Automatic Detection of 3D Vascular Tree Centerlines and Bifurcations in High Resolution Magnetic Resonance Angiography

L. Zhang[1,2], D. L. Parker[2], J. A. Roberts[2], J. Guo[2,3], P. Vemuri[1,2], S. Moon[1,2], F. Noo[2]

[1]Electrical and Computer Engineering, Univeristy of Utah, Salt Lake City, Utah, United States, [2]Radiology, Utah Center for Advanced Imaging Research, Salt Lake City, Utah, United States, [3]Physics, Univeristy of Utah, Salt Lake City, Utah, United States

Introduction: Extraction of the blood vessel skeleton from 3D MRI and MR angiography (MRA) has been investigated by many researchers [1-3] with varying degrees of success. However, the algorithms are in general complicated and time consuming, which becomes a big problem when finding centerlines in 3D high-resolution MRA data, because the data volume is large. Bifurcation determination is another big problem. So we aimed at developing a simple and fast algorithm to automatically extract vessel skeletons from the huge and ever growing amount of image data available in 3D images of the vascular bed. We chose Dijkstra's algorithm [4] as a powerful tool toward this goal. Thus we implemented this algorithm for a 3D vascular tree using an appropriate cost function for vessel voxels. Accuracy of the algorithm is tested on a phantom image with known centerline and bifurcation locations.

Method: This algorithm includes four steps.

Step I - Segmentation

The source image is segmented using the ZBS algorithm [5]. Then a binary voxel mask of the segmented vascular tree is created.

Step II – Cost calculation

The mask is used to find for each vessel voxel the **D**istance **F**rom the closest **E**dge (DFE). The cost function for voxel x is defined as a function of DFE at the voxel:

$$\text{cost}(x) = A \cdot \sqrt{1 - \frac{\text{DFE}(x)}{\text{max\_DFE}(x)}} + 1 \qquad (1)$$

Here A is a constant to amplify the value of the cost, and max_DFE(x) is the local maximum DFE value that voxel x belongs to.

Step III – Dynamic programming

Using the cost function defined by (1), a 3D version of the shortest path detection algorithm is implemented based on Dijkstra's algorithm to find the minimum cost path [6]. Then, all minimum cost paths are back traced toward the root point of the vascular tree. The tracing is performed in the order of decreasing path length and stops whenever the root point is reached or a previously detected centerline path is hit. A centerline path is said to be detected if the traced path is longer than a predefined length threshold. Otherwise, it is treated as a non-centerline path and is not included in the array which saves the centerline paths. A bifurcation is defined as the point where a subsequent centerline path joins a previously detected centerline path.

Step IV – Curve fitting

These originally extracted centerlines are jagged. Therefore, they are smoothed using curve fitting. Before fitting, each centerline path is split into segments between adjacent bifurcations or a bifurcation and an end point. Then, each centerline segment is fit into a parameterized curve using Chebyshev approximation [7], which is very accurate and is easy to implement. There is little chance that the fitted curve will pass outside the vessel. But the Chebyshev fit does not preserve the position of the two end points of a segment, which means the bifurcation locations cannot be preserved. Therefore, we do a cubic spline fit [7] to the chebyshev fitted curve, which include the two end points as control points. Since the spline fit goes through the control points exactly, the bifurcation locations are preserved.

Results: We first applied this algorithm to a simulated coronary artery tree. The result is displayed in Fig. 1 a). The actual centerlines and bifurcations for the phantom are already known and are plotted in Fig. 1 b). The detected centerlines and bifurcations are plotted in Fig. 1 c). In Fig. 1 d), the actual and detected centerlines are plotted together for comparison. Visually, they are indistinguishable except for the areas near bifurcations. The standard deviation of each centerline segment varied from 0.4 to 1 voxel. The mean standard deviation for all segments was 0.55 voxel. The bifurcation displacement varied from 1 voxel to 3 or 4 voxels. The result of applying the algorithm to human data is shown in Fig. 2. The centerlines are visually well centered.

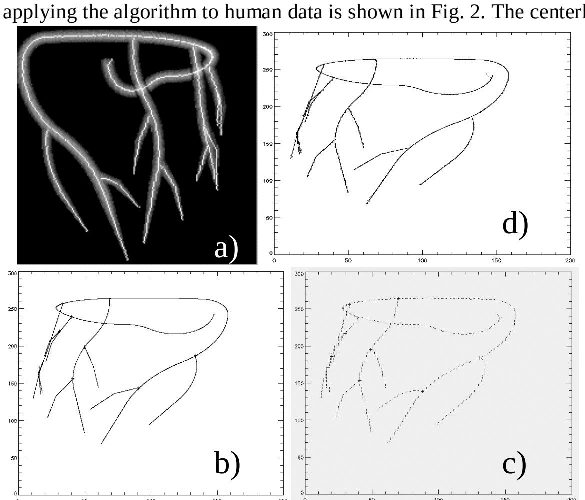

Fig. 1 a) The extracted centerlines overlayed on the DFE image of the phantom tree. **b)** The plot of actual centerlines and bifurcations. **c)** The plot of detected centerlines and bifurcations. **d)** Overlapping of the detected and actual centerlines for comparison.

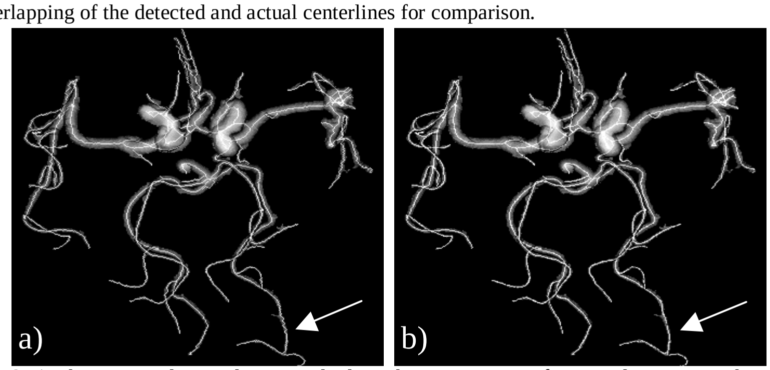

Fig. 2 a) The extracted centerlines overlaid on the DFE image of a vascular tree in a human brain. The arrow points to a rather jagged branch. **b)** The fitted centerlines overlaid on the same DFE image. The arrow shows the smoothing of the jagged branch in a).

Conclusions:

A simple and fully automatic 3D centerline extraction algorithm is implemented based on dynamic programming and curve fitting. Application of the algorithm to a simulated coronary artery phantom showed high accuracy of centerline detection. The accuracy of bifurcation detection is a little lower. Reasonable centerlines and bifurcations, based upon visual assessment, were obtained from application of the algorithm to human MRA data.

References:

[1] T. Pavlidis, *Computer Graphics and Image Procesing*, vol. 13, pp. 142-57, 1980.
[2] D. Guo and P. Richardson, *IEEE Computers in Cardiology*, vol. 25, pp. 441-44, 1998.
[3] Y. Zhou and A. W. Toga, *IEEE TVCG*, vol. 5, pp. 196-209, 1999.
[4] E. W. Dijkstra, *Numerishe Mathemetik*, pp. 269-71, 1959.
[5] D. L. Parker, et al., *JMRI*, vol. 11, pp. 378-88, 2000.
[6] J. Morris, "Dijkstra's Algorithm," 1998.
[7] William H. Press, et al., Numerical Recipes in C, 1992.