
An Automatic Seeding Method For Coronary Artery Segmentation and Skeletonization in CTA

Release 0.00

Chunliang Wang¹ and Örjan Smedby¹

July 8, 2008

¹Linköping University, CMIV

Abstract

An automatic seeding method for coronary artery segmentation and skeletonization is presented. The new method includes automatic removal of the rib cage, tracing of the ascending aorta and initial planting of seeds for the coronary arteries. The automatic seeds are then passed on to a “virtual contrast injection” algorithm performing segmentation and skeletonization. In preliminary experiments, most main branches of the coronary tree were segmented and skeletonized without any user interaction.

Latest version available at the [Insight Journal](http://hdl.handle.net/1926/1338) [<http://hdl.handle.net/1926/1338>]
Distributed under [Creative Commons Attribution License](#)

Contents

1	Introduction	2
2	Background	2
2.1	Fuzzy Connectedness	2
2.2	Virtual Contrast Injection	3
3	Algorithm	3
3.1	Rib Cage Removal	3
3.2	Aorta Tracing	4
3.3	Initial Seeding for Coronary Arteries	5
4	Experiments and Results	6
5	Discussion and Conclusion	7

1 Introduction

During the last decade, coronary CT angiography (CTA) has been undergoing a rapid development. Both the spatial and the temporal resolution have improved significantly. This results in increasing amounts of image data generated by each examination. The typical number of images has increased from one optimal phase of around 100 slices to multiple useful phases of 300-500 slices. Finding an efficient method to extract useful information from this amount of data has become a focus of medical image processing researchers. Coronary artery segmentation and centerline extraction are two important directions in coronary CTA post-processing. In our earlier papers [1, 2], we reported an interactive method – “virtual contrast injection” – for combined coronary segmentation and skeletonization. An open source software using this method has been developed to interactively segment the coronary arteries and extract their centerlines [3]. In preliminary clinical experiments, this method has proved to be efficient and accurate. In previous versions of the software, the users need first to reduce the volume by cropping away most ribs with a cubic box, then plant initial seeds at the roots of the left coronary artery (LCA) and the right coronary artery (RCA), and, after the first round segmentation, interactive seed planting needs to be carried out to achieve accurate results. After that, the centerline extraction will be carried out based on the segmentation results. Although the total procedure time may be acceptable, we have noticed that much time was used for the first two steps (where user input is limited and need not be very accurate) and waiting for the first round segmentation.

Based the robust nature of the “virtual contrast injection” algorithm, we have developed an automatic method to eliminate user interaction completely in the first two steps. The new auto-seeding method includes: rib cage removal, tracing of the ascending aorta, and initial seed planting in the coronary arteries. In preliminary experiments on 16 coronary CTA cases, fully automatic segmentation and centerline extraction was achieved on most of them with reasonable accuracy.

2 Background

In this section we will give a brief review of the “virtual contrast injection” algorithm, which is originally based on the fuzzy connectedness theory, and its extension for skeletonization. More details can be found in our earlier reports [2, 4].

2.1 Fuzzy Connectedness

Just as the coronary arteries can be bluntly separated from the heart during surgery because of the varying connectivity of the wall structures, they can be separated in 3D CTA images based on the connectivity of the contrast agent in the vessel lumen, using fuzzy connectedness theory.

In a 3D image, we define a path p joining two voxels, u and v , as a sequence of distinct voxels $u=w_0, w_1, \dots, w_n=v$, such that for each i , $0 \leq i \leq n$, w_{i+1} is a 26-neighbor of w_i . If $f(w_i)$ is the gray-scale value in voxel w_i , the strength of connectedness of p is determined by the darkest voxel along the path:

$$S(p) = \min_{w_i \in p} (f(w_i)) \quad (1)$$

The connectedness between u and v is the strength of the strongest of all paths joining u and v :

$$C(u, v) = \max_{p(u, v)} (S(p)) \quad (2)$$

2.2 Virtual Contrast Injection

In the virtual contrast injection algorithm, the fuzzy connectedness computation is carried out as a competition between multiple seeds. Two or more seed regions, representing different vascular structures, are specified by the user, and the connectedness map from each seed region is calculated. At this stage, the membership of every voxel can be calculated by comparing its connectedness value to different seeds, assigning the voxel to the seed yielding the highest connectivity. Applying this strategy to all voxels in the image, a natural segmentation by “virtual contrast injection” is achieved [4].

In the current implementation, only a single connectedness map needs to be computed since the algorithm is applied to all seeds simultaneously, using an iterative approach. Every voxel is labeled with a color denoting the current membership of the voxel, using a tree structure that allows colors to propagate from the seeds until the whole volume has been labeled. The competing fuzzy connectedness tree (CFCT) structure also serves to solve an ambiguity problem that may arise when propagation from different seeds occurs along the same path [2]. The whole concept can be compared to contrast agents of different color being injected in the seed regions and circulating within the cardiovascular system while competing with each other. The propagation procedure for a seed is somewhat similar to Region Growing, but uses a non-local propagation criterion.

If only one seed, so-called root seed, is included in a coronary artery segment, by searching upwards in the fuzzy connectedness tree, a path can be found from any leaf voxel u till the root seeds in this segment. As every voxel points to its strongest neighbor, this path always snaps onto the ridge of the fuzzy connectivity map. As long as the highest intensity is found in the center of the vessel, tracing from the distal end of a coronary artery to the root seed will actually follow the centerline of the vessel. To avoid distortion of the centerline, a refining step is added. First, a distance map $distance(v)$ is created to indicate the number of points on the path p_{v-s} connecting the voxel v and the root seed s . Then local optimization of the fuzzy connectedness tree is carried out by searching from the distal end, based on the intensity scene of input image and steered by the distance map. The algorithm is described in detail in [2]. In the latest version of our software, further centerline correction is available using a discrete 2D cross-section segmentation to push the centerline onto the gravity center of the cross-section of the vessels.

3 Algorithm

In this section the details of the auto-seeding algorithm will be described. The algorithm includes three major steps: rib cage removal, ascending aorta tracing and initial seed planting in the coronaries.

3.1 Rib Cage Removal

The method for removing the rib cage that we have used is based on a 2D livewire algorithm. Based on the observation that in the upper axial sections through the heart, the heart is almost completely surrounded by the lungs (cf. Fig. 1, left), we have used 2D livewires to close the anterior or posterior mediastinum (cf. Fig. 1, right). The cost function is designed to be able to attract the wire to the interior edge of the sternum and the azygosophageal recess. Due to the nature of connectivity of the pericardia, an extra cost force regarding the distance between the wires in adjacent slices is introduced, to avoid that edges jump from slice to slice. Thanks to this force, the livewire can make reasonable cuts even in the lower slices through the heart, where

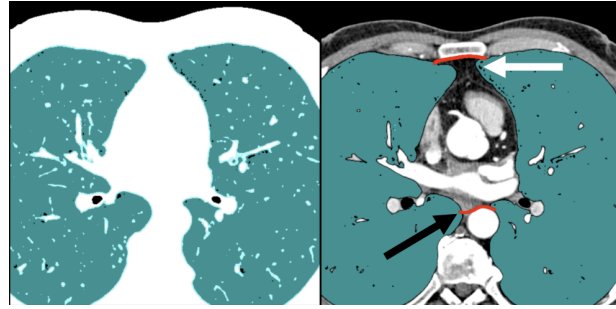


Figure 1: Left: Lung segmentation based on thresholding; Right: red line represents the 2D livewire connecting two lungs in anterior and posterior mediastinum (white arrow: retrosternal space; black arrow: azygoesophageal recess).

there is less lung tissue in the image. The cost function used is described by:

$$C(x) = \alpha \cdot \exp\left(\frac{|f(x) - 100|}{10}\right) + \beta \cdot \exp\left(\frac{50 - g(x)}{10}\right) + \gamma \cdot \exp(d(x)) \quad (3)$$

Here, α , β , γ represents factors for the different forces, which are respectively 1.0, 2.0 and 2.0 in our experiment. $f(x)$ is the intensity at x in the input image; $g(x)$ is the orientation-sensitive gradient magnitude at x , which is positive if the gradient is towards the center of the heart, otherwise negative. The center of the heart area is inherited from the upper slice. $d(x)$ represents the distance from x to the segment border of upper slice.

3.2 Aorta Tracing

Before the tracing of the ascending aorta can start, the location of the aorta needs to be detected. Since the ribs and most of the descending aorta have been removed in the first step, we here use a simple 2D circle detection algorithm (`itk::HoughTransform2DCirclesImageFilter`) to search through the cranial third of the image volume. In each slice, the most prominent three circle-like shapes are recorded, and then all candidate circles are connected to each other if the overlay between circles is more than 90% and the distance between circle centers is less than 3 mm. The rough centerlines of the connected circles are scored based on the circle filter score, the length and the location in the image (right or left relative to the other candidate centerlines). The longest centerline with highest circle score on the right side of the patient is considered to represent the aorta. However, only the cranial end of the centerline is actually used in the following aorta tracing.

In the most cranial slice, a 2D threshold levelset algorithm (`itk::ThresholdSegmentationLevelSetImageFilter`) is applied from the center of the aorta. After the cross-section of the aorta in this slice has been segmented, another slice is chosen by a 1 mm increment in the z-direction of the vessel. The center of gravity from the last cross-section is then passed down to the new slice as a starting seed point for the levelset algorithm. The z-direction in the first slice is assumed to be equal to the z-direction of the patient (cf. Fig. 2, left); later a correction is made every 10 mm in the z-direction by linear regression of the centers of the last ten cross-sections. The 3D cross-section growing will stop if the aortic valve is reached, as detected by the intensity variation in the center of the cross-section becoming significant (cf. Fig. 2, middle).

During the cross-section growing, all voxels in the aortic cross-sections are marked as the root seeds for the coronary arteries. Among these seeds, the virtual contrast injection algorithm will recognize the voxel closest to the coronary artery root as the real root. A barrier will be put in the aortic valve cross-section to

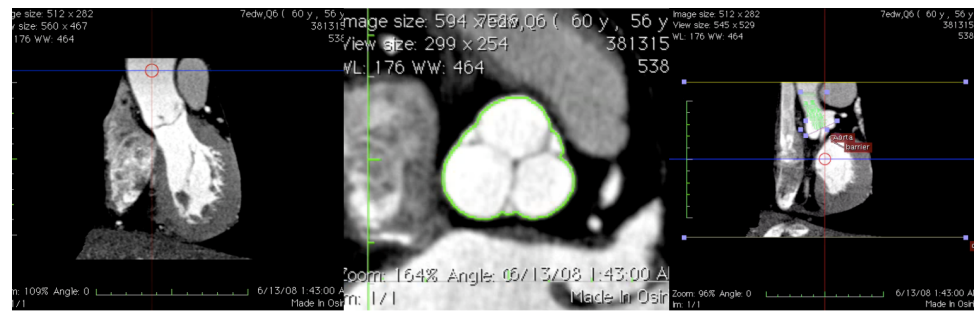


Figure 2: Left: Start slice (blue line of the cross hair) found by 2D circle filter; Middle: Stop slice where the aortic valve was found; Right: Automatic seeds created from aorta tracing (Green: aorta seeds; Yellow lines: seeds for unwanted structures; Purple line: barrier to separate aorta and left ventricle).

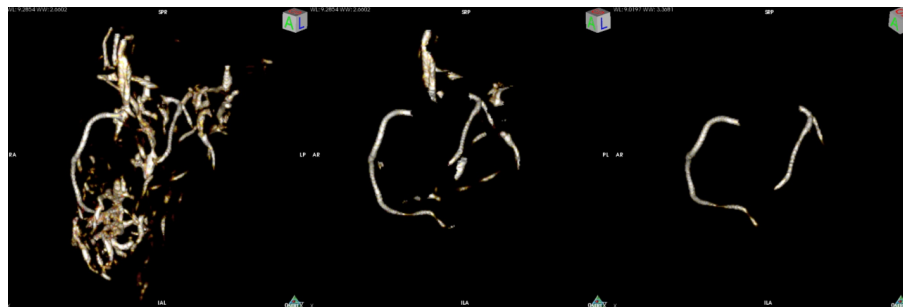


Figure 3: Left: rib-cage removed image filtered with Hessian3DToVesselnessMeasureImageFilter of ITK (sigma=2 mm); Middle: vessel selected by fat mask. Right: The largest connected component on left and right side of the detected aortic valve.

separate ventricles and aorta, and an “other” seed is planted below the aortic valve (cf. Fig. 2, right). Two extra set of “other” seeds are planted in the top and bottom slices of the whole input volume.

3.3 Initial Seeding for Coronary Arteries

In many cases, the aorta seeds are sufficient to segment the coronary arteries. In some cases, though, the coronary arteries may be interrupted due to high intensity artifacts connecting vessels and heart chambers or severe stenosis blocking the contrast connectivity. To correct such errors, extra sets of seeds for those branches are needed. To find some major branches of the coronary arteries, we used here a Hessian- matrix-based vesselness measurement method (`itk::Hessian3DToVesselnessMeasureImageFilter`).

The rib-cage removed data is first resampled to lower resolution with 1 mm isotropic voxels, on which the line-structure filter is applied to detect bright tubular structures. Only a single-scale filter is used, and sigma is set to 2 mm. By thresholding the result, binary shapes of vessel candidates are created (cf. Fig. 3, left). A fat mask is then used to select vessel structures that are surrounded by adipose tissue, as expected for coronary arteries (cf. Fig. 3, middle). The mask is calculated by thresholding the rib-cage removed image data, followed by morphological closure with a 10 mm diameter ball. The largest connected component under the mask on each side of the aortic valve (cf. Fig. 3, right), found in the preceding step, is then marked as the seed for RCA or LCA.

Table 1: Average overlap per dataset

Dataset nr.	OV			OF			OT			Avg. rank
	%	score	rank	%	score	rank	%	score	rank	
8	49.7	26.3	–	17.1	12.6	–	52.5	26.3	–	–
9	85.4	44.1	–	60.1	32.7	–	87.9	43.9	–	–
10	79.5	44.4	–	31.5	15.9	–	83.5	54.2	–	–
11	71.4	36.5	–	49.7	39.1	–	72.9	37.2	–	–
12	77.3	40.0	–	23.4	12.8	–	85.9	43.2	–	–
13	70.8	36.0	–	63.5	49.3	–	74.1	49.6	–	–
14	83.2	42.1	–	48.3	39.6	–	85.2	55.1	–	–
15	88.1	45.0	–	60.6	43.7	–	91.5	58.2	–	–
16	78.1	40.4	–	57.7	41.7	–	83.7	41.8	–	–
17	66.1	37.8	–	8.3	4.7	–	70.4	36.9	–	–
18	89.8	45.7	–	75.7	62.9	–	91.7	60.9	–	–
19	84.7	44.0	–	63.2	38.2	–	84.9	42.5	–	–
20	86.9	48.5	–	44.5	25.2	–	88.7	44.5	–	–
21	87.5	44.5	–	25.0	25.0	–	92.9	59.4	–	–
22	92.1	46.4	–	47.3	23.7	–	96.6	48.3	–	–
23	80.4	40.6	–	62.1	31.5	–	83.2	41.6	–	–
Avg.	79.4	41.4	–	46.1	31.2	–	82.9	46.5	–	–

Table 2: Average accuracy per dataset

Dataset nr.	AD			AI			AT			Avg. rank
	mm	score	rank	mm	score	rank	mm	score	rank	
8	11.56	19.1	–	0.43	39.4	–	11.47	19.7	–	–
9	1.93	25.5	–	0.33	29.5	–	1.77	26.2	–	–
10	5.16	24.8	–	0.38	31.3	–	4.95	25.5	–	–
11	8.36	26.2	–	0.46	35.8	–	8.20	26.7	–	–
12	2.83	20.5	–	0.41	24.7	–	1.77	22.5	–	–
13	11.08	22.4	–	0.40	29.5	–	9.53	23.5	–	–
14	2.86	30.8	–	0.43	36.1	–	2.56	31.5	–	–
15	2.11	29.5	–	0.33	33.0	–	1.32	30.6	–	–
16	5.00	21.4	–	0.38	25.9	–	2.60	22.4	–	–
17	7.32	28.4	–	0.43	38.2	–	6.46	32.8	–	–
18	1.25	28.7	–	0.33	31.1	–	1.04	29.2	–	–
19	2.39	27.2	–	0.42	31.7	–	2.32	27.2	–	–
20	1.35	30.4	–	0.42	34.6	–	0.97	30.8	–	–
21	1.65	26.5	–	0.30	29.6	–	0.93	27.6	–	–
22	1.27	25.0	–	0.39	27.0	–	0.51	26.3	–	–
23	2.60	27.0	–	0.35	33.5	–	2.09	27.6	–	–
Avg.	4.30	25.8	–	0.39	31.9	–	3.66	26.9	–	–

4 Experiments and Results

The autoseeding method was applied as a plugin for OsiriX 3.1 [5], running on a MacBook 2.2Hz Intel CPU, 4G RAM. Sixteen cases provided by the Coronary Artery central lumen line Tracking 2008 website [6] were tested without any user input. Four of the automatic created centerlines for each patient were chosen by either a point A (a point inside the distal part of the vessel) or a point B (a point approximately 3 cm distal of the start point of the centerline), and then compared with the reference centerlines created manually by 3 observers. Out of 64 chosen branches, 50 crossed point A, and all 64 branches crossed point B. The average Overlap with ≥ 1.5 mm vessel (OT) is 82.9%. Detailed results are listed in Table 1-3. The explanation of the evaluation method can be found at [6]. The average processing time is 5 min; detailed efficiency statistics of the new method is listed in Table 4.

Table 3: Summary

Measure	% / mm			score			rank		
	min.	max.	avg.	min.	max.	avg.	min.	max.	avg.
OV	20.0%	99.4%	79.4%	13.6	64.9	41.4	–	–	–
OF	0.0%	100.0%	46.1%	0.0	100.0	31.2	–	–	–
OT	20.0%	100.0%	82.9%	10.0	100.0	46.5	–	–	–
AD	0.25 mm	35.74 mm	4.30 mm	6.9	41.7	25.8	–	–	–
AI	0.24 mm	0.69 mm	0.39 mm	15.2	60.9	31.9	–	–	–
AT	0.25 mm	31.75 mm	3.66 mm	7.4	58.0	26.9	–	–	–
Total							–	–	–

Table 4: Time required(s)

Rib Cage Removal	Aorta Tracing	Initial Seeding	Segmentation	Centerline extraction	total
21.2±1.7	112.7±17.40	88.8±7.4	35.5±2.3	32.3±6.16	290.5±27.7

5 Discussion and Conclusion

With the automatic seeding method, more than 80 percent of major coronary branches were detected without any user interaction. This result is not ideal but may be acceptable if the opportunity of user-steered correction is considered. In the real application environment, further interactive seeding can be carried out by users directly on the automatic segmentation result, which will easily correct the remaining 20 percent of the branches.

An advantage of the CFCT algorithm is that it not only yields the centerlines of the coronary arteries but also a segmentation of the branches. Combined with proper 3D visualization (like MIP or VRT), it will give a good overview of the coronary artery tree and more intuitive 3D information of local lesions.

The newly introduced rib-cage removal function is also helpful for giving a quick overview of the whole heart. Although it is based on 2D segmentation so that the segmentation edges are not smooth in a 3D view the processing time (average 20 s) is acceptable, and with careful parameter tuning it can run even faster (about 5 s) on resampled datasets with lower resolution (1 mm isotropic voxel). Since the purpose of this step is only to cut away big, easy-recognized structures, it is probably not necessary to spend much time on achieving a perfect segmentation, as long as it does not cut off coronary arteries.

The aorta detecting and tracing is the most important step of the whole auto-seeding procedure. In our experience, more than 75 percent of branches can be detected with only the root seeds. Compared with other cross-section-tracing-based algorithms [7] for ascending aorta segmentation, our method does not assume the axis of the aorta to be parallel with the z axis of the patient. Instead, the aortic axis is recalculated every 1 cm, to assure that the cross-section images always follow the aorta, in particular in the aortic valve region, where the axis of the vessel may be almost perpendicular to the z axis of the patient. Stopping at the aortic valve should be reliable, as most coronary CTA data is acquired during the diastolic phase when the valve is closed. With slight modification, we believe it can also be extended to detect the valve during systole.

In the current version, the initial seeds for coronary arteries are less helpful than we expected. The main reason is probably that only the largest connected component of the thresholded vesselness map is used as seed for RCA or LCA. To avoid thresholding, a graph connection method [8], which was used to connect local maxima in distance maps for colon centerline extraction, might be used to create a rough skeleton of the coronary tree. In that way, more branches in the vesselness map could be preserved as seeds and in turn produce more accurate segmentation results.

The accuracy of the centerline extraction is another element that needs to be improved further. Even though the accuracy measurement parameters listed in Table 3, Average distance (AD) and Average distance to the

clinically relevant part of a vessel (AT) [6], may not represent the real performance (as a centerline shorter than the reference line is punished heavily), the parameter Average distance inside vessel (AI), does suggest the centerline is not accurate enough compare to inter-observer accuracy variability. One possible strategy that might be helpful to improve the accuracy is using multi-scale vesselness filter to correct the distortion of the skeleton created from CTCF structure to the ridge of the vesselness measurement, instead of using the Gaussian smoothed intensity scene of the input images as mentioned in section 2.2.

In conclusion, we have presented an automatic seeding method for coronary artery segmentation and skeletonization. In preliminary experiments, it was able to find most main branches without any user interaction. Further work includes reducing the computation time and improving the accuracy of the centerline extraction.

References

- [1] Adam Löfving, Xavier Tizon, Anders Persson, and Örjan Smedby. Virtual contrast injection software tool for selective visualization of vessel structures. *Eur Radiol*, 15:425, 2005. 1
- [2] Chunliang Wang and Örjan Smedby. Coronary artery segmentation and skeletonization based on competing fuzzy connectedness tree. *Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv*, 10(Pt 1):311–8, 2007. 1, 2, 2.2
- [3] Chunliang Wang, Hans Frimmel, Anders Persson, and Örjan Smedby. An interactive software module for visualizing coronary arteries in ct angiography. *International Journal of Computer Assisted Radiology and Surgery*, 3(1):11–18, 2008. 10.1007/s11548-008-0160-6. 1
- [4] Xavier Tizon and Örjan Smedby. Segmentation with gray-scale connectedness can separate arteries and veins in mra. *J Magn Reson Imaging*, 15:438–445, 2002. 2, 2.2
- [5] Antoine Rosset, Luca Spadola, and Osman Ratib. Osirix: an open-source software for navigating in multidimensional dicom images. *J Digit Imaging*, 17(3):205–16, 2004. 4
- [6] Michiel Schaap, Coert Metz, Theo van Walsum, and Wiro Niessen. Coronary artery tracking challenge(<http://cat08.bigr.nl/document/description.pdf>), 2008. 4, 5
- [7] A. Hennemuth, T. Boskamp, D. Fritz, C. Kuhnel, S. Bock, D. Rinck, M. Scheuering, and H. O. Peitgen. One-click coronary tree segmentation in ct angiographic images. *International Congress Series*, 1281:317–321, 2005. 5
- [8] Hans Frimmel, Janne Nappi, and Hiroyuki Yoshida. Fast and robust computation of colon centerline in ct colonography. *Med Phys*, 31(11):3046–56, 2004. 5