Coronary Artery Segmentation and Skeletonization Based on Competing Fuzzy Connectedness Tree

Chunliang Wang and Örjan Smedby

CMIV, Linköping University Hospital
SE-58185 Linköping, Sweden
wcl_sd@hotmail.com, orjan.smedby@cmiv.liu.se

Abstract. We propose a new segmentation algorithm based on competing fuzzy connectedness theory, which is then used for visualizing coronary arteries in 3D CT angiography (CTA) images. The major difference compared to other fuzzy connectedness algorithms is that an additional data structure, the connectedness tree, is constructed at the same time as the seeds propagate. In preliminary evaluations, accurate result have been achieved with very limited user interaction. In addition to improving computational speed and segmentation results, the fuzzy connectedness tree algorithm also includes automated extraction of the vessel centerlines, which is a promising approach for creating curved plane reformat (CPR) images along arteries' long axes.

Keywords: segmentation, fuzzy connectedness tree, centerline extraction, skeletonization, coronary artery, CT angiography.

1 Introduction

In the past few years, noninvasive imaging of the coronary arteries has attracted growing interest. Thanks to the development of image acquisition techniques such as 64-slice scanners for CT angiography (CTA), the spatial and temporal resolution and image quality of the volumetric images have improved remarkably. Compared to the rapid development of the image capture technique, however, the visualization techniques used have not evolved correspondingly. Radiologists and cardiologists still largely depend on viewing original slices, oblique multiplanar reformatting (MPR) and curved plane reformatting (CPR) images, sometimes complemented by a slab maximum intensity projection (MIP) image. Panoramic MIP or volume rendering (VRT) images are less helpful for coronary artery disease diagnosis, due to the concealing effect of contrast medium in adjacent heart chambers and great vessels.

A key method to solve this problem is to segment the coronary arteries in the volumetric datasets. However, due to the close anatomic relationship between coronary arteries and heart chambers and resolution limitations in the images, many automated algorithms, which have been successfully utilized in peripheral vessel extraction, such as pattern reorganization techniques and model-based approaches [1], may fail with coronary artery datasets. Although several algorithms specifically designed for the coronaries have been published [2-4], they tend to have limited success with complicated cases, and important details of coronary artery system may

N. Ayache, S. Ourselin, A. Maeder (Eds.): MICCAI 2007, Part I, LNCS 4791, pp. 311–318, 2007.

be lost during the segmentation. Interactive approaches may solve part of this problem but can be very time-consuming. The concept of fuzzy connectedness (greyscale connectedness), which has been proposed to separate arteries and veins in Magnetic resonance angiography (MRA) [5-7], shows great ability to separate two contrast-filled structures from each other. A previous feasibility study showed that this approach can be applied to 3D CTA data to separate the coronary arteries from other contrast-filled structures and thus permit coronary artery visualization in an angiographic mode similar to invasive X-ray angiography [8].

In this paper we propose a new algorithm based on competing fuzzy connectedness theory, which only requires limited interaction by the user. In addition to overcoming several problems of former algorithms [5-7], the new algorithm includes automated extraction of the vessel centerline, which is useful, e.g., for creating CPR images.

2 Method

In this section, we will first give a brief review of fuzzy connectedness theory, and then present the competing fuzzy connectedness tree algorithm. Finally, we will introduce how it can be extended into a skeletonization algorithm.

2.1 Fuzzy Connectedness and Relative Fuzzy Connectedness Theory

Just as the coronary arteries can be bluntly separated from the heart during surgery due to the varying connectivity of the wall structures, they can also be separated in 3D CTA images depending on the connectivity of the contrast agent in the vessel lumen.

In a 3D image, a path p joining two voxels, u and v, is a sequence of distinct points $u = w_0, w_1, \ldots w_{n-1}, w_n = v$, such that for each i, $0 \le i \le n$, w_{i+1} is a 26-neighbor of w_i. Let $g(w_i)$ be the strength that the voxel w_i can contribute to the path. The strength of connectedness of p is determined by the weakest point along the path:

$$S(p) = \min_{w_i \in p}(g(w_i)) \tag{1}$$

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

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

Although a more sophisticated strategy, such as the Fuzzy Affinity function developed by Udupa and collaborators [5] can be used to calculate the contribution function $g(w_i)$ of each voxel, we have chosen, for convenience and simplicity, to use the gray-scale function $f(w_i)$ directly, i.e. $g(w_i) = f(w_i)$, since our research does not focus on the evaluation of the "cost" function.

An example image (Fig. 1A) contains two seeds, *s1* and *s2* within objects *O1* and *O2*, respectively. With the approach above, the degree of connectedness of a pixel u to each seed can be calculated. Then it can be easily decided to which object u should belong by comparing $C(u, s1)$ and $C(u, s2)$. Applying this strategy to all pixels in the image, a natural segmentation by "relative fuzzy connectedness" is achieved [5].

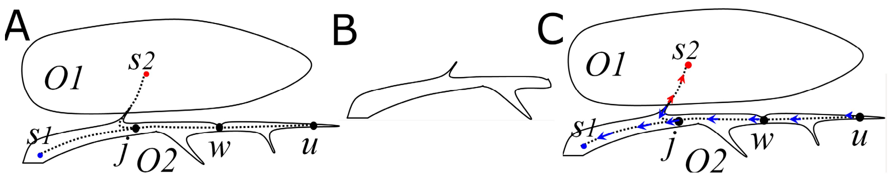

Fig. 1. A: $p_{u\text{-}s1}$ and $p_{u\text{-}s2}$ are the strongest paths from u to $s1$ and $s2$. **B**: segmentation result using relative fuzzy connectedness theory. **C**: competing fuzzy connectedness trees for $p_{u\text{-}s1}$ and $p_{u\text{-}s2}$.

2.2 Shared Path Effect of Relative Fuzzy Connectedness

Although relative fuzzy connectedness can separate most parts of objects correctly, in some cases it will give wrong results. In Fig. 1A, we give an example of such a case. Suppose that $p_{u\text{-}s1}$ and $p_{u\text{-}s2}$ represent the strongest paths from u to $s1$ and $s2$ respectively. Both paths share a segment $p_{u\text{-}j}$, and a point w on $p_{u\text{-}j}$ is the weakest point for both paths. Based on the theory above, all points between w and u will have the same strength of connectedness $g(w)$ to both seeds, and thus the membership of those points will depend on the strategy of the implementation of the algorithm. If *O1* is the object of interest and *O2* is the background, all points between u and w may belong to background (*O2*), even if the points on $p_{j\text{-}w}$ belong to *O1*, as the segmentation result in Fig. 1B shows. With the "SeparaSeed" approach [6], which implemented fuzzy connectedness with a chamfer algorithm [10], the membership of points in $p_{u\text{-}w}$ will depend on which seed will change the "color label" of those points first; i.e., the geometric distance between voxel and seeds will affect the result.

2.3 Competing Fuzzy Connectedness Algorithm Based on Connectedness Tree

To avoid the shared path effect, Udupa and collaborators proposed an Iterative Relative Fuzzy Connectedness algorithm which iteratively refines the competition rules for different objects depending upon the results of the previous iteration [5]. An obvious drawback of that algorithm is the computation time demands caused by the iteration. Here we propose a new algorithm which can avoid the "shared path effect" without adding extra iteration time. The basic idea is to calculate a connectedness tree at the same time as the seeds propagate. As a result, each voxel will point to the neighbor from which it is connected with its "strongest" seed. As shown in Fig. 1C, all points between w and u will be connected with *O1* by pointing to the points on $p_{j\text{-}w}$; thus the potential mistake caused by the shared path will be avoided.

Our competing fuzzy connectedness tree (CFCT) algorithm is described in the following pseudo-code.

Input: A 3D Image $I=(C,f)$, and n seed regions S_j in C
Output: a fuzzy connectedness tree *pointer* and a 3D connectivity map $O=(C, g)$
Auxiliary Data Structures: a 3D array *marker* represents if current voxel should be checked, $N(v)$ denotes the set of neighbor points of v. $N^{+}(v)$ denotes the upper 13 neighbors of v, $N^{-}(v)$ denotes the lower 13 neighbors of v (details in [6])

```
for v∈C, set g(v)=0, pointer(v)=nil, marker(v)=false
for v∈Sj(1•j•n) set g(v)=f(v) and for w∈N(v) set
```

```
   marker(w)=true
3  for v∈ S_j(1•j•n) and S_j is stopping seeds, set g(v)=0
4  repeat
5     for v∈ S_j(1•j•n) from (0,0,0) to (x_max, y_max, z_max)
6        if marker(v)=true
7          find w_max in N+(v) such that g(w_max)= max(g(w))
8          if g(w_max)>g(v)
9             g(v) = min(f(v); g(w_max))
10            pointer(v)= w_max
11            for all w∈ N(v) set marker(w)=true
12         else
13            set marker(v)=false
14    for v∈ S_j(1•j•n) from(x_max, y_max, z_max) to (0,0,0)
15       if marker(v)=true
16         find w_max in N-(v) such that g(w_max)= max(g(w))
17         if g(w_max)>g(v)
18            g(v) = min(f(v); g(w_max))
19            pointer(v)= w_max
20            for all w∈ N(v) set marker(w)=true
21         else
22            set marker(v)=false
23 until no changes in O
```

The extra data structure *marker* limits the comparison within the 26-neighborhood to those voxels whose neighbors have a new optimized connectivity value. This strategy reduces the iteration time, as the number of changed points decreases roughly exponentially with each iteration in the main loop (rows 3-22).

By introducing the array *marker*, the CFCT algorithm has evolved into a variation of Dijkstra's shortest-path algorithm [9]. Here, *marker* is essentially equivalent to the queue *Q*, and setting *marker* (*v*) = true or false, equivalent to a push or pop operation on the queue *Q*. Using an array instead of a queue, multiple duplicated *v* existing in *Q* at the same time are avoided. Memory being allocated beforehand may prevent memory exhaustion during iteration. In practice, the marker array can be merged with the pointer array by using 1 bit of 1 byte, saving even more memory.

A bidirectional scan based on the location of voxels is carried out to decide which voxel should "pop out". This strategy, which was proposed to compute the distance transform [10], helps to accelerate the convergence. In addition, memory is read sequentially, which is faster than accessing memory randomly.

After the construction of the fuzzy connectedness tree, every voxel can get a property *color* by recursively asking its predecessor until a seed is reached. This color map can be used as a segmentation mask, defining a zone containing the anatomically interesting vessels, as well as part of the background

A new kind of seed, the stopping seed, has been introduced to separate the coronary artery from the root of the aorta. A stopping seed is defined as a seed with 0 as initial value (line 3). As it is a seed, its *g* will keep the initial value, so it will never propagate but terminate the propagation of other trees. To cut away the artery totally from the aorta, a stopping plane should be used instead of isolated points.

2.4 Centerline Extraction Algorithm Based on Fuzzy Connectedness Tree

Our centerline extraction algorithm is based on the following observation: if only one seed is included in a coronary artery segment, the CFCT algorithm can always, from any voxel *u* in this segment, find a path connecting *s1* and *u* by searching upwards in the fuzzy connectedness tree (Fig. 2). 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.

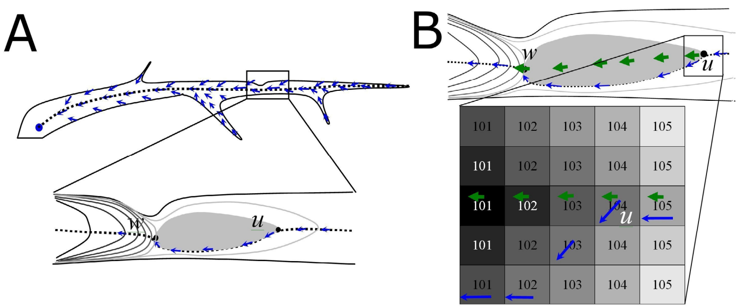

Fig. 2. A: Fuzzy connectedness path is distorted at a flat roof after a stenosis. **B**: Local optimization of the fuzzy connectedness tree, the grey-level represents the intensity in the input image, the number in the grid represents the *distance* to the root seed.

According to fuzzy connectedness theory, the peak intensities will be erased in certain areas of the fuzzy connectivity map. In cases such as Fig. 1, after a weak point *w*, a flat roof will appear (cf. Fig. 2A), because for any voxel *u* after *w*, $C_{u\text{-}s1}$ will be equal to $g(w)$ as long as $g(u) \geq g(w)$. In this area, the fuzzy connectedness path will prefer to follow one edge of the flat region as decided by the scan order.

To avoid distortion of the centerline in such cases, a refining step is added. First, a distance map *distance*(*v*) is created to indicate the number of points on the path $p_{v\text{-}s}$ 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. Suppose voxel *u* is the point we have just found. To decide the next node, rather than using the *pointer*(*u*) directly, we search all neighbors having a distance less than *distance*(*u*) (in Fig. 2B equal to 104), and choose the one with the maximum intensity value in the input image. To reduce noise in the input image, a Gaussian smoothing filter can be used.

In cases with more than one seed in the artery, the connectedness tree is rebuilt after deleting the extra seeds, and all voxels not belonging to a vessel segment are marked as stopping seeds. In the ensuing propagation, only the root seed will have the chance to grow in the coronary artery.

To find the distal endpoint of an artery, a possible strategy is to use the distance map to locate the farthest point, as in [11]. However, the result will be highly dependent on the accuracy of the geometric profile of the segmented artery. When using the strategy above, the distal end will sometimes be located in adjacent tissues with lower intensity. To avoid such mistakes, we use a weight function, when creating the distance map, to assign the step length of a voxel instead of using integer 1. The weight function is defined as: $weight(v) = g(v)/H$, where H is the highest intensity value of the entire input image.

After the longest centerline has been extracted, the centerline of any branch on the main trunk can be found by setting the former centerline as seed and recomputing the weighted distance map. The iteration can be terminated by defining the number of branches desired or by specifying a minimum length of branches (pruning).

3 Results

In 33 clinical coronary CTA datasets (240-450 slices of 0.75mm, 512×512 pixels), we tested both the CFCT algorithm and the older "SeparaSeed" algorithm implemented as plug-ins to Osirix on a Mac G5 (2.5GHz CPU and 2GB RAM). Fig. 3 compares the computation times for each iteration in the main loop for a dataset of 512×512×240 voxels. With the new algorithm, basic seed planting required 2–3 min for an experienced radiologist, the first round segmentation 3–5 min, and interactive seed modification 0–8 min. Using only basic seed planting (with one root seed placed in each coronary artery) resulted in all visible branches being completely segmented in

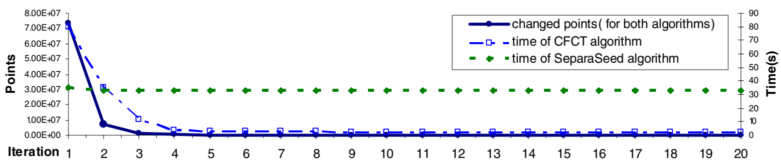

Fig. 3. Comparison of iteration time between "SeparaSeed" and CFCT algorithm

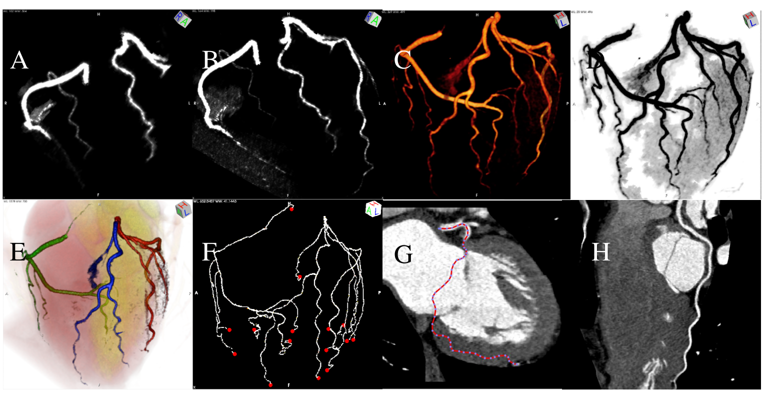

Fig. 4. A: Segmentation result with SeparaSeed. **B**: Result of CFCT algorithm using the same seeds. **C**, **D**: VRT and black-and white inverse MIP images mimicking coronary angiography. **E**: Different vessels shown with different opacity and different color. **F**: A complete skeleton of coronary artery system. **G**, **H**: Centerline extracted with CFCT algorithm and CPR along it.

18 cases, compared to 12 cases when using the same seeds with the "SeparaSeed" algorithm ($p<0.05$; McNemar's Test) (Fig. 4). In total, visually correct centerlines were obtained automatically in 95.3% (262/275) of the visible branches.

4 Discussion

As expected, the CFCT algorithm can run 7–8 times faster than the SeparaSeed algorithm. It can also yield more accurate segmentation results with limited user interaction by avoiding the "shared path effect", thus saving interaction time for planting more seeds. With a VOI definition tool, the whole procedure, including seed planting, segmentation and centerline extraction for the main coronary branches, can be completed within 10–15 min, which may be clinically acceptable. It should be noted that our goal is not an accurate estimation of arterial dimensions; for visualization, suppression of adjacent myocardium is achieved by the rendering algorithm.

The major difference between the CFCT algorithm and Udupa's Relative Fuzzy Connectedness or Iterative Relative Fuzzy Connectedness algorithm is that only one affinity scene is calculated during the whole procedure. By calculating a fuzzy connectedness tree, multiple iterative propagations are avoided. Another speed advantage of the connectedness tree is that previous iteration results can be reused after user modification to recalculate connectedness trees. When new seeds have been planted in the input seed regions, a new round propagation can start running with previous results directly from line 2 of the CFCT algorithm. New trees will grow from those seeds by "snatching voxels" from other trees. If seeds are deleted, the trees arising from those will be removed from the *pointer,* and the connectivity value g of relevant voxels will be reset to 0. After a few iterations, the empty region will be connected to branches extending from nearby trees, resulting in fast convergence. With further improvements, the user may be able to modify seeds at almost interactive speeds.

Compared to centerline extraction algorithms based on Minimum Cost Path Search, an advantage of the fuzzy connectedness tree is that the cost function will not be affected by the Euclidean distance of the "Minimum Cost Path". Since the coronary arteries surround the heart chambers, which may have equal or higher intensity than coronary arteries, the minimum cost path may prefer a short-cut across the narrow barrier (valley) between the heart chamber and arteries, if the cost function is based on intensity. With the fuzzy connectedness tree, this will never happen as long as the strength of the connectedness path between the root seed and distal end is somewhat higher than the lowest intensity of the barrier. Possibly, more accurate centerlines could be obtained by using Hessian-based filters instead of Gaussian filters to correct the distorted centerlines, but in coronary CTA, according to our experience, the latter is sufficient and probably more time-effective.

In summary, this study has shown that the CFCT algorithm is a promising segmentation and skeletonization tool for coronary CTA, requiring only limited interaction by the user. A clinical evaluation of this algorithm will be reported in a separate paper. Other future work includes applying the method to MRA datasets and extending 3D segmentation to 4D datasets for handling dynamic multiple phase CTA.

References

1. Kirbas, C., Quek, F.K.H.: Vessel Extraction Techniques and Algorithms: A Survey. In: Third IEEE Symposium on BioInformatics and BioEngineering, pp. 238–245 (2003)
2. Marquering, H., Dijkstra, J., Koning, P., de, S.B., Reiber, J.H.C.: Towards quantitative analysis of coronary CTA. Int. J. Cardiovasc. Imaging 21, 73–84 (2005)
3. Florin, C., Paragios, N., Williams, J.: Particle Filters. A Quasi-Monte Carlo Solution for Segmentation of Coronaries. In: Medical Image Computing and Computer-Assisted Intervention MICCAI, pp. 246–253 (2005)
4. Khan, M.F., Wesarg, S., Gurung, J., Dogan, S., Maataoui, A., Brehmer, B., Herzog, C., Ackermann, H., Aßmus, B., Vogl, T.: Facilitating coronary artery evaluation in MDCT using a 3D automatic vessel segmentation tool. Eur. Radiol. 16, 1789–1795 (2006)
5. Udupa, J.K., Saha, P.K.: Fuzzy connectedness and image segmentation. Proceedings of the IEEE 91, 1649–1669 (2003)
6. Tizon, X., Smedby, Ö.: Segmentation with gray-scale connectedness can separate arteries and veins in MRA. J. Magn. Reson. Imaging 15, 438–445 (2002)
7. Löfving, A., Tizon, X., Persson, A., Wiklund, G., Smedby, Ö.: Virtual contrast injection – a software tool for selective visualization of vessel structures. Eur. Radiol. 15, 425 (2005)
8. Löfving, A., Tizon, X., Persson, A., Smedby, Ö.: Angiographic visualization of the coronary arteries in computed tomography angiography with virtual contrast injection. The Internet Journal of Radiology 4 (2006)
9. Dijkstra, E.W.: A Note on Two Problems in Connexion with Graphs. Numer. Math. 1, 269–271 (1959)
10. Borgefors, G.: On digital distance transforms in three dimensions. Computer Vision and Image Understanding 64, 368–376 (1996)
11. Bitter, I., Kaufman, A.E., Sato, M.: Penalized-distance volumetric skeleton algorithm. IEEE Transactions on Visualization and Computer Graphics 7, 195–206 (2001)