

A General Framework for Tree Segmentation and Reconstruction from Medical Volume Data

Thomas Bülow, Cristian Lorenz, and Steffen Renisch

Philips Research Laboratories
Roentgenstrasse 24-26, D-22335 Hamburg, Germany

Abstract. Many anatomical entities like different parts of the arterial system and the airway system are of tree-like structure. State of the art medical imaging systems can acquire 3D volume data of the human body at a resolution that is sufficient for the visualization of these tree structures. We present a general framework for the simultaneous segmentation and reconstruction of the abovementioned entities and apply it to the extraction of coronary arteries from multi detector-row CT data. The coronary artery extraction is evaluated on 9 data-sets with known ground truth for the centerlines of the coronary arteries.

1 Introduction

Modern medical imaging devices like multi detector-row computed tomography (MDCT) scanners and magnetic resonance imaging (MRI) devices provide us with high resolution volume data of the human body. Among the anatomic entities with diagnostic relevance that become visible by these means are the vessel trees and the bronchial tree. Depending on the quality of the data wrt. noise, artifacts, spatial resolution etc. tree extraction can be a very challenging task.

In this paper we propose a general framework for the extraction of tree structures from 3D data. Due to the modularity of our approach it can be tailored to specific applications with minor changes. Two implementations of the method – for airways extraction [1], and a preliminary approach to coronary artery segmentation [2] both from MDCT data – have been presented in earlier work. The main contributions of this paper are (i) the presentation of the general framework underlying the abovementioned works (ii) an automated evaluation environment for vessel tree segmentations based on manually extracted ground-truth, and (iii) an extended and improved version of the coronary artery segmentator for MDCT data based on a comparison of several configurations of the general segmentation algorithm. It is crucial that this comparison is not based on visual inspection datasets but rather on the use of an automated evaluation environment.

For the detection of vessels measures of *vesselness* have been designed most of which are based on the eigen-values of the Hesse matrix (second derivative matrix) of the local gray-value structure [4,5]. However, the latter methods are likely to fail at rapid gray-value changes along the vessel which can, e.g., occur in our example application of cardiac MDCT, due to the presence of calcified plaques.

Another class of approaches is based on front propagation. Starting from a seed-point a front is expanded according to a given speed function. This expansion can efficiently be implemented using fast marching and level set methods [3]. Speed functions have been designed according to the abovementioned vesselness measures [7,8]. Besides *segmentation* we are interested in the *reconstruction* of the tree-structure, including proper labeling of connectedness of different segments and parent-child relationships. In the past this has sometimes been dealt with as a separated problem building upon the segmentation results [6].

The framework proposed here allows for the simultaneous segmentation and reconstruction of tree-like structures. Our hierarchical approach consists of three levels ranging from a voxel level to an abstract tree-reasoning level with an intermediate segment level. On each level we have exchangeable modules that allow the adaptation of the methodology to specific applications.

2 The General Framework

There are two major advantages of starting with this general approach rather than dealing with the specific application directly. First, this allows us to adapt the method to different applications easily. Second, it allows to test various configurations for one specific application. This is especially useful since the interplay of different components – each intuitive on its own – can be hard to predict.

2.1 Assumptions

We assume that the tree structure to be segmented stands out against the background according to some image feature $f : (v, I) \mapsto \mathbb{R}$ where $v \in D \subset \mathbb{Z}^3$ is a voxel in the image domain and $I : D \rightarrow \mathbb{R}^N$ is the volumetric data ($N = 1$ for gray scale data). Thus, segments of the tree can in general be segmented locally from the background by a thresholding operation with respect to the local feature value $f(v, I)$. The appropriate threshold can vary within the data volume. Furthermore we assume that segments of the tree roughly comply with certain geometric restrictions, e.g., being elongated cylindrical objects

2.2 The Segmentation Process

We distinguish between *branches* and *segments*, where one branch consists of one or more segments (see Fig. 1). Here, *segments* have merely algorithmic significance. The tree is extracted segment by segment with the main components of the algorithm being *segment initialization*, *segment growing*, and *segment evaluation*. The set of voxels belonging to the tree structure will be denoted by V_{tree} . A segment S is defined as the tuple $S = (V, w, c, p)$ where $V \subset D$ is the set of voxels belonging to S , $w \subset V$ constitutes the *wave front*, $c = (c_1, \dots, c_m)$ with $c_i \in \mathbb{R}^3$ is the centerline of S , and p is a set of parameters used in the expansion and evaluation process.

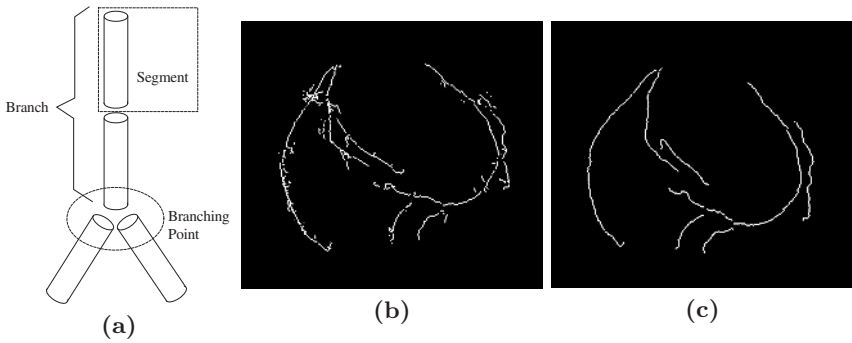


Fig. 1. (a) A *branch* is one component of the tree to be segmented. A branch starts either at the *root* of the tree or at a *branching point* and ends at a *branching point* or at a *termination point*. It can be broken into several *segments*. (b) A raw coronary centerline tree result. (c) The centerline tree after pruning.

Initialization. The segmentation process is started from one or more *seed-voxels* $s_i \in D, i \in \{1, \dots, N\}$. For each seed, a segment $S_i^0 = (s_i, s_i, s_i, p_i)$ is initialized and put into the *segment queue*. The parameter set p_i contains an initial feature threshold t_f based on which acceptable voxels ($f(v, I) > t_f$) are recognized.

The Segment Queue is a sorted list $(S_i^0, S_{i+1}^0, \dots, S_m^0)$ of initialized but yet unprocessed segments. The first segment S_i^0 is removed from the queue, expanded, and its child segments S_{m+1}^0, \dots , if any, are added to the queue. This process is repeated as long as there are segments in the queue or until terminated interactively by the user.

The Wave Front and its Propagation. The prototypical wave front of a segment S can be visualized as a mainly disk-shaped flat subset of V . A propagation function $P^{I,p}$ transforms a wave front w_i into a new wave front w_{i+1} such that for each voxel $v \in w_{i+1}$ the voxel itself or one of its 6-neighbors is contained in w_i . None of the new voxels may be in the set of already segmented voxels V_{tree} , i.e., $\{w_{i+1} \setminus w_i\} \cap V_{tree} = \emptyset$.

Segment Expansion. *Front propagation* is used to expand a segment $S^t = (V, w^t, c, p)$ to $S^{t+1} = (V \cup w^{t+1}, w^{t+1}, (c, center(w^{t+1})), p)$, with $w^{t+1} = P^{I,p}(w^t)$. At this stage the centerline information is built up. We use the notation $P^{I,p}(S^t) = S^{t+1}$. A segment S^t can be reset to its initial state $S^0 = reset(S^t)$.

Segment Evaluation. We evaluate segments in order to determine whether they are part of the desired structure. An application dependent *segment evaluation function* $E(S)$ decides whether a segment should be accepted, rejected or growing should be continued. Depending on the results – especially if S is rejected – the segment can be reset to its initial state and regrown using a different parameter set $p_m = U(p_{m-1}, E(S))$, where U is a parameter update function. Since U can depend on the reason why a segment was rejected we allow for different *reject* and *accept* return values in addition to the *continue* results of E . As soon as the front does no longer consist of a single connected component

the segment is not expanded further, i.e., $E(S) \neq \textit{continue}$. This allows us to keep track of the tree-structure.

Child Segment Initialization. After growing an acceptable segment S one new segment $S_{\textit{new}} = (w_i, w_i, \textit{center}(w_i), \Pi(S))$ per connected front component w_i is initialized. Here Π determines the parameter set of the new segment.

Tree Processing. During the segmentation process the centerline tree is built. Domain knowledge can be used in order to accept/reject segments or whole sub-trees based on the centerline tree. This can be done either during the segmentation or as a postprocessing step pruning the raw centerline tree.

The pseudo-code for the whole process is shown below. Note, that U needs a termination criterion, e.g., a counter that guarantees that $p = U(p, E(S))$ eventually occurs.

```

initialize segments from seed-voxels
put new segments into segment queue
while ( queue not empty )
  retrieve segment  $S$  from queue
  do {  $S = \textit{reset}(S)$ 
      while (  $E(S) == \textit{continue}$  ) {  $S = P^{I,p}(S)$  }
       $p_i = U(p_{i-1}, E(S))$ 
       $p = p_i$ 
    } while (  $p \neq p_{i-1}$  )
   $V_{\textit{tree}} = V_{\textit{tree}} \cup V^S$ 
  initialize child segments and include in segment queue

```

3 Coronary Artery Extraction

In coronary CTA the coronary arteries are imaged by MDCT using an intravenous contrast injection and a special acquisition and reconstruction protocol. Thus, we use the HU-value, i.e., the gray-value as feature $f(v, I) = I(v)$. The extraction of coronary arteries is of use, e.g., in visualization, lumen measurement, computer aided detection (CAD), navigation, and intervention planning. There exists little literature on segmentation of the coronary tree from MDCT data. A previous version of the algorithm presented here has been published [2]. Recently an algorithm for the segmentation of a single vessel with user defined seed-points at the both ends of the vessel has been presented [9].

Difficulties in this domain are that along with the coronary arteries the heart chambers are filled with contrast agent. Due to motion artifacts the delineation between a vessel and a heart chamber may in some cases become almost completely blurred. Furthermore the vessels may be of inhomogeneous gray-value appearance due to calcifications.

The wave-front propagation follows a fast marching approach [3] that simulates the propagation of a wave-front according to the Eikonal equation from geometric optics. We use a binary velocity function with $v = 1$ for voxels above the feature threshold and $v = 0$ else. For the numerical details we refer to [2].

The segment evaluation function for coronary artery segments consists of several modules.

$$E(S) = \begin{cases} reject_{frontsize} & \text{if } \frac{N_{front}}{N_{initial}} > t_{fs} \\ reject_{radius} & \text{else if } \mathcal{R}(S) > t_r \\ reject_{collisions} & \text{else if number of collisions} > t_c \\ E'(S) & \text{else if front split} \\ E'(S) & \text{else if } \frac{\mathcal{L}(S)}{\mathcal{R}(S)} > t_{lr} \\ E'(S) & \text{else if segment is unchanged.} \\ continue & \text{else.} \end{cases}$$

Here $E'(S)$ is a secondary evaluation function that is only computed when the segment is fully grown. It has no *continue* return value. The objective of including E' is that it is only computed once, when the segment is complete, rather than after each expansion step. Here E' is given by

$$E'(S) = \begin{cases} reject_{vesselness} & \text{if } \max_{v \in V} (\mathcal{V}(v, I)) < t_v \\ reject_{splintering} & \text{if } N(\text{frontparts}) > t_s \\ reject_{frontexplosion} & \text{if } \frac{N_{front}}{\mathcal{L}(S)N_{initial}} > t_{fe} \\ accept & \text{else.} \end{cases}$$

The different components in the evaluation functions have the following meaning. Whenever the number of front voxels N_{front} exceeds the number of voxels in the initial front by more than a factor of t_{fs} the segment is rejected. This prevents leakage in many cases. The limitation of the radius $\mathcal{R}(S)$ by t_r has a similar reason. The “number of collisions” is the number of segments in direct vicinity to S . Since, e.g., leakage into the myocardium leads to many collisions of small segments, we limit the number of allowed collisions by t_c . Whenever the front disintegrates or the segment length exceeds its radius by more than t_{lr} the segment expansion is stopped and a final evaluation is done by E' . The same is done in case the segment has not changed in the last expansion step. This happens if there are no voxels above the current gray-value threshold available and consequently the front propagation yields an empty voxel set as new front.

The reason for stopping as soon as the front decomposes is simply that we wish to keep track of the branching structure of the tree. The length/radius ratio is limited in order to save “good” parts of the vessel and not to reject too much whenever the process leaks out of the vessel.

The final evaluation by E' involves a measure of vesselness $\mathcal{V}(v, I)$ that investigates the local gray-value structure and is based on the eigenvalues of the Hessian at different scales [4,5]. Note that we use the maximal vesselness value within the segment rather than the mean vesselness. This helps to account for inhomogeneous appearance of the vessel which leads to low vesselness values for many actual vessel voxels. Also since we use vesselness just as one of several components the threshold t_{vs} is usually chosen to be rather low.

Splintering of the front, i.e., front decomposition into many parts, usually occurs in the presence of noise and can be avoided by raising the gray value

Table 1. 30 vessels compared to ground truth. The table shows the numbers of vessels for which a certain percentage was found for various parameter settings. **(a)** The default parameter setting (see text). **(b)** $N_{nec} = 3$ and $N_{prob} = 1$. **(c)** Number of front components is unlimited. **(d)** Front growth is not restricted. **(e)** Without *front explosion* restriction. **(f)** Maximally 20 segment generations allowed.

		Percentage of vessel found				
		0 – 20%	20 – 40%	40 – 60%	60 – 80%	80 – 100%
Parameter setting	(a)	3	4	3	3	17
	(b)	10	5	3	3	9
	(c)	4	4	5	0	17
	(d)	3	5	9	5	8
	(e)	4	5	3	3	15
	(f)	0	5	8	10	7

threshold. Finally, the segment is rejected if “front explosion” occurs, i.e., if the front grows too fast. This is measured by normalizing the front size ratio $N_{front}/N_{initial}$ by the length of the segment.

The threshold values used in E and E' are all in the parameter set of the segment S . We also include N_{prob} , a counter of *probationary parameter adaptations*, and N_{nec} , a counter of *necessary parameter adaptations*. The parameter update function U for this application acts merely on the three parameters (t_g, N_{prob}, N_{nec}) .

$$\begin{aligned}
 & U((t_g, N_{prob}, N_{nec}), E(S)) \\
 = & \begin{cases} (t_g + \Delta g, N_{prob}, N_{nec} + 1) & \text{if } E(S) = \textit{reject} \text{ and } N_{nec} < \textit{max}_{nec} \\ (t_g - \Delta g, \textit{max}_{prob}, \textit{max}_{nec}) & \text{if } E(S) = \textit{reject} \text{ and } N_{nec} = \textit{max}_{nec} \\ (t_g - \Delta g, N_{prob} + 1, N_{nec}) & \text{if } E(S) = \textit{accept} \text{ and } N_{prob} < \textit{max}_{prob} \\ (t_g, N_{prob}, N_{nec}) & \text{if } E(S) = \textit{accept} \text{ and } N_{prob} = \textit{max}_{prob} \end{cases}
 \end{aligned}$$

For a rejected segment the gray value threshold is increased by Δ_g and N_{nec} is incremented. If the maximum number \textit{max}_{nec} is reached t_g is reduced in order to return to an acceptable state of the segment that might have existed in the previous round. If the segment has been accepted the threshold is reduced probationarily in order to reach distal vessel parts with lower contrast.

4 Results

For evaluation purposes nine cardiac MDCT datasets with manually extracted centerlines of the three main vessels (LAD, LCX, and RCA) were available. For two of the datasets additional marginal and diagonal branches were extracted. This gave us a whole of 30 vessels with known ground truth. For each of the nine datasets two seed-points were placed one in left and right ostium, respectively. The results for each vessel was evaluated by comparison to a ground-truth centerline $c^{true} = (c_0^{true}, \dots, c_N^{true})$. For each point c_i^{true} we checked whether the automatically extracted centerline contains a corresponding point. We allow for a distance of up to 2 mm in order to account for small inaccuracies of the manually and automatically extracted centerlines. Let c^{found} be the found part

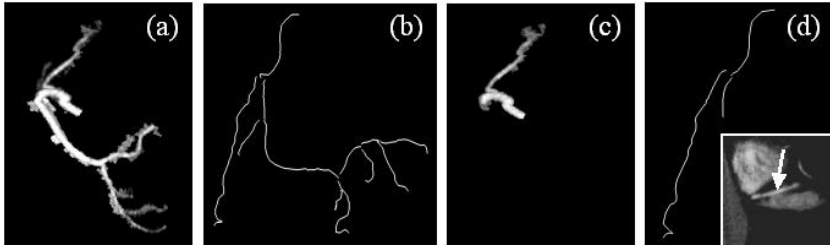


Fig. 2. (a) and (b): MIP and vessel centerlines of a right coronary artery tree segmented using the default parameter setting (see text). (c) and (d): Results for the same dataset at a reduced number of threshold adaptation steps $max_{nec} = 3$, $max_{prop} = 1$. The lower right corner of (d) shows where the segmentation process gets blocked if insufficient threshold adaptation steps are used. In the vicinity of the ventricle the vessel can only be segmented at a higher threshold in this case.

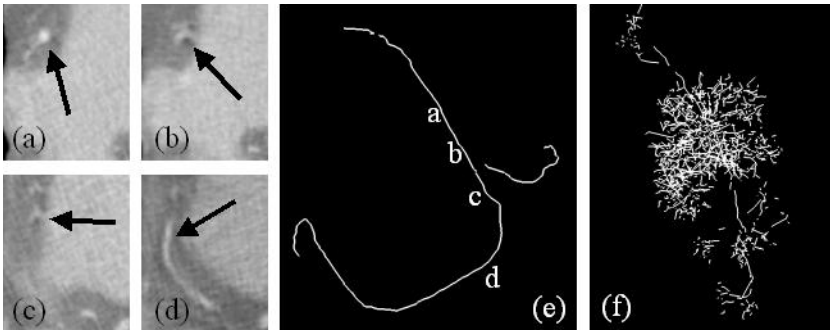


Fig. 3. (a)-(d) Four axial sections from top to bottom. The arrows indicate the RCA. (Clearly visible in (a) and (d), blurred and distorted due to motion artifacts in (c) and (d)). (e) Vessel centerlines after segmentation with default parameter setting. (f) Segmentation without using the vesselness criterion. In this case, severe leakage occurs at blurred portions.

of the centerline corresponding to c^{true} . We judge the result of the segmentation by the ratio $length(c^{found})/length(c^{true})$. We thus identify *true positives*. On the contrary we do not have an automated way to tell for a centerline found by the algorithm but not contained in the ground truth whether it is a *false* or a *true positive*. For this kind of evaluation the results need to be inspected visually. However, the automated results yield some means to rate the quality of an algorithm. For the results of our algorithm please refer to Tab. 1. The default settings used in Tab. 1 (a) were, $t_{fs} = 4$, $t_c = 10$, $t_{lr} = 5$, $t_v = 0.2$, $t_s = 4$, $t_{fe} = 1/mm$, $max_{nec} = 12$, $max_{prob} = 3$, and $\Delta g = 10HU$. From Tab. 1 it can be seen that the default parameter settings as described above yield the best overall results. Reducing the number of threshold adaptation cycles (Tab. 1 (b)) reduces the number of completely found vessels ($> 80\%$) almost by half.

Not restricting front growth (Tab. 1 (d)) leads to more severe leakage and to self blockage, once a segment gets rejected by another criterion. Figures 2 and 3 visualize the effects of different parameter settings (see captions for details).

5 Conclusion and Outlook

It has been shown how the introduced general segmentation scheme can be applied to coronary tree extraction from MDCT data. Due to its modularity we were able to modify the algorithm in order to perform optimally for the given application. This was possible due to the automatic evaluation environment proposed in Sect. 4 in combination with given ground truth for the vessel centerlines. The methodology presented here has also been successfully applied to the extraction of the bronchial tree [1]. In future work we are going to apply the presented approach to other domains, different vessel-trees and other modalities.

Acknowledgement. We thank the clinical science group from Philips Medical Systems CT for helpful discussions and for providing the MSCT data.

References

1. T. Schlathölder, C. Lorenz, I. Carlsen, S. Renisch, T. Deschamps. “Simultaneous Segmentation and Tree Reconstruction of the Airways for Virtual Bronchoscopy” SPIE Conf. on Medical Imaging, Proc. of SPIE Vol. 4684, pp. 103-113 (2002)
2. C. Lorenz, S. Renisch, Th. Schlathölder, Th. Bülow. “Simultaneous Segmentation and Tree Reconstruction of the Coronary Arteries in MSCT Images” SPIE Conference on Medical Imaging, Proceedings of SPIE Vol. 5031, pp. 167-177, (2003)
3. D. Adalsteinsson, J.A. Sethian. “A Fast Level Set Method for Propagating Interfaces”, *J. Comp. Phys.*, 118 (2), pp. 269-277, (1995)
4. C. Lorenz, I.-C. Carlsen, T.M. Buzug, C. Fassnacht and J. Weese. “A Multi-scale Line Filter with Automatic Scale Selection Based on the Hessian Matrix for Medical Image Segmentation”, *Scale-Space '97*, LNCS 1252, pp. 152-163 (1997)
5. A.F. Frangi, W.J. Niessen, K.L. Vincken, M.A. Viergever. “Multiscale Vessel Enhancement Filtering”, *MICCAI 1998*, pp. 130-137 (1998)
6. E. Bullitt, S. Aylward, A. Liu, J. Stone, S.K. Mukherji, Ch. Coffey, G. Gerig, S.M. Pizer. “3D Graph Description of the Intracerebral Vasculature from Segmented MRA and Tests of Accuracy by Comparison with X-ray Angiograms”, *IPMI 99*, Springer, LNCS 1613, pp. 308-321 (1999)
7. O. Wink, W.J. Niessen, B. Verdonck, M.A. Viergever. “Vessel Axis Determination Using Wave Front Propagation Analysis”. *MICCAI 2001*, Springer, LNCS 2208, pp. 845-853 (2001)
8. S. Young, V. Pekar, J. Weese. “Vessel Segmentation for Visualization of MRA with Blood Pool Contrast Agent.” *MICCAI*, Springer, LNCS 2208, pp. 491-498 (2001)
9. S. Wesarg, E. Firlre. “Segmentation of Vessels: The Corkscrew Algorithm”, *SPIE Conference on Medical Imaging, Proc. of the SPIE (2004)* to appear.