxels visited are made negative.

For each positive extremity voxel a node is generated and filled. For each positive face neighbor of the centre voxel of a node a branch wave is generated and filled with the face connected positive vessel voxels until a positive extremity or bifurcation voxel is encountered. A node is generated for the closing extremity or bifurcation voxel and the branch information in the two nodes is updated. The algorithm continues with tracing the branches of the new node.

In case of cycles in the vessel structures, a node may already exist for the closing bifurcation voxel. If so, the bifurcation voxel is negative. Therefore, if no positive extremity or bifurcation voxel is found at the end of a branch wave, the face neighbors of the last positive voxel are inspected for a negative bifurcation voxel. The corresponding node is used to close the branch.

6 Node Geometry Dissected

Correct labelling of voxels requires that voxels of a bifurcation can be distinguished from branch voxels. To this end node geometry is generated. Node geometry contains the position and the shape of the centre region of the bifurcation and the position and the shape of its branch regions. First, a centre sphere is created. The position of the bifurcation voxel is used as centre of the centre sphere. The radius (in voxels) of the centre sphere is derived from the distance transform of the bifurcation voxel.

Next, a branch sphere is created for each branch. The centre of a branch sphere is equal to the position of the voxel of the branch wave so that the branch sphere is just separated from the centre sphere (see Figure 3). The radius of the branch sphere is derived from the distance transform of the centre voxel of the branch sphere. Travelling along and checking each voxel of the branch wave yields the first voxel which fulfils these conditions.

Finally, one branch plane and one centre plane is created for each branch. The branch plane is defined by the centre of the branch sphere and the normal which is given by the direction of the connection line from the centre of the branch sphere to

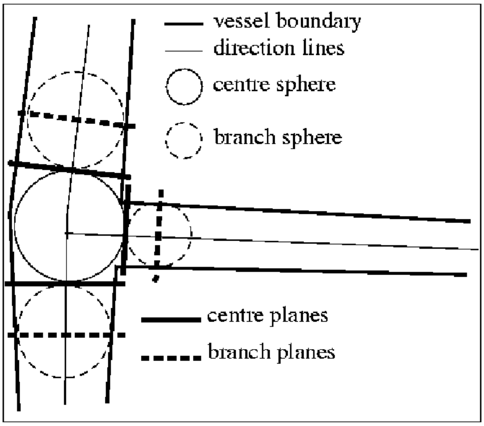

Figure 3: Node geometry

the centre of the centre sphere. The position of the corresponding centre plane is determined by the intersection of the centre sphere and this connection line. Its normal is equal to the normal of the branch plane multiplied with -1.

6.1 Overlapping Bifurcation Spheres

If the distance in voxels along the connecting branch wave between two bifurcation voxels or between a bifurcation and extremity voxel is small compared with the radii of the centre spheres, centre and branch spheres overlap. Identifying the entities at one end with ”first” and at the other end with ”second”, the following cases can be distinguished:

1. If all voxels of a branch wave are inside the first centre sphere, the position of the last voxel is used as centre of the first branch sphere. The radius of this branch sphere is in this case multiplied with minus one to indicate this condition.
2. If some of the voxels of this branch wave are outside the first centre sphere, but the first branch sphere overlaps the first centre sphere even for the last voxel of the branch wave, this last voxel is used as centre of the first branch sphere. This overlap does not harm the final labelling because only the voxels between the branch plane and the corresponding centre plane are members of the branch region (see Section 7.1).
3. If the second node of the branch is a bifurcation and if the centre of the first branch sphere is inside the second centre sphere, the radius of the first branch sphere is multiplied with minus one to indicate this condition. Indeed, if the centre of a branch sphere is inside one of the centre spheres of a branch between two bifurcations, all voxels of the branch are inside the two bifurcations.

Note that if the node at the other end of the branch is an extremity, all voxels outside the bifurcation including the voxels in the neighborhood of the extremity are labelled as branch voxels. So, in this case it is not necessary to test whether the centre of the branch sphere of the bifurcation is inside the centre sphere of the extremity.

The branch and centre spheres are shown in Figure 12. As is clear from this figure, centre spheres are also generated for the extremities. Note the occasional overlap.

7 Final Labelling Dissected

In the fifth step, the vessel voxels get their final label. Starting point is the original segmented voxel volume with only tissue and original vessel voxels and the generated node geometry. The original vessel voxels in a region are labelled by a similar twin wave algorithm as is used for component selection (see Section 3.3). The difference is that the original vessel voxels should not only be face connected, via vessel voxels, to the initial voxel, but should also fulfil constraints depending on the region currently processed.

7.1 The Branch Regions

First, the voxels in the branch regions get the label of their branch. A simplified 2D example of a branch region is given in Figure 4.

The initial voxel for the twin wave algorithm is the centre voxel of the branch sphere. The additional constraints are:

1. The distance of the original vessel voxel to the centre of the branch sphere should be less than or equal to twice the radius of the branch sphere.
2. The original vessel voxel should reside between the branch plane and the centre plane.

The first constraint prevents unrestricted grow of the labelled area of a side branch in case the centre sphere is so small that the centre plane for this side branch does not intersect this branch (is to the left of the right vertical vessel boundaries in Figure 4). In this case a large number of original vessel voxels of the main branch, fulfilling the second constraint,

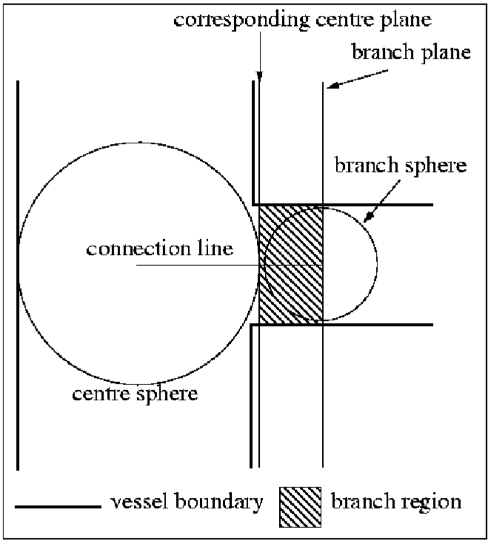

Figure 4: Branch region

are face connected to the centre voxel of the branch sphere.

7.2 The Centre Regions

After the voxels in the branch regions are labelled, the voxels in the centre regions (see Figure 3) are labelled. Each centre region gets a unique label, different from all branch numbers.

The initial voxel for the twin wave algorithm is the centre voxel of the node. The additional constraints are:

1. The distance of the original vessel voxel to the centre of the centre sphere should be less than or equal to the radius of the centre sphere plus the maximum of the branch radii of the current node.
2. The original vessel voxel should reside inside the ”enclosure” of the branch planes of the node.
3. The distance of the original vessel voxel to the position of the current node should be less than or equal to all distances of the original vessel voxel to the positions of the neighbor nodes.

After labelling the original vessel voxels in the branch regions, the original vessel voxels of the centre regions are separated from the remaining original vessel voxels in the branches by the already labelled vessel voxels. But when a branch region is skipped because its branch radius is negative or when the distance between the branch plane and centre plane is very small, the original vessel voxels of the centre region are face connected to the original vessel voxels of the branch. The first two constraints prevent unrestricted grow of the labelled area in these cases.

The third constraint separates the voxels of the current centre region from the voxels of the centre regions of neighbor nodes in case of overlapping centre regions.

7.3 The Halfway Regions

After the voxels in the centre regions are labelled, the voxels of the branches between two branch regions are labelled. The original vessel voxels of the branch wave are used as initial voxels. But using one original vessel voxel of the branch wave as initial voxel for the twin algorithm will normally result in labelling of most other original vessel voxels of the branch wave. So, only one or two vessel voxels of the branch wave will really be used as initial voxel.

Because the remaining original vessel voxels of a branch are separated from the original vessel voxels in other branches by the labelled voxels of the centre regions, no additional constraints are necessary to prevent unrestricted grow of the labelled area.

8 Results and Conclusions

The fully-automatic branch labelling method has been incorporated in a demo program for testing. Five voxel volumes (128x128x128), acquired with the 3D Integris [4], are labelled. The number of components varied between 1 and 19, the number of branches between 7 and 145.

Labelling takes between 10 and 25 seconds on an SGI Octane. The wave propagation step takes between 2 and 12 seconds, the peeling step between 8 and 14 seconds.

The following conclusions can be drawn from the results, the pictures and the experience gained:

1. The new method for fully-automatic branch labelling of voxels in vessel structures gives better results than the wave propagation method.
2. The wave propagation step deteriorates in case of wide short branches (for example an aneurysm), producing spurious extremities.
3. The quality depends on the smoothness of the surfaces of the voxel structures. A very noisy surface results in many short branches. Smoothing the voxel volume can remove these short branches. However, finding the correct

smoothing factor still requires human interaction.

4. The elapsed time for the wave propagation steps depends on the number of components. The elapsed time for the peeling step depends on the number of vessel voxels in the original segmented volume. The elapsed time for the last three steps is negligible compared with the time taken by the first two steps.
5. This new method results also in a set of directed graphs (one for each component) which facilitates fully-automatic vessel tracing from one node (an extremity or bifurcation of the vessel structure) to another node of the same graph.

References

[1] G. Bertrand and G. Malandain ”A New Characterization of three-dimensional Simple Points”. Pattern Recognition Letters, volume 2, number 15, pp. 169–175, February 1994.

[2] J. Bruijns ”Semi-Automatic Shape Extraction from Tube-like Geometry”. Proceedings Vision Modeling and Visualization 2000, Saarbruecken Germany, pp. 347–355, November 2000.

[3] P. Dokldal, C. Lohou, L. Perroton and Gilles Bertrand ”A new thinning algorithm and its application to extraction of blood vessels”. Conference proceedings, BioMedSim ’99, ESIEE, April 1999, France

[4] R. Kemkers, J. Op de Beek, H. Aerts, R. Koppe, E. Klotz, M. Grasse, J. Moret ”3D-Rotational Angiography: First Clinical Application with use of a Standard Philips C-Arm System”. Proc. CAR’98, 1998.

[5] William E. Lorensen, Harvey E. Cline ”Marching Cubes: A High Resolution 3D Surface Construction Algorithm”. Computer Graphics, Vol. 21, No. 4, July 1987.

[6] V. Marion-Poty and S. Miguet ”A new 2-d and 3-d thinning algorithm based on successive border generations”. 4th Conference on Discrete Geometry in Computer Imagery, Grenoble, 1994.

[7] C. Zahlten, H. Juergens, H.-O. Peitgen ”Reconstruction of Branching Blood Vessels From CT-Data”. Proc. of the Eurographics Workshop on Visualization in Scientific Computing, Rostock, June 1994.

A Pictures

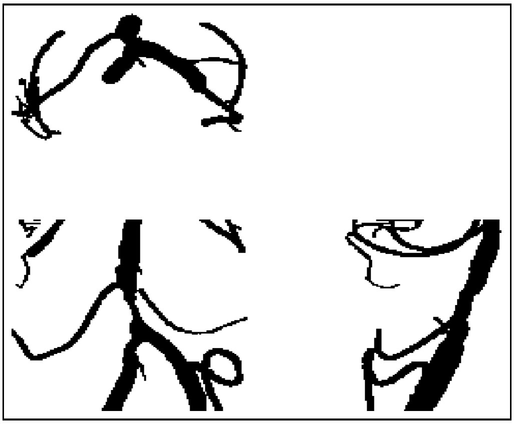

Figure 5: The pixmaps of a segmented volume

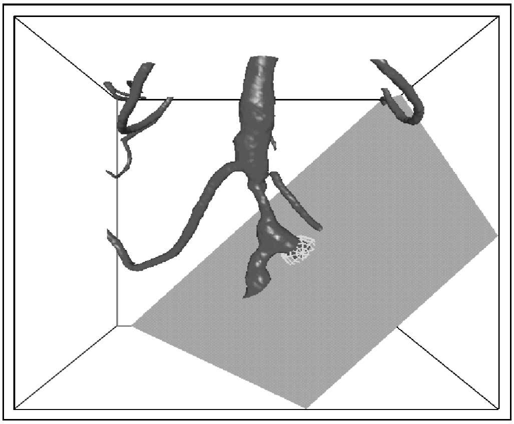

Figure 6: An aligned probe

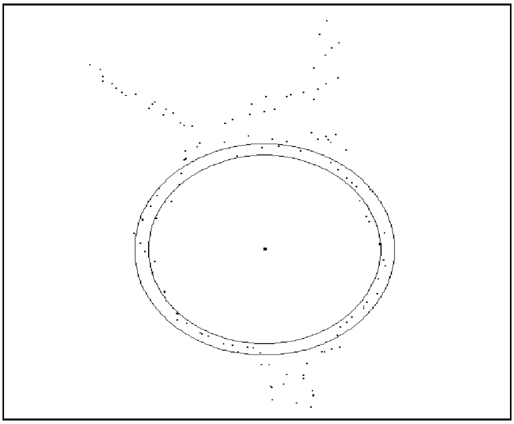

Figure 7: A slice view of a vessel with a neighbor

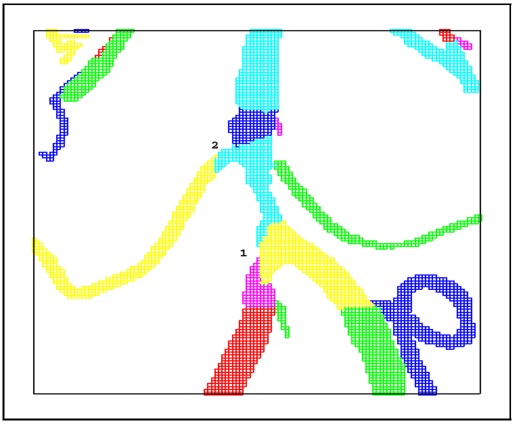

Figure 8: Labelling by wave propagation

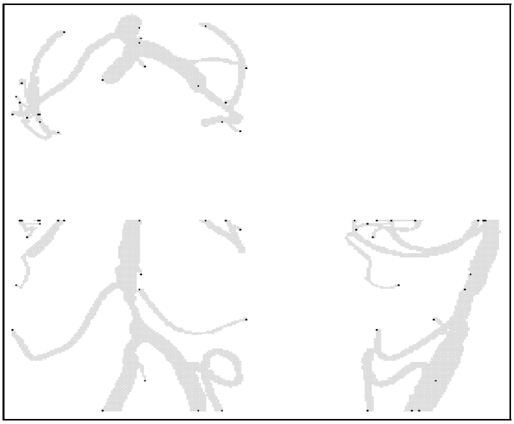

Figure 9: The extremities

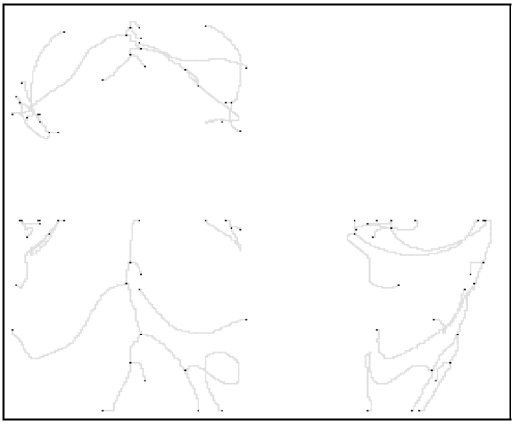

Figure 10: The peeled segmented volume

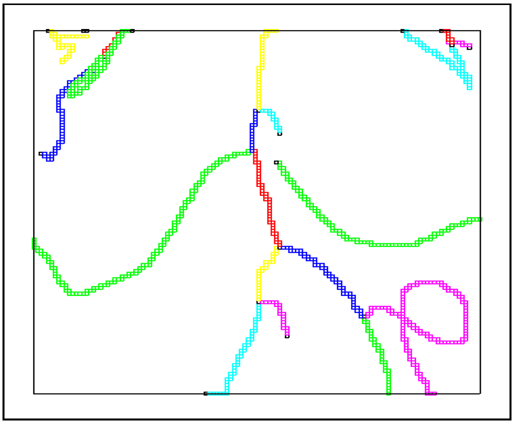

Figure 11: The graphs

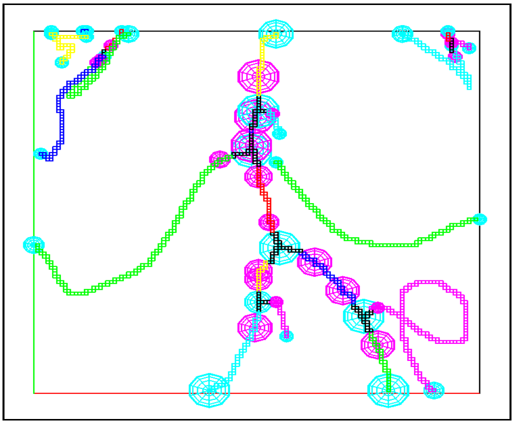

Figure 12: The centre and branch spheres

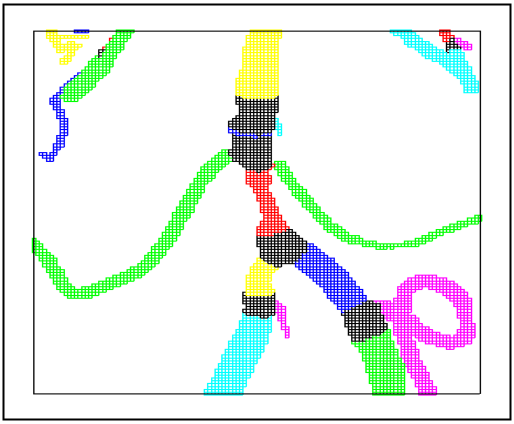

Figure 13: The labelled segmented volume