Coronary Artery Centerline Tracking Using Axial Symmetries

Release 0.01

Engin Dikici[1], Thomas O'Donnell[2], Leo Grady[2], Randy Setser[3], Richard D. White[1]

August 15, 2008

[1] Dept of Radiology, University of Florida College of Medicine, Jacksonville, FL, 32209
[2] Dept of Imaging and Visualization, Siemens Corporate Research, Princeton, NJ 08540
[3] Dept of Diagnostic Radiology, Cleveland Clinic Foundation, Cleveland, OH, 44195

Abstract

We present a method for tracking a coronary artery centerline given a single user supplied distal endpoint. Briefly, we first isolate the aorta and compute its surface. Next, we apply a novel two-stage Hough-like election scheme to the image volume to detect points which exhibit axial symmetry (vessel centerpoints). From the axial symmetry image a graph is constructed. This graph is searched for the optimal path from the user supplied point to any point on the surface of the aorta. Our technique falls under Challenge 2 of the Coronary Artery Tracking Challenge [1].

Contents

1 Introduction and Related Work **1**

2 Our Approach **3**

3 Results **6**

4 Discussion and Conclusions **7**

A Appendix **7**

1 Introduction and Related Work

Segmentation of coronary arteries in volumetric datasets facilitates several forms of useful analysis. For example, the degree of patency of a stenotic lesion influences treatment decisions. And, the planning of the placement of stents and by-pass graphs benefits greatly from the delineation of the vessel paths and

boundaries. Finally, a by-product of segmentation is enhanced visualizations such as fly-throughs, curved reformats, and multiple vessel flattening.

In this paper, we introduce a novel semi-automatic algorithm for segmenting coronary arteries from volumetric datasets which relies on axial symmetries for the detection of centerpoints of vessels.

First, we automatically extract the aorta from the image volume using a fast variant of the isoperimetric algorithm [10]. Then, we apply a two-stage Hough-like election scheme to the volume which enhances axial symmetries. In stage 1, the image is convolved with a gradient and each point votes on the vessel center. Votes are cast in a conical region (with the tip of the cone on the point and its central axis oriented in the gradient direction). The votes are weighted proportional to the intensity of the gradient at the point and how close it is to the central axis of the cone. This creates the "standard Hough" image. In the same manner, a "radius Hough" image is also created. That is: instead of each vote being a simple weighted yes/no, the weighted radius (the distance between the point and the vote) is recorded. The sum of the weights is additionally recorded so that the "radius Hough" image can be scaled to a "normalized radius Hough" image.

In stage 2, we seek agreement between the "standard Hough" and the "normalized radius Hough" images in a second round of voting. For each point in the volume, we create the same conical region of interest as before and search the "standard Hough" image for the most popular location within the region: the centerpoint candidate. We then compute the distance to the centerpoint candidate from the point. If this distance is in agreement with the value for this location in the "normalized radius Hough" image, the candidate receives a vote. The resulting Hough is normalized to form the Axial Symmetry Image.

Finally, we form a graph based on the Axial Symmetry image and search the optimal path from a user supplied endpoint to any point on the aorta surface (an ostium) to recover the vessel of interest.

Fast, accurate vessel segmentation has been a long sought goal in medical image analysis [12]. The centerpoint detection aspect of our approach has its root in the work of Duda and Hart [2]. From the time Duda introduced the usage of rho-theta parameterization for the detection of non-complex objects (lines, circles), many papers have been published which generalize the idea to more complex shapes [3], [4] or even to 3-D [5]. Iterative voting for the localization of the salience was first introduced by Parvin & Yang [6]. They used radial voting for the detection of the centers of blob shaped objects in cell culture systems and stellar data in 2D and 3D. Their approach bears resemblance to ours. We differ, however, in the employment of the radius image, and, to our knowledge, we are the first to develop an iterative voting scheme for the detection of axially symmetric points.

For the computation of vessel centerlines, Dijksta's shortest path *(DSP)* algorithm [13] is a natural fit because of the applicability of the solution to the problem domain. However, there are many caveats that should be addressed in *DSP* based solutions including but not limited to the following; (1) the scale of the graph should be limited to keep the computational costs low, (2) final centerlines should not hug the borders of the tubular object, (3) surrounding non-targeted objects should not define nodes. Egger formulated an adaptively thresholded input image as a graph where non-eliminated voxels define the nodes [7] (Caveat 1). Bitter built a graph from the coarse approximation of the 3D skeleton in their skeleton extraction algorithm [8] (Caveats 1 and 3). Furthermore, they used the coarse object boundary to node distances as a parameter for setting the edge weights (Caveat 2). Wan defined distance from border *(DFB)* metric based on minimum cost spanning trees for more reliable non-hugging paths of their virtual navigation algorithm [9] (Caveat 2).

2 Our Approach

First, the aorta mask, $T(\vec{x})$, is extracted automatically from the image volume by first finding the left ventricle. We used the fast variant of the isoperimetric algorithm [10] that is applied for segmenting only a mask, presented in [11]. The mask in this case was generated by finding a connected component of voxels with intensities crossing a threshold. This threshold was computed using the initial point given inside the aorta. The initial point in the aorta was determined based on spatial and intensity priors relative to left ventricle segmentation. The left ventricle was also segmented using the variant of the isoperimetric algorithm in [10], 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 all of our test datasets.

Then, the Axial Symmetry volume is created using a two-stage algorithm. Stage 1: Three separate images of the same dimensions as the input volume are created. These are: the "standard Hough" image, $\Psi(\vec{x})$, whose purpose is to store the votes cast for vessel centers, a "radius Hough" image, $\Lambda(\vec{x})$, used to compute the expected vessel cross-sectional radii, and $\Sigma(\vec{x})$, a sum image which will be used to normalize the radius image. To collect the votes, we convolve the image, $I(\vec{x})$, with a cone-shaped kernel.

The kernel, based at the point $\vec{x}$, has angular extent $\alpha_{\max}$ and is oriented in the direction of the image gradient $\frac{\nabla I(\vec{x})}{\|\nabla I(\vec{x})\|}$ with a maximal height, h (see Figure) . (Footnote: We employ the gradient direction in order to orient the cone. However, there are other approaches. For example, the direction could come from the two eigen-values associated with the biggest two eigen-vectors of the Hessian matrix at $\vec{x}$. For performance reasons, we employ the gradient filter.)

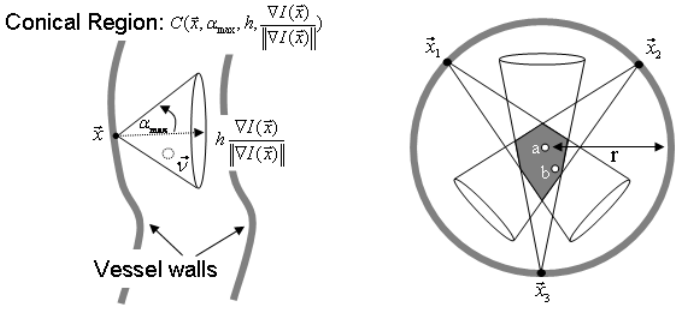

Figure 1 Left: Conical voting region based on a voxel, $\vec{x}$, oriented towards the image gradient. Point $\vec{v}$ lies within the voting region. Right: View of the vessel cross-section with radius r. Voxels are shown voting for a centerpoint of the vessel. The gray region contains the most popular candidates. In the corresponding radius image, $\Lambda(\vec{x})$, point $\vec{a}$, receives weighted increments of value r for all three points $\vec{x}_1, \vec{x}_2, \vec{x}_3$. At point $\vec{b}$, on the other hand, the weighted increments will be different from r (less r than for $\vec{x}_2$ and $\vec{x}_3$, greater than r for $\vec{x}_1$).

A point, $\vec{v}$, within the cone may be described with respect to $\vec{x}$ as: $\vec{v} = R\left(\frac{\nabla I(\vec{x})}{\|\nabla I(\vec{x})\|}\right)\left[rCos\ (\theta)Sin\ (\alpha), rSin\ (\theta)Sin\ (\alpha), rCos\ (\alpha)\right]^T$ where $R\left(\frac{\nabla I(\vec{x})}{\|\nabla I(\vec{x})\|}\right)$ is a rotation matrix aligning the central axis of the cone, $\vec{Z}$, with the image gradient direction; $\alpha:[-\alpha_{\max},\alpha_{\max}]$, is the angle $\vec{v}$ forms with the image gradient vector; $r:[0,\frac{h}{Cos(\alpha_{\max})}]$, is the length of the vector, $\vec{v}$; and $\theta:[0,2\pi]$ serves as a sweep out parameter. These parameters form the conical region, $C(\vec{x},\alpha_{\max},h,\frac{\nabla I(\vec{x})}{\|\nabla I(\vec{x})\|})$.

The kernel K weights a point, $\vec{v}$, by

$$K(\vec{v}(r,\alpha)) = \frac{e^{\frac{-\alpha^2}{2\sigma^2}}}{rCos(\alpha)}$$

Since we assume the centerpoint of the vessel is perpendicular to vessel wall, this weight function penalizes angles deviating from the gradient. Also, the larger the radius, the more pixels to vote for that centerpoint. The $\frac{1}{rCos(\alpha)}$ term compensates for this.

Voting is a two stage process. In the first stage, we create the vote, radius, and sum images and reset their values:

$$\Psi(\vec{x}) = 0, \forall\vec{x},\ \Lambda(\vec{x}) = 0, \forall\vec{x},\ \Sigma(\vec{x}) = 0, \forall\vec{x}$$

Then, for each $\vec{x}$ in $I(\vec{x})$, we align the cone-shaped kernel with the image gradient, $\frac{I(\vec{x})}{\|I(\vec{x})\|}$, sweep out the cone, and increment the vote image with kernel values weighted by the log of the strength of the gradient at $\vec{x}$.

$$\Psi(\vec{x}+\vec{v}) = \Psi(\vec{x}+\vec{v}) + K(\vec{v})\log(|\nabla I(\vec{x})|)$$

In this way, each voxel casts multiple votes for where the centerpoint of the vessel is. On the vessel wall, voxels will cast their votes for a centerpoint which is in the direction (more or less) of the image gradient. Since the image gradient is strong those votes will have stronger weights.

In a similar fashion, votes are collected for the approximate radius of the vessel. Also, a sum of the weights in $\Sigma(\vec{x})$ is stored for normalization.

$$\Lambda(\vec{x}+\vec{v}) = \Lambda(\vec{x}+\vec{v}) + K(\vec{v})|\vec{v}|$$

$$\Sigma(\vec{x}+\vec{v}) = \Sigma(\vec{x}+\vec{v}) + K(\vec{v})$$

Finally, the radius image is divided by $\Sigma(\vec{x})$ to create a normalized radius image $\Lambda'(\vec{x})$.

In stage 2, we create a second vote image, $\mathrm{A}(\vec{x})$, and set its values to zero, $\mathrm{A}(\vec{x}) = 0, \forall \vec{x}$.

For each $\vec{x}$ in $\Psi(\vec{x})$ we find the voxel, $\vec{v}_{\max}$, with the maximum number of votes within the cone-shaped region emanating from $\vec{x}$:

$$\vec{v}_{\max} = \{\vec{v}_i : \Psi(\vec{x} + \vec{v}_i) \geq \Psi(\vec{x}+\vec{v}), \forall\, \vec{v} \in C(\vec{x}, \alpha_{\max}, h, \frac{\nabla I(\vec{x})}{\left\|\nabla I(\vec{x})\right\|})\}$$

If the distance between $\vec{x}$ and the voxel with the maximum votes is approximately equal to the normalized radius,

$$|\, \Lambda'(\vec{v}) - \left\|\vec{v}_{\max} - \vec{x}\right\| \,| < \varepsilon_1$$

then a vote is cast for it in the second vote image

$$\mathrm{A}(\vec{x}+\vec{v}_{\max}) = \mathrm{A}(\vec{x}+\vec{v}_{\max}) + K(\vec{v}_{\max}) \log(\left|\nabla I(\vec{x})\right|)$$

Once all votes are tallied, $\mathrm{A}(\vec{x})$ is normalized to form $\mathrm{A}'(\vec{x})$, the axial symmetry image. High values in $\mathrm{A}'(\vec{x})$ indicate increased likelihood of a point being a centerpoint.

Graph Creation: We then create a graph $G = \{V, E\}$ based on $\mathrm{A}'(\vec{x})$ which will be searched to find the coronary vessels. Vertices, $V = \{B, H, U\}$ where

$B = \{\vec{x} : \nabla T(\vec{x}) > 0\}$, are the aorta surface points.

$H = \{\vec{x} : A'(\vec{x}) > \varepsilon_2\}$, are the axial symmetry voxels over a threshold

$U = \{\vec{u}\}$, is the user supplied distal endpoint.

Edges, E, are created between any two vertices which are less than or equal to 2 centimeters apart in world coordinates (we do this to facilitate bridging occlusions shorter than that length). Edge weights are computed as the line integral (again in world coordinates) between two vertices thru the multiplicative inverse of the axial symmetry volume,

$$w(E_{ij}) = \int_{C_{ij}} \frac{1}{\mathrm{A}'(s)} ds$$

Where C_{ij} is the line segment connecting vertex v_i and v_j.

Vessel Extraction: Finally, we apply Dijkstra's algorithm on G starting from $\vec{u}$, the user supplied endpoint to any point on the aorta surface to recover the vessel of interest. The point on the aorta will be an ostium if the segmentation is correct.

3 Results

We received the following scores when applying our algorithm to the Test dataset.

Table 1: Average overlap per dataset

Dataset	**OV**			**OF**			**OT**			**Avg.**
nr.	%	score	rank	%	score	rank	%	score	rank	**rank**
8	88.7	61.8	–	38.8	40.1	–	91.9	58.6	–	–
9	90.5	56.9	–	50.6	27.5	–	92.2	46.1	–	–
10	90.3	46.6	–	0.0	0.0	–	90.4	45.2	–	–
11	87.8	45.0	–	12.1	16.4	–	87.8	45.0	–	–
12	88.4	45.6	–	16.6	8.3	–	90.7	45.6	–	–
13	97.7	67.6	–	51.1	38.8	–	98.6	61.9	–	–
14	90.5	45.8	–	19.1	12.4	–	90.4	45.2	–	–
15	98.1	76.3	–	25.0	25.0	–	99.1	62.0	–	–
16	91.4	58.5	–	51.2	38.4	–	92.2	71.1	–	–
17	73.3	45.4	–	7.6	4.3	–	73.2	38.7	–	–
18	87.2	54.2	–	20.9	10.7	–	87.2	48.0	–	–
19	99.7	91.1	–	24.8	24.4	–	99.7	51.9	–	–
20	86.5	54.8	–	30.6	16.5	–	86.5	43.4	–	–
21	94.5	58.1	–	23.6	17.7	–	96.5	61.2	–	–
22	94.3	47.6	–	3.3	1.7	–	94.3	47.1	–	–
23	95.4	52.7	–	0.0	0.0	–	95.4	47.7	–	–
Avg.	**90.9**	**56.7**	–	**23.5**	**17.6**	–	**91.6**	**51.2**	–	–

Table 2: Average accuracy per dataset

Dataset	**AD**			**AI**			**AT**			**Avg.**
nr.	mm	score	rank	mm	score	rank	mm	score	rank	**rank**
8	0.61	36.5	–	0.41	38.8	–	0.50	37.6	–	–
9	0.81	22.6	–	0.41	25.0	–	0.76	23.0	–	–
10	0.69	21.4	–	0.46	23.1	–	0.70	20.9	–	–
11	1.02	26.3	–	0.50	28.5	–	1.02	26.3	–	–
12	0.81	20.6	–	0.51	22.5	–	0.77	20.1	–	–
13	0.55	30.8	–	0.37	31.3	–	0.42	30.8	–	–
14	0.76	23.8	–	0.55	25.1	–	0.77	23.6	–	–
15	0.51	22.5	–	0.50	22.8	–	0.51	22.4	–	–
16	0.57	23.6	–	0.44	25.5	–	0.59	22.3	–	–
17	2.86	33.2	–	0.51	38.1	–	2.86	33.2	–	–
18	2.01	21.1	–	0.44	24.0	–	2.01	21.1	–	–
19	0.61	26.1	–	0.61	26.1	–	0.62	26.0	–	–
20	0.96	24.2	–	0.52	27.4	–	0.96	24.2	–	–
21	0.62	21.5	–	0.41	22.5	–	0.58	21.6	–	–
22	0.75	20.8	–	0.56	21.7	–	0.75	20.5	–	–
23	0.56	27.3	–	0.45	28.3	–	0.56	27.3	–	–
Avg.	**0.92**	**25.1**	–	**0.48**	**26.9**	–	**0.90**	**25.0**	–	–

Table 3: Summary

Measure	**% / mm**			**score**			**rank**		
	min.	max.	avg.	min.	max.	avg.	min.	max.	avg.
OV	66.6%	100.0%	90.9%	34.5	100.0	56.7	–	–	–
OF	0.0%	100.0%	23.5%	0.0	100.0	17.6	–	–	–
OT	66.6%	100.0%	91.6%	33.3	100.0	51.2	–	–	–
AD	0.29 mm	7.57 mm	0.92 mm	14.8	64.3	25.1	–	–	–
AI	0.29 mm	0.69 mm	0.48 mm	15.8	65.4	26.9	–	–	–
AT	0.29 mm	7.57 mm	0.90 mm	14.7	64.2	25.0	–	–	–
Total							–	–	–

4 Discussion and Conclusions

The axial symmetry algorithm is computationally expensive. Therefore we apply an automatic cardiac extraction algorithm to isolate the heart. Then, we subdivide the image volume into sub-cubes. Any cube which contains some part of the heart is included in the processing. Even so, the creation of the axial symmetry image typically takes up to five minutes. Once this is calculated, however, extraction of a vessel is done in a few seconds.

Our algorithm has several features. Non-tubular structures will not be both popularly selected in the first voting image, *and* have a consistent radius as measure by the normalized radius image. Therefore, the algorithm will not stray from the vessel. Second, since centerpoint detection is based on gradients rather than intensities we are less prone to failure in cases where the vessel intensity fluctuates. Based on our experiments, we believe we have developed an accurate and robust approach to the vessel segmentation problem.

References

[1] C. Metz, M. Schaap, T. van Walsum, A. van der Giessen, A. Weustink, N. Mollet, G. Krestin and W. Niessen. *3D Segmentation in the Clinic: A Grand Challenge II – Coronary Artery Tracking.* Insight Journal, 2008.

[2] Duda, R. O. and P. E. Hart. *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Comm. ACM, Vol. 15, pp. 11–15 (January, 1972).

[3] D. H. Ballard. *Generalizing the Hough Transform to Detect Arbitrary Shapes*. Pattern Recognition, *v.* 13, n. 2, pp. 111-122, 1981.

[4] Ser P.K., Siu W.C. *A New Generalized Hough Transform for the Detection of Irregular Objects*. JVCIR(6), No. 3, September 1995, pp. 256-264.

[5] K. Khoshelham. *Extending Generalized Hough Transform to Detect 3D Objects in Laser Range Data*. Laser07 (206).

[6] Bahram Parvin, Qing Yang, et.al. *Iterative Voting for Inference of Structural Saliency and Characterization of Subcellular Events*. IEEE TIP07 16(3):615-623

[7] Jan Egger, Zvonimir Mostarkic, Stefan Großkopf, Bernd Freisleben. *A Fast Vessel Centerline Extraction Algorithm for Catheter Simulation*. CBMS 2007: 177-182.

[8] Ingmar Bitter, Arie E. Kaufman, Mie Sato. *Penalized-Distance Volumetric Skeleton Algorithm*. IEEE Trans. Vis. Comput. Graph. 7(3): 195-206 (2001).

[9] Ming Wan, Zhengrong Liang, Qi ke, Lichan Hong, Ingmar Bitter, Arie Kaufman. *Automatic Centerline Extraction for Virtual Colonoscopy*. IEEE Transactions On Medical Imaging, vol.21(12), 2002, pp.1450-1460.

[10] Leo Grady, and Eric L. Schwartz, *Isoperimetric Graph Partitioning for Image Segmentation,* IEEE Trans. Pattern Anal. Mach. Intell. 28(3): 469-475 (2006).

[11] Leo Grady, *Fast, Quality, Segmentation of Large Volumes --- Isoperimetric Distance Trees* ,ECCV (2006), 449—462.

[12] Cemil Kirbas, Francis K. H. Quek. *A Review of Vessel Extraction Techniques and Algorithms*. ACM Comput. Surv. 36(2): 81-121 (2004).

[13] Dijkstra, E. *A Note on Two Problems in Connexion with Graphs*. Numerische Mathematik 1 (1959), 269–271