Journal of Biomedical Informatics 56 (2015) 395–405

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Automated generation of directed graphs from vascular segmentations ☆

Brian E. Chapman [a,*], Holly P. Berty [b], Stuart L. Schulthies [c]

[a] University of Utah, Department of Radiology, 729 Arapeen Drive, Salt Lake City, UT 84108, United States
[b] University of Pittsburgh, Department of Biomedical Informatics, 5607 Baum Boulevard BAUM 423, Pittsburgh, PA 15206-3701, United States
[c] University of Utah, Department of Mathematics, 155 S 1400 E, Salt Lake City, UT 84112-0090, United States

A R T I C L E I N F O

Article history:
Received 19 February 2015
Revised 1 July 2015
Accepted 2 July 2015
Available online 9 July 2015

Keywords:
Medical imaging
Segmentation
Feature recognition

A B S T R A C T

Automated feature extraction from medical images is an important task in imaging informatics. We describe a graph-based technique for automatically identifying vascular substructures within a vascular tree segmentation. We illustrate our technique using vascular segmentations from computed tomography pulmonary angiography images. The segmentations were acquired in a semi-automated fashion using existing segmentation tools. A 3D parallel thinning algorithm was used to generate the vascular skeleton and then graph-based techniques were used to transform the skeleton to a directed graph with bifurcations and endpoints as nodes in the graph. Machine-learning classifiers were used to automatically prune false vascular structures from the directed graph. Semantic labeling of portions of the graph with pulmonary anatomy (pulmonary trunk and left and right pulmonary arteries) was achieved with high accuracy (percent correct $\geqslant 0.97$). Least-squares cubic splines of the centerline paths between nodes were computed and were used to extract morphological features of the vascular tree. The graphs were used to automatically obtain diameter measurements that had high correlation ($r \geqslant 0.77$) with manual measurements made from the same arteries.

1. Introduction

An important imaging informatics task is to help medical imaging evolve from a primarily qualitative to a primarily quantitative discipline. One aspect of this is extracting quantitative and computable features from the image. Being able to do this in a (nearly) automated method would allow prospective collection of quantitative features with minimal impact on current workflow and the retrospective processing of large numbers of cases archived in institutional PACS. While some quantitative feature extraction can be done directly on the original image, typically, extraction involves identifying subregions of the image that constitute particular objects of interest within the image. Sub-images may be geometric subunits of the image or collections of connected voxels that represent an object or feature of interest. A segmentation of a medical image is a binary labeling of the pixels (2D) or voxels (3D) that constitute the image, where each voxel that is part of the object of interest is given one label (e.g. 1) and all other voxels are given another label (e.g. 0).

After segmentation, the labeled voxels are simply an unordered list, and, depending on the complexity of the segmented object, may need to be ordered into substructures in order to facilitate processing or reasoning. In this paper we present a process for ordering a vascular skeleton into the constituent parts of the vascular tree so that quantitative, computable features can be extracted from the original medical images. Our method uses graph-based techniques to recognize critical features within the skeleton (bifurcations, endpoints, and centerlines). Once the skeletal tree structure is recognized, each voxel within the segmentation is mapped to the appropriate graph edge, facilitating characterization of morphological features of specific vascular segments. For this paper we focus on 3D (volumetric) vascular images and how to structure the original unordered list of prior segmented voxels so that vascular-specific features, such as bifurcation angles or segment diameters, can be automatically extracted. The basis for this structuring is extracting the skeleton of the vascular tree.

The vascular skeleton can be extracted directly from the original (gray scale) image based on the curvature properties of the image, using, for example, ridge traversal [1]. However, these techniques are computationally expensive and may not be ideal for extracting the underlying structures in all cases, since the skeletal extraction is explicitly connected to global models that might be difficult to

☆ This work was supported in part by NIH Grants NHLBI R01 HL087119, U54HL108460, by the Department of Biomedical Informatics at the University of Pittsburgh, and by the Office of the Senior Vice President for Health Sciences at the University of Utah.

* Corresponding author.
E-mail address: brian.chapman@utah.edu (B.E. Chapman).

http://dx.doi.org/10.1016/j.jbi.2015.07.002
1532-0464/

match in the presence of confounding anatomical structures. By extracting the skeleton from a segmented image, a wide variety of segmentation techniques can be used to accurately capture the vascular structures of interest. Given a segmented image, researchers have proposed a variety of means of extracting the skeleton, using, for example, wave propagation [2] or tracing optimal paths using Dijkstra's algorithm (e.g. [3]). However, these methods are sensitive to the cost functions selected for the algorithm and the shortest path through a curve is not at the center of a vessel. Alternatively, parallel thinning techniques are model-free, morphology-based approaches to skeleton extraction [4].

However they are created, automatically generated skeletons will almost inevitably require pruning of spurious centerlines. This pruning may be based on simple features such as centerline length [5] or by trying to recognize non-physiological branching angles [6]. Consequently, we explore using machine learning techniques to automatically prune spurious centerlines.

Given a skeleton, the task remains to recognize endpoints and bifurcations in order to define the underlying vascular structures.

The ability to automatically extract the pulmonary arterial structure has a variety of important implications. First, it could help in the development of computer-aided diagnosis algorithms for pulmonary vascular diseases. For example, these automated techniques could be used to quantify vascular geometry as depicted in volumetric medical images (CT or MR), to assist in the diagnosis of pulmonary arterial hypertension (PAH). While manual arterial measurements have been shown to differentiate PAH subjects from normal subjects [7–10], automated feature extraction for PAH diagnosis would aid radiology workflow. Further, automated measurements would allow a more comprehensive disease characterization based on a fuller assessment of the arterial tree, rather than being limited to a few arteries. Similarly, automated vascular tree extraction could help in the design of computer-aided diagnosis of pulmonary embolism by eliminating non-arterial structures prior to the search for filling defects or by limiting the search to a vascular depth that is deemed clinically significant (e.g. excluding sub segmental arteries). These automated techniques could also be used for large-scale, retrospective image-based analysis for quality assurance purposes or for image-based phenotyping for knowledge discovery in conjunction with additional clinical and genomic information.

We used the Python programming language for our tool development, incorporating unmodified third-party, open source image analysis and visualization libraries, such as the Insight Toolkit [11] and the Visualization Toolkit [12] that were accessed through Python wrappers. These tools and all dependencies are easily installed on multiple platforms. We evaluated our methods on a set of 116 CT pulmonary angiography (CTPA) images.

2. Materials and methods

We begin with a description of our data collection and vascular segmentation followed by the vascular graph generation process, where we detail how we map the voxels from the original segmentation to the graph edges to build a complete representation of the vascular structure. Finally, we describe our machine learning approach for pruning the spurious segments from the graph and thus improving the semantic labeling of our models.

2.1. Data collection

For this study, we used a set of 116 de-identified CTPA exams that had been collected for other studies. All images were acquired at the same institution with diagnostic imaging settings using similar multi-row detector helical CT scanners reconstructed with slice thickness ranging from 0.625 mm to 5.0 mm; the modal thickness was 1.25 mm.

2.2. Vascular segmentation and skeleton generation

Automated segmentation of medical images remains one of the most difficult problems in medical image processing [13]. While machine learning techniques for image segmentation have had a great impact on segmentation of traditional 2D scenes, a similar impact has not been seen in medical imaging where the number of available cases is much smaller and the cost of annotating images for training much higher [14]. Consequently, several researchers have introduced unsupervised learning techniques [14,15]. Nonetheless, within the sub-domain of vascular segmentation, the state-of-the art techniques still rely on rules applied to tubular models of vascular structures, which are very good at extracting the peripheral pulmonary arteries but generally cannot capture the central arteries (pulmonary trunk and left and right pulmonary). These central vessels, which are presumably the most informative for diseases such as pulmonary arterial hypertension, are difficult to segment automatically because of their proximity to confounding structures such as the heart and the aorta. Since our primary interest is in structuring a vascular segmentation rather than in developing novel segmentation algorithms, we used the geometric level set algorithm in ITK-SNAP [16] to generate an initial segmentation of the vasculature followed by hand editing of the resulting segmentation using the paintbrush tool in ITK-SNAP. We felt this would produce segmentations similar to a quasi ideal automated technique. Since this work was motivated in part by the problem of automated characterization of pulmonary hypertension, we focused our segmentations on the central pulmonary arteries (pulmonary trunk and left and right pulmonary arteries).

The level-set segmentation required the user to both provide seed points from where to start the segmentation and either an intensity or gradient mapping that drives the evolution of the segmentation. We chose to use intensity maps because our initial experience was that the intensity maps generally produced less leakage of the segmentation into non-vascular structures. Seed points were placed in the pulmonary trunk and the left and right pulmonary arteries (Fig. 1). The segmentation was allowed to proceed until the pulmonary trunk and left and right pulmonary arteries were fully captured. The amount the segmentation bled into non-vascular structures and how far down the vascular tree the segmentation proceeded varied depending on the characteristics of the particular CTPA exam.

Manual editing of the vascular segmentation was done by reviewing the segmentation on a slice-by-slice basis. Using the paintbrush tool in ITK-SNAP, any observed leakages of the segmentation into non-vascular structures were deleted (Fig. 2). We did not, however, delete any vascular structures beyond the central arteries that were included in the segmentation. Consequently the complexity of the pulmonary arterial tree that was captured varied on a case-by-case basis.

2.2.1. Segmentation preprocessing

We observed that imperfections in the segmentation, such as small holes and surface irregularities, could lead to great difficulty in cleanly generating the vascular graph model skeleton. Consequently, some preprocessing of the segmentation had to be performed prior to generating the skeleton and then the graph. We explored using common binary filters to reduce these imperfections prior to 3D parallel thinning. Specifically, we explored using median filtering [17] and morphological closing [18], alone and in combination. Both filters were implemented using ITK filters (itkBinaryMedianImageFilter and itkBinaryMorphologicalClosingImageFilter respectively) with either $(1,1,1)$ or $(2,2,2)$ kernels.