3D Interactive Centerline Extraction

Yong Zhang[1*], Kun Chen[1,2], Stephen T.C. Wong[1]

July 03, 2008

[1]The Methodist Hospital Research Institute, Radiology Department, Weill Medical College, Cornell University, Houston TX
[2]State Key Laboratory of Industrial Control Technology, Institute of Industrial Process Control, Zhejiang University, Hangzhou, P.R.China
[*]Contact: Email: yzhang@tmhs.org

Abstract

This document describes a user-steered method to interactively track centerlines of tubular objects in 3D space. The method is developed as a plug-in of ImageJ using Java language. To evaluate the tracking ability and tracking accuracy, this method has been applied to coronary artery tracking in coronary CT angiography data. Its potential as a user-steered 3D centerline tracking tool has been discussed as well as its limitations and possible improvements.

Contents

Vessel structure detection, segmentation, and visualization are very important problems in the field of medical image analysis. A lot of methods have been proposed to automate the analysis, especially for vessel segmentation [1, 2]. Nevertheless, it still remains a significant challenging problem to develop fully automated methods for vessel segmentation. This is due to the cluttered objects, partial volume effects, intensity inhomogeneity artifact, complex vessel structures, and huge size and high dimensionality in medical images. What's more, the available software tools that can be used for tracking centerline in 3D space are very rare. Based on the above observation, it is still worthwhile to develop a user-steered software tool to extract centerlines of tubular structures in medical images in 3D space.

In this document, we describe a user-steered method to interactively track centerlines of tubular objects in medical images in 3D. To track a centerline, the method requires that users select a starting point, one or multiple intermediate points, and an ending point. Then the rest of the points on or near the centerline will be extracted automatically by the method. We have applied this software tool to the application of coronary artery tracking and evaluate it in terms of tracking capability and tracking accuracy. Its potential usage and limitation have been discussed at the end of the document.

1 Description

Our method for centerline tracking is based on live wire [3-5] and NeuronJ [6]. An Eigen-analysis-based method is used to estimate the local directions of the centerlines of the tubular objects. The proposed method is based on the observation that the local principal directions of a voxel in a 3D image are given by the eigenvectors of the 2nd order derivative matrix (Hessian matrix), given the assumption that the cross sections of the tubular objects satisfied Gaussian or bar-like profile models [7]. This derivative matrix is calculated based on the intensity values around the voxel. The eigenvectors indicating the local principal directions correspond to the smallest absolute eigenvalue among the three eigenvalues of the derivative matrix. In this method, first the lineness of a tubular object is defined based on the eigenvalues of the Hessian, and then the centerlines are estimated and tracked.

The definition of lineness is based on Sato et al.'s work [8]. To encode line morphological information, the lineness is defined as the probability of a voxel belonging to a tubular structure. The lineness is calculated based on the Eigen-analysis of the Hessian matrix of the 3D image. Denoting its three eigenvalues as $\lambda_1, \lambda_2, \lambda_3$ in descending order ($|\lambda_1| \ge |\lambda_2| \ge |\lambda_3|$), the lineness measure $\eta(p)$ for of a voxel p is calculated as,

$$\eta(p) = \begin{cases} |\lambda_2| + \lambda_3, & \text{if } \lambda_1 < 0, \lambda_2 < 0, \lambda_3 \le 0 \\ |\lambda_2| - \dfrac{\lambda_3}{4}, & \text{if } \lambda_1 < 0, \lambda_2 < 0, 0 < \lambda_3 < |\lambda_2| \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

By calculating the above measure for each voxel in the image, a lineness map is created and used to enhance the vessel structures. Notice that the first and second cases in Eq.(1) are not symmetric. This is because when $\lambda_3 > 0$, the corresponding 3D line structure has concavity involved. To avoid true line structures being fragmented at concave locations, the influence of λ_3 should be reduced. Experiments show that by using $|\lambda_2| - \lambda_3 / 4$ the fragmentation can be reduced to better measure the lineness.

The centerlines of arteries are estimated based on the lineness features and the live-wire-based linking algorithm. First, a local cost value $c(p)$ is calculated for each voxel p in the image I based on the lineness $\eta(p)$ computed by Eq. (1),

$$c(p) = 1 - \frac{\eta(p)}{\max_{p \in I}\{\eta(p)\}}. \tag{2}$$

The value of $c(p)$ decreases if the voxel goes near the centerline of the 3-D line structure, and it is minimized on the centerline. Then, the live-wire-based linking algorithm is applied to connect the centerline voxels to construct the centerline of the artery. This linking method requires that the users specify a starting voxel on the tube of interest. The searching algorithm then finds the optimal paths from the starting voxel to all other voxels in the volume. An optimal path means a path with a globally minimal cumulative cost values. The cost function $L(p \to r)$ from voxel p to another neighboring voxel r is calculated as follows,

$$L(p \rightarrow r) = \alpha \cdot c(r) + (1-\alpha) \cdot \frac{1}{2}\left(\sqrt{1-\left|\mathbf{v_p} \cdot \frac{\mathbf{r}-\mathbf{p}}{\|\mathbf{r}-\mathbf{p}\|}\right|} + \sqrt{1-\left|\mathbf{v_r} \cdot \frac{\mathbf{p}-\mathbf{r}}{\|\mathbf{p}-\mathbf{r}\|}\right|}\right), \tag{3}$$

where $\mathbf{v_p}, \mathbf{v_r}$ are the eigenvectors corresponding to the smallest absolute eigenvalues of the Hessian for voxel p and r, respectively. Vectors $\mathbf{p}, \mathbf{r}$ are defined as the vectors from the origin pointing onto voxel p and r, respectively. α is a constant parameter to balance the influences of the two cost components (i.e., the local cost and the linking cost) in Eq.(3). Previous work [9] has shown that the local cost is the major component that determines the optimal path while the linking cost leads to a smoother and tighter tracking. The method is implemented using Java language and developed as a plug-in to the popular ImageJ public domain image processing program.

2 Experimental Results in Coronary Artery Tracking

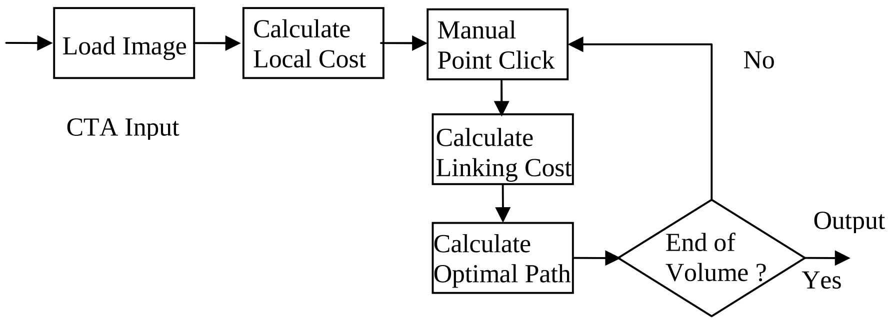
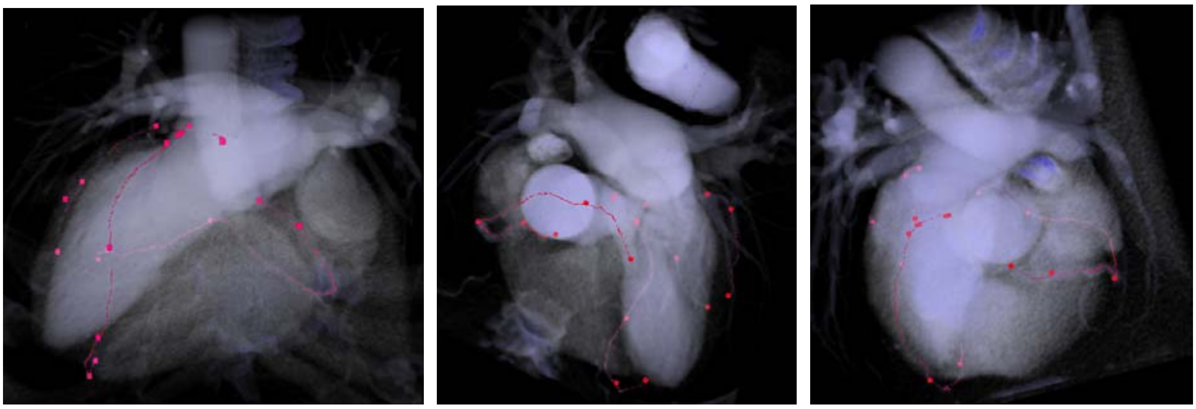


Figure 1 Block diagram for our proposed method

To better evaluate the method and compare with other algorithms, the developed software tool is applied to the coronary artery tracking challenge (CAT08), a grand challenge of 3D segmentation in the clinic in conjunction with MICCAI 2008. According to the description of the CAT08, coronary CTA data for this challenge was acquired in the Erasmus Medical Center Rotterdam, The Netherlands. 32 datasets were randomly selected from a series of patients that underwent coronary CTA. Twenty datasets were acquired on a Siemens Somatom Sensation 64 and twelve datasets on a Siemens Somatom Definition CT scanner. Diastolic reconstructions were used, with reconstruction intervals varying from 250ms to 400ms before the R-peak. Three datasets were reconstructed using a B46f kernel, all others were reconstructed using a B30f kernel. A typical coronary CTA dataset has a resolution of 0.36*mm* x 0.36*mm* x 0.40*mm*.

Our method is categorized as interactive tracking as it requires a series of manually clicked points per vessel as input to completely track the coronary arteries. The procedure to interactively track a coronary artery is illustrated in Figure 1. Once the CTA image is loaded, a one-time calculation of local cost is performed for each voxel in the volume. Then the user can manually pick a series of points to track the artery of interests. Once a point is selected, the program calculates the optimal paths connecting to this point from all of the other voxels. The calculation may take a while since it is in 3D space (in our experiment, to reduce the running time, the original CTA volume is rescaled to *half* of its original sizes with a resolution of 0.72*mm* x 0.72*mm* x 0.80*mm*) Once finished, the user can move the mouse to any voxel in the image volume and an optimal path is visualized to indicate the optimal path connecting the current mouse location and the previous manual selected point. The user may go through a couple of slices to inspect the validity of the tracking. If not satisfied, then the user may manually select another point at an earlier location of the vessel to correct the tracking. This is necessary as vessels may have sharp turns or other complex 3D spatial structures.



(a) (b) (c)

Figure 2 Examples of tracking results for (a) dataset#14; (b) dataset#15; (c) dataset#19.

To evaluate the performance of the method, three overlap measures are used to assess the ability of tracking centerlines and three distance measures are used to determine the accuracy of centerline tracking [10]. Four major coronary arteries are tracked from the coronary CTA image, namely, Right Coronary Artery (RCA), Left Anterior Descending branch (LAD), Left Circumflex artery (LCX), and one large side branch of the main coronary arteries. The interactive method is applied to a group of 16 datasets (Testing 1). Thus a total of 64 coronary arteries are tracked and measured. Table 1 shows the average overlap per dataset. Table 2 shows the average accuracy per dataset. Table 3 summarizes the results of all 64 artery tracking. To describe the type and amount of user-interaction needed, the starting point and ending point provided for each vessel have to be selected by the user. Then one or more intermediate points on or near the artery centerline need to be selected by the user to completely track the whole coronary artery. The actual number of user-clicked points is from 3 to 10, depending on how complex the artery is. For example, 4 points are clicked to track vessel 1 in dataset#15 and 10 points are clicked to track vessel 0 in dataset#18. Figure 2 shows three examples of coronary artery tracking results, with manually clicked points highlighted to indicate the type and amount of user-interaction needed.

Table 1: Average overlap per dataset

Dataset nr.	OV %	OV score	OV rank	OF %	OF score	OF rank	OT %	OT score	OT rank	Avg. rank
8	86.6	56.0	–	61.8	45.4	–	89.3	47.5	–	–
9	92.7	51.8	–	67.7	36.5	–	93.8	46.9	–	–
10	85.3	43.7	–	19.9	10.3	–	87.6	43.8	–	–
11	90.6	51.5	–	33.5	29.3	–	90.6	46.4	–	–
12	85.7	44.3	–	24.7	13.8	–	87.9	44.2	–	–
13	95.3	48.4	–	42.7	26.7	–	95.9	48.0	–	–
14	93.0	47.1	–	38.0	22.1	–	93.4	46.7	–	–
15	96.0	58.8	–	77.6	40.2	–	97.0	48.5	–	–
16	90.2	55.3	–	37.9	20.3	–	90.9	45.4	–	–
17	84.0	67.2	–	35.6	30.2	–	84.0	49.4	–	–
18	91.2	58.6	–	59.5	50.0	–	91.2	58.2	–	–
19	98.6	71.8	–	76.6	68.5	–	98.6	61.8	–	–
20	93.6	55.8	–	45.8	26.3	–	93.7	47.1	–	–
21	88.9	45.2	–	53.1	44.6	–	91.7	58.8	–	–
22	94.8	47.8	–	80.0	40.0	–	95.0	60.0	–	–
23	95.4	48.2	–	64.9	32.9	–	95.4	47.7	–	–
Avg.	**91.4**	**53.2**	–	**51.2**	**33.6**	–	**92.3**	**50.0**	–	–

Table 2: Average accuracy per dataset

Dataset nr.	AD mm	AD score	AD rank	AI mm	AI score	AI rank	AT mm	AT score	AT rank	Avg. rank
8	0.63	30.3	–	0.52	32.6	–	0.61	30.9	–	–
9	0.55	20.6	–	0.47	21.6	–	0.54	20.8	–	–
10	0.67	20.2	–	0.45	22.8	–	0.66	20.7	–	–
11	0.68	24.6	–	0.58	25.9	–	0.68	24.6	–	–
12	0.60	21.9	–	0.47	24.7	–	0.58	22.2	–	–
13	0.53	22.5	–	0.50	23.2	–	0.53	22.8	–	–
14	0.55	27.7	–	0.48	29.1	–	0.55	27.8	–	–
15	0.53	22.8	–	0.49	23.5	–	0.53	23.2	–	–
16	0.62	19.4	–	0.54	20.7	–	0.65	18.1	–	–
17	0.74	37.2	–	0.50	40.8	–	0.74	37.2	–	–
18	0.61	23.3	–	0.44	25.0	–	0.61	23.3	–	–
19	0.59	26.6	–	0.57	26.9	–	0.59	26.6	–	–
20	0.61	24.3	–	0.56	25.4	–	0.61	24.3	–	–
21	0.59	17.9	–	0.48	19.4	–	0.57	18.4	–	–
22	0.67	18.9	–	0.61	19.6	–	0.67	18.8	–	–
23	0.52	23.7	–	0.49	24.2	–	0.52	23.7	–	–
Avg.	**0.60**	**23.9**	–	**0.51**	**25.3**	–	**0.60**	**24.0**	–	–

Table 3: Summary

Measure	% / mm min.	% / mm max.	% / mm avg.	score min.	score max.	score avg.	rank min.	rank max.	rank avg.
OV	53.3%	100.0%	91.4%	39.6	100.0	53.2	–	–	–
OF	2.6%	100.0%	51.2%	1.3	100.0	33.6	–	–	–
OT	53.3%	100.0%	92.3%	32.7	100.0	50.0	–	–	–
AD	0.39 mm	1.28 mm	0.60 mm	13.2	56.0	23.9	–	–	–
AI	0.39 mm	0.72 mm	0.51 mm	15.0	66.4	25.3	–	–	–
AT	0.39 mm	1.28 mm	0.60 mm	13.3	56.0	24.0	–	–	–
Total							–	–	–

3 Discussion

We have developed a software tool to interactively track centerlines of bright tubular objects in dark background. The software tool is able to track the centerline of a tubular object in a user-steered mode and the tracking can be inspected and corrected on the fly in the 3D space. The tool has been applied and evaluated in coronary artery tracking. The evaluation shows that the interactive method has very good tracking ability (91.4% overlap) while the tracking accuracy is somewhat moderate (0.60*mm* average distance or 0.83 voxel average distance along x/y directions and 0.75 voxel average distance along z direction). The moderate tracking accuracy of this tool is due to the rescale of the original image volume and the voxel accuracy (compared to sub-voxel accuracy in computational methods) of the points clicked by the users. Another limitation of this method is that the users need to have adequate knowledge about the tubular targets of interests. Otherwise it may lead to incomplete tracking and large measurement errors. For example, the tracking result of vessel 3 in dataset#17 only has an overlap of 53.3% and the average distance is 1.28*mm*. This is because the user has limited anatomic knowledge of the coronary arteries and stops the tracking at very early stage. It is also possible to study the systematic tracking errors introduced by the implementation and limitation of the interactive method and correct the measurements accordingly. The interactive software tool can be further improved in speed, computation efficiency, and sub-voxel accuracy using advanced programming techniques. Since this tool has very good tracking ability, it can be used to obtain rough centerlines of the tubular objects in the pre-processing step, and then advanced image processing techniques can be applied to refine the centerlines to improve the tracking accuracy.

4 Software Distribution

The software tool is freely available for research purpose. Please contact with the author if interested. The following software needs to be installed:

- ImageJ 1.35i or higher version.

Reference

[1] C. Kirbas and F. Quek, "A review of vessel extraction techniques and algorithms," *ACM Computing Surveys (CSUR),* vol. 36, pp. 81-121, 2004.

[2] A. C. Chung, "Image segmentation methods for detecting blood vessels in angiography," in *The 9th International Conference on Control, Automation, Robotics and Vision (ICARCV)* Singapore, 2006.

[3] W. A. Barrett and E. N. Mortensen, "Interactive live-wire boundary extraction," *Medical Image Analysis,* vol. 1, pp. 331-341, 1996.

[4] E. N. Mortensen and W. A. Barrett, "Interactive segmentation with intelligent scissors," *Graphical Models and Image Processing,* vol. 60, pp. 349-384, 1998.

[5] A. X. Falc˜ao, J. K. Udupa, and F. K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: Livewire on the fly," *IEEE Transactions on Medical Imaging,* vol. 19, pp. 55-62, 2000.

[6] E. Meijering, M. Jacob, J.-C. F. Sarria, P. Steiner, H. Hirling, and M. Unser, "Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images," *Cytometry*, vol. 58A, pp. 167-176, 2004.

[7] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *1st International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI1998)* Boston, MA, 1998.

[8] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis, "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Medical Image Analysis*, vol. 2, pp. 143-168, 1998.

[9] Y. Zhang, X. Zhou, A. Degterev, M. Lipinski, D. Adjeroh, J. Yuan, and S. T. C. Wong, "Automated neurite extraction using dynamic programming for high-throughput screening of neuron-based assays," *NeuroImage,* vol. 35, pp. 1502-1515, 2007.

[10] C. Metz, M. Schaap, T. v. Walsum, A. v. d. Giessen, A. Weustink, N. Mollet, G. Krestin, and W. Niessen, "3D Segmentation in the Clinic: A Grand Challenge II - coronary artery tracking," in *MICCAI Workshop on 3D Segmentation in the Clinic: A Grand Challenge* New York City, USA, 2008.

3D Interactive Centerline Extraction

Yong Zhang[1*], Kun Chen[1,2], Stephen T.C. Wong[1]

July 03, 2008

[1]The Methodist Hospital Research Institute, Radiology Department, Weill Medical College, Cornell University, Houston TX
[2]State Key Laboratory of Industrial Control Technology, Institute of Industrial Process Control, Zhejiang University, Hangzhou, P.R.China
[*]Contact: Email: yzhang@tmhs.org

Abstract

This document describes a user-steered method to interactively track centerlines of tubular objects in 3D space. The method is developed as a plug-in of ImageJ using Java language. To evaluate the tracking ability and tracking accuracy, this method has been applied to coronary artery tracking in coronary CT angiography data. Its potential as a user-steered 3D centerline tracking tool has been discussed as well as its limitations and possible improvements.

Contents

Vessel structure detection, segmentation, and visualization are very important problems in the field of medical image analysis. A lot of methods have been proposed to automate the analysis, especially for vessel segmentation [1, 2]. Nevertheless, it still remains a significant challenging problem to develop fully automated methods for vessel segmentation. This is due to the cluttered objects, partial volume effects, intensity inhomogeneity artifact, complex vessel structures, and huge size and high dimensionality in medical images. What's more, the available software tools that can be used for tracking centerline in 3D space are very rare. Based on the above observation, it is still worthwhile to develop a user-steered software tool to extract centerlines of tubular structures in medical images in 3D space.

In this document, we describe a user-steered method to interactively track centerlines of tubular objects in medical images in 3D. To track a centerline, the method requires that users select a starting point, one or multiple intermediate points, and an ending point. Then the rest of the points on or near the centerline will be extracted automatically by the method. We have applied this software tool to the application of coronary artery tracking and evaluate it in terms of tracking capability and tracking accuracy. Its potential usage and limitation have been discussed at the end of the document.

1 Description

Our method for centerline tracking is based on live wire [3-5] and NeuronJ [6]. An Eigen-analysis-based method is used to estimate the local directions of the centerlines of the tubular objects. The proposed method is based on the observation that the local principal directions of a voxel in a 3D image are given by the eigenvectors of the 2nd order derivative matrix (Hessian matrix), given the assumption that the cross sections of the tubular objects satisfied Gaussian or bar-like profile models [7]. This derivative matrix is calculated based on the intensity values around the voxel. The eigenvectors indicating the local principal directions correspond to the smallest absolute eigenvalue among the three eigenvalues of the derivative matrix. In this method, first the lineness of a tubular object is defined based on the eigenvalues of the Hessian, and then the centerlines are estimated and tracked.

The definition of lineness is based on Sato et al.'s work [8]. To encode line morphological information, the lineness is defined as the probability of a voxel belonging to a tubular structure. The lineness is calculated based on the Eigen-analysis of the Hessian matrix of the 3D image. Denoting its three eigenvalues as $\lambda_1, \lambda_2, \lambda_3$ in descending order ($|\lambda_1| \ge |\lambda_2| \ge |\lambda_3|$), the lineness measure $\eta(p)$ for of a voxel p is calculated as,

$$\eta(p) = \begin{cases} |\lambda_2| + \lambda_3, & if\, \lambda_1 < 0, \lambda_2 < 0, \lambda_3 \le 0 \\ |\lambda_2| - \dfrac{\lambda_3}{4}, & if\, \lambda_1 < 0, \lambda_2 < 0, 0 < \lambda_3 < |\lambda_2| \\ 0 & otherwise \end{cases}. \tag{1}$$

By calculating the above measure for each voxel in the image, a lineness map is created and used to enhance the vessel structures. Notice that the first and second cases in Eq.(1) are not symmetric. This is because when $\lambda_3 > 0$, the corresponding 3D line structure has concavity involved. To avoid true line structures being fragmented at concave locations, the influence of λ_3 should be reduced. Experiments show that by using $|\lambda_2| - \lambda_3 / 4$ the fragmentation can be reduced to better measure the lineness.

The centerlines of arteries are estimated based on the lineness features and the live-wire-based linking algorithm. First, a local cost value $c(p)$ is calculated for each voxel p in the image I based on the lineness $\eta(p)$ computed by Eq. (1),

$$c(p) = 1 - \frac{\eta(p)}{\max_{p \in I} \{\eta(p)\}}. \tag{2}$$

The value of $c(p)$ decreases if the voxel goes near the centerline of the 3-D line structure, and it is minimized on the centerline. Then, the live-wire-based linking algorithm is applied to connect the centerline voxels to construct the centerline of the artery. This linking method requires that the users specify a starting voxel on the tube of interest. The searching algorithm then finds the optimal paths from the starting voxel to all other voxels in the volume. An optimal path means a path with a globally minimal cumulative cost values. The cost function $L(p \to r)$ from voxel p to another neighboring voxel r is calculated as follows,

$$L(p \to r) = \alpha \cdot c(r) + (1-\alpha) \cdot \frac{1}{2}\left(\sqrt{1-\left|\mathbf{v_p} \cdot \frac{\mathbf{r}-\mathbf{p}}{\|\mathbf{r}-\mathbf{p}\|}\right|} + \sqrt{1-\left|\mathbf{v_r} \cdot \frac{\mathbf{p}-\mathbf{r}}{\|\mathbf{p}-\mathbf{r}\|}\right|}\right), \tag{3}$$

where $\mathbf{v_p}, \mathbf{v_r}$ are the eigenvectors corresponding to the smallest absolute eigenvalues of the Hessian for voxel p and r, respectively. Vectors $\mathbf{p}, \mathbf{r}$ are defined as the vectors from the origin pointing onto voxel p and r, respectively. α is a constant parameter to balance the influences of the two cost components (i.e., the local cost and the linking cost) in Eq.(3). Previous work [9] has shown that the local cost is the major component that determines the optimal path while the linking cost leads to a smoother and tighter tracking. The method is implemented using Java language and developed as a plug-in to the popular ImageJ public domain image processing program.

2 Experimental Results in Coronary Artery Tracking

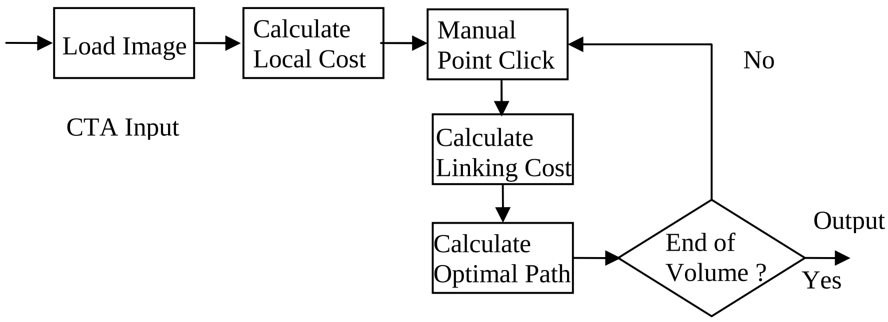
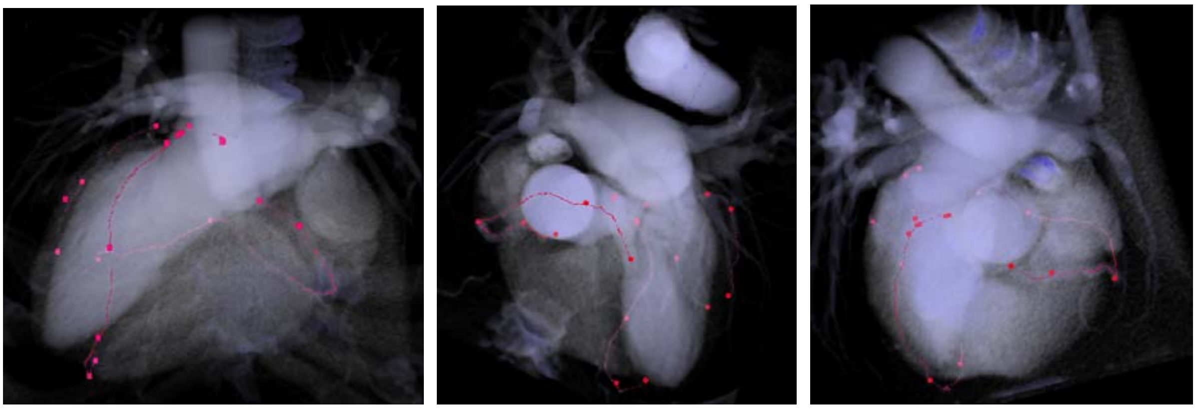


Figure 1 Block diagram for our proposed method

To better evaluate the method and compare with other algorithms, the developed software tool is applied to the coronary artery tracking challenge (CAT08), a grand challenge of 3D segmentation in the clinic in conjunction with MICCAI 2008. According to the description of the CAT08, coronary CTA data for this challenge was acquired in the Erasmus Medical Center Rotterdam, The Netherlands. 32 datasets were randomly selected from a series of patients that underwent coronary CTA. Twenty datasets were acquired on a Siemens Somatom Sensation 64 and twelve datasets on a Siemens Somatom Definition CT scanner. Diastolic reconstructions were used, with reconstruction intervals varying from 250ms to 400ms before the R-peak. Three datasets were reconstructed using a B46f kernel, all others were reconstructed using a B30f kernel. A typical coronary CTA dataset has a resolution of 0.36*mm* x 0.36*mm* x 0.40*mm*.

Our method is categorized as interactive tracking as it requires a series of manually clicked points per vessel as input to completely track the coronary arteries. The procedure to interactively track a coronary artery is illustrated in Figure 1. Once the CTA image is loaded, a one-time calculation of local cost is performed for each voxel in the volume. Then the user can manually pick a series of points to track the artery of interests. Once a point is selected, the program calculates the optimal paths connecting to this point from all of the other voxels. The calculation may take a while since it is in 3D space (in our experiment, to reduce the running time, the original CTA volume is rescaled to *half* of its original sizes with a resolution of 0.72*mm* x 0.72*mm* x 0.80*mm*) Once finished, the user can move the mouse to any voxel in the image volume and an optimal path is visualized to indicate the optimal path connecting the current mouse location and the previous manual selected point. The user may go through a couple of slices to inspect the validity of the tracking. If not satisfied, then the user may manually select another point at an earlier location of the vessel to correct the tracking. This is necessary as vessels may have sharp turns or other complex 3D spatial structures.



(a) (b) (c)

Figure 2 Examples of tracking results for (a) dataset#14; (b) dataset#15; (c) dataset#19.

To evaluate the performance of the method, three overlap measures are used to assess the ability of tracking centerlines and three distance measures are used to determine the accuracy of centerline tracking [10]. Four major coronary arteries are tracked from the coronary CTA image, namely, Right Coronary Artery (RCA), Left Anterior Descending branch (LAD), Left Circumflex artery (LCX), and one large side branch of the main coronary arteries. The interactive method is applied to a group of 16 datasets (Testing 1). Thus a total of 64 coronary arteries are tracked and measured. Table 1 shows the average overlap per dataset. Table 2 shows the average accuracy per dataset. Table 3 summarizes the results of all 64 artery tracking. To describe the type and amount of user-interaction needed, the starting point and ending point provided for each vessel have to be selected by the user. Then one or more intermediate points on or near the artery centerline need to be selected by the user to completely track the whole coronary artery. The actual number of user-clicked points is from 3 to 10, depending on how complex the 3D spatial structure of the artery is. For example, 4 points are clicked to track vessel 1 in dataset#15 and 10 points are clicked to track vessel 0 in dataset#18. Figure 2 shows three examples of coronary artery tracking results, with manually clicked points highlighted to indicate the type and amount of user-interaction needed. The running time for tracking the four arteries in a CTA image with typical rescaled size 256 x 256 x 160 is about 3 to 6 minutes with Intel Core 2 Duo 9300 2.5GHz CPU and 2GB memory, depending on the overall lengths and complexity of the arteries.

Table 1: Average overlap per dataset

Dataset nr.	OV %	OV score	OV rank	OF %	OF score	OF rank	OT %	OT score	OT rank	Avg. rank
8	86.6	56.0	–	61.8	45.4	–	89.3	47.5	–	–
9	92.7	51.8	–	67.7	36.5	–	93.8	46.9	–	–
10	85.3	43.7	–	19.9	10.3	–	87.6	43.8	–	–
11	90.6	51.5	–	33.5	29.3	–	90.6	46.4	–	–
12	85.7	44.3	–	24.7	13.8	–	87.9	44.2	–	–
13	95.3	48.4	–	42.7	26.7	–	95.9	48.0	–	–
14	93.0	47.1	–	38.0	22.1	–	93.4	46.7	–	–
15	96.0	58.8	–	77.6	40.2	–	97.0	48.5	–	–
16	90.2	55.3	–	37.9	20.3	–	90.9	45.4	–	–
17	84.0	67.2	–	35.6	30.2	–	84.0	49.4	–	–
18	91.2	58.6	–	59.5	50.0	–	91.2	58.2	–	–
19	98.6	71.8	–	76.6	68.5	–	98.6	61.8	–	–
20	93.6	55.8	–	45.8	26.3	–	93.7	47.1	–	–
21	88.9	45.2	–	53.1	44.6	–	91.7	58.8	–	–
22	94.8	47.8	–	80.0	40.0	–	95.0	60.0	–	–
23	95.4	48.2	–	64.9	32.9	–	95.4	47.7	–	–
Avg.	**91.4**	**53.2**	–	**51.2**	**33.6**	–	**92.3**	**50.0**	–	–

Table 2: Average accuracy per dataset

Dataset nr.	AD mm	AD score	AD rank	AI mm	AI score	AI rank	AT mm	AT score	AT rank	Avg. rank
8	0.63	30.3	–	0.52	32.6	–	0.61	30.9	–	–
9	0.55	20.6	–	0.47	21.6	–	0.54	20.8	–	–
10	0.67	20.2	–	0.45	22.8	–	0.66	20.7	–	–
11	0.68	24.6	–	0.58	25.9	–	0.68	24.6	–	–
12	0.60	21.9	–	0.47	24.7	–	0.58	22.2	–	–
13	0.53	22.5	–	0.50	23.2	–	0.53	22.8	–	–
14	0.55	27.7	–	0.48	29.1	–	0.55	27.8	–	–
15	0.53	22.8	–	0.49	23.5	–	0.53	23.2	–	–
16	0.62	19.4	–	0.54	20.7	–	0.65	18.1	–	–
17	0.74	37.2	–	0.50	40.8	–	0.74	37.2	–	–
18	0.61	23.3	–	0.44	25.0	–	0.61	23.3	–	–
19	0.59	26.6	–	0.57	26.9	–	0.59	26.6	–	–
20	0.61	24.3	–	0.56	25.4	–	0.61	24.3	–	–
21	0.59	17.9	–	0.48	19.4	–	0.57	18.4	–	–
22	0.67	18.9	–	0.61	19.6	–	0.67	18.8	–	–
23	0.52	23.7	–	0.49	24.2	–	0.52	23.7	–	–
Avg.	**0.60**	**23.9**	–	**0.51**	**25.3**	–	**0.60**	**24.0**	–	–

Table 3: Summary

Measure	% / mm min.	% / mm max.	% / mm avg.	score min.	score max.	score avg.	rank min.	rank max.	rank avg.
OV	53.3%	100.0%	91.4%	39.6	100.0	53.2	–	–	–
OF	2.6%	100.0%	51.2%	1.3	100.0	33.6	–	–	–
OT	53.3%	100.0%	92.3%	32.7	100.0	50.0	–	–	–
AD	0.39 mm	1.28 mm	0.60 mm	13.2	56.0	23.9	–	–	–
AI	0.39 mm	0.72 mm	0.51 mm	15.0	66.4	25.3	–	–	–
AT	0.39 mm	1.28 mm	0.60 mm	13.3	56.0	24.0	–	–	–
Total							–	–	–

3 Discussion

We have developed a software tool to interactively track centerlines of bright tubular objects in dark background. The software tool is able to track the centerline of a tubular object in a user-steered mode and the tracking can be inspected and corrected on the fly in the 3D space. The tool has been applied and evaluated in coronary artery tracking. The evaluation shows that the interactive method has very good tracking ability (91.4% overlap) while the tracking accuracy is somewhat moderate (0.60*mm* average distance or 0.83 voxel average distance along x/y directions and 0.75 voxel average distance along z direction). The moderate tracking accuracy of this tool is due to the rescale of the original image volume and the voxel accuracy (compared to sub-voxel accuracy in computational methods) of the points clicked by the users. Another limitation of this method is that the users need to have adequate knowledge about the tubular targets of interests. Otherwise it may lead to incomplete tracking and large measurement errors. For example, the tracking result of vessel 3 in dataset#17 only has an overlap of 53.3% and the average distance is 1.28*mm*. This is because the user has limited anatomic knowledge of the coronary arteries and stops the tracking at very early stage. It is also possible to study the systematic tracking errors introduced by the implementation and limitation of the interactive method and correct the measurements accordingly. The interactive software tool can be further improved in speed, computation efficiency, and sub-voxel accuracy using advanced programming techniques. Since this tool has very good tracking ability, it can be used to obtain rough centerlines of the tubular objects in the pre-processing step, and then advanced image processing techniques can be applied to refine the centerlines to improve the tracking accuracy.

4 Software Distribution

The software tool is freely available for research purpose. Please contact with the author if interested. The following software needs to be installed:

- ImageJ 1.35i or higher version.

Reference

[1] C. Kirbas and F. Quek, "A review of vessel extraction techniques and algorithms," *ACM Computing Surveys (CSUR)*, vol. 36, pp. 81-121, 2004.

[2] A. C. Chung, "Image segmentation methods for detecting blood vessels in angiography," in *The 9th International Conference on Control, Automation, Robotics and Vision (ICARCV)* Singapore, 2006.

[3] W. A. Barrett and E. N. Mortensen, "Interactive live-wire boundary extraction," *Medical Image Analysis*, vol. 1, pp. 331-341, 1996.

[4] E. N. Mortensen and W. A. Barrett, "Interactive segmentation with intelligent scissors," *Graphical Models and Image Processing*, vol. 60, pp. 349-384, 1998.

[5] A. X. Falc˜ao, J. K. Udupa, and F. K. Miyazawa, "An ultra-fast user-steered image segmentation paradigm: Livewire on the fly," *IEEE Transactions on Medical Imaging*, vol. 19, pp. 55-62, 2000.

[6] E. Meijering, M. Jacob, J.-C. F. Sarria, P. Steiner, H. Hirling, and M. Unser, "Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images," *Cytometry*, vol. 58A, pp. 167-176, 2004.

[7] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *1st International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI1998)* Boston, MA, 1998.

[8] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis, "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Medical Image Analysis*, vol. 2, pp. 143-168, 1998.

[9] Y. Zhang, X. Zhou, A. Degterev, M. Lipinski, D. Adjeroh, J. Yuan, and S. T. C. Wong, "Automated neurite extraction using dynamic programming for high-throughput screening of neuron-based assays," *NeuroImage*, vol. 35, pp. 1502-1515, 2007.

[10] C. Metz, M. Schaap, T. v. Walsum, A. v. d. Giessen, A. Weustink, N. Mollet, G. Krestin, and W. Niessen, "3D Segmentation in the Clinic: A Grand Challenge II - coronary artery tracking," in *MICCAI Workshop on 3D Segmentation in the Clinic: A Grand Challenge* New York City, USA, 2008.