Automatic Coronary Tree Modeling

Hüseyin Tek, M. Akif Gülsün, Soizic Laguitton, Leo Grady, David Lesage and Gareth Funka-Lea

July 7, 2008

Imaging and Visualization Department, Siemens Corporate Research, Princeton, NJ 08540

Abstract

In this paper, we present an automatic method for extracting center axis representations (centerlines) of coronary arteries in contrast enhanced (CE)-CT angiography scans. The algorithm first detects the aorta which is used as an initial mask for ostia detection [3]. Second, the ostia locations are detected via a vessel centerline extraction method [5] which tracks the center axis of the coronaries starting from the aorta surface. The full centerline tree of the coronary arteries are computed via the multi-scale medialness-based vessel tree extraction algorithm [5] which starts a tracking process from the ostia locations until all the braches are reached. The centerline extraction algorithm is a graph-based optimization algorithm using multi-scale medialness filters.

Contents

We present an automatic centerline extraction algorithm for representing coronary arteries in CE-CTA data. The proposed algorithm is based on a centerline tracking algorithm using multi-scale medialness filters [5] and an automatic aorta segmentation algorithm [3]. In clinical applications, a vessel modeling algorithm must be able to produce robust and accurate results in *short time e.g.* in few seconds for a single vessel,

The main focus of this paper is to obtain the centerline representation of coronary arteries automatically in a timely manner. Gulsun and Tek [5] have recently developed a method for extracting centerline tree for coronary arteries in less than 30 seconds on a typical PC. This algorithm requires seed placement for each ostia point. In this paper, we propose to use the segmentation algorithm developed by Grady [3] to automatically segment the aorta, permitting us to detect these ostia points automatically. Specifically, the

ostia points are detected by the centerline tracking algorithm which starts from the aorta surface and stops when two major vessels are detected.

This paper is organized as follows as: In Section 1, we briefly describe the automatic aorta segmentation algorithm. Section 2 describes the multi-scale medialness filters which are used in the centerline tracking algorithm. The centerline detection algorithm is first described for a centerline segment in Section 3 and then for the full centerline tree starting from a single seed point in Section 4. In Section 5, we describe the ostia detection algorithm. Finally, Section 6 presents some results.

1 Automatic aorta segmentation

Our aorta segmentation is performed using the fast variant of the isoperimetric algorithm [3] that is applied to segmenting only a mask, presented in [4]. The mask in this case was generated by finding a connected component of voxels with intensity crossing a threshold. This threshold was computed using the initial point given inside the aorta. An initial point in the aorta was determined based on spatial and intensity priors relative to a left ventricle segmentation. The left ventricle was also segmented using the variant of the isoperimetric algorithm in [3], with an initial point given by searching for bright circular regions at the orientation commonly assumed by the left ventricle. This procedure for segmenting the aorta (and left ventricle) was extremely reliable, producing a correct segmentation in 99.4% of our test datasets and requiring a total of 6.2s of computation time.

2 Medialness Measure From 2D Cross-Sectional Models

In this paper, our goal is to obtain the centerline representations of vessels directly from images without creating a binary vessel mask. Specifically, we propose a novel technique for computing medialness measure which is based on multi-scale cross-sectional vessel modeling. Blood vessels in CTA/MRA have typically circular/elliptic shapes in cross-sectional views even though local variations on them are not too uncommon due to the presence of nearby vessels or pathologies. Ideally, a 2D cross-sectional vessel profile consists of a circular/elliptic bright disk and darker ring around it. Our medialness measure uses this circularity assumption and edge responses obtained from multi-scale filters. Specifically, our medialness response, $m(\vec{x_0})$ at $\vec{x_0}$, is computed from a circle $C(\vec{x_0},R)$ centered at $\vec{x_0}$, with radius R, and is given by

$$m(\vec{x_0}) = \max_R \{\frac{1}{N}\sum_{i=0}^{N-1} E(\vec{x_0} + Ru(2\vec{\pi i}/N))\} \quad (1)$$

where $\vec{u}(\alpha) = sin(\alpha)\vec{u_1} + cos(\alpha)\vec{u_2}$ and $\vec{u_1}$ and $\vec{u_1}$ defines a 2D plane. E measures the normalized edge response which is described below. Krissian *et. al.*, [6] proposed a similar medialness measure where the cross-sectional plane is computed from the eigenvectors of Hessian matrix.

Let us consider a 1-D intensity profile $I(x)$ along a ray $\vec{u_\alpha}$ on a cross-sectional plane of a vessel starting from the location $\vec{x_0}$. Suppose that $\vec{x_0}$ is the center of the vessel with a radius R. Then the cross-sectional boundary of the vessel along the ray should occur at $(\vec{x_0} + R\vec{u_\alpha})$ where the gradient of $I(x)$ has a maxima and the second derivative of $I(x)$ has a zero-crossing. We propose to use the gradient, $\nabla_\sigma I(x)$ for measuring responses at vessel boundaries, in which σ corresponds to the spatial scale of the vessel boundary. These gradients are normalized based on their filter sizes, σ to obtain comparable results between different scales. In general, filter sizes are often selected from the size of vessels for computing gradient responses [6], *i.e.*, larger spatial filters for large vessels. It should be noted that vessel scale, namely R and boundary scale, σ

Distributed under Creative Commons Attribution License

are not always related. For example, the boundary of a large vessel can be detected *better* with small size filters when such vessels are surrounded by other bright structures. Similarly, it is possible that small scale vessels can have long diffused boundaries which cannot be accurately detected via small scale filters.

Let us now define the boundary measure along a ray $\vec{u_\alpha}$ at the location x,

$$b(x) = \max_{\sigma}\{(|\nabla_\sigma I(x)|)\} sign(\nabla_\sigma I(x)) \tag{2}$$

where $sign(x)$ is used to distinguish the rising (dark to bright changes) and falling edges (bright to dark changes). Observe that this boundary measure, $\nabla_\sigma I$ is contrast dependent, *i.e.*, it obtains higher values from high contrast vessels and lower values from low contrast vessels, respectively. Unfortunately, vessels may have significant intensity variations on them - especially vessels in MRA and small size vessels in CTA. In addition, boundaries of bones, calcifications in CTA and vessels next to airways can have strong gradients which usually effect the response of medialness filters. We, in fact, believe that medialness responses should be contrast independent, which can be accomplished by normalizing the boundary measure via the highest gradient obtained for different R values along the ray. Mathematically, we define a normalized boundary measure as $\hat{b}(x) = b(x)/b_{max}$ where b_{max} is the maximum falling edge response along $I(x)$ for $x = \{\vec{x_0} + R_{min}\vec{u_\alpha}, .., \vec{x_0} + R_{max}\vec{u_\alpha}\}$ and R_{min} and R_{max} are the minimum and maximum vessel scales, respectively.

Since the size of vessels to be modeled is not known a priori, our method searches for strong edge responses at the different locations along the ray u_α with different R, $R \in [R_{min}, R_{max}]$. However, observe that for large values of R this produces strong boundary responses at locations which are outside the vessel. In general, there should not be any strong rising edge between $\vec{x_0}$ and $\vec{x_0} + R\vec{u_\alpha}$ where the boundary is searched. If there exists such a strong rising edge, it probably means that the point $\vec{x_0}$ is outside the vessel, thus it should have a lower medialness measure. This is accomplished by first computing the maximum rising boundary response up to the location $\vec{x_0} + R\vec{u_\alpha}$ along the ray and then subtracting this value from the response obtained at $\vec{x_0} + R\vec{u_\alpha}$. Based on these modifications, the final edge response along a ray, $\vec{u_\alpha}$, starting from at $\vec{x_0}$, $E(\vec{x_0} + R\vec{u_\alpha})$ is given as

$$E(\vec{\vec{x_0}} + R\vec{u_\alpha}) = \frac{\max(-b(\vec{x_0} + R\vec{u_\alpha}) - \min_{x\in\{\vec{x_0}, \vec{x_0}+R\vec{u_\alpha}\}}(b(x), 0), 0)}{max_{x\in(\vec{x_0}+R_{min}\vec{u_\alpha}, \vec{x_0}+R_{max}\vec{u_\alpha})}(-b(x), 1)} \tag{3}$$

The proposed medialness measure gives strong responses at the center of a vessel and responses drop rapidly towards vessel boundaries and very small responses are obtained in non-vascular areas, Figure 1. Also, the presence of bright structures does not have strong impact on the responses.

3 Local Center-Axis from Graph-Based Optimization

Medialness map of an image alone cannot be used in analyzing vessels without additional post-processing. Instead, they are constructed to obtain vessel center axis representations which are very useful in visualizing vessels in curved (or ribbon - flattened) multi-planar reformatting (MPR), in quantification of pathologies, in navigation during endovascular interventional treatments, etc. Local vessel center axis between two user selected points is often sufficient for analyzing a segment of a vessel quickly in clinical applications. Thus, in this section, we propose a method for extracting such local center axis representations by integrating the medialness map in a discrete optimization framework. Specifically, we seek to obtain a curve $C(s)$ (center axis) between points p_0 and p_1 which travels through the center of a vessel. This problem can be successfully solved by the *minimum-cost* path detection algorithms [1, 7, 8]: Let $E(C)$ be the total energy along a curve C

$$E(C) = \int_\Omega (P(C(s)) + w)ds \tag{4}$$

Distributed under Creative Commons Attribution License

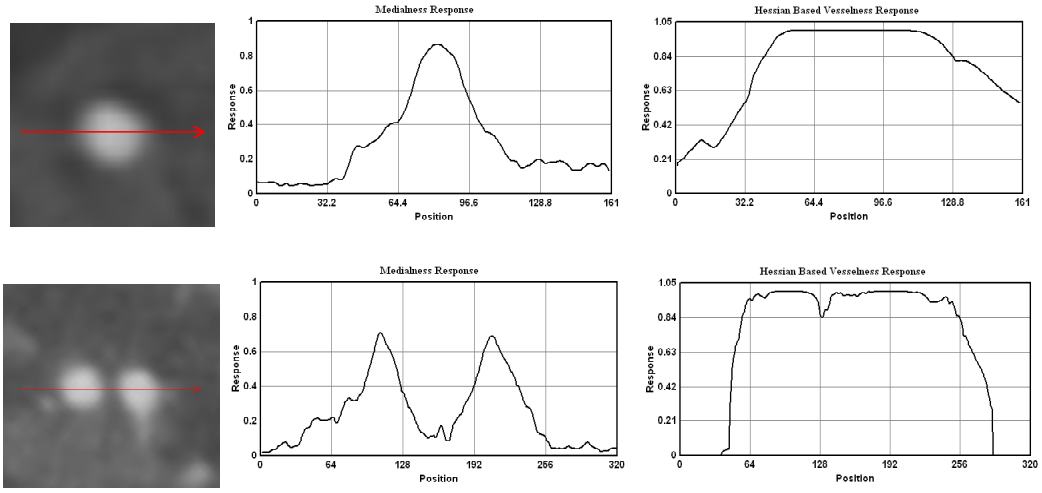

Figure 1: This figure illustrates the medialness responses along a ray on two different examples obtained from our method (middle column) and the Hessian-based method (right column). Observe that unlike Hessian based methods, our technique gives low responses between two nearby vessels.

where $P(C)$ is called potential, w is the regularization term and s is the arch length, *i.e.*, $||C(s)||^2 = 1$. In vessel centerline extraction methods, potential $P(x)$ at x corresponds to the inverse of a medialness measure at that location, namely, $P(x) = \frac{1}{m(x)}$. Let A_{p_0,p_1} represents the set of all curves between p_0 and p_1. The curve with total minimum energy can be computed from the *minimum-accumulative cost*, $\phi(p)$ which measures the minimal energy at p integrated along a curve starting from the point p_0:

$$\phi(p) = \inf_{A_{p_0,p_1}} \{E(C)\} \tag{5}$$

This type of minimization problems has been studied extensively in computer vision for different problems, *e.g.*, segmentation. They are usually solved by either Dijkstra's algorithm [2]. In this paper, we propose to use Dijkstra's algorithm for solving equation (5) in a discrete domain. Specifically, let $G = (N,E)$ be a discrete graph where N and E represent nodes and edges, respectively. The minimum-accumulative cost at the node P_{ij} for a four connected 2D graph is then given by

$$\phi(P_{ij}) = min(\phi(P_{i-1j}) + C^{ij}_{(i-1)j}, \phi(P_{i+1j}) + C^{ij}_{(i+1)j}, \phi(P_{ij-1}) + C^{ij}_{i(j-1)}, \phi(P_{ij+1}) + C^{ij}_{i(j+1)}) \tag{6}$$

where, for example, $C^{ij}_{(i-1)j}$ corresponds to the cost of propagation from point $P_{(i-1)j}$ to P_{ij} which is obtained from the inverse of medialness measure. This above algorithm can be easily implemented by first setting minimum-accumulative cost of all nodes to infinity (or a large value) and then using an explicit discrete front propagation method where propagation always takes places from the minimum value to its neighboring nodes. In our implementation, we use 27-connected lattice in 3D, *i.e.*, diagonal propagations are also included for better accuracy. In addition, the medialness measure is computed orthogonal to the direction of propagation instead of computing at nodes. The discrete path (curve) from a point P_{ij} to source P_0 can then be easily obtained by traversing (backtracking) along the propagation. This algorithm works well even in the presence of nearby vessels, strong calcification and strong contrast change along a vessel and it is computationally efficient. For example, a centerline segment of a coronary artery can be obtained from this algorithm in 3 seconds via two seed placements.

Distributed under Creative Commons Attribution License

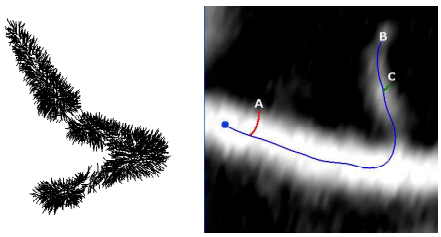

Figure 2: (left) The discrete front and centerlines from these front points. (right) The branch removal process. Observe that front point B is kept while front points A are C are removed.

4 Vessel Tree Modeling

In this section, we extend the local centerline detection algorithm to recover the full vessel tree from a single point, a *source* which may be initialized by an user or another process. Recall that the above algorithm terminates when the front propagation reaches to a *sink*, an end point. When there is no sink point defined for an explicit stopping, the propagation should continue until it reaches to all the branches. The stopping criteria that we choose in our algorithm is based on the medialness measure along a discrete front. Specifically, propagation is forced to stop when the minimum medialness measure along a discrete front at any time drops below a threshold. In our experiments, we found this stopping criteria to be very reliable in clinical applications since our medialness measure is designed to be very low outside vessels. However, the total occlusion cases, where piece of a vessel is totally closed, require starting the propagation on the other side of an occlusion, manually or automatically. We first illustrate how to determine the *correct* vessel centerline tree from the converged propagation.

Suppose that the propagation has converged at time t_f with a set of graph nodes, $F = (P_1, ..., P_K)$, representing a discrete front F, Figure 2. A minimum-cost path between each point P_i of a discrete front, F and the source P_0 can be computed from the minimum accumulative cost map, ϕ, resulting in K different paths. It is obvious that most of these paths are redundant, *i.e.*, a single vessel branch should represented by a single centerline or a single front point. In addition, the existence of a vessel branch can be determined by its length, L_B and its approximate radius, R_B along its centerline, C, *i.e.*, $L_B >> R_B$ [1]. Let us illustrate the basic idea of selecting one centerline for each vessel branch via an example, in Figure 2b which depicts three points A, B, C on a vessel boundary and their corresponding minimum-cost paths. It is clear that the point B with its path C_B represents a branch while the front point A does not since the length of its path is similar to its radius. The front point C may be considered as representing a vessel branch since the length of its minimal path to the source P_0 is significant relative to its average radius. However, the path C_B represents the vessel branch better than the path C_C starting from C. These observations suggest that a front point with the longest path represents a vessel branch better when there are several front points on the same vessel boundary, which is the case after stopping the propagation. This can be implemented very efficiently with the following algorithm:

1. *compute the minimum-cost path C_i and the length L_i for each point P_i in the discrete front set F.*
2. *compute the average radius, R_C along the each path C_i from the scale information contained in the medialness filters.*
3. *order the paths based on their length and store them in a queue, Q_C,* i.e., *maximum is on top.*

[1] The length of a centerline, C, is given by $L_C = \int_C ds$ where s is the arc length.

Distributed under Creative Commons Attribution License

Figure 3: This figure illustrates the results of coronary arteries obtained from our algorithm. Centerlines are drawn in blue and coronary vessel masks are created by using the scales contained in centerline trees.

4. *continue until the queue, Q_C is empty*
   (a) *select the path C from the top of the queue and remove it from the queue.*
   (b) *recompute the path by backtracking until the source, P_0 or the previously computed path on the minimum-accumulative cost map is encountered*
   (c) *mark the path in the minimum-accumulative cost map during the tracking process*
   (d) *recompute the length of the new path, L_C*
   (e) *set the saliency of the path C or its corresponding front point, P as L_C/R_C*
5. *delete the paths whose saliency is less than a user-defined threshold,*

In our experiments, the saliency threshold is set to 2.0, which means that length of a vessel branch should be two times greater than its average radius along its centerline, otherwise it does not appear to be a significant vessel branch. Figure 3 illustrates some examples of vessel centerline tree for coronary arteries and cerebral vessels and others.

5 Automatic Detection of Ostia Locations

The full automatic centerline tree of the coronary arteries can be obtained by first detecting the ostia points automatically and then starting the centerline tracking from these locations. In this paper, we propose to use our centerline tracking algorithm and the aorta mask to obtain these ostia points. Specifically, a front propagation starts from the aorta surface mask and propagates on the discrete grid by minimizing the accumulative costs obtained from the multi-scale medialness filters as described in the previous section. The propagation is forced to stop when the distance from the aorta surface exceeds a threshold e.g., 8 cm. A centerline between the aorta surface and the front point satisfying this distance threshold criteria is computed. The intersection of the centerline and aorta surface is marked as the first ostia point. Second ostia point is detected in a similar fashion. The proposed method has been tested on more than 150 coronary artery data set where the ostia locations are obtained in 6 seconds in average on a 3.2GHz PC. The accuracy of the algorithm was 98.7% .

Distributed under Creative Commons Attribution License

Table 1: Average overlap per dataset

Dataset	OV			OF			OT			Avg.
nr.	%	score	rank	%	score	rank	%	score	rank	rank
8	73.2	40.9	–	57.4	41.7	–	78.2	39.2	–	–
9	90.0	46.5	–	82.3	48.6	–	91.2	45.6	–	–
10	92.3	57.3	–	25.0	12.5	–	92.5	58.7	–	–
11	78.7	40.2	–	33.9	27.5	–	78.7	40.1	–	–
12	89.3	46.1	–	5.1	2.6	–	93.2	46.9	–	–
13	87.8	44.6	–	25.1	12.6	–	89.8	45.0	–	–
14	91.0	46.1	–	72.3	51.8	–	92.7	58.8	–	–
15	83.0	52.3	–	73.4	42.6	–	84.0	54.5	–	–
16	90.3	49.8	–	67.9	46.8	–	94.8	59.9	–	–
17	86.3	54.7	–	47.2	37.4	–	86.9	58.1	–	–
18	86.9	51.3	–	67.9	40.5	–	86.9	43.5	–	–
19	90.6	59.3	–	83.7	66.3	–	90.6	57.8	–	–
20	88.5	54.8	–	52.6	31.8	–	88.5	44.4	–	–
21	94.8	56.7	–	47.1	43.9	–	96.9	61.4	–	–
22	95.3	48.0	–	62.3	31.1	–	96.2	48.1	–	–
23	84.7	42.8	–	6.3	3.1	–	84.7	42.3	–	–
Avg.	**87.7**	**49.5**	**–**	**50.6**	**33.8**	**–**	**89.1**	**50.3**	**–**	**–**

6 Results

The method was evaluated on the 16 CTA datasets of the Testing 1 set of the challenge. Quantitative results are given in Tables 1, 2, 3. Average overlap results (OV and OT) are high (respectively 87.7 and 89.1%), in the order of the inter-observer variability (scores around 50). Average results for OF statistics (before first error) are crippled by a few very low scores. These can be explained by the intrinsic behavior of the method, which, as a minimal path technique, is subject to 'shortcut' effects in the presence of partial or complete occlusions of the vessel. In such cases (dataset 10 for instance), the extracted centerline can temporarily run outside the vessel. This results in a short false positive section, which, given the strict criteria of the challenge, dramatically lowers the OF statistics although the corresponding OV statistics is highly satisfactory. Datasets 12 and 23 suffer from early false positive detections due to the ambiguity in the definition of the starting point and radius of the artery at the ostia. Recall that we use a fully automatic detection of the ostia points. The abnormally low OF scores should not overshadow the fact that even with inaccurate ostia detection, the tracking process robustly extracted the coronaries, as proved by the relatively high OV percentages.

In terms of accuracy, results for true positive points (AI statistics) are satisfactorily in the order of the data resolution. AD and AT statistics are lowered by false positive issues, such as the tracking process 'jumping' into a nearby vein as in dataset 8. Recall that the method evaluated is fully automatic. Our framework allows for easy correction of such cases through minimal user interaction.

Finally, we emphasize the computational efficiency of our approach. Such a criterion is absent from the challenge evaluation but is, in our opinion, essential for the clinical applicability of the method. Fully automatic centerline extraction can be achieved in less than 30 seconds and can be carried out as an offline preprocessing. Furthermore, our framework provides the user with simple tools for correcting and extending automatic results at nearly interactive speeds.

References

[1] T. Deschamps and L.D. Cohen. Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis*, 5(4):281–299, 2001. 3

Distributed under Creative Commons Attribution License

Table 2: Average accuracy per dataset

Dataset	AD			AI			AT			Avg.
nr.	mm	score	rank	mm	score	rank	mm	score	rank	rank
8	4.97	34.3	–	0.39	43.3	–	3.76	36.2	–	–
9	2.17	33.0	–	0.26	36.5	–	2.05	33.4	–	–
10	0.79	31.5	–	0.33	34.1	–	0.78	32.1	–	–
11	3.20	28.0	–	0.40	35.3	–	3.20	28.0	–	–
12	1.18	30.1	–	0.36	33.3	–	0.83	31.8	–	–
13	2.41	31.0	–	0.33	35.1	–	1.87	31.8	–	–
14	1.37	37.7	–	0.32	41.0	–	1.12	38.4	–	–
15	1.88	29.4	–	0.35	34.6	–	1.76	29.8	–	–
16	2.08	30.6	–	0.34	33.2	–	0.78	31.7	–	–
17	1.47	46.1	–	0.38	49.6	–	1.47	46.4	–	–
18	2.27	36.0	–	0.22	41.3	–	2.27	36.0	–	–
19	2.28	39.9	–	0.31	43.7	–	2.28	39.9	–	–
20	1.82	32.5	–	0.40	36.2	–	1.79	32.5	–	–
21	0.67	30.3	–	0.35	31.9	–	0.45	31.3	–	–
22	0.75	33.4	–	0.42	35.0	–	0.68	33.7	–	–
23	1.47	32.1	–	0.34	37.1	–	1.47	32.1	–	–
Avg.	**1.92**	**33.5**	**–**	**0.34**	**37.6**	**–**	**1.66**	**34.1**	**–**	**–**

Table 3: Summary

Measure	% / mm			score			rank		
	min.	max.	avg.	min.	max.	avg.	min.	max.	avg.
OV	53.2%	100.0%	87.7%	28.0	100.0	49.5	–	–	–
OF	0.0%	100.0%	50.6%	0.0	100.0	33.8	–	–	–
OT	53.2%	100.0%	89.1%	28.1	100.0	50.3	–	–	–
AD	0.24 mm	10.15 mm	1.92 mm	17.3	75.3	33.5	–	–	–
AI	0.21 mm	0.62 mm	0.34 mm	24.0	78.8	37.6	–	–	–
AT	0.24 mm	10.15 mm	1.66 mm	18.4	76.4	34.1	–	–	–
Total							**–**	**–**	**–**

[2] E. W. Dijkstra. A note on two problems in connections with graphs. *Numerische Mathematic*, 1:269–271, 1959. 3

[3] Leo Grady. Fast, quality, segmentation of large volumes — Isoperimetric distance trees. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision — ECCV 2006*, volume 3 of *Lecture Notes in Computer Science*, pages 449–462, May 2006. (document), 1

[4] Leo Grady and Eric L. Schwartz. Isoperimetric graph partitioning for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(3):469–475, March 2006. 1

[5] M. A. Gulsun and H. Tek. Robust tree modeling. In *MICCAI*, 2008. (document)

[6] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset. Model based multiscale detection of 3D vessels. In *IEEE Conf. CVPR*, pages 722–727, 1998. 2

[7] Hua Li and Anthony J. Yezzi. Vessels as 4-d curves: Global minimal 4-d paths to extract 3-d tubular surfaces and centerlines. *IEEE Trans. Med. Imaging*, 26(9):1213–1223, 2007. 3

[8] James Alexander Tyrrell, Emmanuelle di Tomaso, Danel Fuja, Ricky Tong, Kevin Kozak, Edward B. Brown, Rakesh Jain, and Badrinath Roysam. Robust 3D modeling of vasculature imagery using superellipsoids. *IEEE Transactions on Medical Imaging*, 2006. 3

Distributed under Creative Commons Attribution License

Automatic Coronary Tree Modeling

Hüseyin Tek, M. Akif Gülsün, Soizic Laguitton, Leo Grady, David Lesage and Gareth Funka-Lea

August 14, 2008

Imaging and Visualization Department, Siemens Corporate Research, Princeton, NJ 08540

Abstract

In this paper, we present an automatic method for extracting center axis representations (centerlines) of coronary arteries in contrast enhanced (CE)-CT angiography scans. The algorithm first detects the aorta which is used as an initial mask for ostia detection. Second, the ostia locations are detected via a vessel centerline extraction method which tracks the center axis of the coronaries starting from the aorta surface. The full centerline tree of the coronary arteries is computed via the multi-scale medialness-based vessel tree extraction algorithm which starts a tracking process from the ostia locations until all the braches are reached. The centerline extraction algorithm is a graph-based optimization algorithm using multi-scale medialness filters.

Contents

We present an automatic centerline extraction algorithm for representing coronary arteries in CE-CTA data. The proposed algorithm is based on a centerline tracking algorithm using multi-scale medialness filters [5] and an automatic aorta segmentation algorithm [3]. In clinical applications, a vessel modeling algorithm must be able to produce robust and accurate results in *short time e.g.* in few seconds for a single vessel,

The main focus of this paper is to obtain the centerline representation of coronary arteries automatically in a timely manner. Gulsun and Tek [5] have recently developed a method for extracting centerline tree for coronary arteries in less than 30 seconds on a typical PC. This algorithm requires seed placement for each ostia point. In this paper, we propose to use the segmentation algorithm developed by Grady [3] to automatically segment the aorta, permitting us to detect these ostia points automatically. Specifically, the

ostia points are detected by the centerline tracking algorithm which starts from the aorta surface and stops when two major vessels are detected.

This paper is organized as follows as: In Section 1, we briefly describe the automatic aorta segmentation algorithm. Section 2 describes the multi-scale medialness filters which are used in the centerline tracking algorithm. The centerline detection algorithm is first described for a centerline segment in Section 3 and then for the full centerline tree starting from a single seed point in Section 4. In Section 5, we describe the ostia detection algorithm. Finally, Section 6 presents some results.

1 Automatic aorta segmentation

Our aorta segmentation is performed using the fast variant of the isoperimetric algorithm [3] that is applied to segmenting only a mask, presented in [4]. The mask in this case was generated by finding a connected component of voxels with intensity crossing a threshold. This threshold was computed using the initial point given inside the aorta. An initial point in the aorta was determined based on spatial and intensity priors relative to a left ventricle segmentation. The left ventricle was also segmented using the variant of the isoperimetric algorithm in [3], with an initial point given by searching for bright circular regions at the orientation commonly assumed by the left ventricle. We have tested the aorta detection algorithm on 150 CTA data sets which are not included in the challange [8]. This procedure for segmenting the aorta (and left ventricle) was extremely reliable, producing a correct segmentation in 99.4% of our test datasets and requiring a total of 6.2s of computation time on a 2.8GHz PC.

2 Medialness Measure From 2D Cross-Sectional Models

In this paper, our goal is to obtain the centerline representations of vessels directly from images without creating a binary vessel mask. Specifically, we propose a novel technique for computing a medialness measure which is based on multi-scale cross-sectional vessel modeling. Blood vessels in CTA/MRA have typically circular/elliptic shapes in cross-sectional views even though local variations on them are not too uncommon due to the presence of nearby vessels or pathologies. Ideally, a 2D cross-sectional vessel profile consists of a circular/elliptic bright disk and darker ring around it. Our medialness measure uses this circularity assumption and edge responses obtained from multi-scale filters. Specifically, our medialness response, $m(\vec{x_0})$ at $\vec{x_0}$, is computed from a circle $C(\vec{x_0}, R)$ centered at $\vec{x_0}$, with radius R, and is given by

$$m(\vec{x_0}) = \max_R \{ \frac{1}{N} \sum_{i=0}^{N-1} E(\vec{x_0} + R\vec{u}(2\pi i/N)) \} \tag{1}$$

where $\vec{u}(\alpha) = sin(\alpha)\vec{u_1} + cos(\alpha)\vec{u_2}$ and $\vec{u_1}$ and $\vec{u_2}$ defines a 2D plane. E measures the normalized edge response which is described below. Krissian *et. al.*, [6] proposed a similar medialness measure where the cross-sectional plane is computed from the eigenvectors of the Hessian matrix.

Let us consider a 1-D intensity profile $I(x)$ along a ray $\vec{u_\alpha}$ on a cross-sectional plane of a vessel starting from the location $\vec{x_0}$. Suppose that $\vec{x_0}$ is the center of the vessel with a radius R. Then the cross-sectional boundary of the vessel along the ray should occur at $(\vec{x_0} + R\vec{u_\alpha})$ where the gradient of $I(x)$ has a maxima and the second derivative of $I(x)$ has a zero-crossing. We propose to use the gradient, $\nabla_\sigma I(x)$ for measuring responses at vessel boundaries, in which σ corresponds to the spatial scale of the vessel boundary. These gradients are normalized based on their filter sizes, σ to obtain comparable results between different scales. In general, filter sizes are often selected from the size of vessels for computing gradient responses [6], *i.e.*,

Distributed under Creative Commons Attribution License

larger spatial filters for large vessels. It should be noted that vessel scale, namely R and boundary scale, σ are not always related. For example, the boundary of a large vessel can be detected *better* with small size filters when such vessels are surrounded by other bright structures. Similarly, it is possible that small scale vessels can have long diffused boundaries which cannot be accurately detected via small scale filters.

Let us now define the boundary measure along a ray $\vec{u_\alpha}$ at the location x,

$$b(x) = \max_{\sigma}\{(|\nabla_\sigma I(x)|)\} sign(\nabla_\sigma I(x)) \tag{2}$$

where $sign(x)$ is used to distinguish the rising (dark to bright changes) and falling edges (bright to dark changes). Observe that this boundary measure, $\nabla_\sigma I$ is contrast dependent, *i.e.*, it obtains higher values from high contrast vessels and lower values from low contrast vessels, respectively. Unfortunately, vessels may have significant intensity variations on them - especially vessels in MRA and small size vessels in CTA. In addition, boundaries of bones, calcifications in CTA and vessels next to airways can have strong gradients which usually effect the response of medialness filters. We, in fact, believe that medialness responses should be contrast independent, which can be accomplished by normalizing the boundary measure via the highest gradient obtained for different R values along the ray. Mathematically, we define a normalized boundary measure as $\hat{b}(x) = b(x)/b_{max}$ where b_{max} is the maximum falling edge response along $I(x)$ for $x = \{\vec{x_0} + R_{min}\vec{u_\alpha}, .., \vec{x_0} + R_{max}\vec{u_\alpha}\}$ and R_{min} and R_{max} are the minimum and maximum vessel scales, respectively.

Since the size of vessels to be modeled is not known a priori, our method searches for strong edge responses at the different locations along the ray u_α with different R, $R \in [R_{min}, R_{max}]$. However, observe that for large values of R this produces strong boundary responses at locations which are outside the vessel. In general, there should not be any strong rising edge between $\vec{x_0}$ and $\vec{x_0} + R\vec{u_\alpha}$ where the boundary is searched. If there exists such a strong rising edge, it probably means that the point $\vec{x_0}$ is outside the vessel, thus it should have a lower medialness measure. This is accomplished by first computing the maximum rising boundary response up to the location $\vec{x_0} + R\vec{u_\alpha}$ along the ray and then subtracting this value from the response obtained at $\vec{x_0} + R\vec{u_\alpha}$. Based on these modifications, the final edge response along a ray, $\vec{u_\alpha}$, starting from at $\vec{x_0}$, $E(\vec{x_0} + R\vec{u_\alpha})$ is given as

$$E(\vec{\vec{x_0}} + R\vec{u_\alpha}) = \frac{\max(-b(\vec{x_0} + R\vec{u_\alpha}) - \min_{x\in\{\vec{x_0}, \vec{x_0}+R\vec{u_\alpha}\}}(b(x), 0), 0)}{max_{x\in(\vec{x_0}+R_{min}\vec{u_\alpha}, \vec{x_0}+R_{max}\vec{u_\alpha})}(-b(x), 1)} \tag{3}$$

The proposed medialness measure gives strong responses at the center of a vessel and responses drop rapidly towards vessel boundaries and very small responses are obtained in non-vascular areas, Figure 1. Also, the presence of bright structures does not have strong impact on the responses.

3 Local Center-Axis from Graph-Based Optimization

The medialness map of an image alone cannot be used in analyzing vessels without additional post-processing. Instead, they are constructed to obtain vessel center axis representations which are very useful in visualizing vessels in curved (or ribbon - flattened) multi-planar reformatting (MPR), in quantification of pathologies, in navigation during endovascular interventional treatments, etc. Local vessel center axis between two user selected points is often sufficient for analyzing a segment of a vessel quickly in clinical applications. Thus, in this section, we propose a method for extracting such local center axis representations by integrating the medialness map in a discrete optimization framework. Specifically, we seek to obtain a curve $C(s)$ (center axis) between points p_0 and p_1 which travels through the center of a vessel. This problem

Distributed under Creative Commons Attribution License

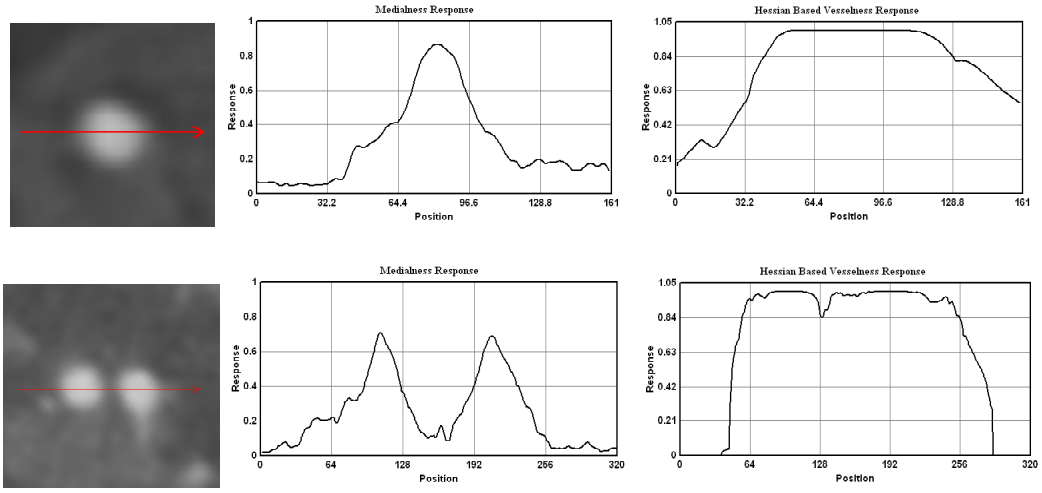

Figure 1: This figure illustrates the medialness responses along a ray on two different examples obtained from our method (middle column) and the Hessian-based method (right column). Observe that unlike Hessian based methods, our technique gives low responses between two nearby vessels.

can be successfully solved by the *minimum-cost* path detection algorithms [1, 7, 9]: Let $E(C)$ be the total energy along a curve C

$$E(C) = \int_{\Omega} (P(C(s)) + w) ds \tag{4}$$

where $P(C)$ is called potential, w is the regularization term and s is the arch length, *i.e.*, $||C(s)||^2 = 1$. In vessel centerline extraction methods, potential $P(x)$ at x corresponds to the inverse of a medialness measure at that location, namely, $P(x) = \frac{1}{m(x)}$. Let A_{p_0,p_1} represents the set of all curves between p_0 and p_1. The curve with total minimum energy can be computed from the *minimum-accumulative cost*, $\phi(p)$ which measures the minimal energy at p integrated along a curve starting from the point p_0:

$$\phi(p) = \inf_{A_{p_0,p_1}} \{E(C)\} \tag{5}$$

This type of minimization problems has been studied extensively in computer vision for different problems, *e.g.*, segmentation. They can solved by the Dijkstra's algorithm [2]. In this paper, we propose to use Dijkstra's algorithm for solving equation (5) in a discrete domain. Specifically, let $G = (N, E)$ be a discrete graph where N and E represent nodes and edges, respectively. The minimum-accumulative cost at the node P_{ij} for a four connected 2D graph is then given by

$$\phi(P_{ij}) = min(\phi(P_{i-1j}) + C^{ij}_{(i-1)j}, \phi(P_{i+1j}) + C^{ij}_{(i+1)j}, \phi(P_{ij-1}) + C^{ij}_{i(j-1)}, \phi(P_{ij+1}) + C^{ij}_{i(j+1)}) \tag{6}$$

where, for example, $C^{ij}_{(i-1)j}$ corresponds to the cost of propagation from point $P_{(i-1)j}$ to P_{ij} which is obtained from the inverse of the medialness measure. This above algorithm can be easily implemented by first setting minimum-accumulative cost of all nodes to infinity (or a large value) and then using an explicit discrete front propagation method where propagation always takes places from the minimum value to its neighboring nodes. In our implementation, we use 27-connected lattice in 3D, *i.e.*, diagonal propagations are also included for better accuracy. In addition, the medialness measure is computed orthogonal to the direction of propagation instead of computing at nodes. The discrete path (curve) from a point P_{ij} to source P_0 can then be easily obtained by traversing (backtracking) along the propagation. This algorithm works well even in the presence of nearby vessels, strong calcification and strong contrast change along a vessel

Distributed under Creative Commons Attribution License

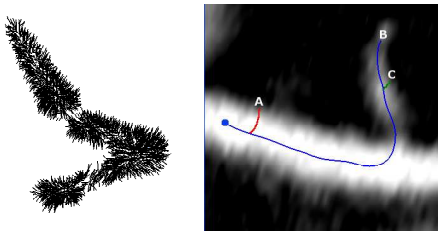

Figure 2: (left) The discrete front and centerlines from these front points. (right) The branch removal process. Observe that front point B is kept while front points A are C are removed.

and it is computationally efficient. For example, a centerline segment of a coronary artery can be obtained from this algorithm in 3 seconds via two seed placements.

4 Vessel Tree Modeling

In this section, we extend the local centerline detection algorithm to recover the full vessel tree from a single point, a *source* which may be initialized by an user or another process. Recall that the above algorithm terminates when the front propagation reaches to a *sink*, an end point. When there is no sink point defined for an explicit stopping, the propagation should continue until it reaches to all the branches. The stopping criterion that we choose in our algorithm is based on the medialness measure along a discrete front. Specifically, propagation is forced to stop when the minimum medialness measure along a discrete front at any time drops below a threshold. In our experiments, we found this stopping criterion to be very reliable in clinical applications since our medialness measure is designed to be very low outside vessels. However, the total occlusion cases, where piece of a vessel is totally closed, require starting the propagation on the other side of an occlusion, manually or automatically. We first illustrate how to determine the *correct* vessel centerline tree from the converged propagation.

Suppose that the propagation has converged at time t_f with a set of graph nodes, $F = (P_1, ..., P_K)$, representing a discrete front F, Figure 2. A minimum-cost path between each point P_i of a discrete front, F and the source P_0 can be computed from the minimum accumulative cost map, ϕ, resulting in K different paths. It is obvious that most of these paths are redundant, *i.e.*, a single vessel branch should be represented by a single centerline or a single front point. In addition, the existence of a vessel branch can be determined by its length, L_B and its approximate radius, R_B along its centerline, C, *i.e.*, $L_B >> R_B$. Let us illustrate the basic idea of selecting one centerline for each vessel branch via an example, in Figure 2b which depicts three points A, B, C on a vessel boundary and their corresponding minimum-cost paths. It is clear that the point B with its path C_B represents a branch while the front point A does not since the length of its path is similar to its radius. The front point C may be considered as representing a vessel branch since the length of its minimal path to the source P_0 is significant relative to its average radius. However, the path C_B represents the vessel branch better than the path C_C starting from C. These observations suggest that a front point with the longest path represents a vessel branch better when there are several front points on the same vessel boundary, which is the case after stopping the propagation. This can be implemented very efficiently with the following algorithm:

1. *compute the minimum-cost path C_i and the length L_i for each point P_i in the discrete front set F.*
2. *compute the average radius, R_C along the each path C_i from the scale information contained in the medialness filters.*

Distributed under Creative Commons Attribution License

Figure 3: This figure illustrates the results of coronary arteries obtained from our algorithm. Centerlines are drawn in blue and coronary vessel masks are created by using the scales contained in centerline trees.

3. *order the paths based on their length and store them in a queue, Q_C,* i.e., *maximum is on top.*
4. *continue until the queue, Q_C is empty*
   (a) *select the path C from the top of the queue and remove it from the queue.*
   (b) *recompute the path by backtracking until the source, P_0 or the previously computed path on the minimum-accumulative cost map is encountered*
   (c) *mark the path in the minimum-accumulative cost map during the tracking process*
   (d) *recompute the length of the new path, L_C*
   (e) *set the saliency of the path C or its corresponding front point, P as L_C/R_C*
5. *delete the paths whose saliency is less than a user-defined threshold,*

In our experiments, the saliency threshold is set to 2.0, which means that length of a vessel branch should be two times greater than its average radius along its centerline, otherwise it does not appear to be a significant vessel branch. Figure 3 illustrates some examples of vessel centerline tree for coronary arteries.

5 Automatic Detection of Ostia Locations

The full automatic centerline tree of the coronary arteries can be obtained by first detecting the ostia points automatically and then starting the centerline tracking from these locations. In this paper, we propose to use our centerline tracking algorithm and the aorta mask to obtain these ostia points. Specifically, a front propagation starts from the aorta surface mask and propagates on the discrete grid by minimizing the accumulative costs obtained from the multi-scale medialness filters as described in the previous section. The propagation is forced to stop when the distance from the aorta surface exceeds a threshold e.g., 8 cm. A centerline between the aorta surface and the front point satisfying this distance threshold criterion is computed. The intersection of the centerline and aorta surface is marked as the first ostia point. Second ostia point is detected in a similar fashion. The proposed method has been tested on more than 150 coronary artery data set where the ostia locations are obtained in 6 seconds in average on a 3.2GHz PC. We have tested the ostia detection algorithm on 150 CTA data sets which are not included in the challange [8]. The accuracy of the algorithm was 98.7% .

Distributed under Creative Commons Attribution License

Table 1: Average overlap per dataset

Dataset	OV			OF			OT			Avg.
nr.	%	score	rank	%	score	rank	%	score	rank	rank
8	73.2	40.9	–	57.4	41.7	–	78.2	39.2	–	–
9	90.0	46.5	–	82.3	48.6	–	91.2	45.6	–	–
10	92.3	57.3	–	25.0	12.5	–	92.5	58.7	–	–
11	78.7	40.2	–	33.9	27.5	–	78.7	40.1	–	–
12	89.3	46.1	–	5.1	2.6	–	93.2	46.9	–	–
13	87.8	44.6	–	25.1	12.6	–	89.8	45.0	–	–
14	91.0	46.1	–	72.3	51.8	–	92.7	58.8	–	–
15	83.0	52.3	–	73.4	42.6	–	84.0	54.5	–	–
16	90.3	49.8	–	67.9	46.8	–	94.8	59.9	–	–
17	86.3	54.7	–	47.2	37.4	–	86.9	58.1	–	–
18	86.9	51.3	–	67.9	40.5	–	86.9	43.5	–	–
19	90.6	59.3	–	83.7	66.3	–	90.6	57.8	–	–
20	88.5	54.8	–	52.6	31.8	–	88.5	44.4	–	–
21	94.8	56.7	–	47.1	43.9	–	96.9	61.4	–	–
22	95.3	48.0	–	62.3	31.1	–	96.2	48.1	–	–
23	84.7	42.8	–	6.3	3.1	–	84.7	42.3	–	–
Avg.	**87.7**	**49.5**	**–**	**50.6**	**33.8**	**–**	**89.1**	**50.3**	**–**	**–**

6 Results

The method was evaluated on the 16 CTA datasets of the Testing 1 set of the challenge [8]. Quantitative results are given in Tables 1, 2, 3. Average overlap results (OV and OT) are high (respectively 87.7 and 89.1%), in the order of the inter-observer variability (scores around 50). Average results for OF statistics (before first error) are crippled by a few very low scores. These can be explained by the intrinsic behavior of the method, which, as a minimal path technique, is subject to 'shortcut' effects in the presence of partial or complete occlusions of the vessel. In such cases (dataset 10 for instance), the extracted centerline can temporarily run outside the vessel. This results in a short false positive section, which, given the strict criteria of the challenge, dramatically lowers the OF statistics although the corresponding OV statistics is highly satisfactory. Datasets 12 and 23 suffer from early false positive detections due to the ambiguity in the definition of the starting point and radius of the artery at the ostia. Recall that we use a fully automatic detection of the ostia points. The abnormally low OF scores should not overshadow the fact that even with inaccurate ostia detection, the tracking process robustly extracted the coronaries, as proved by the relatively high OV percentages.

In terms of accuracy, results for true positive points (AI statistics) are satisfactorily in the order of the data resolution. AD and AT statistics are lowered by false positive issues, such as the tracking process 'jumping' into a nearby vein as in dataset 8. Recall that the method evaluated is fully automatic. Our framework allows for easy correction of such cases through minimal user interaction.

Finally, we emphasize the computational efficiency of our approach. Such a criterion is absent from the challenge evaluation but is, in our opinion, essential for the clinical applicability of the method. Fully automatic centerline extraction can be achieved in less than 30 seconds and can be carried out as an offline preprocessing. Furthermore, our framework provides the user with simple tools for correcting and extending automatic results at nearly interactive speeds.

References

[1] T. Deschamps and L.D. Cohen. Fast extraction of minimal paths in 3d images and applications to virtual endoscopy. *Medical Image Analysis*, 5(4):281–299, 2001. 3

Distributed under Creative Commons Attribution License

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	4.97	34.3	–	0.39	43.3	–	3.76	36.2	–	–
9	2.17	33.0	–	0.26	36.5	–	2.05	33.4	–	–
10	0.79	31.5	–	0.33	34.1	–	0.78	32.1	–	–
11	3.20	28.0	–	0.40	35.3	–	3.20	28.0	–	–
12	1.18	30.1	–	0.36	33.3	–	0.83	31.8	–	–
13	2.41	31.0	–	0.33	35.1	–	1.87	31.8	–	–
14	1.37	37.7	–	0.32	41.0	–	1.12	38.4	–	–
15	1.88	29.4	–	0.35	34.6	–	1.76	29.8	–	–
16	2.08	30.6	–	0.34	33.2	–	0.78	31.7	–	–
17	1.47	46.1	–	0.38	49.6	–	1.47	46.4	–	–
18	2.27	36.0	–	0.22	41.3	–	2.27	36.0	–	–
19	2.28	39.9	–	0.31	43.7	–	2.28	39.9	–	–
20	1.82	32.5	–	0.40	36.2	–	1.79	32.5	–	–
21	0.67	30.3	–	0.35	31.9	–	0.45	31.3	–	–
22	0.75	33.4	–	0.42	35.0	–	0.68	33.7	–	–
23	1.47	32.1	–	0.34	37.1	–	1.47	32.1	–	–
Avg.	**1.92**	**33.5**	**–**	**0.34**	**37.6**	**–**	**1.66**	**34.1**	**–**	**–**

Table 3: Summary

Measure	% / mm min.	% / mm max.	% / mm avg.	score min.	score max.	score avg.	rank min.	rank max.	rank avg.
OV	53.2%	100.0%	87.7%	28.0	100.0	49.5	–	–	–
OF	0.0%	100.0%	50.6%	0.0	100.0	33.8	–	–	–
OT	53.2%	100.0%	89.1%	28.1	100.0	50.3	–	–	–
AD	0.24 mm	10.15 mm	1.92 mm	17.3	75.3	33.5	–	–	–
AI	0.21 mm	0.62 mm	0.34 mm	24.0	78.8	37.6	–	–	–
AT	0.24 mm	10.15 mm	1.66 mm	18.4	76.4	34.1	–	–	–
Total							**–**	**–**	**–**

[2] E. W. Dijkstra. A note on two problems in connections with graphs. *Numerische Mathematic*, 1:269–271, 1959. 3

[3] Leo Grady. Fast, quality, segmentation of large volumes — Isoperimetric distance trees. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision — ECCV 2006*, volume 3 of *Lecture Notes in Computer Science*, pages 449–462, Graz, Austria, May 2006. Springer. (document), 1

[4] Leo Grady and Eric L. Schwartz. Isoperimetric graph partitioning for image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(3):469–475, March 2006. 1

[5] M. A. Gulsun and H. Tek. Robust tree modeling. In *MICCAI*, 2008. (document)

[6] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset. Model based multiscale detection of 3d vessels. In *IEEE Conf. CVPR*, pages 722–727, 1998. 2

[7] Hua Li and Anthony J. Yezzi. Vessels as 4-d curves: Global minimal 4-d paths to extract 3-d tubular surfaces and centerlines. *IEEE Trans. Med. Imaging*, 26(9):1213–1223, 2007. 3

[8] C. Metz, M. Schaap, T. van Walsum, A. van der Giessen, A. Weustink, G. Krestin N. Mollet, and W. Niessen. 3d segmentation in the clinic: A grand challenge ii - coronary artery tracking. *Insight Journal*, 2008. 1, 5, 6

[9] James Alexander Tyrrell, Emmanuelle di Tomaso, Danel Fuja, Ricky Tong, Kevin Kozak, Edward B. Brown, Rakesh Jain, and Badrinath Roysam. Robust 3-d modeling of vasculature imagery using superellipsoids. *IEEE Transactions on Medical Imaging*, 2006. 3

Distributed under Creative Commons Attribution License