Encoding Anatomical Tree Priors for Tubular Structure Extraction for Medical Images Analysis

by

Mengliu Zhao

M.Sc., Capital Normal University, 2012
B.Sc., Nankai University, 2007

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

in the School of Computing Science Faculty of Applied Science

© Mengliu Zhao 2020

SIMON FRASER UNIVERSITY
Spring 2020

Copyright in this work rests with the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.
Approval

Name: Mengliu Zhao

Degree: Doctor of Philosophy

Title: Encoding Anatomical Tree Priors for Tubular Structure Extraction for Medical Images Analysis

Examining Committee: Chair: Manolis Savva
                        Assistant Professor

            Ghassan Hamarneh
            Senior Supervisor
            Professor

            Greg Mori
            Supervisor
            Professor

            Richard Zhang
            Internal Examiner
            Professor

            Albert C. S. Chung
            External Examiner
            Professor
            Department of Computer Science and Engineering
            Hong Kong University of Science and Technology

Date Defended: April 2, 2020
Abstract

Vasculatures and airways in the human body contain anatomical trees, which are tree structures with anatomical properties and geometrical attributes. Since anatomical trees are highly involved in clinical procedures such as disease diagnosis and treatment planning, accurate and automatic annotation and analysis of these structures is extremely important.

In this thesis, after an extensive study of existing literature on 3D tubular tree analysis, we introduce novel techniques encoding anatomical tree priors into vasculature and airway extraction.

We present novel features, e.g., features based on multi-modal von Mises-Fisher distribution, for 3D vasculature bifurcation classification using Random Forest classifier. Then we introduce the first work fitting a parametric 3D geometric model to 3D medical image data of pulmonary vasculature for bifurcation localization. To solve the corresponding optimization problem, we present the modified genetic algorithm with tribes niching technique. For encoding the geometrical variability of anatomical trees and their natural sequential root-to-leaf representation, we propose two deep learning models, the TreeNet and LSTM-Tree for predicting branch direction and bifurcation classification during centerline tree tracking. To overcome the myopic visual search involved in most tree tracking processes, we introduce two novel ways to leverage global prior information, by using tree-level statistics within a Bayesian framework and reframing the tree shape into a pictorial structure. Then we encode anatomical tree priors in the clinical task of age-related macular degeneration classification and retinopathy grading by masking a sequential attention within deep network layers.

Keywords: medical imaging; angiography; vasculature; airways; anatomical trees; segmentation; tracking; machine learning; deep learning; tree prior; anatomical knowledge; hybrid methods
Acknowledgements

I would like to thank Dr. Ghassan Hamarneh for the support and guidance of all the years during my PhD study. Thanks Ghassan for his support and guidance – without his help, I wouldn’t imagine to ever finish this thesis. Ghassan has been an outstanding supervisor, showing me the importance of details, the techniques for planning and the forever pursuit to innovation.

I would also like to thank all other members in my examination committee, Dr. Greg Mori, Dr. Richard Zhang, Dr. Albert C. S. Chung, and Dr. Manolis Savva for their time and great feedbacks on this thesis.

Thanks to all the labmates as well as the alumni of the Medical Image Analysis Lab. I learnt a lot from the reading group discussions, day-to-day information sharing and brainstorming. Special thanks to my collaborators, Dr. Pouya Bastani, Dr. Brandon Miles, Fan Gao, Zahra MiriKharaji, Saeed Izadi and Yiqi Yan.

Finally, great thanks to my family members and friends – your love and support had become such a great comfort during the preparation of this thesis.
Dedication

To the forever advancement of science.
# Table of Contents

Approval ii

Abstract iii

Acknowledgements iv

Dedication v

Table of Contents vi

List of Figures x

List of Tables xii

List of Acronyms xiii

1 Introduction 1

1.1 Motivation ............................................. 1

1.2 List of Thesis Contributions ............................................. 3

1.2.1 Bifurcation Detection via Novel Features and Random Forest 3

1.2.2 Bifurcation Detection via Evolutionary Geometric Deformable Templates 4

1.2.3 Branch Direction Prediction via Multi-Loss Deep Learning Network 5

1.2.4 Anatomical Tree Prediction via Encoding Memory using LSTM 6

1.2.5 Extracting Anatomical Trees via Leveraging Tree Statistics 7

1.2.6 Anatomical Tree Extraction via Pictorial Structures and Minimal Paths 9

1.2.7 Retinal Image Classification via Vasculature-guided Sequential Attention 10

1.2.8 Summary of Contributions ............................................. 12

2 Segmentation of Tubular Structures in Medical Images: A Review 16

2.1 Fundamentals in Vascular Image Analysis ............................................. 16

2.1.1 Medical Image Acquisition with Tubular Structure ............................................. 16

2.1.2 Definition of Segmentation Tasks ............................................. 18
5.3.2 Clinical Dataset ................................................. 74
5.3.3 Competing Methods ........................................... 74
5.3.4 Tracking Details ................................................ 75
5.3.5 Evaluation Metrics ............................................. 75
5.3.6 Experiments ...................................................... 75
5.4 Conclusions and Discussions ................................. 76

6 Anatomical Tree Prediction via Encoding Memory using LSTM 79
6.1 Introduction ......................................................... 79
6.2 Methodology ........................................................ 81
  6.2.1 Problem Definition ........................................... 81
  6.2.2 Central Question ............................................. 82
  6.2.3 ELBO Calculation ............................................ 83
  6.2.4 Adopting LSTM .............................................. 83
  6.2.5 Implementation Details ..................................... 84
6.3 Experiments ........................................................ 84
  6.3.1 Datasets ....................................................... 84
  6.3.2 Competing Methods ........................................ 84
  6.3.3 Prediction Evaluation Results ................................. 85
6.4 Conclusions and Discussions .................................... 86

7 Extracting Anatomical Trees via Leveraging Tree Statistics 87
7.1 Introduction ......................................................... 87
  7.1.1 A Short Review of State-of-the-art Methods on Tree Structure
          Segmentation .................................................. 87
  7.1.2 Incorporating Tree Statistics ................................ 89
7.2 Method ............................................................. 90
  7.2.1 Tree Extraction with Tree Priors ................................ 90
  7.2.2 Bifurcation Classification Criterion ........................ 91
  7.2.3 Daughter Branches Detection Criteria ........................ 92
7.3 Experimental Results ............................................. 93
  7.3.1 Description of the Datasets .................................. 93
  7.3.2 Preprocessing and Initialization ............................. 94
  7.3.3 Testing Results ............................................... 96
7.4 Conclusion .......................................................... 98

8 Anatomical Tree Extraction via Pictorial Structures and Minimal Paths 99
8.1 Introduction ........................................................ 99
8.2 Methodology ......................................................... 100
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.2.1 Bifurcation Detection</td>
<td>101</td>
</tr>
<tr>
<td>8.2.2 Branch Centerline Extraction</td>
<td>103</td>
</tr>
<tr>
<td>8.3 Experiments</td>
<td>104</td>
</tr>
<tr>
<td>8.3.1 Data Description</td>
<td>104</td>
</tr>
<tr>
<td>8.3.2 Evaluation Measures</td>
<td>105</td>
</tr>
<tr>
<td>8.3.3 Experimental Results</td>
<td>105</td>
</tr>
<tr>
<td>8.4 Conclusions</td>
<td>107</td>
</tr>
<tr>
<td>9 Retinal Image Classification via Vasculature-guided Sequential Attention</td>
<td>109</td>
</tr>
<tr>
<td>9.1 Introduction</td>
<td>109</td>
</tr>
<tr>
<td>9.1.1 Motivation</td>
<td>109</td>
</tr>
<tr>
<td>9.1.2 Machine (and deep) Learning for Retinal Image Classification</td>
<td>109</td>
</tr>
<tr>
<td>9.1.3 Attention Mechanisms in Deep Learning</td>
<td>110</td>
</tr>
<tr>
<td>9.1.4 Contributions</td>
<td>110</td>
</tr>
<tr>
<td>9.2 Proposed Method</td>
<td>111</td>
</tr>
<tr>
<td>9.2.1 Baseline CNN with Attention Modules</td>
<td>111</td>
</tr>
<tr>
<td>9.2.2 Vasculature Priors on Attention</td>
<td>111</td>
</tr>
<tr>
<td>9.2.3 Learning the Sequence of Prior-guided Attention Maps</td>
<td>114</td>
</tr>
<tr>
<td>9.2.4 Loss Functions</td>
<td>114</td>
</tr>
<tr>
<td>9.2.5 Implementation Details</td>
<td>115</td>
</tr>
<tr>
<td>9.3 Experiment</td>
<td>115</td>
</tr>
<tr>
<td>9.3.1 Datasets</td>
<td>115</td>
</tr>
<tr>
<td>9.3.2 Results</td>
<td>116</td>
</tr>
<tr>
<td>9.4 Conclusions and Discussions</td>
<td>117</td>
</tr>
<tr>
<td>10 Conclusion and Discussion</td>
<td>119</td>
</tr>
<tr>
<td>10.1 Thesis Summary</td>
<td>119</td>
</tr>
<tr>
<td>10.2 Discussion and Open Questions</td>
<td>120</td>
</tr>
<tr>
<td>Bibliography</td>
<td>123</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Anatomical trees</td>
<td>2</td>
</tr>
<tr>
<td>Figure 1.2</td>
<td>Random forest with novel features pipeline</td>
<td>4</td>
</tr>
<tr>
<td>Figure 1.3</td>
<td>Parametric model with tribes</td>
<td>5</td>
</tr>
<tr>
<td>Figure 1.4</td>
<td>TreeNet model</td>
<td>6</td>
</tr>
<tr>
<td>Figure 1.5</td>
<td>Tree-LSTM pipeline</td>
<td>7</td>
</tr>
<tr>
<td>Figure 1.6</td>
<td>Tree tracking with stat priors</td>
<td>9</td>
</tr>
<tr>
<td>Figure 1.7</td>
<td>Pictorial structure based tree tracking</td>
<td>10</td>
</tr>
<tr>
<td>Figure 1.8</td>
<td>Learning with attention</td>
<td>12</td>
</tr>
<tr>
<td>Figure 1.9</td>
<td>List of all works</td>
<td>14</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>Retinal color fundus photography image</td>
<td>18</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>List of 3D bifurcation models</td>
<td>40</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Bifurcation classification error</td>
<td>54</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Bifurcation localization error</td>
<td>55</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Visual comparison of tracking results</td>
<td>55</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Parametric bifurcation model</td>
<td>59</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Bifurcation model catalogue</td>
<td>61</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Synthetic VascuSynth data</td>
<td>64</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Error distribution plot</td>
<td>67</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Qualitative synthetic results</td>
<td>68</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Qualitative clinical results</td>
<td>69</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>TreeNet within a tracking pipeline</td>
<td>72</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>TreeNet loss model</td>
<td>73</td>
</tr>
<tr>
<td>Figure 5.3</td>
<td>Synthetic examples for TreeNet</td>
<td>74</td>
</tr>
<tr>
<td>Figure 5.4</td>
<td>Qualitative result of tracking with TreeNet</td>
<td>77</td>
</tr>
<tr>
<td>Figure 5.5</td>
<td>Distance error of TreeNet tracking</td>
<td>77</td>
</tr>
<tr>
<td>Figure 6.1</td>
<td>LSTM architecture</td>
<td>83</td>
</tr>
<tr>
<td>Figure 7.1</td>
<td>Branching probability</td>
<td>92</td>
</tr>
<tr>
<td>Figure 7.2</td>
<td>Branching with angular statistics</td>
<td>94</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Figure 7.3</td>
<td>Tree statistics in synthetic images</td>
<td>95</td>
</tr>
<tr>
<td>Figure 7.4</td>
<td>Synthetic 3D data with noise</td>
<td>95</td>
</tr>
<tr>
<td>Figure 7.5</td>
<td>Tree statistics in clinical images</td>
<td>96</td>
</tr>
<tr>
<td>Figure 7.6</td>
<td>Qualitative result of tracking with tree statistics</td>
<td>97</td>
</tr>
<tr>
<td>Figure 8.1</td>
<td>An overview of the pictorial structure tracking pipeline</td>
<td>101</td>
</tr>
<tr>
<td>Figure 8.2</td>
<td>Variation of detected bifurcation number on real data</td>
<td>106</td>
</tr>
<tr>
<td>Figure 9.1</td>
<td>LSTM architecture</td>
<td>112</td>
</tr>
<tr>
<td>Figure 9.2</td>
<td>Example of AMD and retinopathy images and corresponding vessel masks</td>
<td>113</td>
</tr>
</tbody>
</table>
List of Tables

Table 1.1 List of used datasets .................................................. 15
Table 2.1 A list of all the surveyed papers .................................. 25
Table 2.2 Model-driven vs. data-driven ........................................ 28
Table 2.3 Model-driven segmentation papers ................................. 29
Table 2.4 Properties of model-driven methods ............................... 36
Table 2.5 A summary of all evaluation metrics .............................. 41
Table 2.6 Public datasets .......................................................... 46
Table 4.1 GA shape parameters ................................................. 60
Table 4.2 GA transformation parameters ...................................... 60
Table 4.3 GA results on VascuSynth data at SNR=20 ....................... 65
Table 4.4 GA results on VascuSynth data at SNR=13.98 .................. 66
Table 4.5 GA results on VascuSynth data at SNR=10.46 .................. 66
Table 4.6 GA results on clinical data .......................................... 66
Table 4.7 Advantages of GA over competing methods ..................... 69
Table 5.1 TreeNet results on synthetic dataset .............................. 76
Table 5.2 TreeNet direction prediction accuracy ............................ 78
Table 6.1 TreeLSTM prediction accuracy ...................................... 85
Table 6.2 TreeLSTM correlation value improvement ......................... 86
Table 7.1 Length statistics of real data ........................................ 94
Table 7.2 Statistial tree detection with synthetic data ..................... 96
Table 7.3 Statistial tree detection with clinical data ....................... 98
Table 8.1 Pictorial tree detection on clinical data .......................... 105
Table 8.2 Pictorial tree detection with SNR analysis ....................... 106
Table 9.1 Parameters for B-Cosfire vessel filter ............................ 115
Table 9.2 Testing result on AMD dataset with binary classification .... 118
Table 9.3 Testing result on retinopathy dataset with 5 level grading .... 118
## List of Acronyms

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>AMD</td>
<td>Age-related Macular Degeneration</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>ASL MRA</td>
<td>Arterial Spin Labeling MRA</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>COE</td>
<td>Concurrency of Eigenvectors</td>
</tr>
<tr>
<td>COPD</td>
<td>Chronic Obstructive Pulmonary Disease</td>
</tr>
<tr>
<td>CRF</td>
<td>Conditional Random Field</td>
</tr>
<tr>
<td>CT</td>
<td>Computed Tomography</td>
</tr>
<tr>
<td>CTA</td>
<td>Computed Tomography Angiography</td>
</tr>
<tr>
<td>DB</td>
<td>Detected Bifurcation</td>
</tr>
<tr>
<td>DCNN</td>
<td>Dilated Convolutional Neural Network</td>
</tr>
<tr>
<td>DR</td>
<td>Diabetic Retinopathy</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GCN</td>
<td>Graph Convolutional Network</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphic Programming Unit</td>
</tr>
<tr>
<td>GT</td>
<td>Ground Truth</td>
</tr>
<tr>
<td>HOE</td>
<td>Histogram of Eigenvectors</td>
</tr>
<tr>
<td>IVUS</td>
<td>Intravascular Ultrasound</td>
</tr>
<tr>
<td>KNN</td>
<td>K Nearest Neighbors</td>
</tr>
<tr>
<td>LILB</td>
<td>Left Intermediate Lobar Bronchus</td>
</tr>
<tr>
<td>Notation</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>LMB</td>
<td>Left Main Bronchi</td>
</tr>
<tr>
<td>LSLB</td>
<td>Left Superior Lobar Bronchus</td>
</tr>
<tr>
<td>LSTM</td>
<td>Long Short Term Memory</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum a Posteriori</td>
</tr>
<tr>
<td>MRA</td>
<td>Magnetic Resonance Angiography</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>MSP</td>
<td>Minimal Spanning Tree</td>
</tr>
<tr>
<td>OCT</td>
<td>Optical Coherence Tomography</td>
</tr>
<tr>
<td>OOF</td>
<td>Optimally Oriented Flux</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>RBM</td>
<td>Restricted Boltzmann Machine</td>
</tr>
<tr>
<td>RIB</td>
<td>Right Intermediate Bronchus</td>
</tr>
<tr>
<td>RMB</td>
<td>Right Main Bronchi</td>
</tr>
<tr>
<td>ROI</td>
<td>Region of Interest</td>
</tr>
<tr>
<td>RSLB</td>
<td>Right Superior Lobar Bronchus</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal Noise Ratio</td>
</tr>
<tr>
<td>std</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TOF MRA</td>
<td>Time of Flight MRA</td>
</tr>
<tr>
<td>US</td>
<td>Ultrasound</td>
</tr>
<tr>
<td>VMF</td>
<td>von Mises-Fisher</td>
</tr>
<tr>
<td>VOI</td>
<td>Volume of Interest</td>
</tr>
<tr>
<td>WHO</td>
<td>World Health Organization</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Cardiovascular related diseases and chronic respiratory diseases are among the top causes of non-communicable disease deaths globally, which results in 17.9 million and 3.8 million deaths respectively every year, according to a World Health Organization (WHO) report in 2018 [164]. Such diseases need to be diagnosed in an accurate and timely manner. However, accurate diagnosis usually involves 3D non-invasive imaging of anatomical sites such as the brain and the chest. The generated images need to be further segmented so that pathological structures such as vasculatures and airways (or in general, tubular structures) could be clearly seen by the clinicians. The segmentation is usually performed in two ways, either manually by a clinician or automatically by a specially designed algorithm. While manual delineation of vasculatures and airways is tedious and time-consuming, the precision of automated methods is often compromised by the existence of surrounding tissues such as bones and parenchyma. In order to bridge the gap between delineation accuracy and speed, in this thesis, we developed automated and robust computational methods for anatomical tree extraction from 3D medical images.

This thesis is centered around the concept of “anatomical trees”, which has been used in previous literature [62,75,129] without formal definition. In order to distinguish from concepts such as phylogenetic trees and tree data structures in biology and computer science, we propose to use a formal definition of an “anatomical tree”, which is an abstract tree-shape object comprising the following aspects:

1. The abstract tree structure should be related to specific anatomy in the human body, ranging from a certain position or the full body [57], e.g., an airway tree is related to the human lungs rather than the human brain;

2. The abstract tree branches should, at least partially be assigned anatomical labels, see Figure 1.1;
3. Geometric attributes should be associated with abstract tree structure, e.g., branch lengths and angles.

![Diagram of cerebral arteries](image1.png)

(a) Circle of Willis of cerebral arteries  (b) Coronary artery and vein trees.

Figure 1.1: (color figure) Labelling of anatomical trees in clinical images. Figures used with permission. ¹

Once the concept of “anatomical trees” is properly defined, a natural question is how to incorporate this concept as prior knowledge [162] in tubular structure extraction. However, this is no easy task. To further study how tubular structure extraction was performed in existing literature, we provided a literature survey in Chapter 2 of this thesis. In the survey, we divided state-of-the-art works into two main categories: a) model-driven methods, and b) data-driven methods (please find in Section 2.3.1 the description of model-based and data-driven methods, and the corresponding advantages and disadvantages). For model-driven methods, we further divided the models into four sub-categories:

1. Enhancement-based methods;

2. Thresholding-based methods;

¹Figure 1.1b is a derivative of Coronary Circulation from Anatomy and Physiology by OpenStax, distributed under the Creative Commons Attribution License v4.0 license.
3. Graph-based methods;

4. Progressive methods.

In the subsequent chapters, we further developed on the idea of incorporating anatomical

tree priors, by targeting the following tasks:

1. Bifurcation localization: the task of locating bifurcation points in a given image;

2. Branch direction prediction: the task of predicting the direction of a given branch;

3. Statistics and topology embedding: the task of using statistics and topology during
tree tracking;

4. Tree-facilitated disease classification: the task of disease classification using tree-based
knowledge.

To solve the tasks above, a combination of model-based and data-driven techniques was
introduced. The contributions of these techniques are listed in the following section.

1.2 List of Thesis Contributions

1.2.1 Bifurcation Detection via Novel Features and Random Forest

Bifurcation detection is important in medical image analysis for two main reasons:

1. Plaques are easy to accumulate at artery bifurcations, which leads to atherosclerosis
and strokes;

2. Extraction of all vessel branches is crucial for quantification (e.g. branch length,
thickness, tortuosity), visualization, and blood flow simulation purposes.

In this contribution, several novel features are designed for classifying bifurcations in 3D
vascular images using random forest, by capturing the intensity distribution on the spherical
surface, as well as the eigenvector distribution of a Hessian-based filter.

Results with both synthetic and real datasets show that the proposed method
outperforms state-of-the-art methods by at least 5%.

Contributions

1. We devised novel features for 3D bifurcation detection using random forest;

2. We proposed a generic tree tracking pipeline;

3. We achieved bifurcation detection accuracy which outperformed state-of-the-art
methods.
Figure 1.2: Pipeline of combining the proposed bifurcation prediction using novel features in centerline-based tree tracking. The details of this contribution are presented in Chapter 3.


1.2.2 Bifurcation Detection via Evolutionary Geometric Deformable Templates

Given the importance of studying bifurcations in 3D anatomical trees (e.g. vasculature and airway), we propose a bifurcation detector that operates by fitting a parametric geometric deformable model to 3D medical images. Since the model was derived from the analysis of bifurcations in clinical data, it is expected to be useful in locating the bifurcation locations. A fitness function is designed to integrate features along the model skeletons, surfaces and internal areas. To overcome local optima while detecting multiple bifurcations in a single image, we adopt genetic algorithm with a tribes niching technique. Results on both VascuSynth data and clinical CT data demonstrate not only high bifurcation detection accuracy and stability, but the ability to locate parent and children branch directions and vessel wall locations simultaneously.

Contributions

1. We proposed the first work to fit a 3D parametric model to medical image for locating bifurcations;

2. We proposed a multi-objective fitness function for embedding shape and image information;

3. We formulated a novel method for simultaneously detecting multiple bifurcations using genetic algorithm optimization with tribes.
1.2.3 Branch Direction Prediction via Multi-Loss Deep Learning Network: the TreeNet Model

Calculation of blood vessel or airway direction is important for the task of tree tracking in 3D medical images. However, most existing works treat branch direction estimation as only a by-product of vesselness or tubularness computation. In this work, we propose a deep learning framework for predicting tracking directions of anatomical tree structures. We modify the deep V-Net architecture with extra layers and leverage a novel multi-loss function that encodes direction as well as cross sectional plane information. We evaluate our method on both 3D synthetic and 3D clinical pulmonary CT datasets. On the synthetic dataset, we outperform state of the art methods by at least 10% in direction estimation accuracy. For the clinical dataset, we outperform competing methods by 1%-4% in mean direction accuracy and 4%-10% in corresponding standard deviation.

Contributions

1. We proposed the first tree branch direction prediction deep learning method;

2. We built a novel multi-loss function incorporating geometrical and image intensity information on the Frenet frame in a branch-specific way.
1.2.4 Anatomical Tree Prediction via Encoding Memory using LSTM: the Tree-LSTM Model

Extraction and analysis of anatomical trees, such as vasculatures and airways, is important for many clinical applications. However, most tracking methods so far intrinsically embedded the Markovian property, where no memory beyond one tracking step was utilized in the tree extraction process. Motivated by the inherit sequential construction of anatomical trees, vis-à-vis the flow of nutrients through branches and bifurcations, we propose Tree-LSTM, the first LSTM neural network to learn to encode such sequential priors into a deep learning tree extraction method. We also show that, by using LSTM, the variational lower bound of a high-order Markovian stochastic process could be approximated. Our experiments on a CT airway dataset showed that, by adding the LSTM component, the results are improved by at least 11% in mean accuracy relative to state-of-the-art, and the correlation between bifurcation classification accuracy and evidence is improved by at least 15%, which demonstrate the advantage of a unified deep model for sequential tree structure tracking and bifurcation detection.

Contributions

1. We proposed the first work using LSTM in tree branch direction prediction;

2. We derived a formal proof converting the higher-order Markovian process into its variational lower bound;

3. We devised a novel evaluation method for inspecting the correlation between bifurcation classification accuracy and sequential image evidence collected along branches.

Figure 1.5: (color figure) Pipeline and architecture of Tree-LSTM. The details of this contribution are presented in Chapter 6.


1.2.5 Extracting Anatomical Trees via Leveraging Tree Statistics – A Geometrical and Topological Point of View

Using different priors (e.g. shape and appearance) have proven critical for robust image segmentation of different types of target objects. Many existing methods for extracting trees (e.g. vascular or airway trees) from medical images have leveraged appearance priors (e.g. tubular-ness and bifurcationness) and the knowledge of the cross-sectional geometry (e.g. circles or ellipses) of the tree-forming tubes. In this work, we present the first method for 3D tree extraction from 3D medical images (e.g. CT or MRI) that, in addition to appearance and cross-sectional geometry priors, utilizes prior tree statistics collected from the training
data. Intuitively speaking, we expect statistics learnt directly from the dataset to provide guidance to the centerline tree tracking procedure. Our tree extraction method collects and leverages topological tree prior and geometrical statistics, including tree hierarchy, branch angle and length statistics. Our implementation takes the form of a Bayesian tree centerline tracking method combining the aforementioned tree priors with observed image data. We evaluated our method on both synthetic 3D datasets and real clinical CT chest datasets. For synthetic data, our methods key feature of incorporating tree priors resulted in at least 13% increase in correctly detected branches under different noise levels. For real clinical scans, the mean distance from ground truth centerlines to the detected centerlines by our method was improved by 12% when utilizing tree priors. Both experiments validate that, by incorporating tree statistics, our tree extraction method becomes more robust to noise and provides more accurate branch localization.

Contributions

1. We proposed the first method that incorporated topological priors and geometrical statistics of anatomical trees into a tree extraction algorithm;

2. We formulated a novel Bayesian based tracking algorithm;

3. We achieved improvement in both the tree tracking accuracy and stability.
Figure 1.6: (color figure) Tree tracking using statistical priors. Image adapted with permission. 2The details of this contribution are presented in Chapter 7.


1.2.6 Anatomical Tree Extraction via Pictorial Structures and Minimal Paths – a Global Optimal Approach

Extracting centerlines of anatomical trees (e.g., vasculature and airways) from 3D medical images is a crucial preliminary step for various medical applications. We propose an automatic tree extraction method that leverages prior knowledge of tree topology and geometry and ensures globally-optimal solutions. Intuitively speaking, we leverage the fact that the pictorial structure is a topological tree, with "relaxed" geometrical properties. We define a pictorial structure with a corresponding cost function to detect tree bifurcations in anatomical trees with predefined topology. The tree bifurcations are encoded as nodes in the pictorial structure and are associated with an artificial neural network (ANN) based unary term. The geometrical (direction and length) statistics of tree branches are learned from a training set and encoded as geometrical priors for regularizing the pictorial structure

2This work is a derivative of Gross Anatomy of the Lungs from Anatomy and Physiology by OpenStax, distributed under the Creative Commons Attribution License v4.0 license.
edges. Finally, detected bifurcations as well as the ANN tubularity scores, are leveraged to trace globally optimal minimal paths along 3D tree centrelines. Our method outperforms competing state-of-the-art when evaluated on 3D synthesized vasculature and lung airways in CT and our results demonstrate the advantages of incorporating tree statistics and global optimization for this task.

Contributions

1. We proposed the first work to represent tree shapes using pictorial structures in medical image analysis;

2. We proposed to encode both geometrical and topological priors in tree extraction;

3. We achieved the first globally optimal tree extraction solution in airway extraction.

Figure 1.7: (color figure) Pictorial structure based tree tracking framework. The details of this contribution are presented in Chapter 8.


1.2.7 Retinal Image Classification via Vasculature-guided Sequential Attention

Age-related macular degeneration and diabetic retinopathy are diseases of increasing prevalence globally in recent years. Traditionally, diagnosing these diseases relied on manual visual inspection by experts, which was costly, time-consuming and laborious as it required closely examining high-resolution color fundus images. More recently, deep learning

† Joint first author.
networks have shown great potential in predicting diseases from retinal images. However, being purely data-driven, these networks are susceptible to overfitting and their training requires large annotated data. In this contribution, we propose to enrich deep learning-based fundus image classifiers with prior knowledge on special structures in the retina implicated with the disease. In particular, we leverage vessel priors to guide the attention mechanism of deep learning architectures. In addition, we leverage a bi-directional dual-layer LSTM module to learn the inter-dependencies between a sequence of prior-guided attention maps deployed across the depth of the disease classification network. Intuitively speaking, we try to leverage the shape information as well as the inter-dependency of attention features among different layers by using vessel priors and the LSTM module. Results on the clinical datasets show the proposed method could bring performance improvement by as much as 8%.

Contributions

1. We proposed the first work that leveraged vessel tree priors to guide the generation of attention maps;

2. We proposed to use automatically extract vessel masks for generating the attention prior maps;

3. We devised to encode inter-dependency among attention features using LSTM.
Figure 1.8: (color figure) Architecture for attention mechanism. The details of this contribution are presented in Chapter 9.


1.2.8 Summary of Contributions

Overall, we proposed in this thesis different approaches to encode the concept of “anatomical tree” in the analysis of medical images containing tree structures, from the following perspectives (see Figure 1.9):

1. Bifurcation detection.
(a) We proposed the first work using Random Forest and novel features for bifurcation classification in clinical images.

(b) We proposed the first work using 3D parametric models for fitting multiple bifurcation locations simultaneously using genetic algorithm with tribes niching technique.

2. Branch direction prediction.

(a) We proposed the first work using deep learning network for branch direction prediction, from a branch-specific perspective.

(b) We proposed the first work using LSTM for branch direction prediction within a higher-order Markovian perspective.

3. Tree tracking.

(a) We proposed the first work using branch-level statistics for tracking trees in pulmonary images.

(b) We proposed the first work using pictorial structure for searching global optimal trees in pulmonary images.

4. Tree guided disease classification.

(a) We proposed the first work using automatically-extracted tree masks to guided attention maps generation in deep learning-based disease classification.

Other contribution not included in the thesis: We also created an online database (http://vascusynth.cs.sfu.ca) of synthetic data generated by VascySynth, which comprises 120 volumes of binary images with varying tree orders and branch lengths.
Figure 1.9: (color figure) a) List of all works; b) Diagram of all contributions.
<table>
<thead>
<tr>
<th>Ch.</th>
<th>Pub.</th>
<th>Dataset</th>
<th>Type</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ch. 3</td>
<td>[244]</td>
<td>VascuSynth [81]</td>
<td>3D Synthetic</td>
<td>Bifurcation Detection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VESSEL12 [184]</td>
<td>3D Pulmonary Vessel CT</td>
<td></td>
</tr>
<tr>
<td>Ch. 4</td>
<td>[245]</td>
<td>VascuSynth [81]</td>
<td>3D Synthetic</td>
<td>Bifurcation Detection</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VESSEL12 [184]</td>
<td>3D Pulmonary Vessel CT</td>
<td></td>
</tr>
<tr>
<td>Ch. 5</td>
<td>[246]</td>
<td>Synthetic Pathology</td>
<td>3D Synthetic</td>
<td>Direction Prediction</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXACT [2]</td>
<td>3D Airway CT</td>
<td></td>
</tr>
<tr>
<td>Ch. 6</td>
<td>[245]</td>
<td>EXACT [2]</td>
<td>3D Airway CT</td>
<td>Bifurcation Detection</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Direction Prediction</td>
</tr>
<tr>
<td>Ch. 7</td>
<td>[248]</td>
<td>Synthetic Tree</td>
<td>3D Synthetic</td>
<td>Tree Tracking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXACT [2]</td>
<td>3D Airway CT</td>
<td></td>
</tr>
<tr>
<td>Ch. 8</td>
<td>[149]</td>
<td>Synthetic Tree</td>
<td>3D Synthetic</td>
<td>Tree Tracking</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EXACT [2]</td>
<td>3D Airway CT</td>
<td></td>
</tr>
<tr>
<td>Ch. 9</td>
<td>[247]</td>
<td>iChallenge-AMD [3]</td>
<td>Color Fundus Photography</td>
<td>Disease Classification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDRiD [173]</td>
<td>Color Fundus Photography</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1: List of all the datasets used in this thesis. For each chapter (Ch.), the corresponding publication (Pub.), name of the dataset, type, and the related task is shown.
Chapter 2

Segmentation of Tubular Structures in Medical Images: A Review

In this chapter, we review works in recent years on vasculature segmentation in medical images. We begin by introducing fundamental concepts related to vascular image acquisition and the formulation of segmentation tasks. Then we advance into two major categories of segmentation techniques: model-driven and data-driven. Since the task of bifurcation detection is often pursued side-by-side with the segmentation task, we discuss it in the subsequent section. Evaluation methods and public datasets are listed, with a summary at the end of the chapter.

2.1 Fundamentals in Vascular Image Analysis

2.1.1 Medical Image Acquisition with Tubular Structure

A variety of non-invasive and invasive imaging modalities could be used for visualizing tubular structures in the human body. Here we introduce some of the most often used techniques, including computed tomography angiography (CT), magnetic resonance angiography (MRA), intravascular ultrasound (IVUS), retinal fundus photography.

Angiography imaging of Vascular Images in CT and MRI

Computed Tomography (CT) is a method using a computational reconstruction technique to calculate corresponding pixel values of each coordinate in a plane or 3D volume, from X-ray projection raw data. In medical imaging, CT could be used for organs like kidney, brain, lung as well as anatomical regions like head, legs, and arms. Due to the attenuation of X-ray beams, CT reconstructs different pixel intensities according to the body tissue’s Hounsfield scale value [224]. However, as the Hounsfield scale of blood is very close or overlaps with parenchyma and other fluid, it’s necessary to increase the contrast
between these tissues. One way of increasing such contrast is through imaging procedure, by injecting a harmless material into the blood. As this material travels through the blood vessels, a clearer view of the vessels could be captured by CT. The pros and cons of CTA are obvious: it’s fast and cheap (comparing to techniques like MRI), but it also requires contrast agents and is radioactive. More specifically, the problems using the contrast agent include safety issues, time window restrictions and diffusion imbalances. First, the contrast agents most often used in CT scans are usually iodine-based, which are not safe to breast-feeding women and could trigger allergy. The alternatives are non-ionic, but are also more expensive [188]. Second, as the contrast agent diffuses along the blood vessel over the time, there is a limited time window that the patient could be imaged. Third, there is less contrast agent in smaller vessels, which usually makes smaller vessels more difficult to distinguish in the reconstructed images. Since CT scan itself is radioactive, it can harm the patient if taken on a frequent basis.

Magnetic Resonance Imaging relies on the fact that when placed in an external magnetic field, certain atomic nuclei could absorb and emit radiofrequency energy. Since hydrogen atoms are able to generate such detectable radio-frequency signals, human organs containing water, e.g., the brain, muscles are especially suitable to be imaged. MRI angiography may or may not require contrast agent injection. MRA has great advantages over CT: it needs less contrast agent and is not radioactive. On the other hand, MRA is noisy and expensive, and has restrictions on the patient’s body, e.g., patients with certain implant can’t use MRI.

**Intravascular Ultrasound of Vascular Images**

Medical ultrasound is a technique that uses sound waves with a frequency above the audible to humans to body tissues and uses a probe to detect and display the reflections. Comparing to CT and MRI, ultrasound imaging is cheaper and faster, without any radiation. However, ultrasound images usually involve irrelevant high-frequency structures, and the final reconstructed images are not parallel to any anatomical plane. These factors would cause the reconstructed images hard to interpret.

Intravascular ultrasound (IVUS) is an invasive ultrasound imaging technique, which attaches a miniaturized probe to the end of a catheter and the catheter travels through human blood vessels for imaging purposes. IVUS allows real-time visualization of the internal area of blood vessels, measuring vessel wall as well as detecting branching patterns. But IVUS also has drawbacks, it needs a catheter to be inserted inside the patient’s body, which causes an uncomfortable feeling as well as increases the chance to be infected. Unlike CT or MRI scans, IVUS only visualizes a certain blood vessel branch, while the former could visualize the whole vessel tree all at once.
Color Fundus Photography of Retinal Vasculatures

For retinal images, one of the most common tools for acquiring images is the color fundus photography, which uses a light source, charge-coupled device camera and magnifier to take a photo of the retinal fundus with a magnified view. The process is of no radiation so no harm to the human body and the output image is a 3-channel colored image. Although healthy retinas could be photographed as relative clean images, different lighting, diseased retina could result in low contrast images, which makes the blood vessels difficult to be extracted.

![Figure 2.1: (color figure) Retinal color fundus photography image, low contrast.](image)

In summary, the general problems within image acquisition include acquisition difficulty, patient comfort level, acquisition cost, noise level, and contrast level. To further facilitate clinicians to see the pathological structures and make a better and swifter diagnosis, image segmentation is an essential component in the analysis task.

2.1.2 Definition of Segmentation Tasks

Due to the complex shape of the vascular system, we regard the segmentation task of vascular images consisting of these two main formulations: binary segmentation, sequential tree extraction. We also define a subordinate formulation, the bifurcation detection, which often appear side-by-side with vasculature segmentation.

Binary Segmentation  Given image $I$ (could be RGB image, volumetric grayscale image, etc.) defined on domain $\Omega \in \mathbb{R}^n$, which satisfies:

$$I : \Omega \rightarrow \mathbb{R}$$

[1] Image obtained from AMD online challenge: https://amd.grand-challenge.org/
binary segmentation task is to reconstruct a binary function:

\[ u : \Omega \rightarrow \{0, 1\} \tag{2.1} \]

We don’t formally define fuzzy segmentation, such as \( u : \Omega \rightarrow [0, 1] \), since it could be converted to the binary form with thresholding. We also don’t formally specify multi-region segmentation, such as \( u : \Omega \rightarrow \mathcal{Z} \), since the following discussions could be extended to multi-region segmentation result.

To encourage the foreground area of \( u \) (or, where \( u = 1 \)) represents the structures of most interest, \( u \) is generally defined to minimize an energy functional of a similar form:

\[
\min_u \int_\Omega [D(u, I) + B(u)] dx_1^1 dx_2^2 ... dx_n^n, \tag{2.2}
\]

where \( D \) is unary energy term and \( B \) is binary/regularization energy term specific to the application.

**Sequential Tree Extraction** Different from binary segmentation, the sequential tree extraction formulates the problem as the following:

\[
\begin{aligned}
    u_1 &= x_0 \\
    u_{t+1} &= u_t \cup x_t \\
    \text{dist}(x_t, u_t) &= \delta d
\end{aligned} \tag{2.3}
\]

where \( x_t, x_0 \in \Omega \), \( t \) is the time step, \( \delta d \) is the tracking step length (usually a fixed value).

The tracking process starts from an initial set of candidate points, tracking directions, as well as the tracking step length. Extra rules of detecting the cross-sectional plane, bifurcation detection, leakage prevention and stopping criteria are often required.

**Bifurcation Detection** Bifurcation detection could be rephrased into the following format:

\[ u : \Omega \rightarrow \mathcal{R}^{m \times n} \tag{2.4} \]

where \( m \) is the total number of bifurcations found in the image.

The three different formulations will be involved in different methodologies, as shown in Table 2.1. Pros and cons for each formulation are given in the following:

1. Regularity v.s. Structurality. For using binary segmentation formulation, one advantage is that regularization could be easily applied on functionals in the form of Equation 2.2. However, the regularity doesn’t reflect advanced properties on the tree structures, such as branching patterns and the tree hierarchy. Furthermore, to encode regularity on the constants of trees is a difficult task. More discussions will be given in Section 2.3.2)
2. Simultaneity v.s. Sequentiality. For formulations such as binary segmentation and bifurcation detection, another advantage is that the segmentation and detection could be applied to all the points in the image domain simultaneously. However, this is not true for the centerline tree tracking formulation. Furthermore, tree tracking is a sequential event and strongly relies on its previous tracking result.

2.2 List of Papers

In Table 2.1, we give a general overview of all the papers studied in this survey. More detailed discussions will be given in the following sections.
<table>
<thead>
<tr>
<th>No.</th>
<th>Paper</th>
<th>Formulation</th>
<th>Method</th>
<th>Automatic</th>
<th>Year</th>
<th>Anatomy</th>
<th>Modality</th>
<th>EB</th>
<th>Case No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bülow [32]</td>
<td>Binary Segmentation</td>
<td>Front Propagation</td>
<td>Semi</td>
<td>2004</td>
<td>Coronary</td>
<td>CT</td>
<td>✓</td>
<td>9 scans</td>
</tr>
<tr>
<td>2</td>
<td>McIntosh [142]</td>
<td>Binary Segmentation</td>
<td>Artificial Intelligence</td>
<td>Semi</td>
<td>2006</td>
<td>Cerebral</td>
<td>MRA</td>
<td>✓</td>
<td>1 scan</td>
</tr>
<tr>
<td>3</td>
<td>Zhou [253]</td>
<td>Bifurcation Detection</td>
<td>Machine Learning</td>
<td>Semi</td>
<td>2007</td>
<td>Pulmonary Vessel</td>
<td>CT</td>
<td>✓</td>
<td>8 VOI</td>
</tr>
<tr>
<td>4</td>
<td>Graham [76]</td>
<td>Binary Segmentation</td>
<td>Region Growing + Graph Partitioning</td>
<td>Fully</td>
<td>2008</td>
<td>Airway</td>
<td>CT</td>
<td>✓</td>
<td>23 scans</td>
</tr>
<tr>
<td>5</td>
<td>Fabijanska [56]</td>
<td>Binary Segmentation</td>
<td>Region Growing</td>
<td>Fully</td>
<td>2009</td>
<td>Airway</td>
<td>CT</td>
<td>✓</td>
<td>10 scans</td>
</tr>
<tr>
<td>6</td>
<td>Lo [127]</td>
<td>Binary Segmentation</td>
<td>Region Growing</td>
<td>Fully</td>
<td>2009</td>
<td>Airway</td>
<td>CT</td>
<td>✓</td>
<td>40 scans</td>
</tr>
<tr>
<td>7</td>
<td>Feuerstein [63]</td>
<td>Binary Segmentation</td>
<td>Sharpening Filter</td>
<td>Fully</td>
<td>2009</td>
<td>Airway</td>
<td>CT</td>
<td>✓</td>
<td>40 scans</td>
</tr>
<tr>
<td>8</td>
<td>Law [116]</td>
<td>Binary Segmentation</td>
<td>Deformable Surface</td>
<td>Fully</td>
<td>2009</td>
<td>Cerebral</td>
<td>CT</td>
<td>✓</td>
<td>3 scans</td>
</tr>
<tr>
<td>9</td>
<td>Lo [126]</td>
<td>Binary Segmentation</td>
<td>Region Growing</td>
<td>Fully</td>
<td>2010</td>
<td>Airway</td>
<td>CT</td>
<td>✓</td>
<td>250 scans</td>
</tr>
<tr>
<td>11</td>
<td>Bruyninckx [31]</td>
<td>Binary Segmentation</td>
<td>SVM + Minimal Path + Ant Colony Optimization</td>
<td>Fully</td>
<td>2010</td>
<td>Liver Vein</td>
<td>CT</td>
<td>✓</td>
<td>5 scans</td>
</tr>
<tr>
<td>14</td>
<td>Goyal [74]</td>
<td>Binary Segmentation</td>
<td>Thresholding</td>
<td>Fully</td>
<td>2013</td>
<td>Porcine Coronary</td>
<td>Fluorescence Cryomicrotome</td>
<td>✓</td>
<td>105 scans</td>
</tr>
<tr>
<td>15</td>
<td>Breitenreicher [29]</td>
<td>Binary Segmentation</td>
<td>Machine Learning</td>
<td>Fully</td>
<td>2013</td>
<td>Airway</td>
<td>CT</td>
<td>✓</td>
<td>49 scans</td>
</tr>
<tr>
<td>16</td>
<td>Rizi [180]</td>
<td>Binary Segmentation</td>
<td>Fuzzy Connectivity</td>
<td>Fully</td>
<td>2013</td>
<td>Airway</td>
<td>CT</td>
<td>✓</td>
<td>93 scans</td>
</tr>
<tr>
<td>17</td>
<td>Turetken [210]</td>
<td>Binary Segmentation</td>
<td>Multi-direction OOF</td>
<td>Fully</td>
<td>2013</td>
<td>Retinal</td>
<td>Micrograph</td>
<td>✓</td>
<td>14 scans</td>
</tr>
<tr>
<td>18</td>
<td>Tang [208]</td>
<td>Binary Segmentation</td>
<td>Minimal Path + Levelset</td>
<td>Semi</td>
<td>2013</td>
<td>Carotid</td>
<td>CTA</td>
<td>✓</td>
<td>56 scans</td>
</tr>
<tr>
<td>No.</td>
<td>Paper</td>
<td>Formulation</td>
<td>Method</td>
<td>Automatic</td>
<td>Year</td>
<td>Anatomy</td>
<td>Modality</td>
<td>EB</td>
<td>Case No.</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>----------</td>
<td>----</td>
<td>---------</td>
</tr>
<tr>
<td>19</td>
<td>Lu [131]</td>
<td>Binary Segmentation</td>
<td>Sparse Learning</td>
<td>Fully</td>
<td>2014</td>
<td>Coronary</td>
<td>CT</td>
<td>✓</td>
<td>154 scans</td>
</tr>
<tr>
<td>21</td>
<td>Hannink [85]</td>
<td>Enhancement</td>
<td>Wavelet and Spline based Orientation Score</td>
<td>Fully</td>
<td>2014</td>
<td>Retinal</td>
<td>OCT</td>
<td>✓</td>
<td>15 scans</td>
</tr>
<tr>
<td>22</td>
<td>Kitamura [107]</td>
<td>Binary Segmentation</td>
<td>CRF</td>
<td>Fully</td>
<td>2014</td>
<td>Coronary</td>
<td>CTA</td>
<td>✓</td>
<td>48 scans</td>
</tr>
<tr>
<td>23</td>
<td>Cetin [38]</td>
<td>Sequential Tree Extraction</td>
<td>Tracking</td>
<td>Semi</td>
<td>2013</td>
<td>Coronary</td>
<td>CTA</td>
<td>✓</td>
<td>10 scans</td>
</tr>
<tr>
<td>24</td>
<td>Skibbe [200]</td>
<td>Binary Segmentation</td>
<td>Particle Graph</td>
<td>Fully</td>
<td>2015</td>
<td>Cerebral, Carotid and Coronary</td>
<td>CT</td>
<td>✓</td>
<td>33 scans</td>
</tr>
<tr>
<td>25</td>
<td>Hennersperger [88]</td>
<td>Enhancement</td>
<td>Modified Hessian Filter</td>
<td>Fully</td>
<td>2015</td>
<td>Carotid and Abdominal</td>
<td>Ultrasound</td>
<td>✓</td>
<td>30 scans</td>
</tr>
<tr>
<td>27</td>
<td>Liang [124]</td>
<td>Binary Segmentation</td>
<td>Levelset</td>
<td>Fully</td>
<td>2015</td>
<td>Liver</td>
<td>3D Microscopy</td>
<td>✓</td>
<td>54 scans</td>
</tr>
<tr>
<td>28</td>
<td>Mu [156]</td>
<td>Binary Segmentation</td>
<td>Local Threshholding</td>
<td>Fully</td>
<td>2015</td>
<td>Bone Marrow</td>
<td>3D Microscopy</td>
<td>✓</td>
<td>35 scans</td>
</tr>
<tr>
<td>31</td>
<td>Zhao [242]</td>
<td>Binary Segmentation</td>
<td>Bi-Gaussian Filtering and Centerline Connection</td>
<td>Fully</td>
<td>2015</td>
<td>Murine vessel</td>
<td>Micro-CT</td>
<td>✓</td>
<td>2 scans</td>
</tr>
<tr>
<td>32</td>
<td>Chen [44]</td>
<td>Binary Segmentation</td>
<td>Deformable Model</td>
<td>Semi</td>
<td>2015</td>
<td>Femoral Artery</td>
<td>Black Blood MR</td>
<td>✓</td>
<td>6 scans</td>
</tr>
<tr>
<td>33</td>
<td>Wu [232]</td>
<td>Sequential Tree Extraction</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2016</td>
<td>Retinal</td>
<td>OCT</td>
<td>✓</td>
<td>20 images</td>
</tr>
<tr>
<td>34</td>
<td>Lesage [122]</td>
<td>Sequential Tree Extraction</td>
<td>Particle Filtering</td>
<td>Semi</td>
<td>2016</td>
<td>Coronary</td>
<td>CT</td>
<td>✓</td>
<td>61 volumes</td>
</tr>
<tr>
<td>35</td>
<td>Fu [69]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2016</td>
<td>Retinal</td>
<td>OCT</td>
<td>✓</td>
<td>440 images</td>
</tr>
<tr>
<td>36</td>
<td>Sironi [199]</td>
<td>Binary Segmentation</td>
<td>Multiscale Regression</td>
<td>Fully</td>
<td>2016</td>
<td>Cat/Rat Neuron</td>
<td>Microscopy + Micrographs</td>
<td>✓</td>
<td>18 images</td>
</tr>
<tr>
<td>No.</td>
<td>Paper</td>
<td>Formulation</td>
<td>Method</td>
<td>Automatic</td>
<td>Year</td>
<td>Anatomy</td>
<td>Modality</td>
<td>EB</td>
<td>Case No.</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>----------</td>
<td>----</td>
<td>---------</td>
</tr>
<tr>
<td>37</td>
<td>Khorshed [105]</td>
<td>Binary Segmentation</td>
<td>Machine Learning</td>
<td>Fully</td>
<td>2016</td>
<td>Mice Bone Marrow</td>
<td>Microscope</td>
<td>✓</td>
<td>9 scans</td>
</tr>
<tr>
<td>38</td>
<td>Wang [217]</td>
<td>Binary Segmentation</td>
<td>Graph cut</td>
<td>Fully</td>
<td>2016</td>
<td>Renal Artery</td>
<td>CT</td>
<td>X</td>
<td>7</td>
</tr>
<tr>
<td>39</td>
<td>Shin [196]</td>
<td>Binary Segmentation</td>
<td>MRF graph</td>
<td>Semi</td>
<td>2016</td>
<td>Coronary</td>
<td>X-ray Sequence</td>
<td>X</td>
<td>18 sequence</td>
</tr>
<tr>
<td>41</td>
<td>Tan [207]</td>
<td>Binary Segmentation</td>
<td>Gabor Filter</td>
<td>Fully</td>
<td>2016</td>
<td>Retinal</td>
<td>OCT</td>
<td>✓</td>
<td>40 scans</td>
</tr>
<tr>
<td>42</td>
<td>Gu [79]</td>
<td>Segmentation Structure Features</td>
<td>Fully</td>
<td>Generic</td>
<td>Generic</td>
<td>Multiple Sources</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>Lee [119]</td>
<td>Sequential Tree Extraction</td>
<td>Particle Filtering</td>
<td>Semi</td>
<td>2017</td>
<td>Abdominal</td>
<td>CTA</td>
<td>X</td>
<td>10 scans</td>
</tr>
<tr>
<td>45</td>
<td>Ibragimov [95]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2017</td>
<td>Liver</td>
<td>CT</td>
<td>X</td>
<td>72 scans</td>
</tr>
<tr>
<td>46</td>
<td>Chen [43]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2017</td>
<td>Cerebral</td>
<td>TOF MRA</td>
<td>X</td>
<td>46 scans</td>
</tr>
<tr>
<td>47</td>
<td>Tetteh [209]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2017</td>
<td>Retinal</td>
<td>OCT</td>
<td>X</td>
<td>60 scans</td>
</tr>
<tr>
<td>49</td>
<td>Shen [195]</td>
<td>Bifurcation Detection</td>
<td>Region Matching + Particle Swarm Optimization</td>
<td>Fully</td>
<td>2017</td>
<td>Airway</td>
<td>CT Depth Map</td>
<td>✓</td>
<td>334 frames</td>
</tr>
<tr>
<td>50</td>
<td>Jin [100]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2017</td>
<td>Airway</td>
<td>CT</td>
<td>X</td>
<td>40 scans</td>
</tr>
<tr>
<td>51</td>
<td>Meng [144]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Semi</td>
<td>2017</td>
<td>Airway</td>
<td>CT</td>
<td>✓</td>
<td>20 scans</td>
</tr>
<tr>
<td>52</td>
<td>Mosinska [155]</td>
<td>Binary Segmentation</td>
<td>Active Learning</td>
<td>Semi</td>
<td>2017</td>
<td>Retinal</td>
<td>Microscopy</td>
<td>X</td>
<td>Not mentioned</td>
</tr>
<tr>
<td>53</td>
<td>Luo [132]</td>
<td>Binary Segmentation</td>
<td>Symmetry filters</td>
<td>Fully</td>
<td>2017</td>
<td>Cerebral</td>
<td>TOF MRA</td>
<td>X</td>
<td>50 scans</td>
</tr>
<tr>
<td>54</td>
<td>Monti [153]</td>
<td>Binary Segmentation</td>
<td>Iterative thresholding</td>
<td>Fully</td>
<td>2017</td>
<td>Cerebral vein</td>
<td>MR</td>
<td>X</td>
<td>12 scans</td>
</tr>
<tr>
<td>55</td>
<td>Ward [222]</td>
<td>Segmentation</td>
<td>shape-based MRP</td>
<td>Fully</td>
<td>2017</td>
<td>Cerebral</td>
<td>QSM</td>
<td>X</td>
<td>4 scans</td>
</tr>
<tr>
<td>56</td>
<td>Bates [17]</td>
<td>Binary Segmentation</td>
<td>Random Forest and MRF</td>
<td>Fully</td>
<td>2017</td>
<td>Tumor</td>
<td>Multi-photon Microscopy</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>No.</td>
<td>Paper</td>
<td>Formulation</td>
<td>Method</td>
<td>Automatic</td>
<td>Year</td>
<td>Anatomy</td>
<td>Modality</td>
<td>EB</td>
<td>Case No.</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>----------</td>
<td>----</td>
<td>---------</td>
</tr>
<tr>
<td>57</td>
<td>Novikov [163]</td>
<td>Sequential Tree Extraction</td>
<td>Centerline Tracking</td>
<td>Fully</td>
<td>2017</td>
<td>Neck &amp; abdominal</td>
<td>CTA</td>
<td></td>
<td>30 scans</td>
</tr>
<tr>
<td>58</td>
<td>Hong [91]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2017</td>
<td>Coronary</td>
<td>CTA</td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>60</td>
<td>Dasgupta [50]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2017</td>
<td>Retinal</td>
<td>Color Fundus Photography</td>
<td></td>
<td>40 images</td>
</tr>
<tr>
<td>61</td>
<td>Yu [238]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2017</td>
<td>Heart</td>
<td>MR</td>
<td></td>
<td>20 scans</td>
</tr>
<tr>
<td>63</td>
<td>Wu [234]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2018</td>
<td>Retinal</td>
<td>Color Fundus Photography</td>
<td></td>
<td>Multiple Sources</td>
</tr>
<tr>
<td>65</td>
<td>Taha [206]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2018</td>
<td>Renal</td>
<td>CT</td>
<td></td>
<td>236 subjects</td>
</tr>
<tr>
<td>67</td>
<td>Zhao [243]</td>
<td>Binary Segmentation</td>
<td>Region Growing</td>
<td>Fully</td>
<td>2018</td>
<td>Airway</td>
<td>CT</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>68</td>
<td>Moriconi [154]</td>
<td>Binary Segmentation</td>
<td>Filter Bank &amp; Levelset &amp; MST</td>
<td>Fully</td>
<td>2018</td>
<td>Cerebral</td>
<td>Multi-modality</td>
<td></td>
<td>62 scans</td>
</tr>
<tr>
<td>69</td>
<td>Mathai [140]</td>
<td>Binary Segmentation</td>
<td>Levelset + Extended Karlman Filter</td>
<td>Fully</td>
<td>2018</td>
<td>Hand</td>
<td>Ultrasound</td>
<td></td>
<td>35 sequence</td>
</tr>
<tr>
<td>70</td>
<td>Selvao [193]</td>
<td>Binary Segmentation</td>
<td>MRF &amp; Deep Learning</td>
<td>Fully</td>
<td>2018</td>
<td>Airway</td>
<td>CT</td>
<td></td>
<td>32 scans</td>
</tr>
<tr>
<td>71</td>
<td>Zhao [250]</td>
<td>Enhancement</td>
<td>Symmetric Filter</td>
<td>Fully</td>
<td>2018</td>
<td>Generic</td>
<td>CT</td>
<td></td>
<td>Multiple Sources</td>
</tr>
<tr>
<td>72</td>
<td>Chen [45]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2018</td>
<td>Cerebral</td>
<td>CT</td>
<td></td>
<td>15 scans</td>
</tr>
<tr>
<td>73</td>
<td>Phellian [172]</td>
<td>Binary Segmentation</td>
<td>K-means and Levelset</td>
<td>Fully</td>
<td>2018</td>
<td>Cerebral</td>
<td>4D ASL MRA</td>
<td></td>
<td>5 scans</td>
</tr>
<tr>
<td>74</td>
<td>Kandil [101]</td>
<td>Binary Segmentation</td>
<td>Bayesian and Region Growing</td>
<td>Fully</td>
<td>2018</td>
<td>Cerebral</td>
<td>MRA</td>
<td></td>
<td>200 scans</td>
</tr>
<tr>
<td>75</td>
<td>Virzi [216]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Semi</td>
<td>2018</td>
<td>Pelvic</td>
<td>MRI</td>
<td></td>
<td>35 scans</td>
</tr>
<tr>
<td>No.</td>
<td>Paper</td>
<td>Formulation</td>
<td>Method</td>
<td>Automatic</td>
<td>Year</td>
<td>Anatomy</td>
<td>Modality</td>
<td>EB</td>
<td>Case No.</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-------------</td>
<td>--------</td>
<td>-----------</td>
<td>------</td>
<td>---------</td>
<td>----------</td>
<td>----</td>
<td>---------</td>
</tr>
<tr>
<td>76</td>
<td>Na [158]</td>
<td>Binary Segmentation</td>
<td>Line Operator on Superpixels</td>
<td>Fully</td>
<td>2018</td>
<td>Retinal</td>
<td>OCT</td>
<td></td>
<td>≥ 464 scans</td>
</tr>
<tr>
<td>77</td>
<td>Zhang [241]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2019</td>
<td>Retinal</td>
<td>Color Fundus Photography</td>
<td></td>
<td>Multiple Sources</td>
</tr>
<tr>
<td>78</td>
<td>Wolterink [229]</td>
<td>Centerline Tracking</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2019</td>
<td>Coronary</td>
<td>CT</td>
<td></td>
<td>100 scans</td>
</tr>
<tr>
<td>79</td>
<td>Yun [239]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2019</td>
<td>Airway</td>
<td>CT</td>
<td></td>
<td>28 scans</td>
</tr>
<tr>
<td>80</td>
<td>Qin [178]</td>
<td>Binary Segmentation</td>
<td>Thresholding and Tensor Decomposition</td>
<td>Fully</td>
<td>2019</td>
<td>Coronary</td>
<td>X-ray Angiogram</td>
<td></td>
<td>12 sequences</td>
</tr>
<tr>
<td>81</td>
<td>Uslu [214]</td>
<td>Sequential Tree Extraction</td>
<td>Particle Filtering and Deep Learning</td>
<td>Semi</td>
<td>2019</td>
<td>Retinal</td>
<td>OCT</td>
<td></td>
<td>56 scans</td>
</tr>
<tr>
<td>82</td>
<td>Shin [197]</td>
<td>Binary Segmentation</td>
<td>Deep Learning</td>
<td>Fully</td>
<td>2019</td>
<td>Retinal</td>
<td>Color Fundus Photography</td>
<td></td>
<td>Multiple Sources</td>
</tr>
</tbody>
</table>

Table 2.1: Categorical view of all papers surveyed. EB: Explicit bifurcation detection. Automatic: If the algorithm explicitly indicate user input for initialization, it’s regarded as semi-automatic.
2.3 Main Categories of Segmentation Techniques

2.3.1 Previous Surveys

Quite a few surveys have been published in the field of vasculature segmentation during the past fifteen years. Considering the comprehensiveness factor, we exclude some of the survey papers from this review ([51,68,166,171,177]). Three survey papers covering a large variety of imaging modalities and segmentation methods are summarized below.

One of the earliest survey papers in this field was published in 2004 by Kirbas et al. [106], who summarized both 2D and 3D blood vessel segmentation techniques. The authors divided techniques developed before 2004 into six major categories:

1. Pattern recognition-based;
2. Model-based;
3. Tracking-based;
4. Artificial intelligence-based;
5. Neural network-based
6. Miscellaneous

However, the proposed categorization strategy was very confusing to modern readers. For example, the model-based approach only included deformable models. The deformable model category was further divided into two sub-categories: the parametric deformable model and the geometric deformable model. For modern readers, the model-based approach is usually listed in comparison with the data-driven approach, and should cover a wide variety of topics like graph-based models and particle filtering-based models. Furthermore, since the introduction of the level set technique, the distinction between parametric deformable models and geometric deformable models have gradually disappeared.

In 2009, Lesage et al. [121] made a comprehensive review on 3D vessel segmentation techniques. This review divided the segmentation methods into three categories:

1. Region growing;
2. Active contour (level-set);
3. Centerline-based optimization.

This categorization method had pros and cons. The pros included a detailed analysis of region growing and active contour methods, and the introduction of stochastic methods which were not covered in the work of Kirbas et al. [106]. The cons included the neglect on approaches such as non tracking-based graph methods and learning-based methods. Beside
segmentation methods, the authors discussed not only the appearance and geometry models, but also the 3D and 2D features of vessels and bifurcations, which added more perspectives than the work of Kirbas et al. [106].

A more recent review was published by Moccia et al. [152] in 2018, dividing reviewed papers into four categories:

1. Vessel enhancement;
2. Machine learning;
3. Deformable models;
4. Tracking.

Several contributions were made in the work of Moccia et al. [152]. The first contribution was, vessel enhancement was separated as a unique category inside the segmentation workflow, rather than being regarded as a pre-processing step like noise reduction and down-sampling [121]. The second contribution was, the anatomical regions were listed aside with the imaging techniques, as both of them were important factors affecting the segmentation methods. The third contribution was, a comprehensive evaluation metrics and public datasets were given in the paper. However, certain problems exist in the review paper as well. The authors didn’t mention any graph-based methods or bifurcation detection methods. Limited number of Deep learning-based methods were discussed. For the public datasets, it was unclear what type of references were provided. For the evaluation metrics, only metrics for binary segmentation results were provided and it was unknown how to evaluate sequential tree results and bifurcation point results.

In summary, although there is no universal rule for categorizing the segmentation methods, the categories should include methods such as vessel enhancement, machine learning and deep learning, deformable models and level set, centerline tracking-based methods, graph-based methods, region growing, and probabilistic tracking.

In this review chapter, we group the surveyed papers from two perspectives: model-driven and data-driven. The model-driven methods make specific assumptions on the shape or geometrical characteristics of the tubular structures, while data-drive methods mainly leverage the information from the dataset. The advantages of model-based methods include a clear assumption of the structure to be detected. However, from the synthetic data result of Chapter 5 we can see that, these assumptions could become an obstacle when the encountered structures are very different from expected and the model-based methods don’t have a mechanism to learn directly from the data. On the other-side, data-driven methods don’t have a strong assumption how the structure should look like, but learn from the dataset instead – the disadvantage is these type of methods usually need very large datasets and face over-fitting and under-fitting problems.

For model-driven methods, we propose to classify the models into four categories:
1. Enhancement-based;
2. Thresholding;
3. Graph-based;

where the progressive methods could be further divided into four sub-categories:

i. Region growing;
ii. Deformable;
iii. Centerline Tracking;
iv. Particle Filtering.

For data-driven methods, we divide the methods into non deep learning-based for manual feature generation, and deep learning-based methods. The pros and cons of the two categories are listed in Table 2.2. We also review all bifurcation detection techniques in Section 2.4, as this is a subordinate task in vessel segmentation with a general interest. Evaluation metrics and available public datasets are reviewed in Section 2.5.

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model-driven</td>
<td>Needs few/no training data</td>
</tr>
<tr>
<td></td>
<td>Hard to handle shape irregularity</td>
</tr>
<tr>
<td>Data-drive</td>
<td>Learns irregular shapes</td>
</tr>
<tr>
<td></td>
<td>Needs large training set</td>
</tr>
</tbody>
</table>

Table 2.2: Pros and cons of model-driven methods versus data-driven methods.

### 2.3.2 Model-driven Segmentation Methods

<table>
<thead>
<tr>
<th>Enhancement</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hessian-based</td>
<td>Frangi [67], Feuerstein [63], Hannink [85], Jerman [98,99], Hennersperger [88], Annunziata [10], Zhao [242]</td>
<td></td>
</tr>
<tr>
<td>non-Hessian</td>
<td>Turetken [210], Zhou [251], Berks [25], Na [158], Lau [114], Tan [207], Zhao [250], Luo [132], Tagkalakis [205]</td>
<td></td>
</tr>
<tr>
<td>Tresholding</td>
<td>Goyal [74], Brozio [30], Mu [156], Qin [178], Monti [153]</td>
<td></td>
</tr>
<tr>
<td>Graph Based</td>
<td>Bruyninckx [31], Gonzalez [72], Kitamura [107], Skibbe [206], Wang [217], Shin [196], Ward [222], Martin [139], Cao [36], Bates [17]</td>
<td></td>
</tr>
<tr>
<td>Progressive</td>
<td>Region Growing Graham [76], Fabijanska [56], Lo [126,127], Kandil [101], Rumi [180]</td>
<td></td>
</tr>
<tr>
<td>Deformable</td>
<td>Law [116], Tang [208], Liat [124], Chen [44], Peaold [169], Phellian [172], Moriconi [154], Mathai [140]</td>
<td></td>
</tr>
<tr>
<td>Tracking</td>
<td>Bulow [32], Cetin [39], Alvarez [6], Novikov [163]</td>
<td></td>
</tr>
<tr>
<td>Particle Filtering</td>
<td>Lesage [122], Lee [119]</td>
<td></td>
</tr>
</tbody>
</table>
Table 2.3: A summary of model-driven methods.

**Vessel Enhancement**

Enhancement of blood vessels or airways could be seen either as a separate task other than segmentation, or could be seen as itself a segmentation approach. From the former perspective, the enhancement approach could be combined with other segmentation approaches to significantly improve the results [133]. From the latter perspective, after applying the enhancement, thresholding could be further applied to achieve a binary segmentation. Main enhancement filters could be further divided into two groups: 1) multiscale second-order derivative-based (e.g., Hessian) filters, and 2) Non-Hessian based filters.

1. **Multiscale second order derivative based filters.**

   The category of second order derivative filters comprises two sub-categories: i) the Laplacian of Gaussian operators (Feuerstein *et al.* [63]); ii) the combination of eigenvalues of Hessian matrix via multi-scale Gaussian derivatives (Frangi *et al.* [67]).

   The reason why second order derivative based filters could work is given in the following. In the case of 3D images $I$ (assume $I$ is pre-convoluted by a Gaussian kernel, so it’s always differentiable), we use $\lambda_1$, $\lambda_2$ and $\lambda_3$ to represent the three eigenvalues of $H$, which is the Hessian matrix of $I$. With the presence of a tubular shape, the eigenvalues should satisfy:

   \[ 0 \approx |\lambda_1| < |\lambda_2| \approx |\lambda_3| \]  

   (2.5)

   with corresponding eigenvector $\vec{v}_1$, $\vec{v}_2$ and $\vec{v}_3$. Intuitively, $\vec{v}_1$ points to the direction of the vessel branch, while $\vec{v}_2$ and $\vec{v}_3$ span the cross sectional plane perpendicular to the vessel branch.

   Sato’s work [189] was one of the earliest work which applied eigenvalue analysis to tubular and curvilinear structure reconstruction in 3D medical images. The proposed method used only two eigenvalues to enhance vessel structures and couldn’t treat dark lines and bright lines equally. Frangi’s work [67] extended the idea by proposing a unified framework, which integrated all three eigenvalues to distinguish among vessel-like, plate-like and blob-like images with high noise.

   In the work of Feuerstein *et al.* [63], a multiscale sharpening filter was built using the modified Laplacian of Gaussian, on the purpose of separating the airway wall and surrounding tissues in a volume of interest (VOI). Then the airway region inside the VOI was extracted by region growing method.
Jerman et al. [98,99] proposed a modified Frangi’s filter. By using a cutoff threshold on $\lambda_3$ to generate a regularized $\lambda_{p}$, and another cutoff ratio between $\lambda_2$ and $\lambda_{p}$ (so to suppress noise), the authors claimed they could better enhance structures with high curvatures and branching regions, than the original Frangi.

Manniesing et al. [138] introduced a modified Frangi’s kernel to build a PDE based anisotropic diffusion tensor:

$$I_t = \nabla \cdot (D \nabla I)$$

(2.6)

where $D$ is the diffusion tensor constructed using Frangi’s vesselness and corresponding eigenvectors, and $I$ is the image to be enhanced.

In the work of Hannink et al. [85], an improved version of Frangi’s filterness was established by using orientation scores built upon a multiscale cake wavelet and Gaussian derivatives, to enhance both vessels and crossings in 2D retinal images.

In the work of Anunziata et al. [10], the authors presented an anisotropic Gaussian kernel based Hessian filter, which was invariant to rotation, scale and curvature changes.

Hennersperger et al. [88] proposed a modified Frangi filter to accommodate the difference brought by ultrasound imaging of blood vessel, so that ultrasound beam wave directions, ring thickness and speckle noise could be taken into consideration.

In the work of Zhao et al. [242], a Hessian based vesselness was proposed by using bi-Gaussian smoothing on 3D micro-CT images in murine hindlimbs.

2. Non-Hessian based filters. For non-Hessian based filter, we further divide the filters into Gabor-filter related and other methods, such as superpixel-based, gradient flux-based and Wiener filter-based.

(a) Filters based on Gabor Filtering. In the work of Zhao et al. [250], a 3D volumetric image was understood from a Fourier Optics point of view and quadrature filter was used to identify local phase information. The authors adapted the original log-Gabor filter to 3D by incorporating a rotationally symmetric angular Gaussian function with blurring and shifting operations to tolerate for scales and positions.

Lau et al. [114] developed an approach, combining Gabor filter and morphological filter to detect blood vessels in 2D breast tomosynthesis images.

In the work of Tan et al. [207], the authors used a combination of Gabor filter, thresholding and line collation techniques to extract retinal vessels as well as identifying the branches.

Luo et al. [132] designed a 3D symmetry filter, which was based on blurring and shifting operations on a log-Gabor filter.
(b) **Others.** In the work of Turetken *et al.* [210], the idea of Optimally Oriented Flux (OOF) was extended by using two directions on the cross-sectional plane for measuring gradient flux. In this way, a non-circular cross sectional shape could be described (by interpolating the two perpendicular vectors) and the two directions were optimized in an alternative approach.

In the work on Na *et al.* [158], 2D retinal images were enhanced by line operators on multiple discrete directions to find the largest average gray level, on the basis of superpixels.

In the work of Tagkalakis *et al.* [205], finger veins were first extracted using filters based on histogram equalization, brightness normalization and Wiener filtering, before pattern extraction and thresholding for authentication test.

**Thresholding**

Once a probability map or an enhanced image is obtained, a certain method is required to turn the enhanced image into a binary mask, thus the thresholding technique comes into play.

In the work of Goyal *et al.* [74], Frangi enhancement and Otsu/Sauvola thresholding was combined to obtain binary vessel masks in 3D fluorescence cryomicrotome images.

Brozio *et al.* [30] proposed to extract carotid arteries using thresholding inside bounding boxes, followed by connected component operation analysis on each slice.

Local thresholding was used to obtain vasculatures in 3D microscopy images in the work of Mu *et al.* [156].

To segment blood vessel in X-ray sequences, Qin *et al.* [178] proposed a method, in which vessel segmentation was first performed on each X-ray slice, then all the binary masks were combined together and an optimization over all slices was achieved using tensor completion.

In the work of Monti *et al.* [153], vesselness maps from multiple modalities (e.g., QSM and SWI, which were reconstructed from MR images) were iteratively thresholded to obtain a binary mask.

**Graph based methods**

Another major category of model-based mis optimized so to extract vessel tree structures according to certain regularization rules. We categorize these methods as graph-based methods.

Some graph-based methods use classical representations such as Graph-cut, CRF and MRF. For the purpose of renal artery segmentation, Wang *et al.* [217] proposed a method based on Hessian based tensor graph. Since the tensor space is Riemann space, the authors claimed that they could properly define metrics like distance, mean and variance and then use them in a graph cut method to obtain the segmentation result. In the work
of Kitamura et al. [107], coronary lumen and plaque were segmented using a multi-label Conditional Random Field (CRF) model, with higher order shape prior constraint to distinguish coronary lumen and plaque. For the purpose of segmenting coronary vessels in X-ray sequences, Shin et al. [196] proposed to extract vessels from the first frame in an X-ray sequence by either manual or existing automatic methods. The tubular tree was matched to subsequent frames by Caahnfer Matching first, then optimized according to the current frame features using key-points (such as bifurcations and endpoints) correspondence in a local searching style. The vessel hierarchy and local features were kept using Markov Random Field (MRF) optimization. To extract cerebral veins in QSM images, Ward et al. [222] proposed to use Markov Random Field, with tubular shape information encoded using centerline direction into the tree graph. In the works of Bates et al. [17], multi-photon microscopy images were first oversegmented into superpixels, then by learning features using a Random Forest, the outcome map could be further used as the unary term for the MRF model for segmentation purposes.

Another strategy is to treat the natural centerline tree as a tree graph structure. In the work of Gonzalez et al. [72], a dentriteness probability map was first generated by applying Support Vector Machine (SVM) on rotational features, which were obtained using higher order Gaussian derivaties in dendrite stacks. A reduced graph was formed by searching the local maxima in the gerenated probability map. Then a K-cardinal tree was extracted for every subset of K nodes from all the local maxima points and the one with the maximal Bayesian probability is selected as the optimal tree.

Different energy functions could be defined on such graph representations to embed the tree shape. Bruyninckx et al. [31] proposed an energy function on the graph of all bifurcations, to evaluate the total volume size, the Murray’s Law as well as likelihood on pixel-wise image features. Since the optimization on such a graph is NP-hard, the Ant Colony Optimization approach was adopted. A special type of graph based method, particle based Monte Carlo method, was proposed by Skibbe et al. [200] for cerebral vessel extraction. A particle was a vertex with attributes like direction and scale, and energy functional are built using the particles, with constraints to prevent loops and inconsistent edge connections. The energy was optimized using proposals of Monte Carlo type for moving, generating, deleting, rotating and connecting particles.

Progressive methods

Progressive method is a popular category of segmentation methods in medical image analysis. It’s especially useful in tree shape structure extraction, because it can avoid post-processing problems like gap filling [18] and provide connected binary masks.

\[^2\]https://en.wikipedia.org/wiki/Ant_colony_optimization_algorithms
We define progressive methods as any method that starts from given seed points or regions (either given manually or automatically generated), and uses an iterative approach to update the internal region or foreground point set. Four different approaches could be categorized into the progressive group: i) region growing; ii) deformable models; iii) centerline tree tracking; iv) particle filtering.

**Region Growing**  In the work of Fabijanska [56], a two pass region growing method was proposed, using two different growing schemes for higher and lower order airway branches, so to avoid common leakage problems in airway extraction. In the first pass, a seed point was located automatically on the first slice of the central oval area of the trachea, and a strict algorithm defining the intensity level range, and bronchi up to 5 generations were detected. In the second pass, a relaxed growing scheme was performed on a constraint map built using morphological gradient, and bronchi up to 10 generations were found. By doing the two-pass growing method, the authors claimed to achieve a “leaking prevention mechanism”.

Kandil et al. [101] introduced a cerebral vessel segmentation method. The initial segmentation was first obtained by Bayesian classifier. Then local search and thresholding was applied to obtain seeds for a refinement vessel segmentation, by using region growing and majority voting methods.

A growing method was designed by Lo et al. [126, 126], which started by searching for minimal paths between current point and candidate local points. The paths were built on an airway probability map generated using KNN classifier and features like tubularness, direction. The candidate points were selected using a set of rules like airway straightness and non-overlapping topology.

Graham et al. [76] proposed to use a two-fold region growing scheme together with a graph partitioning algorithm. They extracted major airway branches by a first-fold seeded region growing, then extracted finer branches by running a second-fold locally-adaptive growing method on a filtered image. Then the graph partitioning algorithm decided which branch segments should be excluded from the final segmentation.

The fuzzy connectivity method was a special type of growing method, which required a seed region and a cost function for segmentation purposes. In the work of Rizi et al. [180], an assumption of cylindrical shape prior was made by encoding derivation along the angular parameter in polar coordinate expression. By minimizing the cost function built on cylindrical assumption and local membership connectivity, a binary segmentation result could be extracted.

In the work of Zhao et al. [243], a region growing approach was designed for airway segmentation, which adaptive threshold and leakage detection and correction process.
Deformable Model / Front Propagation Methods By mimicking the pressure generated by the liquid on solid surfaces of curved objects, Law et al. [116] proposed to use a deformable surface model to segment tubular objects, by combining surface stress force and the bulk stress force using second order tensors.

In the work of Chen et al. [44], the artery contour was first initialized by user manual segmentation on two slices. Then a parametrical cylinder surface was generated. By evolving the surface on 2D slices, the surface is further evolved using a deformable model with local stiffness and external force fields.

To segment blood vessels in ASL MRA image series, Phellan et al. [172] proposed a method by segmenting the first volume in a 4D using k-means (based on the assumption that only large vessel are contained) and refining the result by level set method. Then the result was transferred to the following images in the series by OR operation and image foresting transform operation.

Liang et al. [124] established a modified Chan-Vese model using vessel wall probability information. Then a local-global vessel association was performed to reduce disconnectivity, by using a Maximum A Posteriori (MAP) method.

Pezold et al. [169] proposed to use Frangi’s vesselness [67] to build an anisotropic total variation energy for segmenting tubular structures in 3D. Global optimization could be achieved by using continuous max flow.

In the work of Tang et al. [208], the centerlines of carotid arteries were first extracted using user given seed points and minimal paths. Then a geometric active contour method with gradient and calcification exclusion terms was used to extract the carotid lumens. Side branches are removed using the envelop of maximum inscribed spheres.

For the purpose of extracting cerebral vessel structures from generic modality images, Moriconi et al. [154] proposed to use a filter bank of steerable Laplacian of Gaussian filters. Then to join filtered results of multiple sources, a level set propogation was used. Eventually, a minimal spanning tree algorithm was used to obtain a tree graph.

Mathai et al. [140] designed a level set segmentation approach on the initial image of the ultrasound sequence, then the result is propagated by using extended Kalman filter.

Particle Filtering Instead of modelling the tubular shape explicitly, a Monte-Carlo genre of methods, called Particle Filtering, tracks the vessel path by randomly sampling over particles in a Bayesian way:

\[ p(s_t|I_{1:t}) = \frac{p(I_t|s_t)p(s_t|I_{1:t-1})}{p(I_t|I_{1:t-1})} \]  

(2.7)

where \( I_t \) is the condition at step \( t \) (each step a group of particles are thrown out randomly. The best one is chosen at the end according to Equation 2.7), and \( p_t \) is the particle at step \( t \).
Condition $I_t$ is usually the image feature, but could also be generated using other methods like Deep Learning [232].

In the work of Lee *et al.* [119], a slice-by-slice particle filtering tracking method was proposed. Multiple particles were thrown at each slice to infer the slice center using Equation 2.7. The segmentation is refined by running an extra few iterations of Chan-Vese algorithm.

Lesage *et al.* [122] introduced a kernel density estimator, and a new sampling technique called the Adaptive Auxiliary Particle Filtering (AAPF). AAPF was used for inferring image cue correlations between consecutive particles.

**Centerline Tree Tracking** Another category of tree extraction algorithms is based on iterative centerline tracking, which usually starts from a given seed point near the root of the tree, predicts the direction of the branch to track along, detects bifurcations to spawn children trackers, and progresses down the tree hierarchy to smaller branches until certain stopping criterion is met.

Bulow *et al.* [32] first introduced a general tree tracking framework, by starting from one or more user-defined seed points. A branch segment queue is formed by iteratively searching for the neighboring pixels of the candidate segment and detecting bifurcations, until a stopping criterion is met.

In the work of Alvarez *et al.* [6], instead of predicting the tracking direction of the aorta, the authors proposed to predict the cross section plane first. By minimizing an ellipse shape and image feature based energy in 3D space, the tracking direction is simultaneously predicted.

Cetin *et al.* [38] proposed to track a coronary tree by iteratively measuring the intensity distribution within an oriented cylinder-sphere combined model and constructed a corresponding tensor representation to optimize for vessel directions, so to obtain the new search front on the coronary branch.

To search for refined orthogonal planes during the centerline tree tracking process, Grelard *et al.* [77] designed a novel tracking process by applying the Voronoi Covariance Measure in a cylindrical neighborhood to search for the optimal plane.

In the work of Novikov *et al.* [163], a centerline tree tracking method was introduced with automatic seeding and sampling-based vesselness calculation. To further connect branch segments, a genetic algorithm-based method was used.

In the work of McIntosh *et al.* [142], a parametric tubular detector based multi-layer Artificial Intelligence (AI) sensor system was proposed to detect 3D vasculatures. By projecting image information on a hemisphere, the AI system makes decisions like growing, bifurcating or terminating.
### Table 2.4: Properties of different model-driven methods in vessel tree extraction.

<table>
<thead>
<tr>
<th>Method</th>
<th>Optimization Free</th>
<th>Tree Hierarchy</th>
<th>Tubularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enhancement</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Threshold</td>
<td>✓</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Graph</td>
<td>x</td>
<td>Pre-processing dependent</td>
<td>x</td>
</tr>
<tr>
<td>Deformable</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centerline Tracking</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Particle Filtering</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Region Growing</td>
<td>✓</td>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

#### 2.3.3 Data-driven Segmentation Methods

Data-driven segmentation methods comprise mainly two categories: i) manual feature generation oriented learning methods; ii) deep learning methods with automatic feature generation.

**Machine Learning with Manual Features**

Several works identifying tubular structures in medical images used hand-crafted features, together with classifiers or regressors such as AdaBoost, SVM or Random Forest. In the work of Havla et al. [87], parameters of the Paul wavelet of order 1 were used with different classifiers (Fishers linear discriminant analysis, logistic regression analysis and support vector machine) to classify arteries from veins in the cerebral vasculatures. Annunziata et al. [9] proposed to aggregate both manual features and learnt context features and feed them into Random Forest classifier for pixelwise classification of 2D curvilinear structures. For the purpose of segmenting retinal vessels from microscopy images, Mosinska et al. [155] proposed an active learning method. The authors identified the edges prone to be misclassified and would possible impose a strong effect in the curvilinear structure graph, then seeked the annotation from an annotator in an iterative way. Breitenreicher et al. [29] used a two-layer classification hierarchy for airway segmentation in CT images. The first layer was built by using 3D Haar features and probabilistic boosting classifier, while the second layer was constructed using Dijkstra algorithm, and branches of high probability values on the final map was aggregated to form a tree. In the work of Lu et al. [131], the learning procedure was broken down into three steps: first, coarse centerlines were extracted by steerable features and dynamic programming; second, multiscale sparse learning was used to predict centerline probabilities; third, coronary vessel surfaces were further extracted by
Markov Random Field segmentation. Caresio et al. [37] proposed 7 hand-crafted features for general feature extraction on thyroid nodule in ultrasound images: vessel length, concavity changes, angle changes, volumetric density, tree numbers, branch numbers and vascularity pattern. A generic pipelines for segmenting linear structure in retinal and neuron images was proposed in the work of Gu et al. [79] using structured and context distance features collected on random patches over the whole image. To delineate neuron structures in microscopy and micrograph images, Sironi et al. [199] proposed a method to first learn the distance transform map through iterative regression, using a learning-based Gradient Boost algorithm [86] on local image patches across multiple scales. Then the centerlines were extracted by finding local maxima after a non-maximum suppression operation. In the work of Berks et al. [25], a random forest based regressor was trained on each slice to predict capillary vessel orientation and widths.

**Machine Learning with Deep Architectures**

On the other hand side, deep learning-based methods have gained huge popularity among researchers ever since 2012 [111], which no longer needed manual design of images features, but to learn directly from the data.

Most works using deep learning for tubular structure extraction were based on convolutional neural networks (CNNs), such as AlexNet, Unet, Inception, auto-encoder, and dilated CNN, while a few other works used non CNN-based architectures like DBN and RNN. Wu et al. [232] used the AlexNet to predict patch-wise PCA coefficients and integrated in a probabilistic tracking pipeline. Unet was used by itself [206] or in combination with other methods such as region growing [100], Deep Supervision [241] or centerline tree tracking [144]. To segment multi-width retinal vessels, Wu et al. [234] proposed to use parallel stacked Unets together. In the work of Tetteh et al. [209], a modified Inception network (with increased sub-layers in the Inception module) was used for feature prediction in 2D retinal images. Wolterink et al. [228] introduced a dilated 3D CNN to predict direction and radius during coronary vessel centerline tracking. For binary segmentation of 3D coronary vessels, Hong et al. [91] designed a new model using stacked auto-encoder and CNN. In the work of Virzi et al. [216], after extracting pelvic vasculature candidates with an iterative tree connection scheme, 3 consecutive patches along the candidate vascular path were input into a modified VGG-16 Network [198]. For non CNN-based architectures, RNN and CNN were combined to generated a deep CRF in the work of Fu et al. [69]. In the work of Uslu et al. [214], a Deep Belief Network was used to predict a trio of binary mask, centerline probability map and edge probability map for parameter estimation in Particle Filtering tracking in 2D retinal image segmentation.

We noticed two phenomenons in the surveyed deep learning papers. One phenomenon is, some of the works chose to combine the power of deep learning with classical methods such as particle filtering [214,232], region growing [100], MRF [95], centerline tree tracking [144,228]
and connected component labeling [239], or simply use the deep network to generate features and use non deep learning-based classifiers for prediction [91].

The other phenomenon is, although the deep models were supposed to be adapted to delineate tubular structures, limited works discussed the possibilities of exploring the tree structures or tubular shapes in the model choices or modification. When it comes to loss function design, most deep models simply used the cross entropy loss [69,100,144,206,234,241], L2 loss [214] or Euclidean loss [198], and none of them have been adapted for tubular shape detection. Some works tried to encode the tree shape in a higher shape space, or to interpret tubularity as a certain form of sparsity. A deep mean field network was introduced by Selvan et al. [193] to optimize a graphical model, which was built on a 7-dimensional space to capture tree structures. In the work of Chen et al. [45], a modified 3D Unet was initiated by adding extra layers, so that the prediction could be performed under different resolutions so to preserve the sparsity of the foreground shape and using a higher weight in the loss for foreground voxels.

2.4 Bifurcation Detection in Vascular Images

Vascular or airway bifurcations are locations where a major branch bifurcate into two (sometimes more) sub-branches (e.g., trifurcation could exist in the right carotid [46]). It’s important to detect bifurcations since they serve as important landmarks (reference) and could provide crucial information for further analysis (such as aneurysm growth [28]). In this section, we summarize different bifurcation models proposed in the literature. Specifically, we are interested in how researchers understand the bifurcation from the feature design and geometrical perspectives.

As analyzed in Section 2.3.3, an important category of segmentation methods were based on classification using manual-designed features. In the work of Srinidhi et al. [202], bifurcations were detected using log-polar transform generated features and classified using random forest classifier. In the work of Zhou et al. [253], bifurcations were found by extracting image features using 2D Gaussian filters on cross sectional planes estimated by Hessian filters and AdaBoost classifier. In the work of Alberti et al. [4], the authors gathered features on the angular sector (cross sectional plane) of the blood vessel and input the features into a multi-scale stacked sequential learning scheme for pixel-wise classification. Then the false positives were removed by morphological filtering and its longitudinal information. In the work of Shen et al. [195], bifurcation were classified in CT depth map images by SVM classifier using parametric and statistical features, and then registered to bronchosopy videos for localization purposes. In the work of Khorshed et al. [105], blood vessels in the mice bone marrow were segmented by multi-resolution thresholding technique. Then bifurcations were detected by running Decision tree classifier on image location as well as morphological features.
For model based methods, most models for bifurcations were based on cylindrical shapes, connected components on hemispherical masks and splitting of geodesic paths. In the work of Cetin et al. [38], a tensor was built by counting the candidate particles in the surrounding cylindrical and spherical areas to model bifurcation in coronary images (Figure 2.2d). In the work of McIntosh et al. [142], a bifurcation was found if an extra circular region was found on the projected hemisphere (Figure 2.2b). An airway bi-/trifurcation (up to 3 sub branches) was located in the work of Feuerstein et al. [63] by determining the number of connected components on the segmented binary surface in the current volume of interest (VOI). In the work of Grélard et al. [77], a bifurcation was simply found as the connected component number on the spherical shell equals to three. A scheme determining the intersection area centerlines and directions was also proposed (Figure 2.2c). In the work of Brozio et al. [30], a point was connected with surrounding candidate points using Dijkstra’s algorithm. If there exists a divergence among the candidate paths, a bifurcation was found (Figure 2.2a). In the work of Bruyninckx et al., potential map was first trained using SVM, and the bifurcations were located as the splitting points of geodesic (from any voxel to the root) bundles. Vessel segments were determined using minimal paths between bifurcation pairs. Eventually the whole tree was extracted by optimizing a graph using Ant Colony Algorithm.

In summary, an overview of the geometrical models used in the literature is given in Figure 2.2.

Other bifurcation detection methods involved clustering, classification with deep networks, or simply use implicit embedding of graphical models. In the work of Lesage et al. [122], bifurcation existence could be observed by the fluctuation of particle numbers. Mean shift clustering was used to divide generated particles for detecting bifurcations and daughter branches. For most graph based (or particle graph based) methods, bifurcations were modeled implicitly as vertices having more than two edges [200]. The work of Wu et al. [232] used an ImageNet classifier [111] to classify whether the current point is a bifurcation or not. If a bifurcation was classified, then a half ellipse was drawn to determine the number of sub-branches and the new locations to start tracking from.
Discussions. There are a few challenges arising from bifurcation detection problem. For one thing, most of the works discussed here are model-driven methods, so the discussion of using data-driven based methods could be further proceeded. Second, especially with data-driven methods, like in Wu’s work [232], after bifurcation classification with a deep model, subsequent tasks like daughter branch locations were still approached with traditional model-based methods.

2.5 Evaluation Metrics and Public Dataset

Once a segmentation result (in the form of binary segmentation, centerlines or bifurcation points) is obtained, we need a certain approach to evaluate the result. In this section, we’ll discuss a variety of evaluation metrics corresponding to different types of segmentation results. A list of public dataset is given in Section 2.5.2.
2.5.1 Evaluation Metrics

As discussed in Section 2.1, three different formulations of segmentation exist. Since the corresponding segmentation results have diverse forms, distinct evaluation metrics are necessary in comparing these results (see Table 2.5). Detailed definition of each evaluation metric will be given in the following sections. We can observe from Table 2.5 that, some metrics could be applied on certain formulations but not others. This indicates pros and cons of each formulation, e.g., binary segmentation is not a good option when tree generation and branch counts are desired from the result.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Binary</th>
<th>Centerline</th>
<th>Bifurcation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dice</td>
<td>✓</td>
<td>Applicable</td>
<td>Applicable</td>
</tr>
<tr>
<td>Jaccard</td>
<td>✓</td>
<td>Applicable</td>
<td>Applicable</td>
</tr>
<tr>
<td>TP/TN/FP/FN</td>
<td>✓</td>
<td>Tolerance Factor</td>
<td>Tolerance Factor</td>
</tr>
<tr>
<td>Surface Distance (maxSDist, meanSDist)</td>
<td>✓</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>✓</td>
<td>Tolerance Factor</td>
<td>Tolerance Factor</td>
</tr>
<tr>
<td>Precision</td>
<td>✓</td>
<td>Tolerance Factor</td>
<td>Tolerance Factor</td>
</tr>
<tr>
<td>Accuracy</td>
<td>✓</td>
<td>Tolerance Factor</td>
<td>Tolerance Factor</td>
</tr>
<tr>
<td>FAR</td>
<td>✓</td>
<td>Applicable</td>
<td>Applicable</td>
</tr>
<tr>
<td>$GT_{percent}$</td>
<td>✓</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>ROC Analysis</td>
<td>✓</td>
<td>Applicable</td>
<td>Applicable</td>
</tr>
<tr>
<td>Hausdorff Dist/Distance Transform</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Branch Count</td>
<td>NA</td>
<td>✓</td>
<td>Applicable</td>
</tr>
<tr>
<td>F-measure</td>
<td>✓</td>
<td>Applicable</td>
<td>Tolerance Factor</td>
</tr>
<tr>
<td>Tree Generation</td>
<td>NA</td>
<td>✓</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 2.5: A summary of surveyed evaluation metrics in: binary segmentation (Section 2.5.1), centerlines extraction (Section 2.5.1) and bifurcation detection (Section 2.5.1). Tolerance factor: the metric has been re-defined with a tolerance factor. Applicable: the metric hasn’t been defined, but the definition is applicable in this type of data.

Without the loss of generality, we define in the following that TP as the numbers of true positives pixels, FN and FP the numbers of false negatives and false positives pixels, in the reference segmentation and predicted segmentation, respectively.
Evaluation on binary segmentation results

1. Dice overlap between the proposed segmentation and the ground truth [31,55,169,208]:

\[
\text{Dice} = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}
\]  

(2.8)

2. Mean of distance transform of the ground truth segmentation (GT) evaluated in the proposed segmentation (DT) [31], and vice versa:

\[
\text{meanDist}_{DT}(GT) = \text{mean}\{\Omega(x) | x \in GT, \Omega = \text{distTransform}(DT)\} 
\]

(2.9)

\[
\text{meanDist}_{GT}(DT) = \text{mean}\{\Omega(x) | x \in DT, \Omega = \text{distTransform}(GT)\} 
\]

(2.10)

3. True positive volume fraction (TPVF), false positive volume fraction (FPVF) and false negative volume fraction (FNVF) [180]:

\[
\text{TPVF} = \frac{|V_{\text{Segmentation}} \cap V_{GT}|}{|V_{GT}|} 
\]

(2.11)

\[
\text{FPVF} = \frac{|V_{\text{Segmentation}} - V_{GT}|}{|V_{GT}|} 
\]

(2.12)

\[
\text{FNVF} = \frac{|V_{GT} - V_{\text{Segmentation}}|}{|V_{GT}|} 
\]

(2.13)

where \(V_{\text{Segmentation}}\) is the binary volume obtained with the proposed method, \(V_{GT}\) is the ground truth binary volume, and the subtraction symbol represents set subtraction.

4. Segmentation Jaccard Index [210] between proposed binary segmentation (DT) and ground truth binary mask (GT):

\[
Jaccard = \frac{|DT \cap GT|}{|DT| + |GT| - |DT \cap GT|} = \frac{|TP|}{|TP| + |FP| + |FN|} 
\]

(2.14)

5. Max and mean surface distance (maxSDist, meanSDist) between proposed segmentation surface (\(S_{DT}\)) and ground truth segmentation surface (\(S_{GT}\)) [208]:

\[
\text{maxSDist} = \max_{x \in S_{DT}} \{\text{dist}(x, S_{GT})\} 
\]

(2.15)

\[
\text{meanSDist} = \text{mean}_{x \in S_{DT}} \{\text{dist}(x, S_{GT})\} 
\]

(2.16)

6. Sensitivity [69,251]:

\[
\text{Sensitivity} = \frac{TP}{TP + FN} 
\]

(2.17)
7. Precision [119, 251]:

\[
Precision = \frac{TP}{TP + FP} \tag{2.18}
\]

8. Accuracy [69, 251]:

\[
Accuracy = \frac{|TP| + |TN|}{|P| + |N|} \tag{2.19}
\]

9. False Alarm Ratio (FAR)

\[
FAR = \frac{FP}{TP + FN} \tag{2.20}
\]


\[
F\text{-measure} = \frac{2 \cdot Precision \cdot Sensitivity}{\text{Precision} + \text{Sensitivity}} \tag{2.21}
\]

11. Percentage of ground truth skeleton (\(GT_S\)) voxels contained in the proposed segmentation (DT) [31]:

\[
\text{GT}_{\text{percent}} = \frac{|DT \cap GT_S|}{|GT_S|} \cdot 100\% \tag{2.22}
\]

12. ROC curve analysis. For probabilistic results: For probabilistic segmentation, ROC analysis with regard to threshold value could be used for performance evaluation. For binary segmentation, the results are first turned into distance transform images, then ROC analysis with regard to threshold value on the could be used for performance evaluation [184]

**Evaluation on centerline results**

1. Hausdorff distance between proposed centerline to ground truth centerline [29]

\[
Hausdorff(DT, GT) = \max_{x \in DT} \min_{y \in GT} \text{dist}(x, y) \tag{2.23}
\]

2. True positive with tolerance factor [210]:

\[
TP_\rho(DT) = \{x | \exists y \in GT s.t. |x - y| \leq \rho\} \tag{2.24}
\]

3. Sensitivity (with distance tolerance) [251]:

\[
\text{Sensitivity}_\rho = \frac{TP_\rho}{TP_\rho + FN_\rho} \tag{2.25}
\]

4. Precision (with distance tolerance) [251]:

\[
\text{Precision}_\rho = \frac{TP_\rho}{TP_\rho + FP_\rho} \tag{2.26}
\]
5. Average distance from ground truth centerline to proposed centerline, and vice versa \[122,251]\]

\[
\text{averageDist}(DT, GT) = \text{mean}_{x \in DT} \min_{y \in GT} \text{dist}(x, y) \tag{2.27}
\]

6. Branch count (BC) \[167\]:

\[
BC = |\{B_i | i = 1, \ldots, n, \text{length}(B_i) \geq \tau\}| \tag{2.28}
\]

where \(B_i\) is the set of centerline points on an individual branch segment, which is detected correctly. \(\tau\) is a preset minimal branch length, which is usually set as 1 mm in clinical cases.

7. Tree length (TL) \[167\]:

\[
TL = \sum_{i=1}^{n} \text{length}(B_i) \tag{2.29}
\]

where \(B_i\) is the set of centerline points on an individual branch segment, which is detected correctly.

8. Tree generations: number of tree generations in airway centerlins \[176\].

**Evaluation on bifurcation detection**

We don’t include patch based bifurcation classification here, as these could be treated as general classification tasks \[4,74\] and only concern methods on bifurcation location detection.

1. Cumulative histogram of detected bifurcation number within a certain radius error \[31\].

2. True positive (TP) with factor \(\rho\) \[196\]: A detected bifurcation (DT) point \((x)\) is true positive, if there is a ground truth (GT) bifurcation point within \(\rho\) pixels;

\[
TP_{\rho}(DT) = \{x | \exists y \in GT \text{s.t.} ||x - y|| \leq \rho\} \tag{2.30}
\]

3. Mean distance between the estimated (DT) and ground truth (GT) point coordinate ( \[196\]).

\[
\text{meanDist}(DT, GT) = \text{mean}\{d(x, GT) | x \in DT\} \tag{2.31}
\]

4. Based on the definition of \(TP_{\rho}\), \(\text{precision}_{\rho}\) and \(\text{recall}_{\rho}\), F-measure is defined as:

\[
F - \text{measure}_{\rho} = \frac{2 \cdot \text{Precision}_{\rho} \cdot \text{Recall}_{\rho}}{\text{Precision}_{\rho} + \text{Recall}_{\rho}} \tag{2.32}
\]
Discussions

A further discussion of the metrics listed in Table 2.5 could be made from the following three perspectives:

**Distance of Sets vs. Set of Distances** With the metrics defined in Equation 2.9, 2.27, 2.31 all in the format of a single value defined on multiple sets of points, it’s hard to measure the the consistency between detected point set and the reference set. This is because for elongated shapes, an accurately detected branch with a few outlier points would result an error value depending on the outliers using distance like Equation 2.23. However, if the outliers could be tolerated in a certain sense (e.g., the outliers were on the branches of less clinical interest), the result would still be highly acceptable. In this manner, using a vector of values instead of a single value, would make more sense.

**Hierarchy in Tree Tracking** For centerline tracking methods, although metrics like Equation 2.28, 2.29 could reflect certain properties of a tree shape, there is no hierarchical information that could be interpreted. From a tree tracking point of view, even though two trees carry the same amount of branch count and generation information, a tree structure with a wrong branching pattern from the root of the tree would definitely result a much higher error.

**The Relationship among Binary, Centerline and Bifurcation Extration** Although there are three different formats of result representations, the binary segmentation, tree centerline and bifurcation locations are not totally independent of each other. A binary segmentation with centerline extraction scheme could obtain the centerline tree and bifurcation locations. The detected bifurcation locations using minimal path schemes could obtain proper centerline paths. The centerline tree with an inverse growing could obtain the binary segmentation result. So there should exist a unified framework for evaluating these different types of results.

### 2.5.2 Public Dataset

**Table of Public Dataset**

A list of public dataset containing 2D/3D vasculatures/airways are given in Table 2.6.
<table>
<thead>
<tr>
<th>Name</th>
<th>Url</th>
<th>Evaluation metric proposed</th>
<th>Segmentation reference</th>
<th>Case number</th>
<th>Anatomy</th>
<th>2D/3D</th>
<th>Pathology</th>
<th>Modality</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARE</td>
<td><a href="http://cecas.clemson.edu/~ahoover/stare/">http://cecas.clemson.edu/~ahoover/stare/</a></td>
<td>✔</td>
<td>Vessel segmentation</td>
<td>≥ 400</td>
<td>Retinal</td>
<td>2D</td>
<td>Multiple diseases</td>
<td>Color Fundus Photography</td>
<td>1975</td>
</tr>
<tr>
<td>Chasedb1</td>
<td><a href="https://blogs.kingston.ac.uk/retinal/chasedb1/">https://blogs.kingston.ac.uk/retinal/chasedb1/</a></td>
<td>✔</td>
<td>Vessel segmentation</td>
<td>28</td>
<td>Retinal</td>
<td>2D</td>
<td>Diabetic retinopathy</td>
<td>Color Fundus Photography</td>
<td>2004</td>
</tr>
<tr>
<td>DRIVE</td>
<td><a href="https://www.isi.uu.nl/research/Databases/DRIVE/">https://www.isi.uu.nl/research/Databases/DRIVE/</a></td>
<td>✔</td>
<td>Vessel segmentation</td>
<td>40</td>
<td>Retinal</td>
<td>2D</td>
<td>Diabetic retinopathy</td>
<td>Color Fundus Photography</td>
<td>2004</td>
</tr>
<tr>
<td>EXACT09</td>
<td><a href="http://image.diku.dk/exact/">http://image.diku.dk/exact/</a></td>
<td>✔</td>
<td>Vessel edges</td>
<td>40</td>
<td>Airway</td>
<td>3D</td>
<td></td>
<td>CT</td>
<td>2009</td>
</tr>
<tr>
<td>3Dircadb1</td>
<td><a href="https://www.ircad.fr/softwares/3Dircadb/3Dircadb1/">https://www.ircad.fr/softwares/3Dircadb/3Dircadb1/</a></td>
<td>✔</td>
<td>×</td>
<td>20</td>
<td>Liver</td>
<td>3D</td>
<td></td>
<td>Liver tumor</td>
<td>2009</td>
</tr>
<tr>
<td>VESSEL12</td>
<td><a href="https://vessel12.grand-challenge.org/">https://vessel12.grand-challenge.org/</a></td>
<td>✔</td>
<td>Annotated points</td>
<td>23</td>
<td>Pulmonary vessel</td>
<td>3D</td>
<td>emphysema, nodules or pulmonary embolisms</td>
<td>CT</td>
<td>2012</td>
</tr>
<tr>
<td>Coronary Artery Branch Detection Validation Database</td>
<td><a href="https://vpa.sabanciuniv.edu/sites/cabdvdv/">https://vpa.sabanciuniv.edu/sites/cabdvdv/</a></td>
<td>✔</td>
<td>Annotated points</td>
<td>10</td>
<td>Coronary</td>
<td>3D</td>
<td></td>
<td>CTA</td>
<td>2013</td>
</tr>
<tr>
<td>REFUGE</td>
<td><a href="https://refuge.grand-challenge.org/">https://refuge.grand-challenge.org/</a></td>
<td>✔</td>
<td>×</td>
<td>1200</td>
<td>Retinal</td>
<td>2D</td>
<td>Glaucoma</td>
<td>Color Fundus Photography</td>
<td>2018</td>
</tr>
</tbody>
</table>

Table 2.6: A list of public 2D/3D datasets for airway/vasculature segmentation evaluation.
Discussions

The issues arose from the current available public datasets include being learning-unfriendly, pathology-limited, modality-limited, and reference-deficient. First, although there is a bulk of literature using learning-based methods in recent year in vasculature extraction 2.3.3, we still see a huge gap between available public resources for training and testing of data-driven method. Second, there is limited data with pathological diseases. While retinal related dataset are the most availabe, the rest are only pulmonary, coronary and carotid related, while cerebral dataset, renal dataset, abdominal dataset are missing. Third, the modalities are also limited, with only color fundus photography and CT images available, while MRA, ultrasound images are not available. The last but not the least, for the public datasets available online, most don’t have the reference segmentation, or only provide some partial annotations as reference, which makes the metrics reviewed in Section 2.5.1 not useful anymore. So an emergent and crucial task in this research field, is to provide more open dataset with annotations to researchers.

2.6 Summary and Discussion

This chapter is a survey of recent literature on tubular structure segmentation in medical images, as well as public dataset and evaluation metrics. For segmentation algorithms, we divided the methods as i) model-based, such as enhancement based, thresholding based, graph-based, augmenting-based; and ii) learning-based methods.

The first question we noticed that, it’s hard to categorize the technique used in a certain paper into a single category. Rather, we see a lot of combined techniques for achieving the segmentation goal, e.g., Lo et al. [167] combined KNN with region growing, Graham et al. [76] combined region growing with graph partitioning, Lee et al. [119] combined particle filtering with Chan-Vese level set method.

The second question is, we also see a surge of using machine learning methods, especially deep network-based methods in recent years, such as Wu et al. [232], Wolterink et al. [229] and Uslu et al. [214]. We see combined approaches of deep learning and other commonly used techniques in 3D medical image analysis, such as Nearest Neighbor Search and PCA analysis [232] and Particle Filtering [214]. Moreover, deep networks could be used to predict a variety of different parameters, such as probability map of vesselness [69], vessel direction prediction [229] and particle item attributes (edge, centerline location, direction, [214]).

For bifurcation detection problems, it’s very hard to say there is a single model that outperforms all others. Brozio’s and Lesage’s models rely on candidate paths. McIntosh’s, Feuerstein’s and Grelard’s models are built on bounding box surfaces and spherical shells, and neglect some structural details (such as branching angle, branch radii ratio). Finely crafted models like Cetin’s rely on strong assumptions on bifurcation shapes and are probably prone to be affected by variations like pathologies and image artifacts.
From the list of available (if not all) public datasets, we can see that the difficulty of vasculature segmentation not only rise from the problem itself but also the lack of 3D data and proper annotations, as gold standard 3D binary segmentation references are extremely difficult to retrieve.

Since binary segmentation results, centerline tracking results and bifurcation detection results are different in morphology, different evaluation metrics are needed for performance measurement. A list of evaluation metrics used in the literature is given in Table 2.5. We can see that, although metrics like Dice and Jaccard are not directly defined on results like centerlines tracking and bifurcation detection, they could be extended using tolerance factors as true positive and sensitivity (see works [210, 251]). However, there are two metrics can’t be extended on binary segmentation results without further post-processing procedures. One is the branch count and the other is the tree generation count. From another perspective, this might give a hint why centerline tracking and bifurcation location should be addressed alongside with vasculature/airway segmentation as these structures carry important topological information.
Chapter 3

Bifurcation Detection via Novel Features and Random Forest

3.1 Introduction

According to World Health Statistics 2012 [230], cardiovascular diseases have lead to the largest proportion (48%) of non-communicable deaths in the world. One of the common cardiovascular diseases is atherosclerosis. When atherosclerosis plaques accumulate inside the artery, the walls become fragile and might cause strokes and cardiac-related deaths. The plaques are more apt to happen at bifurcation locations, thus making it important to detect bifurcations in vascular images. It also matters in problems like blood vessel quantification (e.g., measuring branch length, thickness, tortuosity), visualization, and blood flow simulation, to detect all the bifurcation locations and extract the whole set of vessels [121].

Many bifurcation extraction methods have been proposed in recent years. For 2D retinal fundus images, a classification method was proposed by Saha et al. [186], counting local connectivity in the binarized image. For 3D images, Wette et al. [225] improved the Corkscrew algorithm for bifurcation extraction in coronary arteries, which is a semi-automatic region growing algorithm requiring users input such as start/end points and intensity value bounds. In [102], a bifurcation is treated as a junction: first the vessels are tracked from a single seed given by the user, a skeletonization step is applied, then junctions are detected by examining each pixel’s 26-neighborhood. Brozio et al. [30] evaluated bifurcations by the lengths and shapes of the minimal paths among vessel candidates. Baboiu et al. [13] designed a bifurcation filter based on modeling bifurcations in scale space as the superposition of three straight vessels. Cetin et al. [38] defined a new vessel tensor and detected bifurcations by clustering the direction vectors generated by the tensors. In [142], the authors proposed a vessel crawler organism model, in which image features and prior anatomical information is fed into an artificial cognitive layer that controls the crawling deformations, detection of bifurcations, and spawning of child vessel crawlers. In [21],
carotid bifurcations are detected by mapping surrounding vessel branch information to a known anatomical database.

As pointed out by Lee et al. [117], the asymmetric characteristic of bifurcations made it hard to fix parameters for most algorithms. This has given rise to some machine learning based approaches. In [4], the authors proposed to classify bifurcations in 2D sequential IVUS images using features based on, e.g., Gabor filters and cross-correlation; comparisons were made across different classifiers, e.g., AdaBoost, random forest and Support Vector Machines (SVM). In [253], Gaussian and its 1st and 2nd derivatives on cross-sectional planes were used for detecting bifurcations with AdaBoost.

However, in the aforementioned machine learning based methods, features in [4] were specially designed for IVUS images and hard to be adapted to other modalities. The novelty of this paper lies in the three features designed for bifurcation detection in 3D CT images. In Section 3.2, our new features, as well as four others, are combined as input of a random forest classifier for building a bifurcation detector, which is further incorporated in a general tracking pipeline (Algorithm 1). In Section 3.3, two datasets, VascuSynth [81] and VESSEL12 [184] are examined. It is shown that our proposed features outperform those in [253] by a large margin in cross-validation. Both numerical and visual results show that the proposed bifurcation detector gives superior result to those obtained by the method of Macedo et al. [136].

3.2 Methodology

In this section, we detail three novel features as well as summarize existing ones that we adapted and used for bifurcation detection via a random forest classifier, which is eventually implemented within a general vessel tracking algorithm.

The novel features are designed specifically to characterize bifurcations from non-bifurcation structures. The characteristics of bifurcation structures could be distinguished by three branches (parent branch and two daughter branches) meeting at the same joint. To model these branches, we proposed to model the distributions of foreground intensities on the spherical surface using the multi-modal Von Mises-Fisher. By using the histogram of eigenvector, we intend to model the directional distributions along the three branches.

3.2.1 Von Mises-Fisher (VMF) Distribution.

VMF distribution is a directional distribution defined on a $p$ dimensional unit hypersphere $S^{p-1} \ (p \geq 2)$ [53]:

$$P(X) = c_p(\kappa)e^{\kappa \bar{\theta}^T X}, X \in S^{p-1}, \quad (3.1)$$

where $\bar{\theta}$ stands for the mean direction, $\kappa$ is the concentration coefficient and $c_p(\kappa)$ is a normalizing constant. Similar to the definition of Gaussian Mixture Model (GMM), a
mixture of $C$ VMF distributions could be written as follows:

$$p(X|\Theta) = \sum_{i=1}^{C} \alpha_i p(X|\vec{\theta}_i),$$

(3.2)

where $\alpha_i$ and $\vec{\theta}_i$ represent mixture weights and the corresponding mean directions, respectively for $i = 1, ..., C$.

For each pixel $(x, y, z)$ in a 3D image, we define $N_k(x, y, z)$ as the $k \times k \times k$ neighborhood and $\vec{v}(x', y', z')$ as the vessel direction at each pixel $(x', y', z') \in N_k(x, y, z)$. Then the observation data for determining the mixture distribution (3.2) is $\{\vec{v}(x', y', z') : (x', y', z') \in N_k(x, y, z)\}$; all parameters could be calculated using Expectation-Maximization (EM) [53]. Once mean directions $\vec{\theta}_1, ..., \vec{\theta}_C$ are obtained, we measure the differences between each pair of directions by calculating a $C \times C$ matrix $\Theta$, where $\Theta_{ij} = (\vec{\theta}_i \cdot \vec{\theta}_j) / (||\vec{\theta}_i|| ||\vec{\theta}_j||)$. Standard deviation of the elements of matrix $\Theta$ is denoted as $\theta_{std}$, for measuring the coherence of the vessel directions. We’re using both $\{\alpha_i : i = 1, ..., C\}$ and $\theta_{std}$ as our VMF feature. For a neighborhood containing a single vessel segment, the weights would ideally contain two peak values (positive and negative vessel directions) and $\theta_{std}$ is relatively large (since the direction cosine matrix is sparse); while for a bifurcation, the six weight values would be more homogeneous and $\theta_{std}$ would be small (direction cosine matrix is denser).

### 3.2.2 Histogram of Eigenvectors (HOE).

Inspired by the Histogram of Gradient (HOG) feature [49], we propose a similar feature but using estimated vessel directions instead. For the pixel $(x, y, z)$ and its neighborhood $N_k(x, y, z)$, the HOE feature is obtained through the following steps:

1. Calculate vesselness and vessel direction (eigenvector of Hessian with the smallest eigenvalue), for every pixel in $N_k(x, y, z)$, as in [67];
2. Construct a spherical histogram by casting a vote for each eigenvector on the sphere surface and weight it by the corresponding vesselness [67];
3. Normalize each bin count on the sphere surface so the total count adds up to 1;
4. Define HOE$(x, y, z)$ as the vector whose elements are only the largest $C$ values of the sorted bin counts.

Similar to VMF, the HOE of a bifurcation neighborhood would have six (antipodally symmetric) peaks versus only two for a vessel segment.
3.2.3 Concurrency of Eigenvectors (COE).

To ensure that the bifurcation is at the center of the ROI, the COE feature (with value in [0,1]) is designed as follows:

\[
COE(X) = K \left( \int_{X' \in N_k(X)} |\vec{v}(X) \cdot \vec{XX}'|/|\vec{XX}'|dX' \right)
\]  

(3.3)

\(X = (x, y, z), X' = (x', y', z')\) and \(K = (\int_{X' \in N_k(X)} dX')^{-1}\) Equation (3.3) will reach its maximum when the bifurcation is right in the center of the ROI and decrease elsewhere.

3.2.4 Other Existing Features for 3D Bifurcation Detection.

The following features are also included when training the random forest for bifurcation detection:

1. Spherical Shell Intersection (SSI): This feature is originally described in [235] as the number of connected components between a spherical shell model (composed by two concentric spheres) and its neighborhood \(N_k(x, y, z)\);

2. Convex Hull Ratio (CHR): This feature is defined similar to the “junction-ness degree” in [5], only here we are using vesselness above a certain threshold, \(Th_v\), to determine foreground pixels;

3. Principal Component Analysis (PCA): The PCA feature vector includes the three eigenvalues of the covariance matrix of all vessel directions in the neighborhood;

4. Scale: the Gaussian scale with the maximum vesselness response, as in [67].

In summary, our feature vector that is used to train the random forest is of length \(2C + 8\), including three novel features and four existing features.

For fairness in comparing with other bifurcation detection methods, we propose a simplified tracking framework (Algorithm 1) similar to [38]. Note that the bifurcation detection step (Line 8) can be implemented via our or a competing method. The daughter branch detection (Line 10) is implemented using method in [136]. Centralization step (Line 16) is performed using the method in [38]. The vessel end detection (Line 18) is obtained by thresholding the image intensity by \(Th_I\).

3.3 Evaluation

Our method is evaluated on two datasets, VascuSynth [81] (synthetic, 3D) and VESSEL12 (clinical CT chest images, 3D), against two competing algorithms, [253] and [136]. In all the following experiments, we fix these parameters as: \(\alpha = 0.5, \beta = 0.5, c = 5\)
Algorithm 1: Vessel tracking algorithm

Input: volume image, seed point, initial tracking direction and trained classifier
Output: bifurcation set, centerline set

Initialization: active set (AC) = \{seedpoint\}, bifurcation set (BF) = \emptyset, centerline set (CL) = \emptyset, vesselness, scale and vessel direction

1. while AC ≠ \emptyset do
2.   Active point ← AC;
3.   New tracking starts;
4.   while Current tracking in progress do
5.     Trace along vessel direction;
6.     Classify newly detected point;
7.     if Bifurcation detected then
8.       Detect start points (SP) of two daughter branches;
9.       if Two daughter branches found then
10.      AC ← SP;
11.      BF ← active point;
12.      End current tracking
13.     else Continue with current tracking;
14.     Centralization;
15.     CL ← active point;
16.     if End of vessel reached then Stop;
17.     else Continue with current tracking;

(c = 50 for VESSEL12 dataset), smin = 1, smax = 3 (see [67]) and C = 6, k = 11, Thi = 0.5, Thv = 50. Specifically, we chose C = 6 so that the distribution could pick two opposite directions of the three branches, but the choices of parameters are made empirically.

In our first experiment (Figure 3.1), we evaluate classification performance via leave-one-out cross-validation. Although AdaBoost was used in [253], only random forest is used in this experiment since we want to focus on the contribution of the proposed features. To this end we collected training samples, where each sample is a subvolume (11 × 11 × 11, automatically located in VascuSynth data or manually picked from VESSEL12 data). For VascuSynth, 100 non-bifurcation samples and 100 bifurcation samples are used. For VESSEL12 dataset, 86 non-bifurcation samples and 105 bifurcation samples are used. 200 decision trees are trained each time, with bagging ratio as 0.8 and minimal leaf number as 4. Results using filters from [253] (limit window size as 13 × 13, each with \(x_0 \times y_0 \times (\sigma_x, \sigma_y) \times \theta = 3 \times 3 \times 2 \times 5 = 90\) filters, in consideration of memory consumption; all symbols inherited from [253]) are also listed in Figure 3.1 and the general performance increase is between 5% and 27%.

In our second experiment, we evaluate our method in comparison with a second competing method [136], by replacing the bifurcation detection step in Algorithm 1 (Line 8). The two new tracking algorithms are tested on 10 volumes from VascuSynth dataset (each containing 16 bifurcation points) with 4 different levels of noise (Gaussian, std = 0, 5, 10, 15) and 10 sub-volumes (50 × 50 × 50) selected from the VESSEL12 dataset. For both datasets,
the classifier is trained on the 100 pairs of VascuSynth samples, using 500 trees with bagging ratio as 0.2 and minimal leaf number as 4.

In Figure 3.2, the two sub-figures in the top row show the cumulative histogram of distances from detected bifurcation (DB) points to the closest ground truth (GT) bifurcation location, while the two sub-figures in the bottom row show the accumulated histogram of distances from GT to DB. The first column shows results of VascuSynth – the fact that curves are steeper for our proposed algorithm implies a higher accuracy over the competing algorithm. The second column shows results of VESSEL12 dataset, although the two curves almost overlap at the top, the red is much steeper than the blue at the bottom, which shows a lower tendency of detecting false positives than the competing algorithm. Figure 3.3 shows a visual comparison between tracking results using two bifurcation detectors, green rings indicate errors in bifurcation detection of the competing method of Macedo et al. [136].

3.4 Conclusions and Discussions

In this paper, we proposed three novel features for detecting bifurcation in 3D vascular images: i) the multi-modal von Mises-Fisher coefficients; ii) the histogram of eigenvectors of the Hessian matrix; and iii) the concurrency of eigenvectors of the Hessian matrix. Introducing these features helped to encode statistical prior of the image intensity as well as the image gradient into the bifurcation classification. Synthetic and clinical results showed
Figure 3.2: (color figure) Bifurcation localization error: our algorithm v.s. Macedo et al. [136].

Figure 3.3: (color figure) Visual comparison between two methods: left – proposed; right – competing method [136].
that the proposed method lead to improved bifurcation detection and localization accuracy than state-of-the-art.

In Section 3.3, we mentioned that the choices of certain parameters for the proposed features, such as the number of weights in Equation 3.2, were made empirically. In the future study, a question is to study the effects of tuning these parameters. Also, although we claimed that proposed features are specifically designed for characterizing bifurcations from non-bifurcation structures, future study should involve quantitative evaluation how the proposed features outperform the existing ones.
Chapter 4

Bifurcation Detection via Evolutionary Geometric Deformable Templates

4.1 Introduction

Accurate localization and modelling of bifurcations within 3D medical images is important for fluid dynamics simulation, computer-aided diagnosis, patient-specific surgical planning and other clinical applications. Bifurcation detection is also useful for guiding minimal path- or tracking-based vascular and airway tree segmentation [244]. Besides, bifurcations are crucial in tree labelling and registration algorithms to serve as important landmarks [89]. The geometry of bifurcations is also found to be highly correlated with diseases like atherosclerosis, where plaques accumulate due to slow blood flows, which further induces symptoms like Pulmonary Arterial Hypertension and Chronic Thromboembolic Pulmonary Hypertension in both small and primary blood vessels [94].

Many methods for extracting 3D vasculature/airway bifurcations have been proposed in the past few years, but very few of them explicitly built a bifurcation model for geometrical description and none of them discussed utilizing model-integrated image cues. Some machine learning based pixel-wise classification methods were proposed in [244,253], however, limited geometrical information (e.g. direction) were extracted. Some minimal path based methods either handled bifurcation as principal directions [35], or graph vertices which optimize graph networks using certain length measurement [30]. Some level set based methods handled bifurcations by using direction information extracted using Hessian matrix [194], or required user input for locating branching vessels [137]. A volume growing method was proposed in [84], however, pulmonary veins were simply modeled as cone-like objects, and there was no universal mechanism for detecting bifurcations. An implicit model of 3 super-positioning cylinders was proposed in [14], but the result of the actual codimension bifurcation filter had nothing to do with those cylinders, and like most pixel-wise filtering method, it showed no indication of the parent/children branch directions.
Although a general surface fitting method was proposed in [108], the method was quite limited by requiring some rough skeleton locations as initialization.

In this paper, we argue that a good bifurcation detection method should leverage our knowledge of (i) the unique Y-shaped 3D geometry of bifurcations, (ii) anatomically meaningful shape variability, and (iii) image cues within the bifurcation model, along the surfaces as well as the centerlines and interior regions (we refer to model volumes in the later paragraphs). We adopt an accurate 3D deformable geometrical Y-shaped model parameterized over anatomically-relevant shape variables (e.g. branch radii and angles) [117], with physiologically-driven constraints on these parameters (e.g. radius ratio between parent and child branches [14, 103]). To fit such a single model to a bifurcation, we rely on various image cues (image intensity, gradient magnitude and direction, tubularness) as well as model geometry (centerline location and direction, surface location and normal direction). The above description of the bifurcation and the corresponding fitness calculation, albeit largely comprehensive, makes it infeasible to apply gradient-descent-like optimizers for fitting a single bifurcation model to image data, let alone optimizing numerous models in a 3D volume. We therefore switch to an evolutionary computing method, the genetic algorithm approach with a “tribes”-based niching technique [161, 213] to handle the problem of simultaneous identification of multiple bifurcations. In short, this is the first work that sees bifurcations on a parametrical as well as geometrical and physiological level.

We validated our method on both synthetic and clinical datasets, which shows not only accuracy and stability in quantitative experiments, but also displays elegance with the tribes representation, and demonstrates more merits as simultaneously capturing parent/children branch directions, vessel walls and radii.

### 4.2 Methodology

In this section, we describe a novel approach using 3D parametric shape model for fitting the bifurcation locations in medical images. The intuition behind using the parametric shape model is that, the model itself is derived from the analysis of bifurcations in clinical data, which vice versa, is expected to be useful in locating the bifurcation locations by using proper designed fitness functions.

#### 4.2.1 Modeling the 3D Geometry of a Bifurcation

Borrowed from the pulmonary airway architecture modelling [117], We represent the generic vascular bifurcations as Y-shape geometric objects comprise 7 parts: parent, two transition regions, two curved children branches and two straight children branches (see example in Figure 4.1). The “transition zone” allows for a smooth transition between parent and children branches and we adopt a polynomial of degree $d = 3$ to describe radii
Figure 4.1: (color figure) A bifurcation model comprises 7 typical parts: parent branch; left and right transition regions; left and right curved daughter branch segments; and left and right straight branch segments.

changes. The list of parameters, and the corresponding values we adopted in this work, are given in the Table 4.1. The reader is referred to [117] for a more detailed description of those parameters. A shape catalog is shown in Figure 4.2 with varying left and right transition angles.

Within a 3D image, we apply a 3D affine spatial transformation (translation, rotation and scaling) to the model. The transformation parameters are listed in Table 4.2. Both the shape parameters in Table 4.1 and Table 4.2 will be used to calculate the fitness function, as described in Section 4.2.2.

4.2.2 Multi-objective Fitness Function

Given a particular setting of model shape parameters as well as affine transformation parameters, a unique model is generated in the 3D image space and we will be able to evaluate the plausibility whether a bifurcation exists. To that end, we construct a multi-objective fitness function utilizing appearance cues, such as: intensity, gradient and vesselness, on top of the model centerline, surface and internal areas.

We represent the 3D bifurcation model as $B = (B_C, B_S, B_V)$, where $B_C$ is the centerline, $B_S$ is the surface, $B_V$ is the model volume (volumetric region within the model). We use
<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_p$</td>
<td>parent radius</td>
<td>fixed as 0.5</td>
</tr>
<tr>
<td>$L_p$</td>
<td>parent length</td>
<td>fixed as 2</td>
</tr>
<tr>
<td>$\beta_T$</td>
<td>transition angles</td>
<td>evolved via GA</td>
</tr>
<tr>
<td>$\beta_C$</td>
<td>curved branches angles</td>
<td>fixed as [0, 0]</td>
</tr>
<tr>
<td>$R_\kappa$</td>
<td>curved branches curvature</td>
<td>fixed as [2, 2]</td>
</tr>
<tr>
<td>$R_d$</td>
<td>straight branches radii</td>
<td>evolved via GA</td>
</tr>
<tr>
<td>$l_d$</td>
<td>straight branches lengths</td>
<td>fixed as [2, 3]</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>bifurcation exponent</td>
<td>fixed as 0.67</td>
</tr>
</tbody>
</table>

Table 4.1: Shape parameters. To reduce computational complexity, we fix most of the parameters and only search on $\beta_T$ and $R_d$ axes during optimization.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Explanation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_x$</td>
<td>rotation along x-axis</td>
<td>evolved via GA</td>
</tr>
<tr>
<td>$\theta_y$</td>
<td>rotation along y-axis</td>
<td>evolved via GA</td>
</tr>
<tr>
<td>$\theta_z$</td>
<td>rotation along z-axis</td>
<td>evolved via GA</td>
</tr>
<tr>
<td>$r$</td>
<td>scaling ratio</td>
<td>evolved via GA</td>
</tr>
<tr>
<td>$t_x$</td>
<td>translation along x-axis</td>
<td>evolved via GA</td>
</tr>
<tr>
<td>$t_y$</td>
<td>translation along y-axis</td>
<td>evolved via GA</td>
</tr>
<tr>
<td>$t_z$</td>
<td>translation along z-axis</td>
<td>evolved via GA</td>
</tr>
</tbody>
</table>

Table 4.2: Transformation parameters. All the parameters will be evolved during optimization.

$V$ for Frangi’s vesselness [67], $I$ for image intensity, $\vec{g}$ for image gradient and $\vec{n}$ for inward normal directions on the model surface. The fitness terms could be categorized into three major types: fitness defined along centerlines, on the surfaces, and over the whole model volume.
Figure 4.2: (color figure) Catalogue of sample 3D bifurcation shape models.

**Fitness Terms along Model Branch Centerlines**

a) Average vesselness on model branch centerlines:

\[
F_{V}(B_{C}) = \frac{1}{\text{length}(B_{C})} \int_{B_{C}} V(c) \text{dc}
\]  

(4.1)

b) Average image intensity on model branch centerlines:

\[
F_{I}(B_{C}) = \frac{1}{\text{length}(B_{C})} \int_{B_{C}} I(c) \text{dc}
\]  

(4.2)

**Fitness Terms on Model Surface**

a) Average gradient magnitude on model surface

\[
F_{G}(B_{S}) = \frac{1}{\text{area}(B_{S})} \int_{B_{S}} \|\tilde{g}(s)\|_{2} \text{ds}
\]  

(4.3)
b) Average gradient alignment on model surface

\[ F_A(B_S) = \frac{1}{\text{area}(B_S)} \int_{B_S} < \vec{g}(s), \vec{n}(s) > ds \] (4.4)

**Fitness Terms inside the Model Volume**

a) Mean intensity inside model volume

\[ F_I(B_V) = \frac{1}{\text{volume}(B_V)} \int_{B_V} I(v) dv. \] (4.5)

The multi-objective fitness function is optimized as:

\[ \max_{B(S,C,V)} F(S,C,V) = \max_{B(S,C,V)} \{ \omega_1 F_V(B_C) + \omega_2 F_I(B_C) + \omega_3 F_G(B_S) + \omega_4 F_A(B_S) + \omega_5 F_I(B_V) \}. \] (4.6)

4.2.3 Tribes-based Evolutionary Genetic Algorithm

To find optimal parameters for fitness function (4.6) to locate bifurcations in a 3D image, we must search in a high dimensional space for geometrical parameters as well as transformational parameters (Table 4.1 and 4.2). However, it’s difficult to obtain \( \nabla_{S,C,V} F \) and a heuristic searching method is prone to get trapped in local optima. To overcome these difficulties, we adopt the genetic algorithm (GA) [226], the idea of which originates from natural selection and is efficient in solving constrained optimization problems. With this algorithm, the parameter vector is encoded as a chromosome, and random crossover and mutation is applied to each (or each pair of) chromosome to evolve toward a better offspring [226]. The classical GA was only designed to search for a single global optimum, but for our case, multiple bifurcations/optima typically exist in the same image. To handle multiple reoccurring bifurcations, we also adopt the tribes niching technique [161, 213]. In a nutshell, a tribe is a population of chromosomes with certain similarity - to our case, the similarity is reflected in the 3D translation. For each chromosome within the tribe, we restrict that it can only evolve within a certain distance (along the 3D spatial dimensions) from the tribe center. We initialize several tribes simultaneously and evolve each tribe independently. Eventually very close tribes will be merged with each other and only tribes with the most elite chromosomes will be kept to avoid redundancy and reduce errors. Thus each tribe will represent a single bifurcation in the image, with translation parameters representing the bifurcation location, rotation parameter representing parent and children branch directions, scale/model radii parameter representing the radii for parent and children branches. In short, the genetic algorithm explores the high dimensional parameter space by
creating a population of, in our case, bifurcation models that are spread out, i.e. reoccurring, in the image domain, with varying pose, scale, and shape parameters.

We use the following rules for evolving and merging tribes:

a) if a chromosome after mutation is away from the tribe center for more than distance $R$, the child chromosome is discarded and the parent chromosome will be kept instead;

b) if the distance between two tribes’ centers is below a threshold $\mu$, the two tribes will be merged.

For all the following experiments, we first threshold the vesselness at value 0.1, and then automatically initialize the tribes’ spatial centers on the remaining pixels so that they are $R$ pixels apart. Both $R$ and $\mu$ are set to be 10 pixels.

4.3 Evaluation

4.3.1 Data Description

We use a synthetic dataset as well as a clinical CT dataset for evaluation purposes. The synthetic dataset comprises 20 volumes generated by the VascuSynth 3D vascular tree simulation software [81,97] with different Gaussian noise levels (SNR=20; 13.98; and 10.46, Figure 4.3). Each volume is around $30 \times 30 \times 30$ voxels in size and each contains two bifurcations. The clinical dataset is composed of computed tomography (CT) pulmonary volumes\(^1\), and each volume is cropped to an ROI of size $50*50*50$. Sixty two (62) bifurcations and the directions of the 3 neighboring branches per bifurcation were annotated manually by a graduate student knowledgeable with 3D CT data and anatomical trees.

4.3.2 Quantitative Results

Evaluation Metrics

We use $GT$ to represent the set of all ground truth point locations. $\forall x \in GT$, $\vec{V}_{GP}(x), \vec{V}_{GL}(x), \text{ and } \vec{V}_{GR}(x)$ are used to represent ground truth parent, left and right children branch directions at $x$. $DT$ is the set of all detected bifurcation locations, and $\vec{V}_{DP}(y), \vec{V}_{DL}(y), \text{ and } \vec{V}_{DR}(y)$ are the detected parent, left and right children branch directions at detected location $y$. For the propose method, the detected bifurcation location is the coordinate of the center point on the cross sectional plane between the parent branch and the transition zones.

The evaluation metrics are defined as follows:

\(^1\)https://vessel12.grand-challenge.org/
Figure 4.3: Volume rendering of the VascuSynth images with different noise levels: (a) original noise-free binary image; (b) SNR = 20; (c) SNR = 13.98; (d) SNR = 10.46.

a) Coherence metric: \( \forall x \in GT, \)

\[
M_C(x) = \min_{y \in DT} 1/(\vec{V}_{GP}(x) \cdot \vec{V}_{DP}(y) + \vec{V}_{GL}(x) \cdot \vec{V}_{DL}(y) + \vec{V}_{GR}(x) \cdot \vec{V}_{DR}(y)) \tag{4.7}
\]

b) Distance metric: \( \forall x \in GT \)

\[
M_D(x) = \min_{y \in DT} ||x - y||_2 \tag{4.8}
\]

c) Dispersion metric: \( \forall x \in GT, \ y \in DT, \) if \( ||x - y||_2 < R, \) \( R \) is the tribe radius (see Section 4.2.3),

\[
M_P(x, y) = ||x - y||_2 \tag{4.9}
\]

Intuitively, the coherence metric \( M_C \) measures the total difference between the directions of the ground truth (GT) branches and the directions of the three branches at the detected bifurcation. The distance metric measures the distance between the detected bifurcation
location and the GT bifurcation location. For the proposed method, the detected bifurcation location is determined using the transformed model origin as described in [117]. The distance metric considers the proximity of only the single detected bifurcation that is closest to the GT and ignores the other false positives, which results in a scalar for each \( x \in GT \). The dispersion metric, on the other hand, returns a vector approximating the distribution of all the detected bifurcations around each GT location, e.g., considering both “good” and “bad” detections.

To determine the weight values in (4.6), we do a brute force search over a training set for the maximal fitness value, searching each weight between \([0.5, 5.0]\) with step= 0.5. For the synthetic data, the weights are chosen as \( \{\omega_1 = 1, \omega_2 = 2, \omega_3 = 2, \omega_4 = 0.5, \omega_5 = 2\} \). For the clinical dataset, the weights are chosen as \( \{\omega_1 = 1, \omega_2 = 0.5, \omega_3 = 2, \omega_4 = 1, \omega_5 = 0.5\} \).

### Results of Experiments on Synthetic Data

The results of noise tests with the VascuSynth synthesized data are shown in Tables 4.3, 4.4 and 4.5. The proposed method outperforms all competing methods, ISBI’07 [253], ISBI’14 [244] and a scale space detector (MMBIA’12) [14] with regard to metrics \( M_C \) and \( M_P \). We approximate the normal distributions of metric value \( M_P \) in Figure 4.4 (a-c) and the red curves represent our proposed results. The red curves are steeper and further left than the rest of the curves, which means the bifurcation points detected by the proposed method are more accurate and stable than those obtained by the competing methods. In both experiments with synthetic and clinical data, the tribes are empirically initialized with Frangi’s vesselness [67] \( \geq 0.1 \), each tribe with radius= 10 and at least 10 pixels apart with each other, so to balance computation time and detection accuracy.

<table>
<thead>
<tr>
<th>SNR=20</th>
<th>Proposed</th>
<th>ISBI’07</th>
<th>ISBI’14</th>
<th>MMBIA’12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_D ) mean(std)</td>
<td>2.75(1.18)</td>
<td>0.57(0.72)</td>
<td>1.07(2.35)</td>
<td>13.58(9.14)</td>
</tr>
<tr>
<td>( M_C ) mean(std)</td>
<td><strong>0.35 (0.01)</strong></td>
<td>NA</td>
<td>0.55(0.21)</td>
<td>NA</td>
</tr>
<tr>
<td>( M_P ) mean(std)</td>
<td><strong>2.75(1.18)</strong></td>
<td>7.44(1.92)</td>
<td>5.2(2.66)</td>
<td>6.71(3.29)</td>
</tr>
</tbody>
</table>

Table 4.3: Results on VascuSynth data at SNR=20. ISBI’07 and MMBIA’12 do not return direction information, so the values are not applicable (NA).

### Results of Experiments on Clinical Data

The results of experiments with clinical data are shown in Table 4.6. Again, the proposed method outperforms all competing methods, ISBI07 [253], ISBI14 [244] and MMBIA’12 [14] with regard to metrics \( M_C \) and \( M_P \). We approximate the normal distributions of metric value \( M_P \) in Figure 4.4 (d) and the red curves represent our proposed results. The red curves
are steeper and closer to the left than the rest of the curves, which means the bifurcation points detected by the proposed method are more accurate and stable than those obtained by the competing methods.

<table>
<thead>
<tr>
<th>SNR=13.98</th>
<th>Proposed</th>
<th>ISBI’07</th>
<th>ISBI’14</th>
<th>MMBIA’12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_D$ mean(std)</td>
<td>3.73(4.91)</td>
<td>0.57(0.72)</td>
<td>0.97(1.71)</td>
<td>14.78(7.78)</td>
</tr>
<tr>
<td>$M_C$ mean(std)</td>
<td>0.36(0.09)</td>
<td>NA</td>
<td>0.54(0.21)</td>
<td>NA</td>
</tr>
<tr>
<td>$M_P$ mean(std)</td>
<td>2.66(1.08)</td>
<td>7.47(1.92)</td>
<td>5.29(2.58)</td>
<td>6.38(2.50)</td>
</tr>
</tbody>
</table>

Table 4.4: Results on VascuSynth data at SNR=13.98.

<table>
<thead>
<tr>
<th>SNR=10.46</th>
<th>Proposed</th>
<th>ISBI’07</th>
<th>ISBI’14</th>
<th>MMBIA’12</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_D$ mean(std)</td>
<td>3.87(4.86)</td>
<td>0.64(0.72)</td>
<td>0.81(0.97)</td>
<td>12.80(4.50)</td>
</tr>
<tr>
<td>$M_C$ mean(std)</td>
<td>0.36(0.08)</td>
<td>NA</td>
<td>0.52(0.18)</td>
<td>NA</td>
</tr>
<tr>
<td>$M_P$ mean(std)</td>
<td>2.81(1.03)</td>
<td>7.44 (1.92)</td>
<td>5.61(2.52)</td>
<td>7.79(1.43)</td>
</tr>
</tbody>
</table>

Table 4.5: Results on VascuSynth data at SNR=10.46.

4.3.3 Qualitative Results

Figures 4.5 and 4.6 show sample results obtained by the proposed method. In Figure 4.6 (c-d), the yellow dots are classified by ISBI’14 as bifurcations, and we can observe many are faraway from the true bifurcations. This illustrates that, even though in Tables 4.3-4.5, the proposed method has larger $M_D$ values, it still outperforms the competing methods by giving more precise results with much fewer false positives.

Finally, we note that the proposed method provides more merits compared to competing methods as summarized in Table 4.7. As shown in Figure 4.6, the proposed method not only reports whether a point is a bifurcation but also shows where the parent branch and children
Figure 4.4: (color figure) “Error” distributions of VascuSynth and clinical data. (a)-(c): Normal distribution estimated using $M_P$ mean and std in Table 4.3-4.5. (d): Normal distribution estimated using $M_P$ mean and std in Table 4.6.

branches lie, a type of result none of the competing methods could achieve. Although the ISBI’14 method implicitly returns branch directions in its “vmf” feature, by observing the $M_C$ values from Table 4.3-4.6, we can see they are far from accurate. Also in Figure 4.6 we can see the model surfaces nicely fit to the vessel walls with accurate parent/children scales estimation, while the competing methods only returns a single scale value at a given point and provide no estimation to wall locations.

4.4 Conclusions and Discussions

We presented the first bifurcation detection method that relies on constructing and fitting a 3D deformable Y-shaped template to volumetric medical image data. Using a tribes
Figure 4.5: (color figure) An example of the Vascusynth data results (SNR=10.46). 9 tribes were initialized and 2 tribes were found. The original noise-free image is rendered purple using an iso-surface. The best chromosome in each tribe is rendered in red (parent), green (transition) and blue (daughters).

Based evolutionary computing optimization, we were able to detect multiple bifurcations in a volume simultaneously. No regularization or shape prior terms were necessary in the fitness function as such priors are already encoded in the underlying shape model. Our method not only detects the locations of the bifurcations, but also reports the parameters of the 3D shape model (e.g. parent and child directions and radii), which may be useful for fluid simulations. Future work will involve applying dense deformations of the bifurcation model surface to better match edges as well as reconstructing the complete anatomical tree by detecting inter-bifurcation branches.
Figure 4.6: (color figure) An example of the clinical results, comparing the proposed method and ISBI'14. (a)-(b): results of proposed method. 30 tribes were initialized in (a) and 5 tribes were found eventually. 36 tribes were initialized in (b) and 6 were found eventually. The original image is rendered in iso-surface as purple. Surface: red is the parent branch; green is the transition zone; and blue shows the children branches. (c)-(d): results of ISBI'14 method. Yellow dots are classified as bifurcations, many of which are false positives.

<table>
<thead>
<tr>
<th></th>
<th>Proposed</th>
<th>ISBI'07</th>
<th>ISBI'14</th>
<th>MMBIA'12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vessel Walls</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Centerlines</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Parent/Child scales</td>
<td>✓</td>
<td>Single</td>
<td>Single</td>
<td>Single</td>
</tr>
<tr>
<td>Directions</td>
<td>✓</td>
<td>×</td>
<td>Implicit</td>
<td>×</td>
</tr>
<tr>
<td>Parent/Child Topology</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

Table 4.7: Comparisons among the proposed method and three competing methods. For the competing methods, only a single scale value is returned at each location, while the proposed method provides scale estimation for all parent and children branches. The ISBI'14 method doesn’t return the directions of the neighboring branches directly, but its “vmf” feature (c.f. [244]) implicitly implies the branch directions.
Chapter 5

Branch Direction Prediction via Multi-Loss Deep Learning Network: the TreeNet Model

5.1 Introduction

Tree extraction is a crucial task in 3D medical image analysis, and accurately extracted circulatory and respiratory trees can be further utilized in surgery planning, registration and tree space analysis [15, 62, 66]. However, the automation and accuracy of tree extraction still remains an open problem due to the natural complexity and variability of the topologies of anatomical tree structures [104], the various imaging reconstruction artifacts [182], varying image intensities along branches, the similarity between tubular structure lumen and background tissue lumen, and the changing geometry due to different pathologies [16, 83, 227].

One major category of tree extraction algorithms is based on iterative tracking, which usually starts from a given seed point near the root of the tree, predicts the direction of the branch to track along it, detects bifurcations to spawn children trackers, and progresses down the tree hierarchy to smaller branches until some stopping criteria are met [244]. Although there have been several works that tackle the important bifurcation detection step of the tracker [39, 142], very few are specifically designed to determine the direction of the branch at a given point. Most works simply treat the problem of direction estimation as a by-product of vesselness or tubularness calculation [98, 115, 204].

Most existing methods on vesselness (with explicit/implicit direction prediction) rely on certain assumptions made on the geometry of tubular structures. Most Hessian based vessel enhancement filters, e.g., Frangi et al. [67] and Jerman et al. [98, 204], assumed the vessels were elongated structures. Cetin et al. [39] measured intensity distribution within an oriented cylinder-sphere combined model and constructed a corresponding tensor representation to optimize for vessel directions, however, their success relied on a good match between the cylinder and the actual vessel shape. Law et al. [115] used a gradient...
based tensor to model oriented flux flow and the vessel direction was also approximated by the eigenvector – intrinsically their assumption of vessel shapes were still straight tubes. However, in clinical datasets, especially those exhibiting various pathologies or abnormalities such as narrowing (e.g., in COPD [227]), aneurysms [16], and high tortuosity (which might indicate diseases like arterial hypertension and strokes [83]), the aforementioned shape assumptions might no longer hold true, which results in incorrect direction estimates.

In contrast to the above deterministic methods, stochastic and learning based tracking methods provide more flexibility by adjusting the prediction retrospectively during the tracking process, or by using prior information learnt from the training data [249]. Lee et al. [119] proposed to use particle filtering to track vessel contours slice by slice, with the per-slice contour obtained by the Chan-Vese model. Lesage et al. [120] also proposed to use particle filtering method, but in contrast, utilized geometric flow, image features, as well as radius and direction prior distributions to perform Bayesian inference. In these tracking processes, vessel directions were found implicitly by subtracting neighboring points along the detected centerlines. On the other hand, a significant number of machine learning based methods ignore directional information completely by focusing on pixel-wise classification [105, 244, 253].

The fast development of deep learning methods provides vast opportunities in exploring the structures in 3D images from coarse to fine scales [148], however, limited work has been done on analyzing 3D vasculature images, and none of them estimated tree branch directions. Mirikharaji et al. [23] proposed to use an artificial neural network trained on 2D patches to learn the probability map of airway bifurcation locations; instead of tracking new branches, they connected the bifurcations by minimal paths to form the whole tree. Fu et al. [69] proposed to combine a CNN architecture with a conditional random field model to achieve a smooth binary segmentation for retinal vessels, but their method was only performed on 2D retinal images and no vessel direction was estimated. Chen et al. [43] proposed to use a convolutional autoencoder for voxel-wise cerebral arteries segmentation while completely ignoring directional information.

We claim the following contributions are made in this paper:

1. The proposed method, which extends V-Net [148], is the first tree branch direction prediction deep learning method;

2. The proposed multi-loss function is novel and specially designed for tracking 3D tubular structures;

3. The proposed model is trained and tested in a branch-specific way, which takes advantage of the “anatomical tree statistics” [149, 249] and fully utilizes statistical and geometrical information.
5.2 Methodology

5.2.1 Architecture

Our proposed deep learning architecture is an improvement of V-Net [148]. The choice of V-Net is two-fold:

1. The encoding-decoding process of V-Net helps to propagate contextual information into higher resolution layers – in our case, the context information is the tubularity of the neighboring points;

2. Our multi-loss function (introduced below) relies on cross sectional plane information, so the prediction process implicitly involves plane segmentation and reconstruction.

We rescale all input volumes to $64 \times 64 \times 64$ voxels with histogram equalization. We use batch normalization and add three extra fully connected layers (FCs, with output channels 1024, 64 and 4) at the end of the V-Net and output a 4-element vector $\langle \vec{v}, R \rangle$ where $\vec{v}$ predicts the direction of the vessel in the center of the input cube, and a radius $R$ that serves as intermediate input in training the loss layer. The overall methodology is illustrated in Figure 5.1.

5.2.2 At Testing Time

A region of interest (ROI, or 3D patch, which we use in the context interchangeably) is generated and input into the network (as shown in red dotted box in Figure 5.1), and the output is the predicted vector of the corresponding branch (shown as $\vec{v}_{gt}$ in Figure 5.2).

![Figure 5.1: (color figure) The proposed architecture and tracking process.](image-url)
Loss Function

We define the following multi-loss function:

\[
L(\vec{v}_{dt}^i, R_{dt}^i) = \omega_1 L_{dir} + \omega_2 L_{mask} + \omega_3 L_{image} + \omega_4 L_{radius} \tag{5.1}
\]

\[
L_{dir}(\vec{v}_{dt}^i, R_{dt}^i) = -|\vec{v}_{gt}^i \cdot \vec{v}_{dt}^i|^2
\]

\[
L_{mask}(\vec{v}_{dt}^i, R_{dt}^i) = ||B_{gt}^i - B_{dt}^i||_2^2
\]

\[
L_{image}(\vec{v}_{dt}^i, R_{dt}^i) = ||I_{gt}^i - I_{dt}^i||_2^2
\]

\[
L_{radius}(\vec{v}_{dt}^i, R_{dt}^i) = |R_{gt}^i - R_{dt}^i|^2
\]

where \(i\) is the training index, \(gt\) refers to ground truth value, \(dt\) refers to model prediction. \(I_{gt}^i\) and \(I_{dt}^i\) are corresponding \(gt\) and predicted cross sectional planes (going through the center voxel). \(B_{gt}^i\) and \(B_{dt}^i\) are the ground truth and predicted (using radius \(R_{dt}^i\) and circular expression) branch masks on the cross sectional planes, \(R_{gt}^i\) and \(R_{dt}^i\) are the ground truth and predicted radii, as illustrated in Figure 5.2. The four terms \(L_{dir}, L_{image}, L_{mask}\), \(L_{radius}\) capture the errors in, respectively, direction estimation, cross sectional image plane reconstruction, branch internal area estimation and radius estimation. We normalize \(L_{image}\) and \(L_{mask}\) by the patch cube size and set the weights empirically to \(\omega_1 = \omega_2 = \omega_3 = \omega_4 = 1\). The total loss is optimized over \(v_{dt}^i\) and \(R_{dt}^i\). Since accurate direction prediction leads to accurate cross section plane prediction, using multiple loss terms should theoretically increase the direction prediction accuracy.

Tree Tracking

We follow the tracking procedure in [244], which starts from a given seed point in a branch and tracks along vessel/airway detected by the proposed architecture. Additional tracking details are given in Section 5.3.
5.3 Evaluation

5.3.1 Synthetic Dataset

We use three different types of synthetic dataset to mimic pathologies such as narrowing and aneurysms, and high tortuosity [16, 83, 227]: i) Occlusion, ii) Torus and iii) Leakage. Examples are shown in Figure 5.3. We augment the data by rotating the volumes along each axis randomly between $[0, 60^\circ]$, using two radius values, translating along each dimension separately by three values ($[-1, 0, 1]$, so 9 cases in total) and adding Gaussian noise with standard deviation from 0.005 to 0.1 (20 cases). This brings the total number of image volumes per each category to 360. We then run a 3-fold cross validation ensuring that augmentations of any volume are not split across the train and test sets.

![Synthetic examples (noise free).](image)

5.3.2 Clinical Dataset

The clinical dataset is from the Extraction of Airways from CT (EXACT) 2009 challenge. Sixteen training volumes with binary segmentations were provided by the organizers. We extracted two categories of data: 1) ROIs, each is a cuboid containing one of the following 7 anatomical structures: trachea, right main bronchi (RMB), left main bronchi (LMB), right superior lobar bronchus (RSLB), right intermediate bronchus (RIB), left superior lobar bronchus (LSLB) and left inferior lobar bronchus (LILB); 2) patches, each is a cube randomly sampled from the branch centerlines, with radii twice the mean radii of the branch, intensities normalized to $[0, 1]$, and augmented by adding Gaussian noise with standard deviation $[0.01, 0.04]$ with step size $= 0.01$. We perform a 4-fold cross validation on the patients for training and testing.

5.3.3 Competing Methods

We compare with 4 state-of-the-art algorithms: i) OOF [115]; ii) Tensor [39]; iii) Jerman [98, 204]; 4) Particle filtering [119]. Since particle filtering doesn’t directly predict the vessel direction, we use a primitive tracking method to first trace the branch centerline, then estimate the directions. For multiscale methods, the scale ranges are set according to mean branch scales learnt from the dataset, and all other parameters are set according to the
original paper (for airways, i.e., dark-on-bright, some parameters are inverted accordingly). Note that although i) and ii) are not direction prediction methods per se, they leverage direction estimates to filter branches, which makes the comparison fair.

5.3.4 Tracking Details

Both the proposed method and the competing ones use the same initial seeds, which are selected from the ground truth branch centerlines. By calculating the mean radii \( \bar{R} \) of the corresponding branch, the ROI radii are set automatically as \( 2 \bar{R} \).

5.3.5 Evaluation Metrics

Two metrics are used to evaluate the results. For the tracking method, we use the asymmetric distance function proposed in \[244\] to compare the ground truth centerline to the extracted centerline:

\[
D(C_1, C_2) = \{ \min_{s_2 \in C_2} \text{dist}(s_1, s_2) | \forall s_1 \in C_1 \}
\]  \hspace{1cm} (5.2)

where \( \text{dist}(s_1, s_2) \) is 3D Euclidean distance between voxels \( s_1 \) and \( s_2 \), \( C_1 \) the ground truth centerline and \( C_2 \) the detected centerline (by either the proposed method or particle filtering). \( D(C_1, C_2) \) returns a set of distance values for all the points on \( C_1 \), so a smaller mean value and standard deviation would indicate a better result. For other competing methods, since they return a per-voxel direction estimate, we use the following symmetric accuracy metric:

\[
\text{accu}(\vec{v}_1, \vec{v}_2) = \vec{v}_1 \cdot \vec{v}_2
\]  \hspace{1cm} (5.3)

where \( \vec{v}_1 \) and \( \vec{v}_2 \) are the branch direction vectors to be compared.

5.3.6 Experiments

The evaluation result on the synthetic dataset is shown in Table 5.1. We can see a marked improvement in the proposed method over the competing ones by at least 10% in mean direction accuracy. For the Occlusion category, all competing methods performed poorly, since all these methods assume that the foreground is always luminous. For the Torus category, we can see the Tensor method \[39\] performing the worst, as it modeled the blood vessel as cylindrical tubes, which were very different from the torus shapes in the given images. On the contrary, the Tensor method performed much better than other competing methods on the Leakage category, as a long cylinder might overcome the small leakage (but not good enough to overcome the occlusion) and found the correct direction. It is worth noting that, although the Jerman filter could achieve good enhancement results at tortuous and bulging branches \[98, 204\], the filter was not designed to deal with the direction estimation task.
We observe that by removing only one of the loss terms (other than $L_{\text{dir}}$) actually performs worse than using only $L_{\text{dir}}$. This is not surprising when we remember that the cross sectional plane and the direction together serve as the Frenet frame, so removing one term would invalidate the frame representation. Since $L_{\text{image}}$ contains the most information on the cross sectional plane, removing it leads to the worst performance. The improvement in prediction accuracy of the multi-loss function supports our hypothesis that all four terms contribute to the result, given their complementary nature.

Figure 5.4a shows an example where an airway centerline tree is extracted using our proposed method (red curves) and compared to the ground truth tree centerlines (yellow curves). Figure 5.4b shows a qualitative comparison between the tracking result, along branch LIB, between the particle filtering and the proposed method. The mean and standard deviation of distance errors of each branch are shown in Figure 5.5. The proposed method achieves a lower error mean and standard deviation on every anatomical branch.

The results in Table 5.2 are consistent with the synthetic data results. The proposed method outperforms all the competing methods on all branches.

We run our experiments on a Nvidia GTX GeForce 12 GB TITAN GPU, and the processing time per patch at testing stage is 0.04 second.

<table>
<thead>
<tr>
<th>Year</th>
<th>Occlusion</th>
<th>Torus</th>
<th>Leakage</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOF [115]</td>
<td>2008</td>
<td>0.11(0.07)</td>
<td>0.69(0.28)</td>
</tr>
<tr>
<td>Tensor [39]</td>
<td>2015</td>
<td>0.47(0.1)</td>
<td>0.48(0.037)</td>
</tr>
<tr>
<td>Jerman [98, 204]</td>
<td>2016</td>
<td>0.44(0.46)</td>
<td>0.62(0.48)</td>
</tr>
<tr>
<td>Proposed with $L_{\text{dir}}$ only</td>
<td>0.95(0.06)</td>
<td>0.96(0.09)</td>
<td>0.97(0.04)</td>
</tr>
<tr>
<td>Proposed w/o $L_{\text{image}}$</td>
<td>0.90(0.07)</td>
<td>0.95(0.12)</td>
<td>0.94(0.04)</td>
</tr>
<tr>
<td>Proposed w/o $L_{\text{mask}}$</td>
<td>0.93(0.07)</td>
<td>0.93(0.13)</td>
<td>0.97(0.04)</td>
</tr>
<tr>
<td>Proposed w/o $L_{\text{radius}}$</td>
<td>0.94(0.07)</td>
<td>0.93(0.10)</td>
<td>0.97(0.05)</td>
</tr>
<tr>
<td>Proposed</td>
<td><strong>0.97(0.02)</strong></td>
<td><strong>0.97(0.06)</strong></td>
<td><strong>0.99(0.04)</strong></td>
</tr>
</tbody>
</table>

Table 5.1: Three fold cross validation result on synthetic dataset.

5.4 Conclusions and Discussions

We proposed the first deep learning architecture for estimating anatomical tree branch directions, which is a critical step for the common tracking-based tree extraction methods. Our proposed loss function is unique in that it follows the geometry of the target structure.
Figure 5.4: (color figure) (a): Whole tree extracted. (b): Centerlines tracked by proposed algorithm and competing particle filtering algorithm on LIB.

Figure 5.5: (color figure) Distance error bar between GT centerlines and detected centerlines.

(i.e. the curvilinear tree branches) by using branch direction agreement and cross sectional image information, based on a Frenet frame of reference.

Although the dataset we designed in Section 5.3 was assumed to reflect pathologies such as narrowing and aneurysms, we didn’t evaluate how well these data reflect the real cases. Given the large performance improvement gap between the synthetic data experiment and clinical data experiment in Table 5.1 and Table 5.2 (especially comparing the proposed method with Jerman et al. [98, 204]), the synthetic dataset likely reflect only the extreme
Table 5.2: Direction accuracy (mean and standard deviation) on airway branches of different levels.

<table>
<thead>
<tr>
<th></th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trachea</td>
<td>Trachea</td>
<td>Trachea</td>
</tr>
<tr>
<td>OOF [115]</td>
<td>0.19(0.18)</td>
<td>0.24(0.24)</td>
<td>0.29(0.26)</td>
</tr>
<tr>
<td>Tensor [39]</td>
<td>0.86(0.15)</td>
<td>0.60(0.20)</td>
<td>0.78(0.15)</td>
</tr>
<tr>
<td>Jerman [98,204]</td>
<td>0.91(0.17)</td>
<td>0.93(0.16)</td>
<td>0.90(0.18)</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.92(0.11)</td>
<td>0.95(0.07)</td>
<td>0.93(0.08)</td>
</tr>
</tbody>
</table>

Table 5.2: Direction accuracy (mean and standard deviation) on airway branches of different levels.

cases of pathological changes. In a summary, future study on how to effectively mimicking the pathologies should be made.
Chapter 6

Anatomical Tree Prediction via Encoding Memory using LSTM: the Tree-LSTM Model

6.1 Introduction

Vascular and airway trees, i.e., anatomical trees, of circulatory and respiratory systems are extremely important clinically as they are related to fatal diseases like ischaemic heart disease, stroke, chronic obstructive pulmonary disease, and lower respiratory infection, which were identified as the top four causes of death globally, according to a report from World Health Organization in 2018 [231]. In the field of medical image analysis, anatomical tree extraction from 3D medical imaging data is a crucial task, as accurate extraction of anatomical trees could be further utilized in tasks such as surgery planning [192], early diabetic diagnosis [54], and tumor type classification [37].

Most tree extraction approaches fall into one of the following categories [121,249]: a) active contour/surface methods (or deformable energy minimizing models), which expand, commonly via a variational framework, a curve or surface to fit the desired vessel/airway boundaries; b) tracking methods (described below); c) minimal path approaches, which determine the path of a vessel/airway branch as the route with minimal energy between given points at both ends of the branch; d) graph based methods, which represent the anatomical tree as a tree graph and model the tree extraction as a combinatorial optimization problem; e) machine learning and deep learning methods that learn various aspects of the tree extraction process (e.g., bifurcation detection) from training data; and f) other hybrid methods.

The tracking methods cover a large portion of the tree extraction literature. At a high level, this class of approaches involves starting from one or multiple seed points, followed by iteratively detecting tracking directions, and identifying new candidate branch points (usually along the branch centerline), until the whole tree is tracked. Cetin et al. [38] proposed to use a tensor based direction prediction technique for coronary vessel tracking.
with a “shape prior” encoding the tube-like geometry of branches. Wang et al. [219] proposed to use Bayesian estimation in the tracking process and the vessel likelihood was estimated iteratively using gradient flux on the cross sectional plane. Lesage et al. [122] proposed to use particle filtering, which adopted the sequential importance sampling technique to sample candidate particles in a recursive way. Lee et al. [119] also proposed to use particle filtering for vessel tracking, only that they leveraged a Chan-Vese model at each prediction. There have also been a rising number of works encoding machine/deep learning techniques [229, 244, 246] in the tree tracking process.

Examining most tracking methods from the stochastic modelling perspective, we note that they share a common underlying first-order Markovian model assumption, which means the current prediction is only directly affected by the last tracking step (commonly the image features therein), ignoring direct relations with previous points along the tracked trajectory [38, 119, 122, 219, 229, 244, 246]. Even though a perfect Monte-Carlo estimation in the particle filtering approaches is supposed to involve all previous particles for importance sampling, however, in most particle updating scheme implementations [119, 122], a first-order Markovian assumption is usually adopted to reduce complexity. Very limited works have tried to encode higher-order Markovian models or global information into the tracking process. Zhao et al. [249] proposed adopting a Bayesian inference formulation to encode branch-wise statistics (tree topology, branch length, and angle geometrical priors), however, in the form of a prior term instead of the interaction among previous predictions.

Deep learning architectures have been explored extensively in recent years in the field of computer vision and medical image analysis in general, and to a lesser degree for anatomical tree extraction. The work of Wu et al. [232] used an AlexNet classifier with PCA and nearest neighbor search to classify whether the current point in a 2D retinal image is a bifurcation. Charbonnier et al. [41] proposed to combine the results of three ConvNets on orthogonal planes to detect leaks in airway segmentation. Wolterink et al. [229] used a seven-layer dilated convolutional neural network (DCNN) to predict coronary vessel directions in 3D patches. Zhao et al. [246] proposed to use a modified V-net [148] for airway direction prediction, by using a custom multi-loss function. Selvan et al. [193] proposed to use a K-nearest neighbour voxel-wise classifier for airway segmentation, then a 10-layer mean field network was used later for tree graph refinement.

Although long short-term memory (LSTM) architectures have been applied to computer vision and medical image analysis problems to perform inference on sequential data by exploiting temporal information in 2D+time and 3D+time data [42, 203], no prior work has been done using LSTM for vasculature and airway structure prediction. To the best of our knowledge, the only work that adopted LSTM for anatomical tree analysis is the work of Wu et al. [233], in which they labelled an already segmented abstract representation of a coronary vessel tree using a bidirectional LSTM per branch, whereas the focus of our work is to segment or extract trees form a 3D images.
In this work, we argue that the tree tracking process goes beyond a first-order Markovian and leverage the global information along the pseudo-temporal (sequence) dimension in combination with the proven capabilities of deep networks to automatically learn appearance features. To this end, we propose Tree-LSTM, the first neural network LSTM architecture to learn to encode such sequential priors into a deep learning tree extraction method. Our proposed method is the first work in the field of anatomical tree analysis with the following contributions:

1. A formal derivation of the higher-order Markovian process;
2. A LSTM-equipped CNN is used for approximating the higher-order Markovian process;
3. A novel evaluation method is proposed for inspecting the correlation between bifurcation classification accuracy and sequential image evidence collected along branches.

By using the proposed architecture, we show that the bifurcation and direction prediction accuracy could be improved by a large margin.

6.2 Methodology

Tree tracking is predominantly formulated as an iterative centerline tracking problem (see Section 2.1), wherein the vessel/airway direction and bifurcation existence at each tracked point is inferred, and used to estimate the next candidate centerline point, and child branches are spawned whenever a bifurcation point is encountered. This type of inference is a lower-order Markovian formulation.

However, a lower-order Markovian fails to encode information collected during the tracking process, or the "memory", beyond one tracking step. To overcome this problem, we propose to encode the memory using a higher-order Markovian formulation, which is eventually converted to a LSTM mechanism (Section 6.2.4). We'll also see in Section 6.3.4 how this higher-order Markovian relationship is reflected in the correlation between class evidence and the prediction accuracy.

6.2.1 Problem Definition

Let $C_t$ be a random variable whose value, e.g., branch direction or bifurcation presence, needs to be estimated at the branch centerline point of step $t$. A realization of $C_t$ is denoted $C_t^*$, and $I_t$ be the corresponding image feature learnt from a sequence of image patches. Then the Maximum A Posteriori formulation for estimating $C_t$ is given by:

$$C_t^* = \arg \max_{C_t} P(C_t|C_{t-1}, C_{t-2}, ..., C_1, I_t, I_{t-1}, ..., I_1). \tag{6.1}$$
In the following, we show that although the maximization of this posterior probability can be intractable (Section 6.2.2), we are able to maximize its variational lower bound instead (Section 6.2.3), a common and effective trick utilized in variational inference [26]. We find the solution to this lower bound problem using LSTM (Section 6.2.4).

6.2.2 Central Question

We break down the approximation of Equation 6.1 into several steps. First we show two statements (with proof), which essentially state that the optimization in Equation 6.1 could be reformulated as a simpler optimization (that of the right hand side of Equation 6.4).

**Statement 1.** If \( X \) and \( Y \) are conditionally independent variables given \( Z \), \( P(X), P(Y), P(Z) \) and \( P(X,Y) \) are prior probabilities, and \( P(Z) \) is constant, then

\[
P(Z|X,Y) \propto P(Z|X)P(Z|Y).
\]

**Proof.** By using Bayes’ theorem and the definition of conditional independence,

\[
P(Z|X,Y) = \frac{P(X,Y|Z) \cdot P(Z)}{P(X,Y)} \propto P(X,Y|Z) = P(X|Z) \cdot P(Y|Z)
\]

\[
= \frac{P(Z|X) \cdot P(X)}{P(Z)} \cdot \frac{P(Z|Y) \cdot P(Y)}{P(Z)} \propto P(Z|X) \cdot P(Z|Y).
\]

\[\text{(6.2)}\]

Please note the proof in (6.3) does have a strong assumption that \( P(Z) \) is constant. However this assumption is intrinsically true in our application, as we use \( P(Z) \) to predict the existence of bifurcation or branch direction, modelled by a uniform distribution in the absence of prior knowledge.

Defining \( \mathbb{I}_t = \{I_t, I_{t-1}, ..., I_1\} \) and \( C_t = \{C_t, C_{t-1}, ..., C_1\} \), and replacing \( X, Y, Z \) with \( I_t, (\mathbb{I}_{t-1}, C_{t-1}) \), \( C_t \), respectively, we attain Statement 2.

**Statement 2.** If \( I_t, \mathbb{I}_{t-1} \) and \( C_{t-1} \) are conditionally independent given \( C_t \), then

\[
P(C_t|I_t, \mathbb{I}_{t-1}, C_{t-1}) \propto P(C_t|I_t) \cdot P(C_t|\mathbb{I}_{t-1}, C_{t-1}),
\]

\[\text{6.4}\]

i.e, the inference of candidate \( C_t \) could be separated into inference from image features at time step \( t \) and the inference from all previous states and their respective image features. Since our goal is to find a feasible algorithm to approximate (6.1), we must approximate \( P(C_t|\mathbb{I}_{t-1}, C_{t-1}) \) instead. This could be achieved by calculating its variational lower bound (ELBO) using Jensen’s inequality.
6.2.3 ELBO Calculation

We first introduce hidden variables $H_t = \{h_t, h_{t-1}, ..., h_1\}$ to encode non-observable variables (i.e., beyond $I$ and $C$ of the tracking process). We denote the prior distribution of $H_{t-1}$ by $q$. Then, since the logarithm function is concave, by Jensen’s inequality, $\mathbb{E}(\log(X)) \leq \log(\mathbb{E}(X))$, which we use to arrive at (rewriting $I_{t-1}, C_{t-1}$ as $\Theta$ for readability):

$$
\log P(C_t|\Theta) = \log \int_{H_{t-1}} P(C_t, H_{t-1}|\Theta) = \log \int_{H_{t-1}} P(C_t, H_{t-1}|\Theta) \frac{q(H_{t-1})}{q(H_{t-1})}
$$

$$
= \log \left( \mathbb{E}_q \left[ \frac{P(C_t, H_{t-1}|\Theta)}{q(H_{t-1})} \right] \right) \geq \mathbb{E}_q \left[ \log \left( \frac{P(C_t, H_{t-1}|\Theta)}{q(H_{t-1})} \right) \right] = \mathcal{L}B. \quad (6.5)
$$

Since $\mathbb{E}_q[\log P(C_t, H_{t-1}|\Theta)] = \mathbb{E}_q[\log(P(C_t|H_{t-1}, \Theta) \cdot P(H_{t-1}|\Theta))]$ and $\log(x/y) = \log x - \log y$, the variational lower bound $\mathcal{L}B$ of (6.1) can be further simplified to:

$$
\mathcal{L}B = \mathbb{E}_q[\log(P(C_t|H_{t-1}, \Theta))] + Q(H_{t-1}, \Theta), \quad (6.6)
$$

where $Q(H_{t-1}, \Theta) = \mathbb{E}_q[\log(P(H_{t-1}|\Theta))] - \mathbb{E}_q[\log(q(H_{t-1}))]$.

6.2.4 Adopting LSTM

Given their ability to encode sequential data, we adopt LSTM to learn the sequential information within the tracking process. Unlike the memoryless Markovian assumptions, the adoption of the LSTM network naturally encodes all information from previous steps in $h_{t-1}$, which approximates corresponding terms in (6.4) and (6.6) as $P(C_t|I_t) \approx P(C_t|I_t, h_{t-1})$ and $P(C_t|H_{t-1}, \Theta) \approx P(h_{t-1}|H_{t-1}, \Theta)$. See the proposed architecture in Figure 6.1.

---

Figure 6.1: (color figure) Schematic representation of Tree-LSTM. Left column: A sequence of patches from an image. Green box: predicts $P(h_{t-1}|H_{t-1}, \Theta)$. Blue box: predicts $P(C_t|I_t, h_{t-1})$. Right column: final output.
6.2.5 Implementation Details

We use two CNN implementations, TreeNet [246] and DCNN [229], to extract features \( I_t \), which are then fed into the LSTM cell of Sak et al. [187]. We use two fully connected layers (with 1024 and 64 nodes respectively) before the output layer of the network and use the 1024-dim vector as \( I_t \). The CNN model is first trained then kept fixed during LSTM training. For the LSTM network, a hidden state vector length of 32 is used with a sequence length \( L = 10 \). Our model uses a separate LSTM for each task as \( C_t \) encodes either the presence of bifurcation (\( \mathcal{L}_{\text{CLASS}} \)) or the tracking direction (\( \mathcal{L}_{\text{DIR}} \)), with corresponding losses:

\[
\mathcal{L}_{\text{CLASS}} = \sum_{t=1}^{L} |c_t - c^g_t|
\]

\[
\mathcal{L}_{\text{DIR}} = -\sum_{t=1}^{L} \left\langle \vec{v}_t, \vec{v}^g_t \right\rangle,
\]

(6.7)

where \( c \) is the predicted class (\( c = 0 \) for single branch, \( c = 1 \) for bifurcations) and \( \vec{v} \) is the predicted direction(s) (1 direction predicted for \( c = 0 \) and 3 directions predicted for \( c = 1 \)). Superscript \( g \) implies ground truth values. We use stochastic gradient descent for optimization with momentum 0.999, exponential decay ratio 0.6, and initial learning rate \( 1e^{-8} \) for direction prediction and \( 1e^{-5} \) for bifurcation classification. The implementation is based on TensorFlow v1.12 on Nvidia GTX 1080Ti GPUs.

6.3 Experiments

6.3.1 Datasets

We use the EXACT\(^1\) challenge public training dataset with 4-fold cross validation. Three major airway branches – trachea, left main bronchi (LMB) and right main bronchi (RMB) are extracted for evaluation.

6.3.2 Competing Methods

For evaluation, we use three state-of-the-art works: Zhao et al. [244] (RF); TreeNet [246]; and DCNN [229] (the last layers of TreeNet and DCNN are modified for classification and regression purposes). To isolate the benefit of adding LSTM, we evaluate the two CNN-based methods, TreeNet and DCNN, without and with LSTM. We change the input patch size to 32 and use the same loss function (\[246]\)) for both TreeNet and DCNN, and remove the middle layer of TreeNet for computational speed up. For evaluation, we measure bifurcation classification accuracy (\( \text{ACC}_{\text{CLASS}} \)) between output class (\( c \)) and ground truth class (\( c^g \)), and direction prediction accuracy (\( \text{ACC}_{\text{DIR}} \)) between output direction (\( \vec{v} \)) and

\(^1\)http://image.diku.dk/exact/
ground truth class ($\vec{v}^g$)

$$ACC_{\text{CLASS}} = 1 - |c_t - c_t^g|$$

$$ACC_{\text{DIR}} = \langle \vec{v}_t, \vec{v}_t^g \rangle.$$ 

(6.8)

6.3.3 Prediction Evaluation Results.

From Table 6.1a, we see that TreeNet+LSTM outperforms TreeNet by 21%, and DCNN+LSTM outperforms DCNN by 17%. RF performs better than TreeNet and DCNN alone on almost all cases, and performs no worse than TreeNet+LSTM on the trachea. This is not surprising as RF was designed for the purpose of bifurcation classification whereas TreeNet and DCNN were both designed for direction prediction. From Table 6.1b, adding LSTM, boosted TreeNet’s performance by 11% and DCNN by 18%, on average.

<table>
<thead>
<tr>
<th>Branch</th>
<th>Trachea</th>
<th>LMB</th>
<th>RMB</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF [244]</td>
<td>0.74(±0.49)</td>
<td>0.67(±0.45)</td>
<td>0.61(±0.45)</td>
</tr>
<tr>
<td>TreeNet [246]</td>
<td>0.35(±0.49)</td>
<td>0.54(±0.50)</td>
<td>0.57(±0.50)</td>
</tr>
<tr>
<td>TreeNet+LSTM</td>
<td>0.61(±0.39)</td>
<td>0.74(±0.44)</td>
<td>0.75(±0.44)</td>
</tr>
<tr>
<td>DCNN [229]</td>
<td>0.74(±0.46)</td>
<td>0.61(±0.50)</td>
<td>0.45(±0.50)</td>
</tr>
<tr>
<td>DCNN+LSTM</td>
<td>0.80(±0.41)</td>
<td>0.78(±0.43)</td>
<td>0.72(±0.45)</td>
</tr>
</tbody>
</table>

(a) Bifurcation classification accuracy (mean± std)

<table>
<thead>
<tr>
<th>Branch</th>
<th>Trachea</th>
<th>LMB</th>
<th>RMB</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF [244]</td>
<td>n/a†</td>
<td>n/a†</td>
<td>n/a†</td>
</tr>
<tr>
<td>TreeNet [246]</td>
<td>0.71(±0.12)</td>
<td>0.79(±0.11)</td>
<td>0.80(±0.08)</td>
</tr>
<tr>
<td>TreeNet+LSTM</td>
<td>0.82(±0.12)</td>
<td>0.92(±0.06)</td>
<td>0.90(±0.05)</td>
</tr>
<tr>
<td>DCNN [229]</td>
<td>0.76(±0.15)</td>
<td>0.57(±0.21)</td>
<td>0.84(±0.13)</td>
</tr>
<tr>
<td>DCNN+LSTM</td>
<td>0.87(±0.12)</td>
<td>0.93(±0.05)</td>
<td>0.92(±0.07)</td>
</tr>
</tbody>
</table>

(b) Direction prediction accuracy (mean± std)

Table 6.1: Detection accuracy. †Not applicable as RF does not predict direction.

6.3.4 Assessing LSTM’s Ability to Leverage Sequential Data

In Section 6.2, we hypothesized that LSTM is applicable to higher-order Markovian inference due to $P(C_t|H_{t-1}, I_{t-1}, C_{t-1}) \approx P(C_t|h_{t-1}, I_t)$, which suggests the model directly learns information from $H_{t-1}$ as a whole. To this end, we wish to validate that Tree-LSTM
prediction accuracy improves with increased evidence along an $L$-long sequence of patches. So, we define an evidence support measure within the sequence as $B = \sum_{i=1}^{L} \beta_i$ where $\beta_i = 1$ indicates the presence of a bifurcation in the ground truth data and 0 otherwise. Now, we bin our data based on $B \in \{1, 2, \cdots, 10\}$ and measure, for every evidence bin, the average bifurcation classification accuracy at the last or $10^{th}$ patch, i.e. $P(C_{10}|H_9, I_9, C_9)$. Table 6.2 records the correlation values between average bifurcation classification accuracy and evidence, which clearly shows how adding LSTM improves the correlation substantially between $\sim 15\%$ and $67\%$. Intuitively speaking, an increased correlation value $\rho$ means, by seeing more evidence in the sequence (e.g., bifurcations found in $C_9$), the accuracy predicting $C_{10}$ would be increased, which is consistent with our initial assumption in (6.1).

<table>
<thead>
<tr>
<th>Branch</th>
<th>Trachea</th>
<th>LMB</th>
<th>RMB</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>TreeNet</td>
<td>DCNN</td>
<td>TreeNet</td>
</tr>
<tr>
<td>with LSTM?</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>$\rho$</td>
<td>0.09</td>
<td>0.61</td>
<td>0.12</td>
</tr>
<tr>
<td>improvement</td>
<td>25.74%</td>
<td>39.54%</td>
<td>15.98%</td>
</tr>
</tbody>
</table>

Table 6.2: Correlation values $\rho$ between average classification accuracy and degree of evidence for different branches. The percentage improvement in $\rho$ when using LSTM (with TreeNet and DCNN) is also reported (calculated as $(\rho_{LSTM} - \rho_{no})/2$, denominator 2 is the max possible change in $\rho$).

6.4 Conclusions and Discussions

To utilize predictions of all previous points along a tracked vessel/airway centerline, we extended the commonly adopted first-order tree branch tracking assumption to a higher-order Markovian process. We estimated the Bayesian variational lower bound of the proposed formulation and used the CNN+LSTM architecture to optimize tracking. We showed the advantage of using LSTM in tracking real clinical data where the proposed method outperformed the state-of-the-art by at least 11\%. The improvement in correlation values between bifurcation classification accuracy and amount of branch sequence evidence is improved by at least 15\%.

Future work should involve investigating the impact of choosing different lengths for the sequence as well as choosing over different loss functions, such as L2 loss and cross entropy loss, for the bifurcation classification task.
Chapter 7

Extracting Anatomical Trees via Leveraging Tree Statistics – A Geometrical and Topological Point of View

7.1 Introduction

Every year, millions of people are affected by circulatory and respiratory system problems. According to the World Health Organization, more than 3 million people died of chronic obstructive pulmonary disease (COPD) in 2012, which is equal to 6% of all deaths globally that year [165]. (Semi-)automatic tree structure segmentation from 3D images is important for both vasculature and airway analysis [61], diagnosis and pre-operative planning [145]. However, due to low contrast (especially in low-dose CT), similar attenuation coefficients (i.e., CT voxel values) of air and pulmonary parenchyma along with distorted tubular shapes (e.g., narrowed airways with COPD) and imaging artifacts, tree structure extraction becomes a challenging problem [177].

7.1.1 A Short Review of State-of-the-art Methods on Tree Structure Segmentation

Many methods for segmenting tree structures, such as airways and vasculature based on centerline tracking and machine learning have been proposed in recent years [177], but only a few were based on statistics and none of them utilized topological priors from an anatomical point of view, such as airway tree hierarchy, or geometrical statistics, such as branch angles and lengths, like we do in this work. Following the categories from Chapter 2, we discuss in the following some existing literature on tree-extraction from 2D and 3D images from data-driven (machine learning) prospective as well as model-driven perspectives:

1. Progressive methods:
(a) Region growing;
(b) Active contour;
(c) Centerline-based Tree Trackers;
(d) Minimal paths;

2. Graph based methods;

Machine Learning

This class of method relies on training machine learning systems to predict the class (or the probability of the class) of each pixel or voxel, where the two possible classes are tree (or branch) vs. background. Once the prediction map is obtained, different methods are used to join likely branch voxels into one contiguous tree segmentation. A voxel-wise classification method was proposed by Lo et al. [126,128], in which they used a KNN classifier to first generate an airway probability map followed by vessel tracking using region growing. Probability Boosting Tree (PBT) classifier was used in [252] instead to obtain the coronary vessel paths. There are also other works proposing novel features [244,253] for classifying bifurcations.

Region Growing

Most of the airway extraction methods, mentioned in the comparative study paper of Lo et al. [126], were based on region growing, which makes it still hard to integrate statistical information like tubular geometry or airway morphometry. For example, Feuerstein et al. [63] proposed a region growing method to extract the airways using LoG enhanced images. Graham et al. first extracted the airways by region growing, then filtered the segmentation into small airway segments, and eventually removing spurious false airway tree branches by optimizing a graph-partitioning problem [76].

Active Contour

As for active contour methods, most still use information like the gradient and Hessian, and it’s not obvious how to encode anatomical tree topology in these frameworks. Wang et al. [220] proposed to use 4D curves to extract tree structures, generalizing snake contours with cylinder models and propagated the front by gradient vector flow (GVF) force, the eigenvector of the Jacobian matrix and region intensity. Shang et al. [194] adopted a combined approach, first segmenting wider vessels by propagating the active contour front using intensity information, then segmenting smaller vessels with Hessian eigenvectors and eigenvalues and investigating the characteristics of principal curvatures.
Centerline-based Tree Trackers

These methods are typically initialized with a seed at the root of the tree, then the tracker advances from this seed down through the tree. To explore different branches, these trackers either (a) explicitly locate bifurcations and split accordingly into children branch trackers \([38, 64, 142, 219]\); or (b) explore different possible paths thus implicitly defining tree bifurcations, e.g., the particle filtering based coronary vessel tracking method of Florin et al. \([64]\). A key computational module in such trackers is a search for the next point along the tree branch to advance to. This in turn typically requires two types of “regularization”: The first is to regularize the path to avoid sudden abrupt turns by the tracker, e.g., via Kalman filtering, as proposed in the 2D retinal blood vessel tracking method of Chutatape et al. \([47]\). The other is to regularize the data via image de-noising and tube (vesselness or ridge) filtering, as proposed by the 3D tracking method of Kumar et al. \([112, 113]\).

Minimal Paths

Here each branch of a tree-like structure is one that “optimally” connects a pair of (start-end) points. Typically, manual seeding combined with image processing is used to detect likely start-end points. Optimality, on the other hand, is obtained using a potential map that is constructed based on image evidence, e.g. paths that pass through pixels with a strong vesselness (Frangi et al. \([67]\)), are more favourable, as proposed by Soleimanifard et al. \([201]\). In their work, the energy functional satisfied the Eikonal equation and was optimized using level sets. In the work by Breitenreicher et al. \([29]\) the potential map was based on extracting Haar features at different scales.

Graph based Methods

A graph based method was proposed in Bauer et al. \([19, 20]\), detecting tube-like structures in the beginning and then connecting them into a tree graph. In the work of Hu et al. \([92]\), graph-cut was performed locally in a neighboring sphere, segmenting vessel points by intensity differences. In \([168]\), accurate airway wall extraction was obtained by an optimal graph construction method, which was based on an initial coarse airway segmentation provided by an algorithm proposed by Lo et al. \([126]\).

7.1.2 Incorporating Tree Statistics

Although in some of the works mentioned above, statistical methods like Bayesian inference were applied, as in the work of Wang et al. \([219]\), no tree statistics were actually used. In the work of Florin et al. \([64]\), the inference rule was simply applied on inferring parameters from the last tracking step, so no global statistical information was embedded. On the other hand, topological information like branching types (in mice airways) has been analyzed in the work of Grothausmann et al. \([78]\), however, the aim of their work is not
about airway extraction and the information extracted is yet unknown how to further help in airway (human’s, especially) segmentation. In general, our key contribution in this paper is fully utilizing tree topology information and geometrical statistics to aid the segmentation process, while none of the methods surveyed have explored.

Tree-like structures in living beings, like the Circle of Willis in the brain, the airways in the lung and the abdominal arteries, are not perfectly identical across a population. Nevertheless, these trees conform to particular topological and geometrical patterns largely consistent across the normal (non-pathological) adult population. In particular, the first several generations of branching (e.g. root, first two children, the four grandchildren) are known to respect a well-defined hierarchy, which our approach is designed to leverage.

In this paper, we propose a new tree tracking method that incorporates, not only image features (or appearance priors), but also, and for the first time, geometrical and topological tree priors to improve tracking and bifurcation detection of tree-structures in 3D medical images (such as magnetic resonance imaging, MRI, or computed tomography, CT). The geometrical priors include branch length and angular statistics, which are learnt from segmented training images. Our tracker conforms to a topological prior that enables leveraging the appropriate branch- and level-specific geometrical statistics. For example, the trachea always branches into left and right main bronchi, while left main bronchus further develops into superior and inferior lobar bronchus, and right main bronchus into superior and intermediate lobar bronchus. We implement our method by adopting a Bayesian formulation that incorporates the aforementioned tree-priors. Quantitative and qualitative results show that by using geometrical and topological priors, the accuracy and stability of tree extraction is significantly improved.

7.2 Method

7.2.1 Tree Extraction with Tree Priors

At a high level, statistics learnt directly from the dataset should provide guidance to the centerline tree tracking procedure. In this section, we describe how a Bayesian formulation could be adopted to infer bifurcation locations while tracking tree centerlines, with topological and geometrical priors extracted from the dataset.

The proposed tracker works by continually advancing through the image to map the airway/vessel centerlines, and bifurcating when a branching point is detected. Branch length statistics play the important role of weighting the probability to bifurcate, which is inferred from the image features (i.e., the likelihood), by probabilities inferred from the geometrical priors, via Bayes’ theorem. Branch angular statistics further help to locate daughter branches by giving lower penalties to branching directions that agree with angular priors. The tree topology, on the other hand, is embedded as follows: As our tracker progresses from the tree root downward along the branches into 3D images, we continuously
update a tree data structure with a crawler pointer or tag indicating where on the tree model our tracker is, which branch (trachea, right main bronchi, ...) or at which level (level 1, 2, 3, ...) is currently actively being tracked. This allows us to pull the corresponding level- and branch-specific geometrical priors.

To extract the full tree, a key decision of where, down along each tracked branch, the tracker must bifurcate (e.g. where to bifurcate the root into children, or the each child into grandchildren). Additionally, once the decision to bifurcate is made, the initial tracking directions of the two child branches must be resolved. The next two sections focus on describing the details of these key modules.

7.2.2 Bifurcation Classification Criterion

We denote $B_i$ as the occurrence of bifurcation at the tree level $i$, $I$ the image features, $L_i$ the random variable representing the detected branch length, and $l$ the prior level- and branch-specific length value. We assume $I$ and $L_i$ are independent, since branch length is a global geometrical property, while the image features are locally defined. $P(I)$ and $P(L_i \leq l)$ are prior distributions, since they are not relevant to $B_i$, they could be replaced by constant values here. We also assume $P(I|B_i)$ to be of uniform distribution. Thus, the probability of finding a bifurcation at tree level $i$ with detected branch length $L_i$ is as follows:

\[
P(B_i|I, L_i \leq l_i) = \frac{P(B_i)P(L_i \leq l_i|B_i)}{P(L_i \leq l_i, I)}
\]

\[
= C' \frac{P(B_i|I)P(I)P(L_i \leq l|B_i)P(I|B_i)}{P(L_i \leq l_i, I)}
\]

\[
= CP(L_i \leq l_i|B_i)P(B_i|I).
\]  

The probability map $P(B_i|I)$ could be generated by a general bifurcation detection classifier. $P(L_i = l|B_i)$ can be collected from training data or based on expert knowledge of anatomy (see section 7.3.1) and $P(L_i \leq l|B_i)$ is its cumulative density function (CDF). If we define weight $\omega = CP(L_i \leq l|B_i)$, $C$ is some constant, then:

\[
P(B_i|I, L_i \leq l_i) = \omega P(B_i|I).
\]  

The intuition behind Equation (7.2) is that, when the tracked branch length is much shorter than the mean length, it is less likely to have a bifurcation detected ($\omega < 1$); but as the length of the tracked branch exceeds the mean length, the probability of finding a bifurcation will increase nonlinearly (up to the scale $C$ ($C > 1$), since we expect a bifurcation will eventually be found unless an end point is reached).
See Figure 7.1 as an example of where, \( \omega \geq 1 \) after reaching mean branch length \( l = 60 \) and is non-linearly increasing (at a rate dependent on the standard deviation, 15, in this case).

Figure 7.1: (color figure) Probability of a parent branch to bifurcate given prior distribution of the parent branch length. Branching weight \( \omega = CP(L_i \leq l|B_i) \) (as in Equation 7.2) increase according to branch length, where \( C = 2, P(L_i \leq l|B_i) \) is Gaussian cumulative density function, given mean branch length \( l = 60 \), standard deviation 15.

### 7.2.3 Daughter Branches Detection Criteria

The probability \( P \in [0, 1] \) of bifurcating, given image evidence and prior level- and branch-specific lengths, calculated in section 2.2, will now be converted to a decision to bifurcate. One naïve approach is to bifurcate whenever \( P > 0.5 \). However, this approach does not take into account any evidence whether this candidate location \( x_0 \) for a bifurcation will result in two plausible child branches. To determine whether two branches can be identified at \( x_0 \), we follow the following procedure that leverages the branch angular statistics (Figure 7.2).

1. Denote the parent branch direction at \( x_0 \) as \( \vec{p} \). First threshold the neighboring points so that any point \( y \) behind \( x_0 \) is discarded. This is computed by checking whether \( \cos(<y - x_0, \vec{p}>) \leq 0 \).

2. Denote the mean direction and standard deviation of each daughter branch as \( \mu_i \) and \( \theta_i \) (training of \( \mu_i \) and \( \theta_i \) is found in section 7.3.1), respectively for branches \( i = 1, 2 \). Then threshold the neighboring points to be within 3 standard deviations \( \theta \) (99.7% confidence interval) of the daughter branch angles. This is done by checking:
\[ \cos (\mu_i + 3\theta_i) \leq \cos (\langle y - x_0, \vec{p} \rangle) \]  
\[ \cos (\mu_i - 3\theta_i) \geq \cos (\langle y - x_0, \vec{p} \rangle), i = 1, 2. \]  
(7.3)

If the criterion in Equation (7.4) is satisfied, point \( y \) is kept; otherwise \( y \) is discarded (Figure 7.2).

3. The remaining neighboring points are clustered into two parts using k-means.

(a) If at least one cluster is empty, then the current point \( x_0 \) is rejected as a bifurcation, since we can’t find its two daughter branches. The tracker will keep tracking the centerline of the current branch.

(b) Otherwise, there are two cluster centroids \( c_i, i = 1, 2 \). Then we check if the following two criteria are satisfied:

i. mean image intensity along the path \( x_0 \) and \( c_i \) is within the given intensity threshold;

ii. mean image intensity along the path \( c_1 \) and \( c_2 \) is beyond the intensity threshold, indicating there is a clear separation between the daughter branches.

If both criteria above are satisfied, then \( x_0 \) is accepted a bifurcation and \( c_1, c_2 \) will be set as starting points of tracking the daughter branches.

### 7.3 Experimental Results

#### 7.3.1 Description of the Datasets

**Synthetic Data**

We generated 330 binary volumes, each of size 256*256*256 and containing a binary tree structure with 4 levels. The tree statistics are illustrated in Figure 7.3 and include: mean and standard deviation (std) of branch lengths that describe \( P(L_i = l | B_i) \), as well as mean angles \( \mu_i \), and angle standard deviation (std) \( (\theta_i \) in section 7.2.3), all assuming Gaussian distribution (it’s worth to note the trees are not symmetric). Spatially-variant Gaussian noise was added to approximate low-dose CT acquisition \( [130] \), with standard deviation \( \gamma \* e^{0.5(f+1)/3} \) (so that std is nonlinearly proportional to image intensity and std \( \leq 1 \) when \( \gamma \leq 1 \)), where \( f \) is image intensity, and different noise levels \( \gamma \in \{0.6, 0.8, 1.0\} \) (Figure 7.4).

**Clinical Data**

The clinical dataset was obtained from the 2009 MICCAI challenge \([2, 167]\). We used all 16 ground truth segmentation results provided by the organizers to generate ground
Figure 7.2: (color figure) Illustration of daughter branches detection using angular statistics. 
p is the current branch direction. The red dashed circle is the neighborhood of \( X_0 \) (current search point). \( \mu_1 \) and \( \mu_2 \) are prior mean daughter directions, \( \theta_1 \) and \( \theta_2 \) their respective standard deviations. The pink areas are filtered neighboring points; any point outside will be discarded. \( C_1 \) and \( C_2 \) are the resulting clustered centroids that act as roots for tracking the daughter branches.

truth centerlines by running the fast marching algorithm [1], and then calculate length and angle priors from these centerlines. Seven branches: trachea, right main bronchi (RMB), left main bronchi (LMB), right superior lobar bronchus (RSB), right intermediate bronchus (RIMB), left superior lobar bronchus (LSB) and left inferior lobar bronchus (LIFB) were manually labeled by a graduate student. See branch and angular statistics in Table 7.1 and Figure 7.5.

<table>
<thead>
<tr>
<th></th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trachea</td>
<td>RMB</td>
<td>LMB</td>
</tr>
<tr>
<td>mean length</td>
<td>133.29</td>
<td>51.16</td>
<td>75.75</td>
</tr>
<tr>
<td>std length</td>
<td>36.40</td>
<td>19.32</td>
<td>18.61</td>
</tr>
</tbody>
</table>

Table 7.1: Length statistics (in units of voxels) of real data.

7.3.2 Preprocessing and Initialization

In this paper, we manually set the starting point for the tracker in the synthetic data experiment. For the clinical data experiment, we adopt an approach similar to [118,223], selecting a region of interest (ROI) by first choosing the uppermost 3/4 slices and a 256*256 square region in the center of each slice. The volume is then filtered by a 5*5 median filter on each slice. The 30th slice counting from the top is used for detecting the seed point. By
Figure 7.3: (a): (color figure) Tree structure in a synthetic data; (b): Statistics of branch lengths; (c): Statistics of angles.

<table>
<thead>
<tr>
<th></th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
<th>Level 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>80</td>
<td>60</td>
<td>40</td>
<td>30</td>
</tr>
<tr>
<td>std</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Angle 1</th>
<th>Angle 2</th>
<th>Angle 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>45°</td>
<td>45°</td>
<td>30°</td>
</tr>
<tr>
<td>std</td>
<td>5°</td>
<td>5°</td>
<td>5°</td>
</tr>
</tbody>
</table>

Figure 7.4: Synthetic 3D data with noise. Noise level from left to right, top to bottom: 0, 0.6, 0.8, 1.0.

detecting dark circles of radii between [5, 50] pixels, we choose the one closest to the center of the slice as the seed point.
7.3.3 Testing Results

Synthetic Data

Performance on the synthetic data was measured based on the number of detected branches not leaking into background regions. The results are summarized in Table 7.2, which show that, even as the noise level increases from 0.6 to 1.0, the overall performance using statistical priors shows a steady improvement of at least 13% with statistics.

<table>
<thead>
<tr>
<th>Noise Level</th>
<th>0.6</th>
<th>0.8</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>WOS</td>
<td>976</td>
<td>948</td>
<td>878</td>
</tr>
<tr>
<td>WS</td>
<td>1250</td>
<td>1200</td>
<td>1000</td>
</tr>
<tr>
<td>Improvement</td>
<td>28.69%</td>
<td>26.58%</td>
<td>13.90%</td>
</tr>
</tbody>
</table>

Table 7.2: Number of branches detected in synthetic data. WOS: method without statistics. WS: method using statistics.

Clinical Data

Evaluation was performed using leave-one-out cross-validation. Accuracy was measured by calculating the shortest Euclidean distances from the centerlines of the given ground
truth segmentations (i.e. ground truth centerlines) to the detected centerlines. Qualitative results in Figure 7.6 show improved bifurcation detection and tree topology representation when using tree statistics. Quantitative results in Table 7.3 show that the mean distance from ground truth centerlines to detected centerlines (i.e. the error) is reduced by 12% by adding statistical priors. By running a paired t-test, we show that the result is of statistical significance with $p \leq 0.05$. Overall, we noted that by adding statistic priors, the tracking accuracy is significantly improved.

In both synthetic and clinical data experiments, we replace $P(B_i | I)$ in equation 7.2 by the Random Forest classifier as described in [244] with the same image features. The classifier is trained on synthetic samples in the synthetic data experiment and on real samples in the clinical data experiment.

Figure 7.6: (color figure) Centerline detected in real data. The pink surface is the binary segmentation of the volumes provided by the EXACT challenge. The yellow centerline corresponds to trachea (1st level), blue centerlines correspond to RMB and LMB (2nd level) and black centerlines correspond to RSB, RIMB, LSB and LIFB (3rd level). Figures in the bottom row are the zoomed-in cyan regions in the top row. Note the missing bifurcation highlighted by cyan and green circles in the left vs. properly detected in the right.
<table>
<thead>
<tr>
<th></th>
<th>Mean Distance</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WOS</td>
<td>10.49</td>
<td>20.06</td>
</tr>
<tr>
<td>WS</td>
<td>9.36</td>
<td>19.46</td>
</tr>
<tr>
<td>Improvement</td>
<td>12.07%</td>
<td>3.08%</td>
</tr>
</tbody>
</table>

Table 7.3: Distance (in units of voxels) from ground truth centerline in real clinical data to detected centerlines. WOS: method without statistics. WS: method using statistics. The improvement is of statistical significance with $p \leq 0.05$.

### 7.4 Conclusion

In this work, we presented the first method in the medical image analysis field that incorporates priors of anatomical trees into a tree extraction algorithm. We developed a Bayesian based algorithm for tracking tree centerlines, utilizing statistical geometry priors as well as tree topology priors (including tree hierarchy, branch angle and length statistics). By testing on synthetic images of different noise levels and real CT chest scans, the proposed method showed a clear improvement in terms of both stability and accuracy. This is supported in our experiments over real clinical data. While in our diagrams (e.g., Figure 7.3 and 7.5) and the results, we adopted a binary tree representation (containing only bifurcations), known to constitute the majority of real anatomical branchings patterns, our approach could be naturally extended to trees with trifurcations and more generations (levels) of branching. Future works will involve training and applying our method to tree-like vasculature with other tree topologies, such as the Circle of Willis in the brain to highlight areas of pathology, e.g. aneurysms. A possible weakness of our approach is that the tracking is limited by the adoption of a myopic, local decision making process; a multi-hypothesis extension, allowing the tracker to explore different paths, may yield further robustness to noise.
8.1 Introduction

Branching tree-like anatomical structures are abundant in the human body (i.e., vascular and airway trees of circulatory and respiratory systems) and analyzing their properties is important for various clinical applications, e.g., diagnosis and surgical planning. A necessary precursor to morphological tree analysis is segmenting the trees from 3D medical images. However, 3D segmentation of tree structures is challenging due to, e.g., insufficient contrast between vessels or airways and background, neighbouring/touching tissue, and geometrical variability. Extracting the trees amounts primarily to identifying the bifurcations and the curvilinear paths between them.

Several previous works on segmentation of tree-like structures relied on local, voxel/patch-level information. For example, Frangi et al. proposed a filter based on local Hessian matrix [67]; Law et al. estimated branch direction based on optimal local inward flux orientation [115]; Schneider et al. used steerable filters and random forest for pixel-wise classification [190]; and Wu et al. proposed a deep learning framework to classify local patches for tracking [232].

Tracking based methods, on the other hand, provide better structural information, but they generally fail to build a global tree structure. Lesage et al. proposed a particle filtering method to track coronary vessels, which incorporate vessel geometry using flux based features [122]. Macedo et al. [135] proposed a centerline-tracking method, on top of a 2D feature based bifurcation detector.

Incorporating prior knowledge, like geometry and topology, into optimization based image segmentation algorithms has been proven useful for obtaining more accurate and
plausible results [162]. However, these priors typically introduce non-convexities in the objective functions.

Although tree-like structures were extracted in [179, 181, 200, 211, 212], in contrast to our work, their trees need to be seeded in tubularity measurement maps and the initial tree topologies are seeding-dependent. In [211, 212], edge pair-wise geometrical, instead of topological, prior dominated the optimization process, which makes it impossible to maintain a desired, fixed anatomical structure. While in [179, 181, 200], the topological priors were interpreted as 2-tuple or 3-tuple of neighboring edges, instead of constructing the whole anatomical tree structure. To segment coronary vessels on 2D xray sequences, M’hiri et al. used temporal and spatial prior inherited from an earlier xray image, but the method is difficult to extend to 3D [147]. Beriault et al. proposed a CRF framework that used brain structures (e.g., basal ganglia) and sinuses locations as anatomical priors for segmenting the cerebral vasculature [34]. But none of these methods addressed the global branching aspect of anatomical tree structures.

Our goal in this work is to perform 3D tree extraction while satisfying these two important objectives:

1. Encode the geometrical and topological priors of trees;
2. Ensure a globally optimal tree extraction solution.

In this paper, we achieve both objectives by adopting, for the first time, pictorial structures for tree extraction.

Pictorial structure were introduced into the computer vision community in 2005 (Felzenszwalb et al. [60]) and an extensive literature (e.g., Belagiannis et al. [22], Belagiannis et al. [22]) has been established based on this concept since then. To ensure global optimality, pictorial structures require a model with a tree-like topology. This property makes them a natural and ideal fit to the problem of anatomical tree extraction.

8.2 Methodology

At a high level, our automatic approach comprises the following two key steps (Figure 8.1):

1. Detecting bifurcations;
2. Extracting centerlines of branches connecting bifurcations.

Step 1 is achieved via fitting the pictorial structure to the 3D image data by globally optimizing an energy function with an artificial neural network (ANN) derived unary term and a geometrical statistics based binary term. Step 2 is achieved via a globally optimal minimal path extraction.

The reasons we adopt the pictorial structure include the following:
1. The bifurcation locations (which are set to be the pictorial components) are connected with local visual properties, which can be easily encoded as unary term of the pictorial structures;

2. We want to encourage a "relaxed" relationship among the bifurcation locations, which could be replaced by statistical information, instead of a fixed relationship. This can be encoded easily using the binary term of the pictorial structure;

3. When given in a tree form, the pictorial structure could be optimized with a global optimization method, avoiding local optima.

Figure 8.1: (color figure) An overview of the proposed method.

8.2.1 Bifurcation Detection

We formulate the problem of bifurcation detection in 3D as a pictorial structure optimization. A pictorial structure models a deformable object by a set of connected parts. This technique finds the instances of an object in an image by measuring the matching cost for each part and a deformation cost between each two connected components. Felzenszwalb et al. [60] restricted the connection of components to form an acyclic graph $G = (V, E)$, where each vertex $v_i$ corresponds to a component and each edge $e_{ij} = (v_i, v_j)$ models a connection between vertices $v_i$ and $v_j$. We encode the 3D anatomical tree bifurcations as the nodes of the pictorial structure whereas branch directions and lengths statistics are learned as geometrical priors for regularizing pictorial edges. Let $I(x)$ be an $N$ dimensional image and $x \in \mathbb{R}^N$, we optimize an energy function over the location of $n$ nodes in $N$ dimensional space, as follows:

$$L^* = \arg \min_{L = \{L_1, \ldots, L_n\}} \left( \sum_{i=1}^{n} U(L_i | I) + \sum_{e_{ij} \in E} B(L_i, L_j) \right) \quad (8.1)$$
where $U(I, L_i)$ is the unary term penalizing locating $v_i$ at location $L_i$ and $B(L_i, L_j)$ is the binary term penalizing the deformation of the vector $L_{ij} = L_j - L_i$ away from geometrical priors learned from training data. By leveraging the generalized distance transform [58], the pictorial energy function in (8.1) is efficiently and globally minimized.

**Unary Term via an Artificial Neural Network**

We train a three layer neural network stacked with Restricted Boltzmann Machines (RBM) to build a distance map and use it as the unary term in (8.1). RBM is a two-layer network of visible and hidden units with no intra-layer connections and symmetrically weighted inter-layer connections. Instead of initializing the network by small random weights, we pre-train the network unsupervisely using RBMs [90]. RBMs compute the joint probability of visible and hidden units and provide a high level representation of data in an unsupervised manner. We construct the network by stacking three RBMs, considering hidden units of preceding ones as visible units of following RBMs. We then fine-tune an ANN, end-to-end, to predict tree voxels by minimizing the total cross entropy between predicted and ground truth segmentations of a training dataset. To encourage a detected bifurcation to be close to the center of its neighboring branches, we instead compute and predict a distance map from segmented edges rather than the segmentation maps themselves.

**Binary Term from Geometrical Statistical Priors**

We learn distribution of branch angles and lengths of anatomical trees from the skeletons of the ground truth segmentations of a training dataset. Anatomical branch angles and lengths are encoded as three dimensional displacement vectors pointing from bifurcations at lower generations of the tree to upper generations. We model the joint prior distribution of locations of two pictorial connected components as a multivariate Gaussian. The mean vector $\mu_{ij}$ and covariance matrix $\Sigma_{ij}$ of a displacement vector between nodes $v_i$ and $v_j$ are estimated from the training data. By applying singular value decomposition, i.e., $\Sigma_{ij} = U'_{ij}M_{ij}^{-1}U_{ij}$, we write the following joint likelihood estimation in the form of the Mahalanobis distance:

$$-\log p(L_i, L_j) \propto d_{ij}(L_i, L_j) = [T_{ij}(L_i) - T_{ji}(L_j)]'M_{ij}^{-1}[T_{ij}(L_i) - T_{ji}(L_j)]$$

(8.2)

where $L_i$ and $L_j$ are the locations of nodes $v_i$ and $v_j$, respectively; $T_{ij} = U'_{ij}(L_i - \mu_{ij})$ and $T_{ji} = U'_{ij}(L_j)$ are rigid (i.e. six degrees of freedom) spatial transformations [60]. $M_{ij}$ is a diagonal matrix weighting the deformation cost of connection $e_{ij}$. 
Optimization of Pictorial Structure using Generalized Distance Transform

The general idea of optimizing Equation 8.1 is to convert it to its generalized distance transform and apply the corresponding global optimization.

We define \( Q(L_i) \) recursively as the following, using the default parent/children relationship in the acyclic graph setting (see how to construct the acyclic graph in Section 8.2.1):

\[
Q(L_i) = \min_{L_j} \sum_j U(L_j | I) + \sum_j B(L_i, L_j) + \sum_{L_j \in C_i} Q(L_j)
\]  

(8.3)

where \( C_i \) is the children point set of bifurcation \( i \). This formulation could be further converted into the generalized distance transform representation:

\[
Q(L_i) = D_f(T_{ij}(L_i))
\]

(8.4)

where \( \Omega \) is the image definition domain and

\[
f(x) = U(T_{ij}^{-1}(L_j)(x)) + \sum_j B_{C_j}(T_{ij}^{-1}(x))
\]

(8.5)

\[
D_f(z) = \min_{x \in \Omega} ||z - x||_2 + f(x)
\]

(8.6)

Optimization of Equation 8.6 could be found in the work of Felzenszwalb et al. [58] and a detailed description of converting the pictorial structure into the generalized distance transform is given in Felzenszwalb et al. [59].

Forming the Acyclic Graph

To efficiently find a global solution to (8.1), we must pick a set of connections between pictorial components that form an acyclic graph (tree). One natural option is to adopt the anatomical tree connectivity as the pictorial structure tree connectivity (option 1). However, it is more informative to connect pairs of nodes with a consistent behaviour across the training data [58]. So, alternatively (option 2), we construct a complete weighted graph over all vertices, assign to edge \( e_{ij} \) a weight \( w_{ij} \) equal to 2-norm of covariance matrix \( \|\Sigma_{ij}\|_2 \), and finally find the Minimum spanning tree (MSP) of this graph by Prim’s algorithm [175]. We found that the detected tree structure is the same as anatomical tree.

8.2.2 Branch Centerline Extraction

We use a globally optimal minimal path extraction, based on the fast marching method, to extract the centerlines of all tree branches. While most minimal path extraction methods are semi-automatic and require users to provide a path’s start and end points [110,123,146,
In tubular structures, if the speed function of the minimal path algorithm is homogeneous or has a small variation near the actual centerline, the shortest path is detected as the Euclidean path instead of the medial path or centerline. To ensure that the detected paths pass along the centers of branches, we adapted Deschamps’ path centering algorithm [52]. Deschamps first extracted a rough centerline for tubular structures and then used the detected centerline to achieve a rough binary segmentation of vasculatures. A distance transform of the detected edges in the segmented vasculatures is computed and is fed to the minimal path algorithm as a new speed function.

In this paper, instead of segmenting using estimated centerline, we use the output mask \( M \) from our ANN as an approximate segmentation. Then we perform the same distance transform procedure as mentioned before and generate the corresponding speed function \( D = \text{DistanceTransform}(M) \). Given the speed function \( D \) and detected bifurcation \( L_i \) as starting points, the centerlines could be located as geodesic paths:

\[
G(p) = \inf_{g(0)=L_i} \int_0^L D(g(s)) ds 
\]

(8.7)

where \( G \) is solved using the fast marching algorithm to solve the Eikonal equation:

\[
\begin{aligned}
\left| \nabla G \right| &= D \\
G(L_i) &= 0
\end{aligned}
\]

(8.8)

8.3 Experiments

8.3.1 Data Description

Synthetic Data

We generated 50 volumes, each of size \( 150^3 \) and containing a binary tree structure with 4 levels. The tree statistics are set as: mean branch lengths of \( \{50, 40, 30, 20\} \) (voxels) and standard deviation (std) of 2; the mean angles between the neighboring levels are \( \{\pi/4, \pi/6, \pi/12\} \) with std \( = \pi/36 \).

Clinical CT Data

We used 19 chest computed tomography (CT) scans\(^1\). We performed 3-fold cross validation to train the model. The ANN is trained with 240,000 2D patches of size \( 29 \times 29 \) along axial planes around each voxel. We chose our ANNs based on empirical evidence and

\(^1\)http://image.diku.dk/exact/
previous works. Other ANN designs, or even other non-ANN methods for calculating the unary term, may yield even better results. A grid search is performed to set and fix the ANN hyperparameters. Our pictorial model consists of the seven components of the first four levels of an airway tree (i.e., trachea, left and right main bronchi, etc.)

8.3.2 Evaluation Measures

Bifurcation Detection Evaluation

The performance of the proposed approach is assessed by the following two metrics:

1. $N_D$: number of detected bifurcations with distance less than $D$ from the ground truth locations;
2. $M$: mean distance between the ground truth bifurcations and the corresponding closest detected bifurcations.

Branch Extraction Evaluation

We measure how well the detected curvilinear centerlines match the ground truth centerlines by computing $\mu_D$: the average distance between centerlines [134].

8.3.3 Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>$M$ (mm)</th>
<th>$\mu_D$ (mm)</th>
<th>bif. det./path init.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Tracker [135]</td>
<td>9.41 ± 6.84</td>
<td>9.21 ± 9</td>
<td>manual root seed</td>
</tr>
<tr>
<td>C MP + ANN + DT</td>
<td>NR†</td>
<td>3.09 ± 1.5</td>
<td>manual bif. det.</td>
</tr>
<tr>
<td>D ours (pict w/o stats + MP)</td>
<td>14.54 ± 16.54</td>
<td>4.87 ± 4.84</td>
<td>automatic</td>
</tr>
<tr>
<td>E ours (pict with stats + MP)</td>
<td>8.39 ± 7.41</td>
<td>3.51 ± 2.4</td>
<td>automatic</td>
</tr>
</tbody>
</table>

†NR: Not reported since bifurcations are manually selected.

Table 8.1: Performance of different methods on clinical data with measure $M$ and $\mu_D$. Distance unit in mm and values shown in format mean ± std. (MP: minimal path).

Advantage of Statistics

To confirm the advantage of incorporating tree statistics, we removed the tree statistics of real data and globally optimized the objective function. In practice, we mimicked a uniform geometrical prior by scaling up the covariance matrix elements by a factor of 20.
Rows D and E in Table 8.1 show that incorporating statistics improves $M$ and $\mu_d$ by 42% and 28%, respectively.

**Robustness to Noise**

To evaluate the robustness of our bifurcation detector to noise, we added three levels of Gaussian noise with SNR= $[10, 5, 3.3]$ to the noise-free synthetic data. A distance map from the edges of the tree mask is used as the unary term. Table 8.2 shows that our method is stable even to high level of noise. For example, in columns 3 and 4 of Table 8.2, SNR is doubled while $M$ is increased about one voxel.

<table>
<thead>
<tr>
<th>SNR</th>
<th>$M$ (voxel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>∞ (noise-free)</td>
<td>5.19 ± 3.30</td>
</tr>
<tr>
<td>10</td>
<td>5.29 ± 3.23</td>
</tr>
<tr>
<td>5</td>
<td>6.69 ± 11.29</td>
</tr>
<tr>
<td>3.3</td>
<td>7.83 ± 9.30</td>
</tr>
</tbody>
</table>

Table 8.2: Effect of SNR on measure $M$ for synthetic data ($mean \pm std$).

Figure 8.2: (color figure) Variation of $N_D$ on real data for proposed method and Tracker [135].
Advantage of Globally Optimal Model Fitting

We examined the drop in performance when a gradient descent local optimizer approach is used. As expected, the bifurcation localization result is highly sensitive to initialization even if the initialization is close to the ground truth locations. This sensitivity can be attributed to not having a reliable and clean (noise-free) data term, causing the local optimizer to get trapped in local optima. It is also worth noting that our algorithm is linear in both the number of branching points and the number of possible locations for each node. We also compare the proposed method to two competing methods, a tracker based on a bifurcation estimator (Tracker) [135] and the model-based optimally oriented flux method (OOF) [115].

Comparing to Tracker

The root seed point of Tracker is manually set in the trachea trunk. Since the tracker doesn’t have a built-in anatomical tree topology, we match each ground truth bifurcation to its closest one among all the detected bifurcations. Figure 8.2 reports $N_D$ as a function of $D$ for the proposed method (blue curve) and Tracker (brown curve) on real data. The two different plateau levels of the two curves illustrate that not all the bifurcations are detected by Tracker. Also, rows A and E in Table 8.1 illustrate that the proposed method outperforms Tracker, reducing the error by 10% in $M$ and 62% in $\mu_D$.

Comparing to OOF

To trace the centerlines using the tubularity score of OOF, we had to manually select bifurcations before using the fast marching algorithm to generate the path between those bifurcations. A Naïve comparison of rows B and E in Table 8.1 shows that OOF outperforms our method by about 3 mm in $\mu_D$ on real data. However, our proposed method doesn’t need initialization and is fully automatic. So, for a fair comparison, we used the same manually selected bifurcations and the centerline extraction approach in section 8.2.2. The result is reported in row C. Now, comparing rows B and C shows that, using the same set of bifurcations, the minimal path on distance transform of ANN output (i.e., our approach) outperforms OOF. Moreover, since the variation of $\mu_D$ for rows B, C and E is less than the average voxel size of our clinical data (0.67*0.67*0.95 $mm^3$), these experiments confirm that by detecting bifurcations using the pictorial structure, the tracing algorithm becomes fully automatic while the accuracy of the centerline detection remains practically unchanged.

8.4 Conclusions

We presented the first global method for extracting tree-like structures from 3D medical images while encoding geometrical tree priors. The global model-to-data fitting made
centerline tracing free from any initialization and the incorporation of priors made the method more robust to noise. Incorporating fixed topological priors for consistent branches is advantageous of this paper. In the existence of topological variability, e.g. pathology cases or generations deep down the tree; our method is not designed to handle these cases. Nevertheless, we note that the pictorial algorithm [60] is stable to occlusions, so even when the tree model has a fixed topology, it should still be able to locate actual trees with slight topological variations. Future work will involve integrating the minimal path optimization within the pictorial algorithm and encoding more elaborate branch statistics (e.g. medial curvature and radii). It is also interesting to explore automatic ways to detect pathological deviations from priors that are supported by image evidence, as these may indicate pathology. Since the human body contains many fine capillary structures, it’s worth analyzing whether the proposed method would work on locating the bifurcations in these structures as well.
Chapter 9

Retinal Image Classification via Vasculature-guided Sequential Attention

9.1 Introduction

9.1.1 Motivation

The prevalence of eye diseases has been on the rise during the past years, both globally and regionally. According to a recent Lancet publication [27], more than 216 million people suffer from moderate to severe visual impairment. The fact sheet from the US National Eye Institute shows around 1.3 million of Americans are blind and the figure is expected to rise to 2.2 million by 2030 [159]. Two of the leading causes of blindness are age-related macular degeneration (AMD) and diabetic retinopathy (DR) [159]. Early diagnosis and treatment of these diseases are crucial in vision preservation, which makes automatic and accurate classification of retinal images extremely important [185, 191].

9.1.2 Machine (and deep) Learning for Retinal Image Classification

Limited attempts to build automatic classification systems have been made using traditional machine learning methods that relied on hand-crafted features. In the work of Roychowdhury et al. [183], AdaBoost was used for feature reduction in a two-step hierarchical binary DR classification (with DR or without) approach albeit with low specificity (53%) according to their reported results. Wang et al. [221] proposed to combine multi-scale features and feature selection algorithms for AMD classification, but their work focused on optical coherence tomography images, and use a fairly small dataset with only 45 patients, while multiple images come from the patient at different scans.

The recent success of deep learning-based visual recognition for numerous applications has sparked renewed interest in addressing the task of retinal disease classification and grading from fundus images. In the work of Gulshan et al. [80], the Inception network
was used for retinopathy grading (5 levels). Pratt et al. [174] used a 13-layer convolutional neural network (CNN) for retinopathy grading (5 levels). In the work of Gargeya et al. [71], a deep network with 5 residual blocks was constructed for feature generation, then the output feature is input into a decision tree with other metadata information for binary retinopathy classification.

Previous fundus imaging-based deep learning methods for AMD and DB classification and grading relied on a purely data-driven tuning of network architectures yet lacked any disease-specific customization to encode existing prior knowledge, such as anatomical structural changes associated with the progression with of specific diseases. One form of AMD, the wet AMD has been known to be associated with abnormal growth of blood vessels in the eyes [40]. By examining retinal photography images, McGowan et al. [141] discovered correlations between AMD and, not only blood vessels around the macular area but also the blood vessel caliber across the whole fundus image. Also, a recent study by Jackson et al. [96] discovered a high prevalence of vascular abnormalities in conjunction with AMD. However, to the best of our knowledge, automatic deep learning AMD classification methods completely ignored any vascular priors. On the other hand, DR is caused by retinal blood vessel changes due to diabetes and, to a certain extent, could also be linked to vessel overgrowth on the retina [65,109]. This literature shows how the development of AMD and DB is highly correlated with changes in retinal blood vessel structures and suggests that automatic methods, deep learning or otherwise, could leverage such prior information.

9.1.3 Attention Mechanisms in Deep Learning

Several deep learning methods with attention mechanism have been proposed in the past few years. Wang et al. [218] proposed to add intermediate deconvolution layers to extract attention maps, combining them with the last layer for prediction. In the works of Mnih et al. [151] and Xu et al. [236], attention features were extracted from different locations within an image and stacked into a sequence that is fed into an LSTM framework. Similarly, to handle cancer classification from large histopathology images, BenTaieb and Hamarneh proposed an attention mechanism that adaptively selects only a limited sequence of image locations for further processing [24]. But none of these work leveraged any disease-specific prior information. For diagnosing melanoma, Yan et al. used skin lesion masks to guide attention maps across different layers of the VGG architecture [237], but their work relied on expert-delineated (not automatically-generated) prior masks, nor did they learn the patterns of a sequence of attention maps.

9.1.4 Contributions

Our work is the first:
1. To leverage anatomical knowledge (in the form of vascular priors) to guide the attention maps for retinal disease classification from fundus images;

2. To automatically extract the attention prior maps (rather than requiring manually-segmented images);

3. To encode the inter-dependency among attention features (deployed across the depth of the network), which we accomplish via a novel bi-directional, dual-layer LSTM.

We perform evaluation on two clinical datasets with cross-validation and an ablation study. The experimental results show that, by using the proposed vasculature priors and the LSTM attention formulation, the results are improved by as much as 8%.

9.2 Proposed Method

The proposed architecture is illustrated in Figure 9.1a. In the following we describe the baseline architecture (Section 9.2.1), the proposed vasculature priors (Section 9.2.2), LSTM module (Section 9.2.3) and the corresponding loss functions (Section 9.2.4).

With the proposed network, we expect to encode not only the shape prior for guiding the generation of attention maps, but also the inter-dependency among different layers of the attention features. The inter-dependency among the features comes from the fact that these features are generated by different layers of the same network. To explore this inter-dependency property, we propose to use a double-layer bi-directional LSTM network. With more layers of attention involved, the inter-dependency property leveraged by LSTM should impose a stronger effect over the final result, which will be shown in Section 9.3.

9.2.1 Baseline CNN with Attention Modules

We adopt the baseline CNN architecture proposed by Yan et al. [237], which extends VGG-16 [198] with two additional attention layers and one penultimate global feature vector (obtained via global average pooling). The attention features and global feature vector are combined and input into one dense layer for classification. We denote the intermediate features generated from \( n \) different intermediate convolutional layers as \( \{F^1, F^2, ..., F^n\} \), where \( F^k = \{f^k_1, f^k_2, ..., f^k_n\} \), and \( f^k_i \) is the feature of channel \( i \). We further denote the global feature from the last convolutional layer as \( G \), the layer-wise attention maps as \( \{M^1_{attn}, M^2_{attn}, ..., M^n_{attn}\} \), and the corresponding attention feature vectors as \( \{A^1, A^2, ..., A^n\} \).

9.2.2 Vasculature Priors on Attention

A prior image \( M_{\text{prior}} \) is a binary image mask used to guide the attention features. \( M_{\text{prior}} \) can be generated using an automatic method or manual delineation. In this work, our goal
is to use the vasculature structure to guide the attention feature maps. As expert manual delineation is time-consuming, we set $M_{\text{prior}}$ to be the retinal vasculature mask extracted from the input image using the automatic B-Cosfire vessel filtering [12], which is based on calculating the geometric mean of multiple difference of Gaussian filters. Examples of the generated vessel masks could be found in Figure 9.2.

To guide attention feature maps across different scales, $M_{\text{prior}}$ is processed with adaptive average pooling into different sizes as $\{M_{\text{prior}}^{1}, M_{\text{prior}}^{2}, \ldots, M_{\text{prior}}^{n}\}$ (see details in Yan et
al. [237]), corresponding to the sizes of the attention maps \( \{M_{\text{attn}}^1, M_{\text{attn}}^2, ..., M_{\text{attn}}^n\} \). \( M_{\text{attn}}^i \) is guided by \( M_{\text{prior}}^i \), for \( i = 1 \cdots n \), by maximizing the similarity between the two via a minimizing Dice-based loss (defined in Section 9.2.4).

The attention feature vector \( A^k \) is calculated as follows:

\[
\begin{align*}
\mathcal{G} &= \text{Upsample}(W_g \otimes G) \\
\mathcal{F}^k &= W_f \otimes F^k \\
M_{\text{attn}}^k &= \sigma(W \otimes \text{ReLU}(\mathcal{F}^k + \mathcal{G})) \\
a_i^k &= M_{\text{attn}}^k \odot f_i^k \\
A^k &= \text{GlobalAveragePool}\{a_1^k, a_2^k, ..., a_n^k\}
\end{align*}
\]

where \( W_g, W_f, W \) are convolutional filter weights, \( \sigma \) is the sigmoid function, and \( \otimes \) and \( \odot \) are the convolution and element-wise multiplication operators, respectively. In Equation 9.1, global feature is convolved by a 256 channel filter and upsampled to match the size of \( F^k \). In Equation 9.2, intermediate feature \( F^k \) is convolved with a 256 channel filter. In Equation 9.3, the attention map is generated using \( \mathcal{G} \) and \( \mathcal{F}^k \) from the previous two
steps. Then in Equation 9.4, the attention map is multiplied with intermediate feature $f^k_i$ at channel $i$ on an element-wise way, to preserve the intermediate information. Then the attention vector $A^k$ is obtained by global average pooling of $\{a^k_1, a^k_2, ..., a^k_n\}$.

### 9.2.3 Learning the Sequence of Prior-guided Attention Maps

In the work of Yan et al. [237], $A^1, A^2, G$ were aligned into a single vector before being input into a dense layer for classification. In contrast, we wish to encode the inter-dependency among all the $n$ learnt attention feature vectors across all layers, $A$, and the global feature vector $G$, using a bi-directional dual-layer LSTM model.

In a LSTM module, given inputs $x_t, h_{t-1}, c_{t-1}$ at time step $t$, $W_{xi}, W_{hi}, W_{xf}, W_{hf}, W_{xo}, W_{ho}, W_{xc}, W_{hc}$ the learnt weights, and $b_i, b_f, b_o, b_c$ the corresponding biases, the following update scheme is used:

\[
i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + b_i) \quad (9.6)
\]
\[
f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + b_f) \quad (9.7)
\]
\[
o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + b_o) \quad (9.8)
\]
\[
g_t = \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c) \quad (9.9)
\]
\[
c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (9.10)
\]
\[
h_t = o_t \odot \tanh(c_t). \quad (9.11)
\]

To this end, $x_1, x_2, ..., x_t$ in Equation 9.6-9.9 are replaced by $A^1, A^2, ..., G$. To deal with different sizes of $A^k$, we tile $A^k$ until they are of the same size as $G$. The output of the LSTM is then fed into the classification layer (Figure 9.1a). Illustration of inputs and output of LSTM could be found in Figure 9.1a)

### 9.2.4 Loss Functions

Similar to Yan et al. [237], we first use a modified cross entropy loss, called focal loss [125] with $\gamma = 2.0$, to deal with class imbalance:

\[
L_F = -(1 - p_i)^\gamma \log(p_i) \quad (9.12)
\]

where $p_i$ is the estimated probability for the class with label $i$. In a $N$-class classification problem, $p_i$ has to satisfy $\sum_{i=1}^{N} p_i = 1$.

As described in Section 9.2.2, we wish to use vessel priors to guide the learning of attention layers (Figure 9.1b). To this end, we define the following prior loss, based on the Dice similarity coefficient:

\[
L'DSC = 1 - 2 \frac{|M^i_{attn} \cap M^i_{prior}|}{|M^i_{attn}| + |M^i_{prior}|} \quad (9.13)
\]
where $M^i_{attn}$ is the intermediate feature at layer $i$ and $M^i_{prior}$ the corresponding prior mask.

The final loss is the sum of the focal loss and the weighted prior loss, for the total of $n$ attention maps:

$$L = L_F + \sum_{i=1}^{n} w_i \cdot L^{i}_{DSC}. \quad (9.14)$$

### 9.2.5 Implementation Details

For the proposed architecture, we use Adam optimizer with $\beta = (0.9, 0.999)$ and set the initial learning rate to $10^{-4}$, weight decay ratio $\epsilon = 10^{-8}$, total epoch = 20, and batch size = 20. The weight loss $w_i$ in (9.14) is set empirically to 0.1 for the AMD dataset and 0.001 for the retinopathy dataset. Finally, the parameters for B-Cosfire filter are summarized in Table 9.1.

<table>
<thead>
<tr>
<th></th>
<th>$\sigma$</th>
<th>$\rho$</th>
<th>$\sigma_0$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sym</td>
<td>asym</td>
<td>sym</td>
<td>asym</td>
</tr>
<tr>
<td>Retinopathy</td>
<td>5.0</td>
<td>5.0</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>AMD</td>
<td>5.0</td>
<td>5.0</td>
<td>20</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 9.1: Parameters for B-Cosfire vessel filter. sym: symmetric filter parameters; asym: asymmetric parameters. See corresponding parameter meaning in the the original paper [12].

### 9.3 Experiment

We test the proposed method on two public datasets (iChallenge-AMD [3] and IDRiD [173]) and evaluate the performance of competing methods using Accuracy, Precision, Recall and F1-score. We also perform ablation studies assessing the value of the vessel priors and the LSTM formulation.

#### 9.3.1 Datasets

**iChallenge-AMD Dataset**

We obtained training data from iChallenge-AMD, a recent AMD classification challenge, which contains 398 images: 87 AMD images and 311 non-AMD images, for the purpose of binary classification. All images are color fundus images of resolution $2124 \times 2056$. We performed 3-fold cross validation and augmented the training set 2 times by random cropping, scaling and rotation.
IDRiD Dataset

We obtained 516 images from this dataset: 413 training and 103 testing data, for the purpose of retinopathy grading. International 5-level diabetic retiopathy (DR) grading is provided: (i) no apparent retinopathy, (ii) mild non-proliferative DR, (iii) moderate non-proliferative DR, (iv) severe non-proliferative DR and (v) proliferative DR. All images are color fundus images of resolution 4288 * 2848.

Dealing with the Class Imbalance Problem

For the AMD dataset, since 311 non-AMD images and 87 AMD images were provided as the training set, we oversample AMD image by 3 times to avoid class imbalance. For retinopathy dataset, we oversample class (ii) seven times, class (iv) two times and class (v) three times.

9.3.2 Results

Baseline Experiments

We carry out baseline experiments using the architecture in the work of Yan et al. [237] without prior information or LSTM, and only with two layers (layer 3 and 4) of attention features.

Assessing the advantage of using the vessel priors

To test the hypothesis that adding vessel priors would help guiding the attention map, we compare the baseline architecture with and without vessel priors.

From Table 9.2 we see that adding vessel priors to the baseline architecture improves the prediction results by as much as 2%.

Comparing rows 1 – 2, 4 – 5, 7 – 8 and 10 – 11 in Table 9.3 shows that simply adding the vessel priors to the baseline architecture improves the results of almost all evaluation metrics by as much as 6%. The improvement was observed regardless of the number of layers equipped with an attention module.

Assessing the Advantage of Using the LSTM

Here we set out to evaluate whether the proposed LSTM module could leverage the inter-dependency of the attention sequence.

From Table 9.2 we see that by adding the LSTM module (proposed), the precision for predicting AMD is improved by as much as 6%.

Comparing rows 2 – 3 from Table 9.3 shows that, when the attention is limited to only one layer (layer 4), LSTM is not effective since there is almost no sequential information in the training data.
Furthermore, comparing rows 2 – 3, 5 – 6, 8 – 9 and 11 – 12 in Table 9.3 shows that, when the length of the attention sequence is no less than 2 (attention from more than 2 layers), by incorporating LSTM, the results improve by at least 3% for all metrics and as much as 6%. In this sense, we can further claim that leveraging the LSTM on a sequence, even with limited length, is useful in exploring the inter-dependency among attentions from different layers.

**Overall performance of the proposed method**

By combining the vessel priors and the LSTM module for learning inter-dependency from the attention sequence, Table 9.2 and Table 9.3 (comparing rows 1 – 3, 4 – 6, 7 – 9 and 10 – 12) both show the results are improved by as much as 8%.

**Comparing with the state-of-the-art**

Comparing the results of our proposed method to state-of-the-art AMD classification methods [33, 170], and to the DR grading accuracy results reported on the challenge, our proposed method achieves comparable accuracy. However, none of the state-of-the-art methods leverage any attention mechanism, so we expect that their performance will improve with vessel-guided priors and attention sequence modelling, similarly to how our baseline methods improved with these extensions.

Furthermore, as shown by Yiqi et al. [237], including priors can help rendering the regions relevant for classification, thus contributing to a more intuitive and interpretable deep learning model.

**9.4 Conclusions and Discussions**

In this paper we propose a new architecture using vessel priors to guide the attention sequence in deep learning networks. To leverage the inter-dependency among the attention sequence, a bi-directional dual-layer LSTM module is used. Experiments using two clinical dataset, with binary AMD classification and 5-level retinopathy grading tasks, clearly demonstrate the advantages of the proposed architecture. Moreover, our ablation study with both datasets show how the proposed architecture works with varying lengths of the attention sequence, which could be easily extended when there is an even deeper network involved. Numerical results with multiple evaluation metrics are reported and the performance improvement produced by the proposed architecture reaches as much as 8%.

In the future study, we can explore this LSTM-leveraged mechanism on different types of CNN models, as well as on possible negative cases, where adding the prior maps might be generating a negative impact on the final result. It’s worthwhile to explore the relationship between the morphology of the vessel prior and the performance improvement, e.g., to
distinguish whether the whole vessel region contributes toward the final result, or only a certain part of the vessel region is involved. Also it’s worth to study how the quality of automatically extracted vessel masks impact the performance improvement, e.g., if we can obtain vessel segmentation annotations from some experts, would a more accurate vessel extraction scheme return better results? For multi-class classification problems like retinopathy grading, it’s still unknown why the performance is much lower than binary classification task such as AMD classification. One possibility is, the provided retinopathy dataset is highly unbalanced and one future direction could be exploring how to deal with this high unbalance.

### Methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Vessel Priors</th>
<th>LSTM</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>AMD</td>
<td>NAMD</td>
<td></td>
</tr>
<tr>
<td>[237]</td>
<td>x</td>
<td>x</td>
<td>94 ± 2%</td>
<td>89 ± 6%</td>
<td><strong>95 ± 2%</strong></td>
<td>81 ± 7%</td>
</tr>
<tr>
<td>[237] + vessel</td>
<td>✓</td>
<td>x</td>
<td>94 ± 2%</td>
<td>91 ± 4%</td>
<td><strong>95 ± 2%</strong></td>
<td>82 ± 5%</td>
</tr>
<tr>
<td>Proposed</td>
<td>✓</td>
<td>✓</td>
<td><strong>95 ± 3%</strong></td>
<td><strong>97 ± 5%</strong></td>
<td><strong>95 ± 2%</strong></td>
<td>82 ± 12%</td>
</tr>
</tbody>
</table>

Table 9.2: Testing result on AMD dataset with binary classification. AMD: images labeled as age-related macular degeneration; NAMD: without disease. Attention comes from layer 3 – 4 of the architecture (see details of the baseline architecture in [237]). Mean ± standard deviation are reported among the 3 groups for cross-validation.

<table>
<thead>
<tr>
<th>Row</th>
<th>Methods</th>
<th>Attention</th>
<th>Vessel Priors</th>
<th>LSTM</th>
<th>Accuracy</th>
<th>Mean Precision</th>
<th>Mean Recall</th>
<th>Mean F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[237]</td>
<td></td>
<td>x</td>
<td>x</td>
<td>56%</td>
<td>54%</td>
<td>56%</td>
<td>55%</td>
</tr>
<tr>
<td>2</td>
<td>[237] + vessel</td>
<td>layer 4</td>
<td>✓</td>
<td>x</td>
<td>59%</td>
<td>59%</td>
<td>59%</td>
<td>57%</td>
</tr>
<tr>
<td>3</td>
<td><strong>Proposed</strong></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td><strong>59%</strong></td>
<td><strong>58%</strong></td>
<td><strong>59%</strong></td>
<td><strong>58%</strong></td>
</tr>
<tr>
<td>4</td>
<td>[237]</td>
<td></td>
<td>x</td>
<td>x</td>
<td>60%</td>
<td>57%</td>
<td>60%</td>
<td>58%</td>
</tr>
<tr>
<td>5</td>
<td>[237] + vessel</td>
<td>layers 3-4</td>
<td>✓</td>
<td>x</td>
<td>62%</td>
<td>60%</td>
<td>62%</td>
<td>60%</td>
</tr>
<tr>
<td>6</td>
<td><strong>Proposed</strong></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td><strong>68%</strong></td>
<td><strong>63%</strong></td>
<td><strong>68%</strong></td>
<td><strong>65%</strong></td>
</tr>
<tr>
<td>7</td>
<td>[237]</td>
<td></td>
<td>x</td>
<td>x</td>
<td>65%</td>
<td>65%</td>
<td>65%</td>
<td>63%</td>
</tr>
<tr>
<td>8</td>
<td>[237] + vessel</td>
<td>layers 2-4</td>
<td>✓</td>
<td>x</td>
<td>65%</td>
<td>63%</td>
<td>65%</td>
<td>63%</td>
</tr>
<tr>
<td>9</td>
<td><strong>Proposed</strong></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td><strong>68%</strong></td>
<td><strong>67%</strong></td>
<td><strong>68%</strong></td>
<td><strong>67%</strong></td>
</tr>
<tr>
<td>10</td>
<td>[237]</td>
<td></td>
<td>x</td>
<td>x</td>
<td>61%</td>
<td>59%</td>
<td>61%</td>
<td>60%</td>
</tr>
<tr>
<td>11</td>
<td>[237] + vessel</td>
<td>layers 1-4</td>
<td>✓</td>
<td>x</td>
<td>64%</td>
<td>61%</td>
<td>64%</td>
<td>61%</td>
</tr>
<tr>
<td>12</td>
<td><strong>Proposed</strong></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td><strong>67%</strong></td>
<td><strong>67%</strong></td>
<td><strong>67%</strong></td>
<td><strong>65%</strong></td>
</tr>
</tbody>
</table>

Table 9.3: Testing result on retinopathy dataset with 5 level grading.
Chapter 10

Conclusion and Discussion

10.1 Thesis Summary

Tubular structures in medical images, especially 3D images, are hard to extract due to their elongated shapes and complex tree structures, the varying intensity within the tree and adjacent structures, and their sparse presence in the whole image volume. To improve tree extraction as well as disease classification accuracy, we proposed in this thesis a framework for incorporating anatomical priors (Figure 1.9). Specifically, we formulated the tree-based anatomical priors in a tree tracking framework from the following three perspectives:

1. Tree knowledge formulated as hand-crafted features and statistics;
2. Tree structures formulated as parametric shapes and pictorial structures;
3. Tree priors formulated in deep learning models, such as multi-loss functions built on the Frenet frame, higher-order Markovian process optimized using LSTM, and tree masks for attention-guidance in deep networks.

In Chapter 3, we proposed a machine learning method and novel hand-crafted features for bifurcation detection. The novel features were based on multi-modal spherical distributions and eigenvectors of the Hessian in 3D space, which provided the key novelties of the chapter. In Chapter 4, we constructed a parametric bifurcation shape and converted the bifurcation detection problem into a shape model fitting problem. Furthermore, we proposed to use the genetic algorithm with a multi-objective fitness functions, as well as a tribes niching technique, for searching multiple solutions to the model fitting problem simultaneously. In Chapter 5, we proposed the first deep learning model for predicting tree branch directions, using a multi-loss function defined over the Frenet frames that was naturally formed by the tubular directions with corresponding cross-sectional planes. In Chapter 6, we improved on the general myopic tree trackers, by introducing a sequence-based higher-order Markovian using an LSTM. In Chapter 7, we integrated length statistics into a Bayesian prediction and angular statistics into daughter branch searches, which
further showed effectiveness in increasing whole tree tracking stability. In Chapter 8, we developed a tree detection algorithm, which leveraged not only unary probability map generated by a deep learning network, but also the binary term generated using tree statistics from the whole dataset. Both the unary map and the binary term were fitted into a pictorial structure with global optimization. Finally, in Chapter 9, we focused on how to encode the anatomical tree into disease classification, by using automatically-generated vessel tree masks as priors guiding attention in deep learning networks. Moreover, we rephrased the attention from multiple layers from the deep network as a sequence and showed the superiority of using LSTMs to explore the inter-dependency among attention features.

10.2 Discussion and Open Questions

Following the contributions in this thesis, we give a list of different directions and related questions to be further explored in future research.

**Extensiveness vs. Expensiveness.** The multi-loss function in Chapter 5, the multi-objective fitness function in Chapter 4, and the design of manual features in Chapter 3 shared the similarity of using multiple terms where each term described a single aspect of the desired outcome such as prediction accuracy. To further improve these outcomes, one possible future research direction is to extend the functions and features by adding new terms so to encode more image and geometrical information – we denote this type of approach as “extensive” approaches. Similar extensiveness could be applied to the deep learning architecture by adding more layers, or applied to the genetic algorithm by using more tribes or populations. Several dimensions of extensiveness lead to a non-linear increase in computational cost without clear convergence guarantees. An uninformed pursuit of extensiveness can increase the risk for overfitting or dataset memorization at the cost of generalization [48, 73]. In this thesis, the balance between extensiveness and expensiveness was only achieved by empirical efforts. A well-versed strategy of balancing between the two should be explored in the future.

**How to Further Extract Statistics?** Although branch-wise length statistics, as well as branch-by-branch angular statistics have been encoded explicitly, pictorial structure in this thesis, how to incorporate the statistics in the tree space/manifold [7]. Understanding tree statistics and tree space is another field to be explored. Tree manifold information, especially moving from binary-branching tree topology to generic tree topology is a challenging task, which requires careful design of manifold metric so to address geodesic calculation as well as gradient computation.
Reconsider Accuracy. As we mentioned in Section 1.1, the motivation for accurate and automatic delineation of vascular tree structures in medical images is to facilitate the disease diagnosis procedure of clinicians. However, an endless pursuit of delineation accuracy indicates an underlying belief – a more accurate delineation could either facilitate understanding/visualization of the diseased area or produce more accurate analysis numbers (e.g., airway volumes), which eventually leads to better disease diagnosis. This underlying belief should be further explored. Specifically, the exploration of whether there exists a positive correlation between segmentation accuracy and disease prediction would be valuable. Furthermore, the evaluation of segmentation accuracy using different metrics (see Section 2.5.1) would be desired.

Further Discussion on Differential Geometry and Deep Network. We proposed a unique multi-loss function in Chapter 5, which was built on the Frenet frame of reference. By using the Frenet frame, we did observe an improvement on the overall performance, but we didn't discuss its further implications. Since a set of geometrically meaningful parameters like the curvature the torsion could be defined on the Frenet frame, a natural question is whether a deep network would be able to learn these parameters. If the deep network could effectively learn these parameters, the next question would be whether a model making predictions on tubular branches benefit from the learning.

To Improve on Deep Learning Losses. Since the loss function is an important component of a deep network, further improvement on the performance of a deep network involves the improvement on loss functions. Here we give a list of ways to further improve the loss functions: i) using convex losses; ii) using regularization terms in losses; iii) exploring the parameter space; and iv) using spatial relationships in losses. First, as using convex functionals has been proved as a successful strategy in model-driven segmentation methods [8], we can ask a similar question whether using a convex loss function in a deep network would further improve its performance, e.g., whether using a convexified Mumford-Shah functional would improve over a non-convex one. Second, as using regularization terms has been used for encoding prior knowledge in model-based methods [162], we also ask how could similar regularization terms be added to the loss functions and how much they would help. Third, as the relationship between image manifold and parameter space has been explored [143], we would ask similar questions on exploring the relationship between the parameter space involved in deep models and image manifold, by revisiting the loss functions. Last, as topological relationship could be encoded within functionals of model-driven segmentation methods [160], we would further ask how can we encode similar topology information in the loss function of a deep network, for the purpose of tree structure extraction.
Reflection on the Data-driven Era. The success of deep learning/ data-driven methods has already made a huge impact in the field of medical image analysis. However, there exists a gap between the deep models and classical model driven methods such as levelset, graph cut, region growing, etc. Although different approaches have been adapted (see Chapter 2) to combine multiple methods together, there seems to be a lack of a unified framework. More specifically speaking, the expected unified framework should combine the advantages of both classical model-driven methods (e.g., data free, can embed shape prior) and data-driven models (can accommodate outliers) together, towards a formal hybrid framework of model and data-driven paradigms. Furthermore, data-driven methods not only could be combined with model-driven methods, but could also be combined with other data-driven methods as well, e.g., by employing neural architecture search (NAS) using deep models [254].

Unsupervised/semi-supervised methods. From the list given in Table 2.6, we can see the question that most of the dataset contained no or only highly sparse annotations, while some of the annotations that were provided referred to the pathology instead of the anatomy, e.g., lung images containing only nodule annotation, not airways or vasculatures. With these sparse or non-anatomical based annotations given, a first relevant question is whether we can leverage partial information. To be specific, we are asking the following sub-questions: i) how much would partial annotation, such as sparse points annotated along certain vessel branches, provide guidance toward the extraction of the whole vessel tree? ii) for non-anatomical structures like the lung nodules, since there exists certain spatial relationship between the nodules and blood vessels [70], we would ask how much can these spatial relationship knowledge contributes into the vessel analysis procedure?

The second question is, since tree structures themselves bear certain information, whether it is possible to solely learn from tree structures in case the annotations do not exist at all. The third question is, if we are able to collect annotations from other sources, e.g., serious gaming and crowdsourcing [93], how could we modulate these annotations which usually come with high noise [150].

On Generating Realistic Synthetic Dataset Beside obtaining annotations from crowdsourcing, another solution for annotation insufficiency is to create synthetic data with realistic attributes. Based on existing synthesizing software and synthetic dataset [82, 97], a relevant question is, how to improve on the software so to make the generated images more realistic looking. Another question is, how to leverage the deep models for generating realistic looking images.
Bibliography


