

DISSERTATION

SUBMITTED

TO THE

COMBINED FACULTY FOR THE NATURAL SCIENCES AND MATHEMATICS

OF

HEIDELBERG UNIVERSITY, GERMANY

FOR THE DEGREE OF

DOCTOR OF NATURAL SCIENCES

Put forward by

Diplom-Informatiker Wei Liao

Born in Hubei, China

Oral examination: _____

**Minimal Path Methods
for Segmentation and Analysis
of 2D and 3D Line Structures**

Advisor: PD Dr. Karl Rohr

Dedicated to my beloved parents and family.

Abstract

Image segmentation plays a vital role in many applications of computer vision. Segmentation is not only an important task in its own right, but also a prerequisite for many further image analysis steps. Consequently, segmentation is one of the most active research areas of computer vision. In this thesis, line structures are considered, which have quite different characteristics compared to common objects in natural 2D images: Line structures are much thinner and longer, and often they have little color or texture information such as blood vessels in medical images. To cope with these challenges, minimal path methods are commonly used. In this thesis, two new methods are introduced which are extensions of existing minimal path methods.

The first method is a novel hybrid approach for automatic 3D segmentation and quantification of high-resolution 7 Tesla magnetic resonance angiography (MRA) images of the human cerebral vasculature. Our approach consists of two main steps. First, a 3D model-based approach is used to segment and quantify thick vessels and most parts of thin vessels. Second, remaining vessel gaps of the first step in low-contrast and noisy regions are completed using a 3D minimal path approach, which exploits directional information. We present two novel minimal path approaches: The first is an explicit approach based on energy minimization using probabilistic sampling, and the second is an implicit approach based on fast marching with anisotropic directional prior.

The second method we introduce is a novel minimal path method for the segmentation of 2D and 3D line structures. Minimal path methods perform propagation of a wavefront emanating from a start point at a speed derived from image features, followed by path extraction using backtracing. Usually, the computation of the speed and the propagation of the wave are two separate steps, and point features are used to compute a static speed. We introduce a new continuous minimal path method which steers the wave propagation progressively using dynamic speed based on path features. We present three instances of our method, using an appearance feature of the path, a geometric feature based on the curvature of

the path, and a joint appearance and geometric feature based on the tangent of the wavefront. Such features have not been used in previous continuous minimal path methods. We compute the features dynamically during the wave propagation, and also efficiently using a fast numerical scheme and a low-dimensional parameter space. Our method does not suffer from discretization or metrication errors.

We conducted quantitative and qualitative experimental evaluations of our methods using 2D and 3D images from different application areas, including synthetic images, retinal images, satellite images of streets, rivers, and bridges, and 3D 7T MRA images of human brain vessels.

Zusammenfassung

Bildsegmentierung spielt eine zentrale Rolle in vielen Anwendungen der Bildverarbeitung. Segmentierung ist nicht nur eine wichtige Aufgabe an sich, sondern auch eine Voraussetzung für viele weitere Schritte der Bildanalyse. Folglich ist die Bildsegmentierung einer der aktivsten Forschungsbereiche in der Bildverarbeitung. In dieser Dissertation werden Linienstrukturen betrachtet, die sehr unterschiedliche Eigenschaften im Vergleich zu gewöhnlichen Objekten in 2D natürlichen Bildern besitzen: Linienstrukturen sind viel dünner und länger, und sie haben oft kaum Farb- oder Texturinformationen wie beispielsweise Blutgefäße in medizinischen Bildern. Um mit diesen Herausforderungen zurechtzukommen, werden häufig Methoden der kürzesten Pfade benutzt. In dieser Arbeit werden zwei neue Methoden vorgestellt, die Erweiterungen bestehender Methoden der kürzesten Pfade sind.

Die erste Methode ist eine neue hybride Methode für automatische 3D Segmentierung und Quantifizierung von hochauflösenden 7 Tesla Bildern der Magnetresonanzangiographie (MRA) der menschlichen Gehirngefäße. Unsere Methode besteht aus zwei Hauptschritten. Im ersten Schritt wird eine 3D modellbasierte Methode angewendet, um dicke Gefäße und die meisten Teile der dünnen Gefäße zu segmentieren und zu quantifizieren. Im zweiten Schritt werden die Lücken zwischen den Gefäßen aus dem ersten Schritt in kontrastarmen und verrauschten Bildregionen durch eine 3D Methode der kürzesten Pfade vervollständigt, die Richtungsinformationen benutzt. Wir präsentieren zwei neue Verfahren, die auf Methoden der kürzesten Pfade basieren: Das erste Verfahren ist eine explizite Methode und basiert auf Energieminimierung mit probabilistischem Sampling. Das zweite Verfahren ist eine implizite Methode und basiert auf der Fast-Marching-Methode mit anisotropischer A-priori-Richtungsinformation.

Die zweite Methode, die wir vorstellen, ist eine neue Methode der kürzesten Pfade für die Segmentierung von 2D und 3D Linienstrukturen. Methoden der kürzesten Pfade benutzen die Ausbreitung einer Wellenfront, die von einem Startpunkt mit einer Geschwindigkeit ausgeht, die durch Bildmerkmale bestimmt ist.

Anschließend wird der Pfad durch Zurückverfolgung extrahiert. Häufig sind die Berechnung der Geschwindigkeit und die Ausbereitung der Wellenfront zwei getrennte Schritte. Dabei werden punktbasierte Merkmale benutzt, um eine statische Geschwindigkeit zu berechnen. Wir stellen eine neue kontinuierliche Methode der kürzesten Pfade vor, die die Ausbereitung der Wellenfront schrittweise steuert, und zwar mit einer dynamischen Geschwindigkeit, die pfadbasierte Merkmale benutzt. Wir stellen drei Instanzen unserer Methode vor, die ein Merkmal des Erscheinungsbildes des Pfades, ein geometrisches Merkmal basierend auf der Krümmung des Pfades, sowie ein Verbundmerkmal aus Erscheinungsbild und Geometrie der Tangente der Wellenfront benutzen. Solche Merkmale wurden in früheren kontinuierlichen Methoden der kürzesten Pfade nicht benutzt. Die Berechnung der Merkmale erfolgt dynamisch während der Ausbereitung der Wellenfront. Außerdem ist diese Berechnung effizient, indem eine schnelle numerische Methode und ein niedrigdimensionaler Parameterraum verwendet werden. Unsere Methode wird nicht durch Diskretisierungs- und Metrisierungsfehlern negativ beeinflusst.

Wir haben quantitative und qualitative experimentelle Auswertung unserer Methoden mit 2D und 3D Bildern durchgeführt, einschließlich synthetischer Bilder, Retinabilder, Satellitenbilder von Straßen, Flüssen, und Brücken, sowie 3D 7T MRA Bilder der menschlichen Gehirngefäße.

Acknowledgement

First of all, I want to thank my advisor PD Dr. Karl Rohr for his constant guidance, time, and support during the last years.

In addition, I also thank PD Dr. Stefan Wörz for his support.

I also want to thank my cooperation partners. I am thankful to Prof. Dr. Zang-Hee Cho and Dr. Chang-Ki Kang from the Neuroscience Research Institute at the Gachon University, Incheon, Korea, for providing medical images and for their hospitality during our three visits in Korea. I want to thank Dr. Hendrik von Tengg-Koblogk and Dr. Philipp Hoegen from Heidelberg University, German Cancer Research Center (DKFZ) Heidelberg, and Inselspital Bern, for providing medical images and for discussions.

I thank my colleagues from the Biomedical Computer Vision Group (BMCV) at the BioQuant and IPMB and from other groups. This includes my first office mates in Heidelberg: Dr. Nathalie Harder, Dr. William Godinez; my office mates after moving to another office in TP3: Marco Tektonidis, Vasil Tsimashchuk, Dr. Astha Jaiswal, Jing Yang, Stephen Krämer; my office mates after returning to the office in BioQuant: Simon Eck, Dr. Clarissa Liesche, Artjom Zern, Dr. Sharib Ali, Qi Gao, and other colleagues: Dr. Andreas Biesdorf, Jan-Philip Bergeest, Yu Qiang, Thomas Wollmann, and Christian Ritter. I also wish to thank all other people from the division Bioinformatics at Heidelberg University and DKFZ Heidelberg, especially Prof. Dr. Roland Eils, Manuela Schäfer, and Corinna Sprengart.

Moreover, I would like to thank my colleagues Dr. Jun Zhou, Dr. Lizhen Liu, Chen Chen, Dr. Huangshan Chen, Dr. Clarissa Liesche, and Dr. Yongsheng Cheng, who accompanied me when running long distances after work.

Also, I thank my friends Fengjun Xu, Tingyu Liu, Yuanxue Gao, Jing Yang, Dr. Huangshan Chen, Dr. Jun Zhou, Dr. Cheng Yang, and Dr. Shun Lu for helping me endure the most difficult times and for creating so many joyful moments.

I thank the Faculty of Mathematics and Computer Science at Heidelberg University for accepting me as PhD student. I also thank Prof. Dr. Christoph Schnörr for agreeing to be a member of the examination committee.

Finally, I want to express my deepest gratitude to my parents and family.

The work in this thesis was supported by the Deutsche Forschungsgemeinschaft (DFG), project QuantVessel (RO 2471/6).

Contributions

Publications

The methods and results in this thesis were published in peer-reviewed international journals and at international conferences.

Peer-Reviewed Journal Articles

- [1] **W. Liao**, S. Wörz, C.-K. Kang, Z.-H. Cho, and K. Rohr, “Progressive Minimal Path Method for Segmentation of 2D and 3D Line Structures,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, no. 3, pp. 696–709, 2018.
- [2] P. Hoegen, S. Wörz, M. Müller-Eschner, P. Geisbüsch, **W. Liao**, K. Rohr, M. Schmitt, F. Rengier, H.-U. Kauczor, H. von Tengg-Kobligk, “How Precise Are Preinterventional Measurements Using Centerline Analysis Applications? Objective Ground Truth Evaluation Reveals Software-Specific Centerline Characteristics,” *Journal of Endovascular Therapy*, vol. 24, no. 4, pp. 584–594, 2017.
- [3] **W. Liao**, K. Rohr, C.-K. Kang, Z.-H. Cho, and S. Wörz, “Automatic 3D Segmentation and Quantification of Lenticulostriate Arteries from High-Resolution 7 Tesla MRA Images,” *IEEE Transactions on Image Processing (TIP)*, vol. 25, no. 1, pp. 400–413, 2016.
- [4] S. Seo, C.-K. Kang, S. Kim, D. Yoon, **W. Liao**, S. Wörz, K. Rohr, Y.-B. Kim, D. Na, and Z.-H. Cho, “Measurements of Lenticulostriate Arteries Using 7T MRI: New Imaging Markers for Subcortical Vascular Dementia,” *Journal of the Neurological Sciences*, vol. 322, pp. 200–205, 2012.
- [5] C.-K. Kang, S. Wörz, **W. Liao**, C.-A. Park, Y.-B. Kim, C.-W. Park, Y.-B. Lee, K. Rohr, and Z.-H. Cho, “Three Dimensional Model-Based Analysis

of the Lenticulostriate Arteries and Identification of the Vessels Correlated to the Infarct Area: Preliminary Results,” *International Journal of Stroke*, vol. 7, pp. 558–563, 2012.

Peer-Reviewed Conference Articles

- [1] S. Wörz, P. Hoegen, **W. Liao**, M. Müller-Eschner, H.-U. Kauczor, H. von Tengg-Kobligk., and K. Rohr, “Framework for Quantitative Evaluation of 3D Vessel Segmentation Approaches using Vascular Phantoms in Conjunction with 3D Landmark Localization and Registration,” in *Proc. SPIE Medical Imaging 2016: Image Processing (MI’16)*, vol. 9714. San Diego, CA/USA: SPIE Bellingham, WA/USA, Feb.-Mar. 2016.
- [2] **W. Liao**, K. Rohr, and S. Wörz, “Globally Optimal Curvature-Regularized Fast Marching for Vessel Segmentation,” in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’13)*, ser. Lecture Notes in Computer Science, vol. 8149. Nagoya, Japan: Springer Berlin Heidelberg, Sept. 2013, pp. 550–557.
- [3] **W. Liao**, S. Wörz, and K. Rohr, “Globally Minimal Path Method Using Dynamic Speed Functions Based on Progressive Wave Propagation,” in *Proc. Asian Conference on Computer Vision (ACCV’12)*, ser. Lecture Notes in Computer Science, vol. 7725. Dajeon, Korea: Springer Berlin Heidelberg, Nov. 2012, pp. 25–37.
- [4] **W. Liao**, K. Rohr, C.-K. Kang, Z.-H. Cho, and S. Wörz, “Automatic Human Brain Vessel Segmentation from 3D 7 Tesla MRA Images using Fast Marching with Anisotropic Directional Prior,” in *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI’12)*. Barcelona, Spain: IEEE Computer Society, 5 2012, pp. 1140–1143.
- [5] **W. Liao**, S. Wörz, and K. Rohr, “Vessel Segmentation using an Iterative Fast Marching Approach with Directional Prior,” in *Proc. SPIE Medical Imaging 2012: Image Processing (MI’12)*. San Diego, CA/USA: SPIE Bellingham, WA/USA, Feb. 2012.
- [6] **W. Liao**, K. Rohr, C.-K. Kang, Z.-H. Cho, and S. Wörz, “A Generative MRF Approach for Automatic 3D Segmentation of Cerebral Vasculature from 7 Tesla MRA Images,” in *Proc. IEEE International Symposium on Biomedical*

Imaging: From Nano to Macro (ISBI'11). Chicago, IL/USA: IEEE Computer Society, 3–4 2011, pp. 2041–2043.

Poster

W. Liao, K. Rohr, C.-K. Kang, Z.-H. Cho, and S. Wörz, “A Probabilistic Minimal Path Technique for Automatic 3D Segmentation of Cerebral Vasculature from 7 Tesla MRA Images,” *Poster Competition of the Summer School of Advanced Methods in Biomedical Image Analysis*, Erasmus Program, Masaryk University, Czech Republic, August 2011.

Awards

Two prizes have been awarded for the work in this thesis.

Student Travel Award of MICCAI (2013)

This prize was awarded for the paper *Globally Optimal Curvature-Regularized Fast Marching for Vessel Segmentation*, since it was among the best 30 out of 131 student papers at the 16th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'13) in Nagoya, Japan, September, 2013.

Best Poster Award (2011)

This prize was awarded for the poster *A Probabilistic Minimal Path Technique for Automatic 3D Segmentation of Cerebral Vasculature from 7 Tesla MRA Images* at the Poster Competition of the Summer School of Advanced Methods in Biomedical Image Analysis, Erasmus Program, Masaryk University, Czech Republic, August 2011.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Applications	2
1.2.1	Blood Vessels in 2D Medical Images	2
1.2.2	Blood Vessels in 3D Medical Images	3
1.2.3	Streets, Rivers, and Bridges in Satellite Images	6
1.3	Contribution of this Thesis	7
1.3.1	Direction-Preserving Minimal Path Methods	7
1.3.2	Progressive Minimal Path Method	9
1.4	Thesis Organization	11
2	Review of Approaches for Segmentation of Line Structures	13
2.1	Structure Enhancement	13
2.1.1	Differential Measures	14
2.1.2	Machine Learning	17
2.2	Deformable Models	18
2.2.1	Parametric Intensity Models	19
2.2.2	Active Contours and Level Set Method	20
2.3	Graph Cuts	23
2.4	Minimal Paths	25
2.4.1	Fast Marching with Anisotropic Speed	26
2.4.2	Domain Lifting	27
2.4.3	Line Graph Approaches	31
2.4.4	Directional Information	32
2.4.5	Additional Keypoints	35
2.4.6	Dynamic Speed	36
2.4.7	Integer Programming	38
2.5	Hybrid Methods	39

2.5.1	Geometry Refinement	40
2.5.2	Structure Completion	41
3	Background of Minimal Path Methods	43
3.1	Introduction to Minimal Path Methods	43
3.1.1	Interpretation as Wave Propagation	43
3.1.2	Advantages of Minimal Path Approaches	45
3.1.3	Algorithmic Taxonomy for Minimal Path Methods	46
3.2	Fast Marching Method	47
3.2.1	Propagation of the Wavefront	48
3.2.2	Extracting the Line Structure	49
3.3	Problems of Classical Minimal Path Approaches	50
4	Direction-Preserving Minimal Path Methods	53
4.1	Overview of the 3D Segmentation Approach	54
4.2	3D Model-Based Approach	55
4.2.1	3D Parametric Intensity Model	55
4.2.2	Automatic Initialization	56
4.2.3	Partly Missing Connections	56
4.3	3D Probabilistic Minimal Path	57
4.3.1	Energy Function	57
4.3.2	Sampling Scheme	59
4.4	3D Fast Marching with Anisotropic Directional Prior	60
4.4.1	Standard Fast Marching Approach	60
4.4.2	Anisotropic Directional Prior	62
4.4.3	Interpretation of Anisotropic Directional Prior	65
4.5	Iterative 2D Fast Marching	65
4.5.1	Iterative Framework	66
4.5.2	Speed Functions	67
4.5.3	Termination Criteria	69
5	Experimental Results of Direction-Preserving Minimal Path Meth-	71
	ods	
5.1	Synthetic 3D Images	71
5.1.1	Branching Vessels	71
5.1.2	Parallel Vessels	76
5.2	Real 3D 7T MRA Images	77
5.2.1	Materials	77

5.2.2	Segmentation Results	77
5.2.3	Quantitative Evaluation	81
5.2.4	Clinical Studies	82
5.2.5	Stroke	82
5.2.6	Vascular Dementia	83
5.2.7	Aging Effect	83
5.3	Synthetic 2D Images	83
5.4	Real 2D Retinal Images	84
6	Progressive Minimal Path Method	89
6.1	Minimal Path Method with Dynamic Speed	89
6.1.1	Dynamic Speed	90
6.1.2	Progressive Minimal Path Framework	91
6.2	Instances of the Progressive Minimal Path Method	95
6.2.1	Appearance Feature	95
6.2.2	Geometric Feature	97
6.2.3	Joint Appearance and Geometric Feature	100
7	Experimental Results of Progressive Minimal Paths	105
7.1	Appearance Feature	105
7.1.1	2D Synthetic Images	105
7.1.2	2D Real Images	109
7.2	Geometric Feature	112
7.2.1	2D Synthetic Images	112
7.2.2	2D Retinal Images	113
7.2.3	3D Synthetic Images of Vessels	114
7.2.4	3D 7T MRA Images of Brain Vessels	118
7.3	Joint Appearance and Geometric Feature	120
8	Summary and Conclusion	125
8.1	Direction-Preserving Minimal Path Methods	125
8.2	Progressive Minimal Path Method	127

List of Tables

5.1	Quantitative results for 508 gaps (errors in mm).	79
5.2	Overview of the six subjects in the stroke study.	82
5.3	Evaluation of the iterative fast marching approach and comparison with Li-Yezzi.	87
7.1	Comparison of quantitative results for retinal images.	114
7.2	Mean errors and standard deviations (in voxels) of the centerline positions for all experiments with 3D synthetic images.	117
7.3	Quantitative results for 508 gaps in 3D 7T MRA images (errors in mm).	118

List of Figures

1.1	Location and shapes of the retina and the retinal vessels.	3
1.2	2D fundus images of the human retina.	3
1.3	Location and shapes of the lenticulostriate arteries.	4
1.4	Comparison of 3D 3T MRA and 3D 7T MRA images of the region around the LSAs.	5
1.5	Typical challenges for vessel segmentation in 3D 7T MRA images of the cerebral vasculature.	6
1.6	Satellite images.	7
1.7	Examples of different application areas of minimal path methods . .	8
1.8	Comparison of minimal path methods.	10
2.1	Hessian-based vesselness measure for line structures in 2D images. .	15
2.2	Flux-based vesselness measure for 3D images.	16
2.3	The filter bank.	18
2.4	Result of the filter bank.	19
2.5	Illustration of the level set method.	21
2.6	Principle of segmentation using graph cuts.	24
2.7	Using graph cuts for 3D segmentation.	24
2.8	Shape of the wavefront using different speed functions after time τ , starting at the point p	26
2.9	Comparison of the results using isotropic and anisotropic speed. . .	27
2.10	Domain lifting for 2D images.	28
2.11	Minimal path method in with radius as an additional dimension of the parameter space.	28
2.12	Segmentation of a retinal vessel and comparison with a classical minimal path approach.	29
2.13	Segmentation result for vessels in two 3D CTA images.	29
2.14	Minimal path method with radius and orientation as additional dimensions of the parameter space.	30

2.15	For line structures with crossing, both classical minimal approach (a) and the approach with radius as additional dimension (b) cannot follow the natural path.	30
2.16	Three representations of the same path.	32
2.17	Neighborhoods in graphs for 2D images.	33
2.18	Three stages of the wave propagation for vessel segmentation.	33
2.19	Three stages of the wave propagation for vessel segmentation.	34
2.20	In each iteration, the locally optimal path starting from x_0 inside circular ROI is determined.	34
2.21	Minimal path approach with automatically inserted key points.	36
2.22	Minimal path approach with automatically inserted key points and only one user-specified point.	37
2.23	Dynamic speed.	38
2.24	Result of classical minimal path approach and the approach with dynamic speed.	38
2.25	Segmentation result.	39
2.26	Segmentation result.	40
2.27	Segmentation of liver vessels in a 3D CT image.	41
2.28	Segmentation of vessel networks in two 2D retinal images.	42
3.1	Steps of minimal path methods.	44
3.2	Problem of classical minimal path approaches.	51
4.1	MIP of a region of a 3D 7T MRA image of the brain.	54
4.2	Comparison of our approach with previous minimal path approaches.	57
4.3	2D sketch of 3D sampling within a cone.	59
4.4	2D sketch of 3D anisotropic effect.	61
4.5	Effect of the directional prior.	66
4.6	Two retinal images of the DRIVE dataset.	66
4.7	Sketch of the iterative vessel segmentation approach.	67
5.1	Synthetic 3D branching vessel.	72
5.2	Determination of mean error.	73
5.3	Mean error and standard deviation of the centerline positions for different <i>angles</i> for 3D branching vessels.	74
5.4	Mean error and standard deviation of the centerline positions for different <i>gap lengths</i> for branching vessels	75

List of Figures

5.5	Mean error and standard deviation of the centerline positions for different <i>noise levels</i> for branching vessels	75
5.6	Synthetic 3D parallel vessels.	76
5.7	Mean error and standard deviation for different <i>gap lengths</i> for parallel vessels.	77
5.8	Segmentation and quantification result of vessels in the LSA region from a real 3D 7T MRA image using FM-ADP.	78
5.9	Comparison of the segmented LSAs.	79
5.10	Segmentation results for a branching vessel in a real 3D 7T MRA image.	80
5.11	Segmentation results for parallel vessels in a real 3D 7T MRA image.	81
5.12	Comparison of different fast marching methods for a synthetic curve with noise, and a loop.	84
5.13	Comparison of different fast marching methods for an image from the DRIVE dataset.	85
5.14	Quantitative evaluation using retinal images from the DRIVE dataset.	86
5.15	Example results for our approach and Li-Yezzi.	88
6.1	One step of the wave propagation with dynamic speed.	92
6.2	Left: Retinal vessels crossing each other (image from DRIVE dataset). Right: Segmentation results.	97
6.3	Wave propagation of our progressive minimal path approach.	99
6.4	Principle of tangent-based bridge detection.	102
7.1	Segmentation of a curved line structure with many gaps due to artifacts (left column), and a spiral with a high noise level (right column).	107
7.2	Dependency of Q_{inside} on the length Γ of the local paths γ_{local}	108
7.3	Dependency of Q_{inside} on the threshold T_v of the mean vesselness of the local paths γ_{local}	108
7.4	Examples of synthetic images with four different types of 2D line structures.	109
7.5	Dependency of Q_{inside} on the noise level for four different types of 2D line structures.	110
7.6	Segmentation of real images.	111
7.7	Synthetic images of a loop (left) and an open curve which is crossed by a straight line (right).	112
7.8	Four different examples of retinal vessels from the DRIVE dataset.	113

7.9	(a) Sketch of <i>parallel vessels</i> with a gap. (b) 2D section of an example of a 3D synthetic image based on (a).	115
7.10	(a) Sketch of <i>branching vessel</i> . (b) 2D section of an example of a 3D synthetic image based on (a).	116
7.11	Mean error and standard deviation of the centerline positions for different gap lengths for <i>parallel vessels</i> in 3D synthetic images using different minimal path methods.	116
7.12	Mean error and standard deviation of the centerline positions for different gap lengths for <i>branching vessels</i> in 3D synthetic images using different minimal path methods.	117
7.13	Percentage of the total number of completed gaps for ten 3D 7T MRA images of the LSA region of the human brain using different minimal path methods.	119
7.14	Percentage of the total number of completed gaps for 3D 7T MRA images: Average over all 10 3D 7T MRA images.	120
7.15	Segmentation of vessels of the whole brain from a 3D 7T MRA image using our progressive method in conjunction with the model-based approach.	120
7.16	Bridge detection in satellite images with different types of bridges, rivers, and environments.	121
7.17	Bridge detection in a satellite image.	122

List of Algorithms

3.1	Fast marching method	48
3.2	Backtracing of the fast marching method using discrete neighborhood.	49
3.3	Backtracing of the fast marching method using Runge-Kutta method.	50
6.1	Progressive minimal path framework	94
6.2	IsSatisfied for joint appearance and geometric feature	101

Chapter 1

Introduction

1.1 Motivation

Image segmentation plays a vital role in many computer vision applications. On the one hand, segmentation is an important task in its own right. On the other hand, for many further analysis steps, high quality segmentation is needed. Consequently, segmentation is one of the most active research areas of computer vision. Intensive research has been carried out on segmentation methods and there has been considerable progress, both using classical methods (e.g., [1–5]), and more recent deep learning methods (e.g., [6–10]).

However, the majority of the research work has been devoted to the segmentation of 2D natural images, this includes usually objects of everyday life. For example, the major benchmark databases for image analysis [11–16] contain mainly 2D natural images. Most of the objects in these images have some common characteristics. For example, their shapes are usually relatively regular and hardly contain thin and elongated parts. This regularity is implicitly assumed by many segmentation methods which employ regularization methods. Also, the objects have rich color and texture information, and consequently color and texture models can be used to facilitate segmentation. Furthermore, images in these databases have relatively high quality, i.e. they contain low noise and have high image contrast.

In contrast to common objects in everyday life acquired by a video camera, line structures have different characteristics, which necessitates other segmentation methods. Due to their inherently thin and long structure, many methods using smoothness-based regularization are not suitable, because usually regularization tends to shorten the boundary. This phenomenon is known as “shrinking bias” and will be discussed later in the literature review in Chapter 2. Furthermore, a typical

class of line structures are blood vessels or other fiber-like structures in medical images. Such images contain no color information and relatively little texture, and the noise level is high and the image contrast is often low. These characteristics make the segmentation of line structures very challenging, and methods, which have been applied successfully to segment common objects in everyday life, are typically not well suited. Consequently, methods are needed which are specifically designed for the task of segmenting line structures. In this thesis, the aim is to develop such methods.

1.2 Applications

The segmentation of line structures has many applications. For example, the methods and algorithms developed in this thesis have been applied to three tasks, namely the segmentation of blood vessels in 2D medical images, the segmentation of blood vessels in 3D medical images, and the segmentation of rivers and streets in 2D satellite images. These applications are described below.

1.2.1 Blood Vessels in 2D Medical Images

2D medical imaging is very widely used in clinical practice. Commonly used modalities include X-ray radiography, 2D ultrasound, and optical imaging using fundus photography. These modalities have relatively low cost, short imaging time, and relatively low radiation dosages, and most of them provide images with relatively high resolution.

In this thesis, segmentation of blood vessels from 2D retinal images is studied. These images are acquired using fundus photography. The locations of the retina and the vessels are shown schematically in Fig. 1.1, and two fundus images of the retina containing a network of vessels are shown in Fig. 1.2.

The analysis of the morphology of retinal vessels is crucial for the diagnosis of diseases such as diabetes [18], stroke [19], and glaucoma [20], since these diseases lead to noticeable changes of the geometry of retinal vessels. For example, for patients of diabetes the retinal vessels can have increased tortuosity [21]. For patients with open-angle glaucoma, the diameters of retinal vessels change significantly [22, 23]. These geometric properties can be measured by first segmenting the retinal vessels in the images.

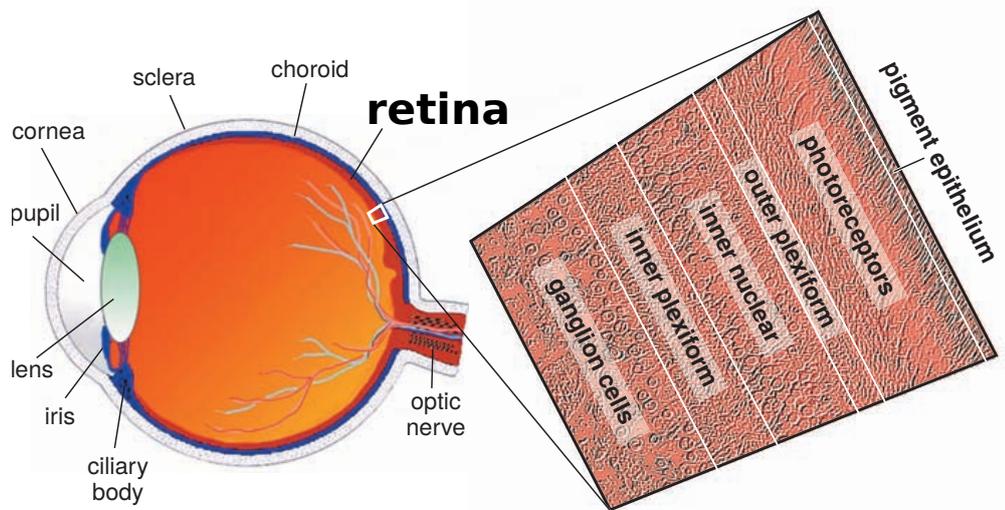


Figure 1.1: Location and shapes of the retina and the retinal vessels (from [17]).

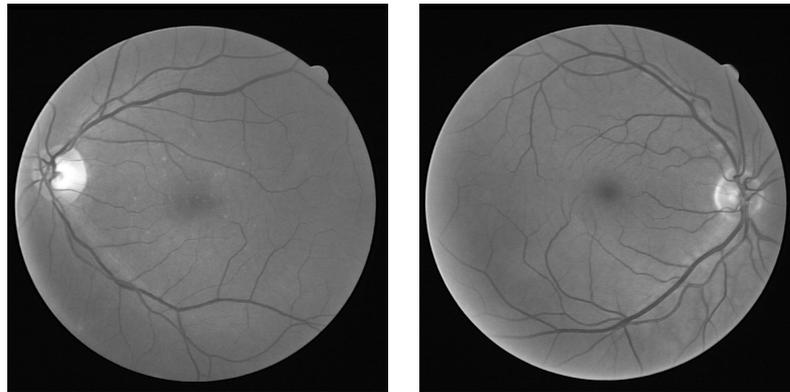


Figure 1.2: 2D fundus images of the human retina (from [24]).

1.2.2 Blood Vessels in 3D Medical Images

Using 3D medical imaging, the entire volumes are acquired so that it is possible to observe 3D structures directly, instead of relying only on 2D projections. Common 3D imaging modalities are computed tomography (CT), magnetic resonance imaging (MRI), single photon emission computed tomography (SPECT), and positron emission tomography (PET). To acquire images of blood vessels, usually special CT or MRI are used, i.e. computed tomography angiography (CTA) or magnetic resonance angiography (MRA), respectively. Among these two modalities, CTA has shorter imaging time and lower cost. In addition, bones, vessels, and soft tissues can be imaged at the same time. However, the radiation of CTA is relatively high. In contrast, MRA is more expensive and needs longer imaging time, but it has the advantage that it is non-invasive, i.e. there is no radiation for the patient.

With the advent of MRI scanners with high field strength, such as the 7 Tesla (7T) scanners, the imaging resolution of MRA has been increased significantly [25].

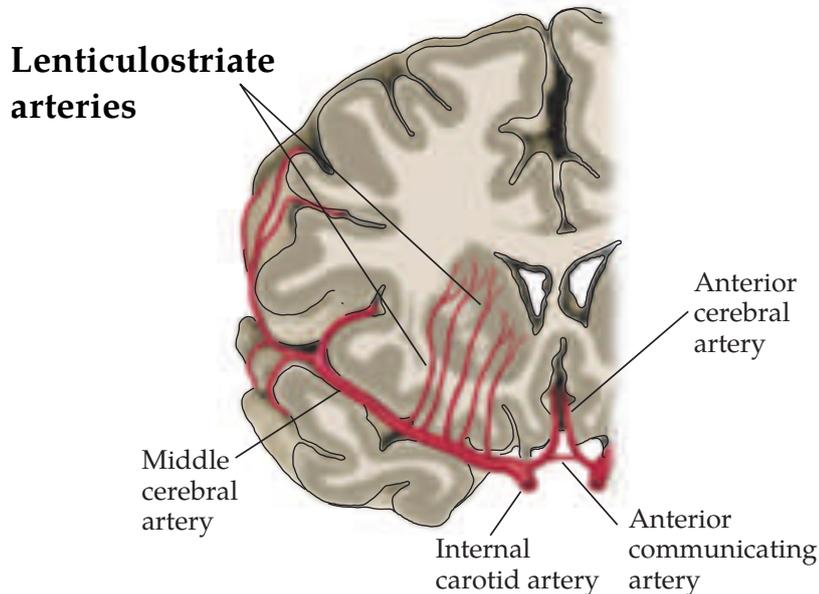


Figure 1.3: Location and shapes of the lenticulostriate arteries (from [26]).

In this thesis, the segmentation of human brain vessels from 3D 7T MRA images is considered. Analysis of these vessels is important for the diagnosis of different serious diseases. For example, symptomatic and silent stroke or vascular dementia can be caused by abnormalities of small cerebral vessels [27, 28]. To determine pathological changes, the vessel trees need to be *segmented* and *quantified*. Among the available imaging modalities, magnetic resonance angiography (MRA) is widely used to acquire 3D images of the cerebrovascular system. With the recently introduced 7 Tesla (7T) MRA, high-resolution 3D images can be acquired non-invasively. These images contain considerably more thin vessels compared to 1.5T or 3T MRA images [25]. In this work, we are concerned with the analysis of lenticulostriate arteries (LSAs), which form a complex vascular system. The LSAs originate from the middle cerebral arteries, and are the major microvessels supplying blood to the basal ganglia and internal capsule. Fig. 1.3 shows the location and shapes of the LSAs, and Fig. 1.4 shows 2D maximum intensity projections (MIPs) of cerebral vessels of the LSA region in 3D images of the same subject using 3T MRA (Fig. 1.4a, 1.4c) and 7T MRA (Fig. 1.4b, 1.4d). The basal ganglia and internal capsule are susceptible to diseases like ischemic and hemorrhagic cerebral stroke [29]. Although it is well known that LSAs are involved in such types of stroke, it remains unclear how the morphology of LSAs changes due to stroke [27].

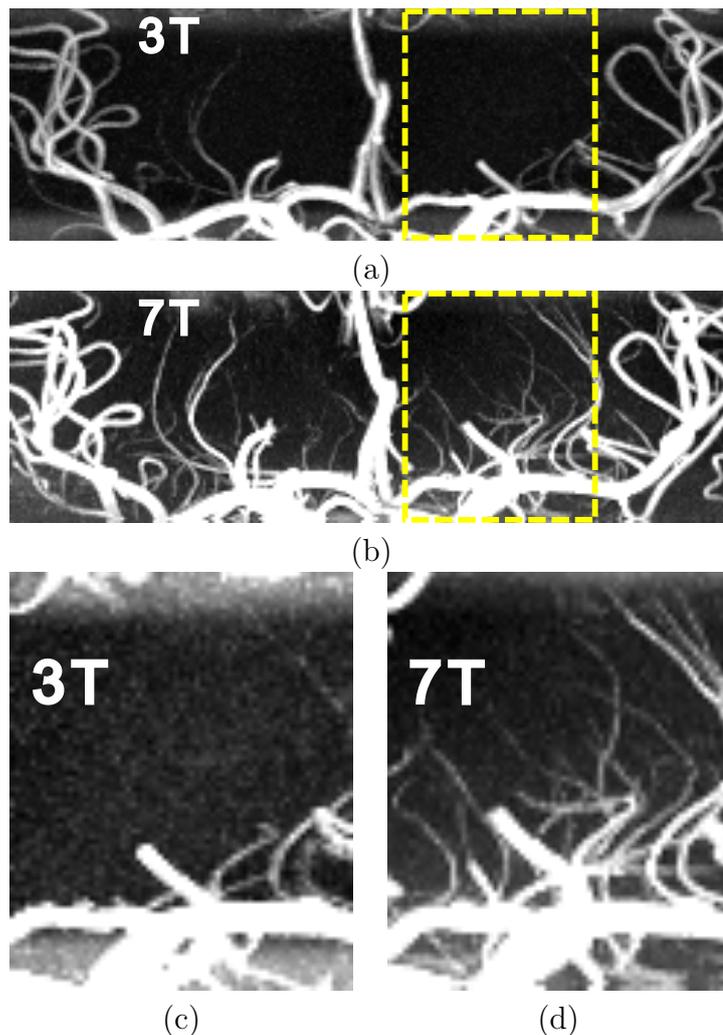


Figure 1.4: Comparison of 3D 3T MRA and 3D 7T MRA images of the region around the LSAs. The images are acquired from the same subject. (a) and (b) show 2D MIPs of the LSA region in 3T and 7T images, respectively, and (c) and (d) show zoomed views of the sections marked in (a) and (b), respectively. The contrast has been enhanced to improve the visibility of thin vessels.

In this thesis, we consider the automatic segmentation and quantification of LSAs in 3D 7T MRA images. This is a challenging task since many LSAs are relatively thin and long, comprise parts with high curvature, and are often located close to each other (Fig. 1.4d). Furthermore, although 7T MRA contains more vessels, the signal-to-noise ratio is only slightly higher than that of 3T MRA [30]. Typical challenges of 7T MRA data include high noise level, highly curved vessels, low image contrast, and tubular artifacts (Fig. 1.5).

Concerning the segmentation of cerebral vessels from 3D 7T MRA images, only few approaches exist. In [31], segmentation is performed manually, which is very time-consuming and not suitable for a large number of images. A semi-automatic

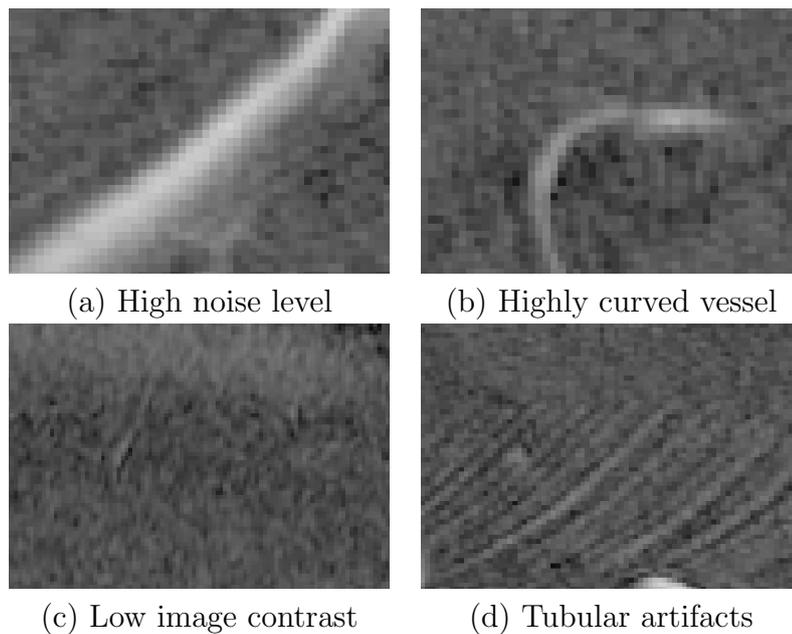


Figure 1.5: Typical challenges for vessel segmentation in 3D 7T MRA images of the cerebral vasculature. The contrast in these images is enhanced to improve the visibility.

approach is proposed in [32], but there the vessels still need to be segmented interactively slice by slice. Existing automatic segmentation approaches are based on vesselness measures (e.g., [33, 34]) or region growing initialized using high-intensity voxels (e.g. [35]). However, these approaches do not take into account the shape of vessels. For example, in noisy images, the segmentation may leak out of the area of interest. In addition, important properties of vessels such as the radius are not quantified. Also note that in high-contrast regions, directional information can be estimated reliably (e.g., using a model-based approach), which can help to segment vessels in neighboring low-contrast and noisy regions. However, none of the previous approaches incorporated such reliable directional information.

1.2.3 Streets, Rivers, and Bridges in Satellite Images

Analysis of remote sensing images is important for a large variety of applications, for example, precision agriculture, assessment of biodiversity, creation of maps like Google Maps, and quantification of damages caused by natural catastrophes. The images are usually acquired using unmanned aerial vehicles (UAV), airplanes, or satellites [36]. Commonly used imaging modalities are based on optical photography or infrared photography.

In this thesis, we focus on the interactive segmentation of streets, rivers, and

1.3. Contribution of this Thesis

the detection of bridges over rivers in images taken from satellites using optical photography. The user specifies the start and end points to segment the streets or rivers. In addition, the bridges over a river are automatically detected during the segmentation of the corresponding rivers, under the reasonable assumption that the bridges are roughly perpendicular to the river underneath them. Examples of the satellite images containing streets, rivers, and bridges are shown in Fig. 1.6.



Figure 1.6: Satellite images of (a) Streets (b) A river with bridges.

1.3 Contribution of this Thesis

In this thesis, two different methods for the segmentation of 2D and 3D line structures are introduced. These methods significantly extend the minimal path framework, which is commonly used for the segmentation of line structures. The effectiveness and versatility of our methods are demonstrated by applying them to a variety of tasks in different application areas, i.e., segmentation of 2D and 3D vessels, segmentation of streets and rivers, as well as detection of bridges, as shown exemplarily in Fig. 1.7.

1.3.1 Direction-Preserving Minimal Path Methods

The first method introduced in this thesis employs direction-preserving minimal paths. This method is used in a novel hybrid approach for automatic 3D segmentation of cerebral vessels from high-resolution 7T MRA image data. Additionally, the method is also used for semi-automatic segmentation of retinal vessels in 2D fundus images. Our approach exploits tubular shape information to avoid the leaking problem of minimal path methods. Compared to two-step approaches where first a binary segmentation is determined and then a quantification is computed, in our approach segmentation and quantification are performed simultaneously in

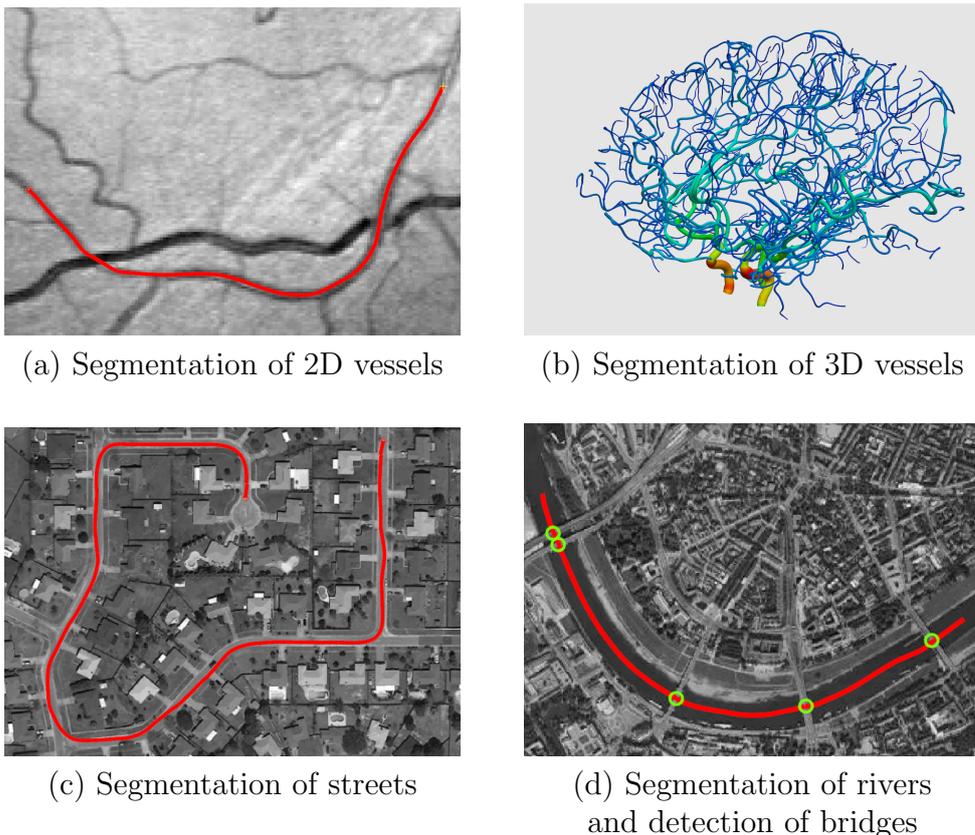


Figure 1.7: Examples of different application areas of minimal path methods. (a) Segmentation of vessels in 2D retinal images. (b) Segmentation of vessels in 3D 7T MRA images of the human brain. (c) Segmentation of streets in 2D satellite images. (d) Segmentation of rivers and detection of bridges in 2D satellite images.

one single step. Furthermore, our approach can cope well with thin vessels in 3D 7T MRA images and 2D retinal images. The work was published in [37–40]. Our main contribution is twofold.

First, we present two different approaches to incorporate prior directional information into minimal path methods: A sampling-based probabilistic approach and a fast marching approach with anisotropic directional prior. These approaches allow taking into account the important information about the *initial direction* of a vessel while previous minimal path approaches (e.g., [41–44]) do not use such information. Compared to [45], our approach imposes only *soft constraints*, which allows tolerating some deviation from the estimated initial direction.

Second, we propose a novel *hybrid* approach for automatic 3D segmentation and quantification in high-resolution 7T MRA images of the human cerebral vasculature. Our approach combines a minimal path approach with robust model-based vessel segmentation [46]. The model-based approach is used to segment thick ves-

sels and most parts of thin vessels. However, in low-contrast and noisy regions, usually there exist gaps between segmented vessels. To complete these gaps, we exploit the directional information obtained from the model-based approach and use it in the minimal path approaches. Our hybrid approach allows quantification of relevant properties of vessels such as the vessel length and local vessel radius, and is fully *automatic*. We conducted an extensive evaluation of our hybrid 3D vessel segmentation approach using over 2300 synthetic images and 40 clinical high-resolution 3D 7T MRA images. We quantitatively compared the results with ground truth and with a previous minimal path approach. Furthermore, our approach was applied to data from two clinical studies on stroke and vascular dementia, and the results have been evaluated by neuroscientists. Also, we applied our method to the segmentation of vessels in 2D retinal images.

1.3.2 Progressive Minimal Path Method

The second method introduced in this thesis is a novel progressive minimal path method for efficient segmentation of line structures from 2D and 3D images. Our approach uses a dynamic speed based on new path features to better steer the propagation of a *wavefront*. The trace of the wavefront is then used to compute the line structures. The work was published in [47–49]. Our main contribution is twofold.

First, we introduce a novel *efficient* algorithmic framework for minimal paths. In previous minimal path approaches, the speed of the wavefront at a point \mathbf{x} is computed using *point features* (e.g., vesselness using derivative filters [50–52]). There also exist more elaborate features based on filterbanks or neural networks obtained using machine learning (e.g., [53, 54]), but they are computationally expensive, especially for 3D images, and such features have not been used in minimal path approaches. In contrast, we introduce *path features* by aggregating point features along a local path $\gamma_{\text{local}}(\mathbf{x})$ starting at \mathbf{x} . The propagation of the wavefront is *constrained* using these path features. Our experiments show that path features obtained by aggregating standard point features (e.g., vesselness) improve the result significantly compared to point features. An important property of path features is that they need to be computed on-the-fly, i.e. *during* the progressive propagation of the wavefront. This is not possible in classical minimal path approaches using static speed. For example, for the line structure in Fig. 1.8a, previous approaches with a point feature (e.g., [50]) usually lead to a short cut (Fig. 1.8b), since the total length of all gaps due to artifacts along the desired

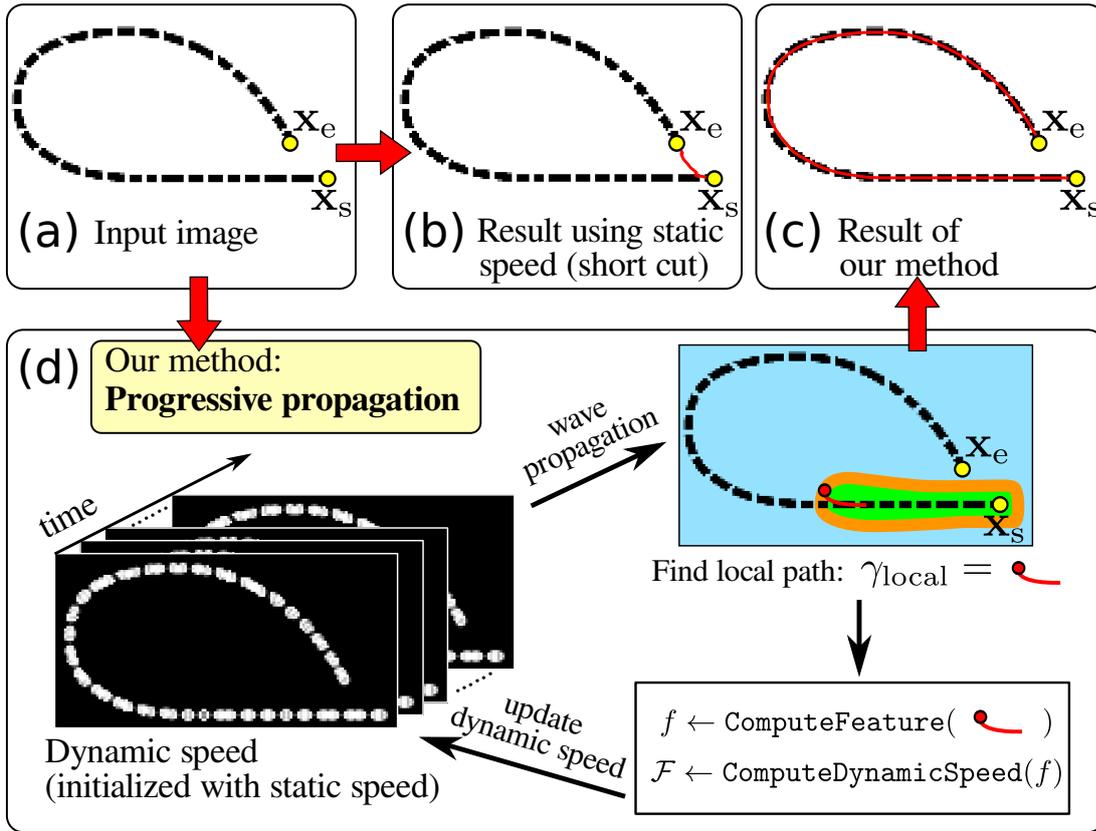


Figure 1.8: Comparison of minimal path methods: (a) A line with artifacts. (b) Using a static speed, the result (red curve) between x_s and x_e is usually a short cut. (c) Using our progressive minimal path method the line is correctly segmented. (d) Sketch of our approach: The dynamic speed is initialized with a static speed and updated during the propagation of the wavefront, based on the functions `ComputeFeature` and `ComputeDynamicSpeed` (see Sect. 6.1.2 and Algorithm 6.1 below). Green, blue, and orange regions represent image points where the arrival time of the wavefront is finally determined (\mathcal{R}_A , Alive), not computed yet (\mathcal{R}_F , Far), or temporarily determined (\mathcal{R}_T , Trial), respectively.

path is much larger than the length of the short cut. But by aggregating the very same point features to path features of γ_{local} and updating the speed function dynamically (Fig. 1.8d), the correct result can be found (Fig. 1.8c, see Sect. 6.2.1 below for more details). Note that the path features in our method can be computed much more efficiently than elaborate features used, for example, in machine learning approaches (reviewed below in Sect. 2.1.2). Our method uses isotropic speed, so we can employ a simple and accurate numerical scheme for arbitrary dimensions. Compared to domain lifting (reviewed below in Sect. 2.4.2), the parameter space of our method has the same dimensionality as the image, i.e. no additional dimension for features is necessary. Furthermore, with our curvature

constrained method we do not need to introduce additional nodes or edges as in methods using curvature regularization [55,56] (reviewed below in Sect. 2.4.3), and features involving a large (arbitrary) number of points can be computed efficiently. Note that the path features in our method are not computed for all points of an image but only for points that are actually reached by the wavefront. This reduces the computation time significantly, especially for 3D images. In our method we use a wider range of features compared to approaches with static speed based on directional information, or key points (reviewed below in Sect. 2.4.5), or previous approaches with dynamic speed using only the normal of the wavefront (reviewed below in Sect. 2.4.6). Compared to [57–59], we use path features instead of point features, and we exploit a larger spectrum of features. Path features cannot be used in these previous approaches. In our work we used a continuous formulation to avoid metrication or discretization errors, but our framework can also be applied using a discrete formulation.

Second, we present several instances of our framework using different new path features, which are computed dynamically during the propagation of the wavefront using a *continuous* formulation. The features comprise an *appearance* feature of the path, a *geometric* feature (the curvature of the path), and a *joint* appearance and geometric feature (appearance of the tangent of the wavefront). Our method was applied to image data from different application areas: Retinal vessel segmentation in 2D images, brain vessel segmentation in 3D images, and segmentation of rivers and streets as well as detection of bridges in satellite images. To the best of our knowledge, this is the first attempt to combine segmentation based on minimal paths with object detection in a principled way. We conducted extensive evaluations using both 2D and 3D images, including both synthetic and real data (satellite images, retinal images, and 3D 7T MRA images).

1.4 Thesis Organization

This thesis is organized as follows. In Chapter 2 we give a review of methods for the segmentation of line structures. In Chapter 3, the theoretical background of minimal path methods, especially the fast marching method, which is the foundation of the methods, is introduced. In Chapter 4, two direction-preserving minimal path methods are presented, and the experimental results of these methods for synthetic and real 3D 7T MRA images of brain vessels, as well as for 2D retinal images, are described in Chapter 5. In Chapter 6, a progressive minimal path method is presented, and the results of this method for synthetic and various 2D

and 3D real images are described in Chapter 7. Finally, Chapter 8 concludes the thesis.

Chapter 2

Review of Approaches for Segmentation of Line Structures

There exist a large number of segmentation approaches for common objects in everyday life acquired by a video camera. These methods mainly deal with regular objects in images. For a general overview and introduction to segmentation methods in computer vision we refer to classical textbooks [60–65].

In many applications besides natural natural scene analysis, the objects to segment resemble 2D or 3D line structures. These objects include blood vessels, fibers of biological structures, and streets and rivers in satellite images. General segmentation methods typically do not perform well for line structures, since these objects are often very thin and long, usually with little color information or texture. Furthermore, the images, especially medical images, often have low contrast and high noise level, and the line structures are often distorted, i.e., they are blurry or contain gaps due to artifacts.

This chapter provides a review of approaches for segmentation and analysis of 2D and 3D line structures, including approaches for enhancement of line structures (Sect. 2.1), approaches based on deformable models (Sect. 2.2), approaches based on graph cuts (Sect. 2.3), minimal path approaches (Sect. 2.4), and hybrid methods (Sect. 2.5). Further surveys of the segmentation methods for line structures, or more specifically for blood vessels, can be found in [66–69].

2.1 Structure Enhancement

In this section, we review methods which are used to enhance the line structures. Usually, the enhancement of line structures is a preprocessing step which extracts

salient features. These features describe line structures better than the image intensity or edge information, while suppressing background and other objects. The result of enhancement is usually used as input by other approaches for further segmentation and analysis.

2.1.1 Differential Measures

Approaches based on *differential measures* exploit partial derivatives of the images. There are mainly two types of such measures, based on the Hessian matrix or the image flux.

In the first type of approaches, which are based on the Hessian matrix, second-order partial derivatives are employed. A vesselness measure is derived using the eigenvalues and eigenvectors of the Hessian matrix (e.g., [50, 51, 70–72]). For example, for a 2D image g , the *Hessian matrix* is computed for each image point \mathbf{x} as:

Hessian matrix

$$\mathcal{H}(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 g(\mathbf{x})}{\partial x \partial x} & \frac{\partial^2 g(\mathbf{x})}{\partial x \partial y} & \frac{\partial^2 g(\mathbf{x})}{\partial x \partial z} \\ \frac{\partial^2 g(\mathbf{x})}{\partial x \partial y} & \frac{\partial^2 g(\mathbf{x})}{\partial y \partial y} & \frac{\partial^2 g(\mathbf{x})}{\partial y \partial z} \\ \frac{\partial^2 g(\mathbf{x})}{\partial x \partial z} & \frac{\partial^2 g(\mathbf{x})}{\partial y \partial z} & \frac{\partial^2 g(\mathbf{x})}{\partial z \partial z} \end{bmatrix} \quad (2.1)$$

Depending on the ratio between the eigenvalues of \mathcal{H} , the local shape of the structure can be estimated.

The second type of approaches, which are based on the flux, employ the first-order partial derivative (e.g., [52, 73–75]). For a curve \mathcal{C} and a vector field \mathcal{V} , the inward *flux* of \mathcal{V} through \mathcal{C} is defined as [52]:

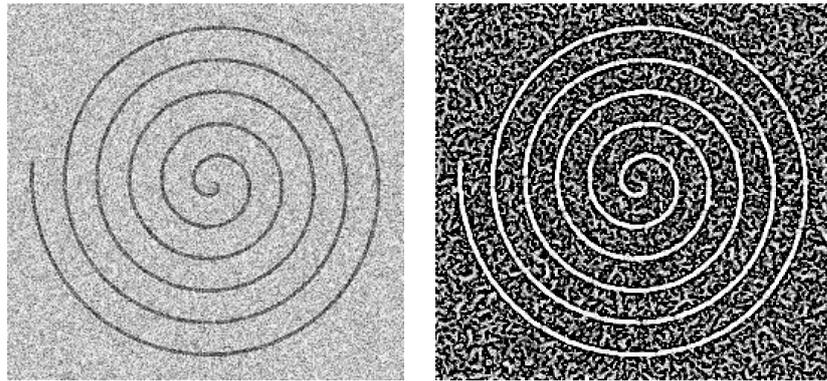
Flux

$$f(\mathbf{x}, \mathbf{v}; r) = \int_{\partial S_r} ((\nabla(G * g)(\mathbf{x} + r \cdot \mathbf{n}) \cdot \mathbf{v}) \cdot \mathbf{v}) \cdot \mathbf{n} \, ds \quad (2.2)$$

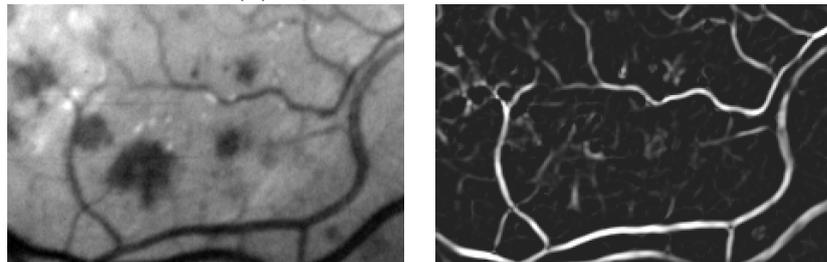
In (2.2), the flux f depends on the position \mathbf{x} , the direction \mathbf{v} , and the radius r of a potential vessel. S_r is the circle with the radius r centered at \mathbf{x} , ∂S_r is the boundary of S_r , G is the Gaussian function, and \mathbf{n} is the normal at the boundary of S_r . The optimal direction \mathbf{v}_{opt} is determined by

$$\mathbf{v}_{\text{opt}} = \arg \min_{\mathbf{v}} f(\mathbf{x}, \mathbf{v}; r). \quad (2.3)$$

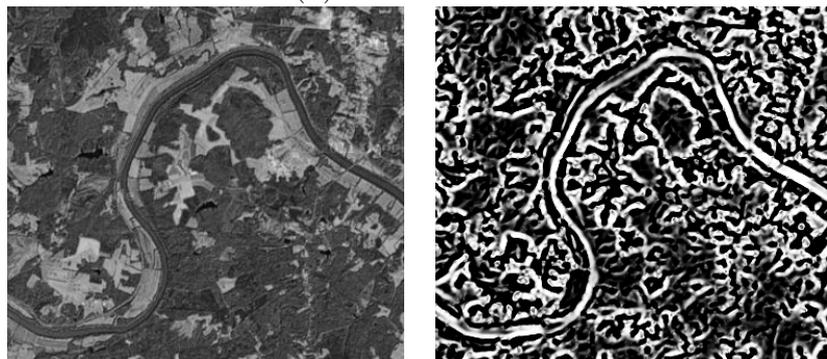
2.1. Structure Enhancement



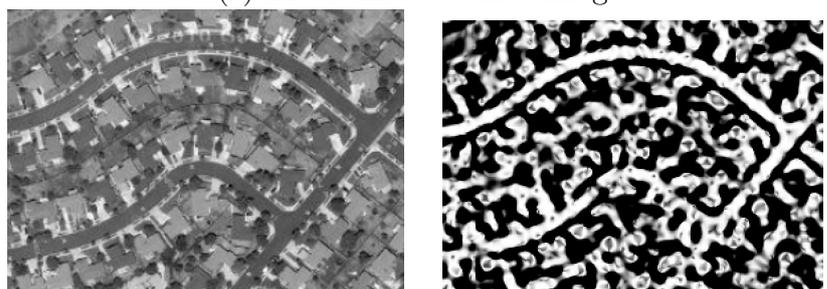
(a) Synthetic line structure



(b) Retinal vessels



(c) Rivers in a satellite image



(d) Streets in a satellite image

Figure 2.1: Hessian-based vesselness measure [50] for line structures in different types of 2D images.

Often, these methods are applied in a multiscale manner (e.g., [50–52]). Differential measures have been applied to 2D and 3D images (e.g., [33, 34]). For example, Fig. 2.1 shows the vesselness map for a wide variety of line structures in

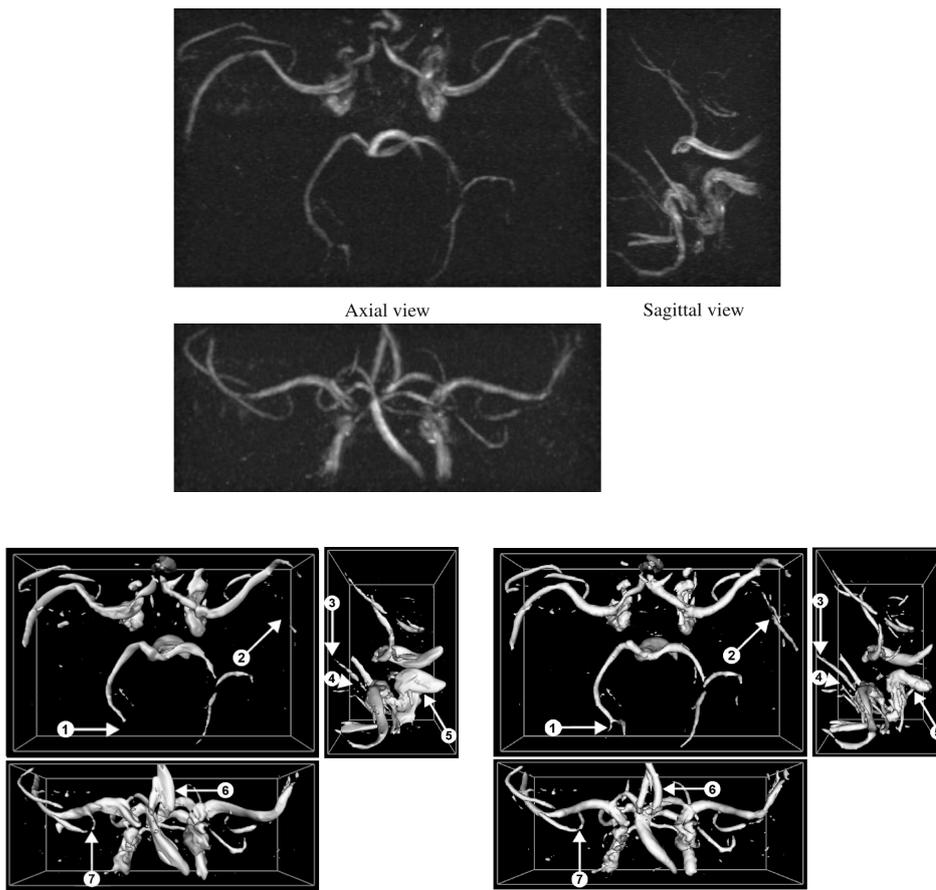


Figure 2.2: Flux-based vesselness measure [52] for vessels in a 3D MRA image (from [52]).

2D images, including images containing synthetic lines, medical images of retinal vessels, and satellite images of rivers and streets. Examples of 3D vessel segmentation in medical images are shown in Fig. 2.2. The differential measures are not limited to the analysis of vessels. For example, they can also be used to enhance streets or rivers in satellite images. In contrast to machine learning based methods (see Sect. 2.1.2 below), which require a large number of annotated images for training, differential measures usually have only a few parameters, which can be estimated from the image data without requiring a training procedure. Therefore, differential measures are widely used for the analysis of a broad spectrum of line structures. However, in low-contrast and noisy regions, partial image derivatives are difficult to estimate, and therefore these approaches alone are not well suited for coping with the challenging 3D medical images, in particular for thin vessels. Therefore in many approaches the differential measures are computed first, and the result is used for further analysis using other methods, such as graph cuts or minimal path methods.

2.1.2 Machine Learning

In approaches based on machine learning, extraction of line structures is typically achieved using annotated training images. In recent years, deep convolutional neural networks (DCNN) have achieved superior results for segmentation tasks compared to classical machine learning methods, while the latter methods are usually easier to train. In the following, we review the methods using DCNN, followed by a discussion on classical machine learning methods.

Using neural networks, non-linear functions can be approximated efficiently, and the backpropagation algorithm is often used for parameter optimization. In most of classical neural network methods, the number of layers is relatively low, and the layers are fully connected, i.e., all neurons in one layer are connected to all neurons in the next layer. In contrast, in DCNN a large number of layers are used, many of which are convolutional layers, i.e., a neuron is connected to a limited number of neurons in the next layer, and all neurons for the same convolution operation share the weights. Using DCNN, networks are often trained end-to-end, i.e., the user does not need to manually design features to use, since the features are also learned automatically by DCNN. Popular frameworks for DCNN include, for example, TensorFlow [76], Caffe [77], MXNet [78], CNTK [79], Torch [80], Keras [81], Matconvnet [82], Lasagne [83]. Originally used for image classification (e.g., [6, 13]), DCNN has now been applied to further application areas such as object detection (e.g., [84, 85]) and semantic segmentation (e.g., [10, 86]). Some methods focus on segmentation of line structures [87] for different application areas such as retinal vessels [88], coronary artery [89], streets in satellite images [53]. However, to train DCNN usually a large number of images are needed, which are often difficult to obtain, especially in the case for medical images. Furthermore, the annotation of the training images can be very time consuming and prone to error, especially for 3D images such as tomographic images. Also, training of DCNN is not easy and very time consuming.

Most classical machine learning approaches require less parameters than DCNN. In several approaches, features are determined manually and used as input for a classifier. For example, in [90] boosting trees are used based to extract coronary arteries in CT images. In contrast, there exist also approaches where the features themselves are learned. For example, in [91], features are defined using a filter bank, which is learned from the data, including streets in satellite images, neurons in brightfield image stacks, and neurons in *in vivo* two-photon image stacks. Compared to DCNN, the method in [91] requires much less parameters, and con-

sequently significantly less training samples are needed to compute the filter bank. The number of filters is further reduced in [54] by using separable filters (see Fig. 2.3), and the results for a 2D retinal image are shown in Fig. 2.4. However,

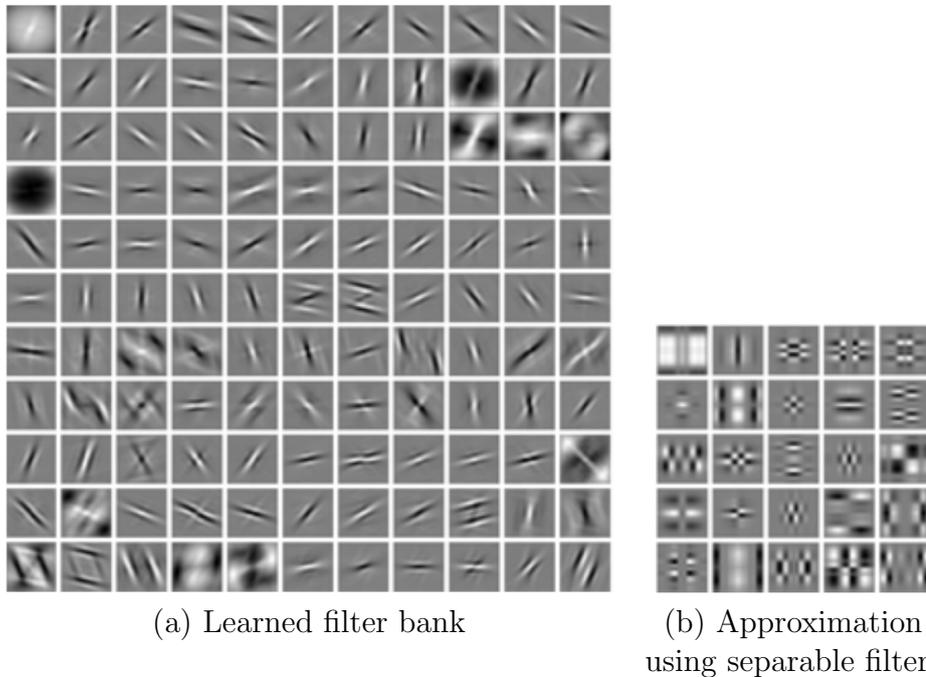


Figure 2.3: The filters in (a) are approximated using a smaller number of separable filters in (b) (from [91]).

in these approaches still large amounts of training data are required, and these approaches are typically still very time-consuming. For 2D images, in [53] the computation time for training is 3 days, and in [54] a large number of filters is used (121 filters, each with 14 orientations). For 3D images, the computation time for training and application would increase even further. Also, the discretization of filter orientations inevitably introduces discretization errors.

2.2 Deformable Models

Deformable models are curves or surfaces which are fitted to objects in an image. There are two main types of deformable models: Parametric intensity models and active contour (level set) methods. Using level set methods, more global properties can be imposed using regularization. In contrast, with parametric intensity models it is possible to segment and quantify the structures simultaneously.

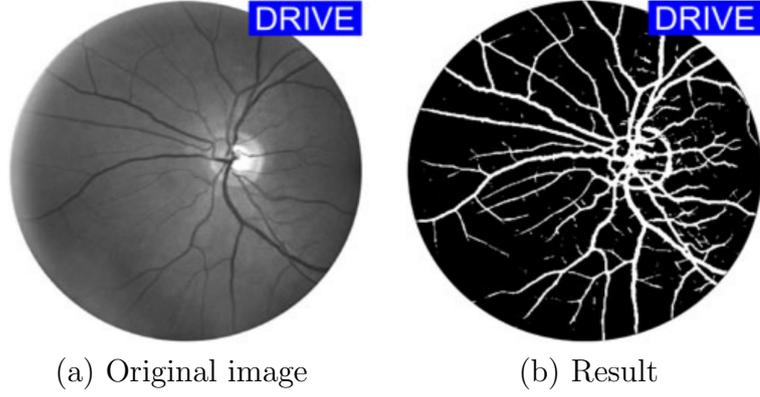


Figure 2.4: Result of the filter bank in [91]. (a) Original 2D retinal image from the DRIVE dataset [24]. (b) Result using the filter bank trained using retinal images (from [91]).

2.2.1 Parametric Intensity Models

Using parametric intensity models (e.g., [46, 92–95]) the properties of line structures are described analytically. By optimizing a cost function for fitting these models to the image data, the values of properties can be determined. For example, in [46] the cost function is defined within a spherical region of interest (ROI) as given below.

Parametric intensity model in [46]

$$\sum_{\mathbf{x} \in \text{ROI}} (g_M(\mathbf{x}, \mathbf{p}) - g(\mathbf{x}))^2 \rightarrow \min, \quad (2.4)$$

where

$$g_M(\mathbf{x}, \mathbf{p}) = a_0 + (a_1 - a_0) g_{Cyl}(\mathcal{R}(\mathbf{x}, \boldsymbol{\alpha}, \mathbf{x}_0), R, \sigma), \quad (2.5)$$

with parameters $\mathbf{p} = (R, a_0, a_1, \sigma, \alpha, \beta, \gamma, x_0, y_0, z_0)^T$. $g(\mathbf{x})$ is the image intensity at the position \mathbf{x} , R is the radius of a vessel, a_1 and a_0 are the intensity levels of the vessel and the surrounding tissue, σ is the image blur, and g_{Cyl} is a 3D cylindrical model. \mathcal{R} is a rigid transform, which consists of a rotation using $\boldsymbol{\alpha} = (\alpha, \beta, \gamma)^T$ and a translation using $\mathbf{x}_0 = (x_0, y_0, z_0)^T$.

Since usually a relatively large ROI is used, these approaches are robust against noise. Also, no post-processing is needed to determine the parameters such as vessel radius and orientation, since the values of these parameters are obtained by optimization during the segmentation. However, since the model is optimized for each ROI individually, there is no measure for the overall optimality. Furthermore,

global constraints, such as regularity of the centerline of the line structures, are not imposed.

2.2.2 Active Contours and Level Set Method

Methods based on active contours and level sets rely on the evolution of curves (for the 2D case) and surfaces (for the 3D case). After being initialized in the vicinity of the objects to segment, the curves or surfaces evolve towards the true boundary of the objects. Originally, the active contour model is formulated using an energy function which consists of three terms [1]:

$$\int_0^1 (\alpha \cdot \|\mathcal{C}_p(p)\|^2 + \beta \cdot \|\mathcal{C}_{pp}(p)\|^2 + g_d(\mathcal{C}(p))) dp. \quad (2.6)$$

For a curve \mathcal{C} , the first two terms specify the internal energy computed using the first and second derivatives of \mathcal{C} , and the third term specifies its external energy computed by a function g_d . These terms are computed explicitly for each point on \mathcal{C} , and \mathcal{C} evolves to minimize the total energy. However, using this method, the contour needs to be parameterized explicitly, which has two consequences. First, it is difficult to handle topological changes during the curve evolution. Second, tracing points on the curve leads to numerical difficulties.

To avoid these difficulties, the level set method was introduced [96]. In the case of 2D images, the 2D contour \mathcal{C} is embedded as the zero level set of a 3D function u . During the evolution of u , \mathcal{C} naturally changes as a consequence of the changes of u . In Fig. 2.5, the evolution of u is shown in the second row, and the blue plane represents the zero level. The shape of the corresponding \mathcal{C} , which is the intersection of u and the blue plane, is shown in the first row. The evolution of a 2D curve \mathcal{C} and its 3D embedding function u are directly related to each other [98]. If \mathcal{C} evolves according to

$$\frac{\partial \mathcal{C}}{\partial t} = \beta \cdot \mathcal{N}, \quad (2.7)$$

where t is the time, β is the speed of the curve evolution, and \mathcal{N} is the normal of \mathcal{C} , then u evolves according to

$$\frac{\partial u}{\partial t} = \beta \cdot \|\nabla u\| \quad (2.8)$$

For example, a commonly used model is the geodesic active contour:

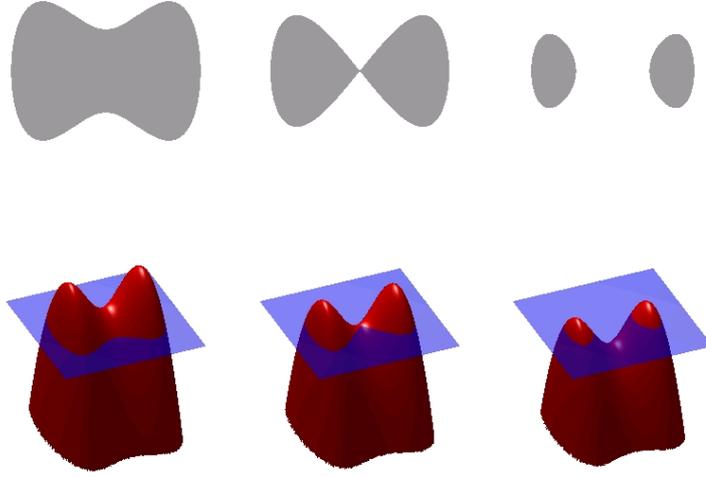


Figure 2.5: Illustration of the level set method (from [97]). The evolution of the 2D curve \mathcal{C} (first row) is the result of the evolution of the 3D embedding function u (second row).

Geodesic active contour in [98]

$$\int_{\mathcal{C}} g_d(\|\nabla g(\mathcal{C}(s))\|) ds \quad (2.9)$$

where g is the input image, and g_d is a strictly decreasing function. For example,

$$g_d(\|\nabla g(\mathcal{C}(s))\|) = \frac{1}{1 + \|\nabla g(\mathcal{C}(s))\|}. \quad (2.10)$$

The curve \mathcal{C} evolves according to

$$\frac{\partial \mathcal{C}}{\partial t} = g \cdot \kappa_{\mathcal{C}} \cdot \mathcal{N} - (\nabla g \cdot \mathcal{N}) \cdot \mathcal{N}, \quad (2.11)$$

and the corresponding embedding function u evolves according to

$$\frac{\partial u}{\partial t} = g \cdot \kappa_u \cdot \|\nabla u\| - \nabla g \cdot \nabla u, \quad (2.12)$$

where $\kappa_{\mathcal{C}}$ and κ_u are the curvature of \mathcal{C} and u , respectively. For a comprehensive overview and a systematic treatment of active contours and level set methods, see the monographs [99–103].

Besides the geodesic active contours, the active contour method has also been

extended in various other ways. For example, in [2], the gradient at edges is propagated into regions with low gradient, resulting in a gradient vector field (GVF). Compared to the classical snakes [1], snake models based on GVF allow the initial contour to be further away from the actual object boundary. Also, strong concavity can be better segmented.

So far, the methods discussed in this section are edge-based methods, in which the energy function is determined using the contour only. In contrast, region-based methods employ the properties of regions to define the energy function. For example, in [104], the intensities of the region inside and outside the contour are used instead of the gradient on the contour, resulting in active contours without edges (ACWE). Using region-based information, objects can be segmented even if their boundary is blurry or discontinuous.

During the evolution of curves or surfaces, the level set function often develops irregularity which cause numerical difficulties. To overcome the problem of irregularity, often re-initialization is used [102, 105], which keeps the level set function as a signed distance function, but this introduces other problems. From a theoretical point of view, the re-initialization may change the level set function [106]. From a practical point of view, there are no obvious rules about how re-initialization should be used [107]. In [108], a new method is introduced so that no re-initialization is needed any more, and the level set function is always regularized automatically.

The level set formulation is based on partial differential equations (PDE), so naturally level set methods are usually implemented using finite difference methods. However, due to their close relationships to PDEs [109, 110], morphological operators can also be used to approximate PDEs for curve and surface evolution [111]. For example, geodesic active contours [98] and active contour without edges [104] can both be implemented using morphological operators. In this way, there are no numerical stability issues which are typical for previous numerical methods like finite difference. Also, the evolution of the level set function does not need re-initialization.

However, typically active contour approaches penalize the length of segmented objects, resulting in the “shrinking bias” [112], i.e., they tend to yield results where the circumference is small compared to the enclosed area. For example, for a thin and long 2D line structure, classifying the entire structure as background would cause the energy function of the geodesic active contour (2.9) to be minimized to zero, even though such a result is obviously wrong. Several remedies to this problem have been proposed. For example, in [113] an expanding force is added

to the energy function, and in [73, 114] differential measure is used to enhance the local structure. However, these approaches do not change the nature of the problem, which is the length-based regularization in the level set function itself.

2.3 Graph Cuts

Methods based on graph cuts are used to solve optimization problems formulated using Markov random fields (MRF). MRF is a powerful framework for the probabilistic modeling of images [99, 100, 115–117]. For the optimization of MRF models, many optimization approaches exist, such as simulated annealing [118], iterated conditional modes [119], graduated non-convexity [120], loopy belief propagation [121], tree-reweighted message passing [122], quadratic pseudo-boolean optimization [123]. These optimization methods are reviewed in [124]. Approaches based on graph cuts have been successfully used in many branches of computer vision, for example, segmentation [3, 4], stereo reconstruction [125], image restoration [125], determination of geodesics [126], classification [5, 127, 128]. For an overview of the theory and application of these methods, see [129–131].

The computation of graph cuts is a classical problem in combinatorial optimization. To use graph cuts for segmentation, an energy function is defined using a graph, which consists of nodes and edges on a regular discrete image grid (2D or 3D). Typically, the energy function consists of two terms: The *data term*, also referred to as *fidelity term*, keeps the result in accordance with the input image, while the *prior term*, also referred to as *smoothness term* or *regularization term*, imposes the prior knowledge on the result. In most MRF models using graph cuts, the prior knowledge is the assumption that the result should be smooth, i.e., the noise of the result should be low. Each node corresponds to a pixel, and there are two special *terminal nodes*, one for the object and one for the background. The user provides seed points for the object and the background, respectively (see the pixels B and O in Fig. 2.6a). Each node is connected with its neighboring nodes on the image grid, as well as with the terminal nodes (Fig. 2.6b). By computing the minimum cut of the graph, all nodes are separated into two partitions, one connected with the terminal node for the object, the other connected with the terminal node for the background (Fig. 2.6c). This separation corresponds to a binary segmentation of the original image (Fig. 2.6d). The same principle can also be used for 3D images. For example, Fig. 2.7a shows one slice in an 3D CT image, along with seed points provided by the user. The segmentation result is shown in Fig. 2.7b. A remarkable feature of methods based on graph cuts is that in many

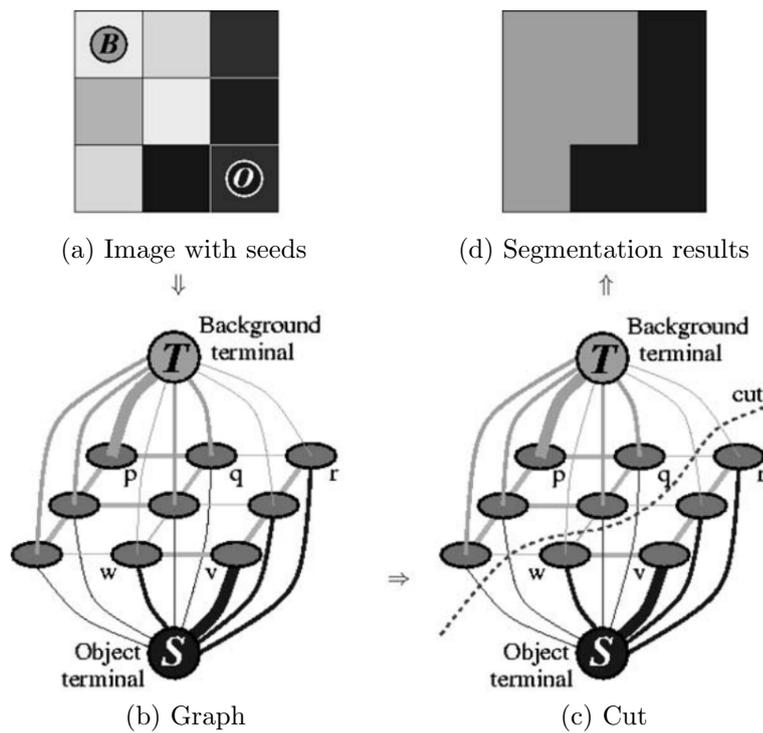


Figure 2.6: Principle of segmentation using graph cuts. (a) The user-specified seed points for the object and the background. (b) Constructed graph. (c) Minimum cut for the graph. (d) Final binary segmentation (from [3]).

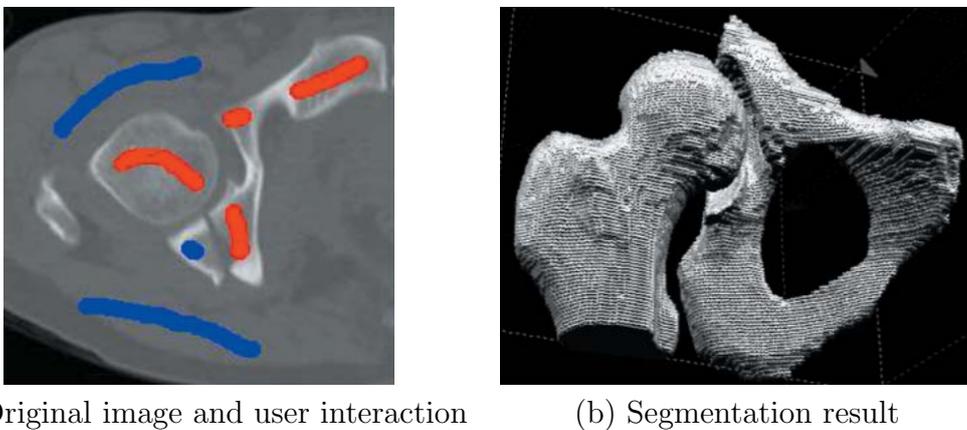


Figure 2.7: Using graph cuts for 3D segmentation. (a) A slice of a 3D CT image, along with user-specified seed points. (b) Segmentation result (from [3]).

cases [3], the global optimum of the energy function can be found in polynomial time. In contrast to methods based on PDEs, such as level set methods, there is no concern about numerical difficulties, since the algorithm is inherently discrete, and therefore there is no numerical approximation error during the computation.

The algorithms for the computation of the minimum cut are studied and com-

pared in [132]. [133] specifies the characteristics of energy functions which can be minimized using graph cuts, identifies them with *submodular* functions in the combinatorial optimization, and shows how to algorithmically construct a graph for any energy function which consists of submodular functions. [124] shows how to handle energy functions with non-submodular functions using quadratic pseudo-boolean optimization (QPBO). By doing so, a partial solution is computed, i.e., for a subset of pixels the optimal label can be determined, while other pixels are assigned a dummy label, meaning that the label at these pixels cannot be determined. In [125], it is shown that also problems with more than two labels can also be solved using graph cuts, even though the result is not the exact global optimum, but a very close approximation of it. Besides graph cuts [3], random walker is also a graph-based segmentation approach [134]. Methods based on graph cuts and random walker can all be considered as special instances of a more general framework of power watershed [135].

Unfortunately, similar to level set methods, methods based on graph cuts also suffer from the shrinking bias [112, 130]. Therefore, when applying these methods to segment vessels, often heuristics (e.g., automatic clustering [136]) are used to reduce the shrinking bias (e.g., [137–141]). Instead of segmenting vessels themselves, some methods aim at segmenting regions containing vessels [142]. Other methods counteract the shrinking bias using more elaborate results of graph theory (e.g., [7, 143–145]).

2.4 Minimal Paths

Using minimal path methods, a line structure is considered as a path between given start and end points such that an energy function is minimized. The energy of a path is defined as the integral of a potential function along the path. There exist mainly two widely used classes of methods, which formulate and solve the minimal path in different ways. On the one hand, the Dijkstra’s algorithm [146] uses a *discrete* formulation, and employs a graph algorithm for optimization. On the other hand, the fast marching method [147] [41] uses a *continuous* formulation and employs a partial differential equation for optimization. In both cases, finding the optimal path can be achieved by first propagating a wavefront emanating from the start point. At each pixel the speed of the wavefront is inverse to the potential. During the propagation, the arrival time of the wavefront at each pixel is recorded in an action map. Upon reaching the end point, gradient descent is used to trace the optimal path from the end point back to the start point. Usually the global

minimum of the energy function can be obtained in polynomial time. The details of these algorithms are described in Chapter 3.

The original minimal path methods have several limitations. For example, a shorter path with pixels of high potential (i.e., low speed) may have lower energy than a longer path with pixels of low potential (i.e., high speed), but usually the latter is the desired result, and the former is referred to as a *short cut*. Also, often only the centerline of the line structure is determined, but there is no information about the radius. Furthermore, to regularize the resulting curves, only length is used, and other properties, such as curvature, are ignored. To overcome these limitations, several improvements have been proposed, which are reviewed in the following.

2.4.1 Fast Marching with Anisotropic Speed

Fast marching approaches based on *anisotropic speed* exploit the anisotropic nature of line structures to avoid short cuts. Examples for anisotropic speed are the Riemannian speed and the Finsler speed. As shown in Fig. 2.8, each of these speed functions can be intuitively characterized by the shape of the wavefront after time τ , starting from one single point p .

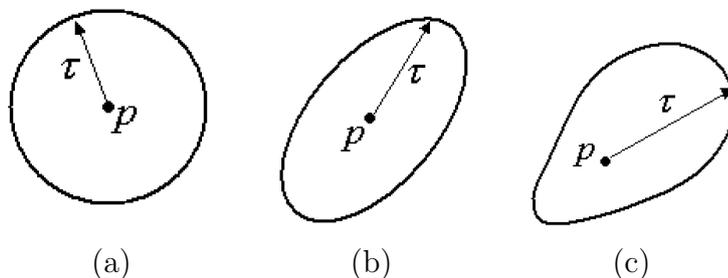


Figure 2.8: Shape of the wavefront using different speed functions after time τ , starting at the point p . (a) Isotropic speed (b) Riemannian speed (c) Finsler speed (from [148]).

In the classical fast marching method, the speed function is isotropic. When start and end points are close to each other but the correct path is long, such isotropic speed function can cause short cut. An example of such structures is shown in in Fig. 2.9. At each pixel, let \mathcal{F}_{\parallel} and \mathcal{F}_{\perp} denote the speeds of the wavefront in the directions *parallel* and *perpendicular* to the line structure, respectively. In the isotropic case in Fig. 2.9a, $\mathcal{F}_{\parallel} = \mathcal{F}_{\perp}$ for each pixel. Consequently, a short cut is found. In contrast, by using anisotropic speed in Fig. 2.9b, so that \mathcal{F}_{\parallel} becomes much higher than \mathcal{F}_{\perp} , the natural shape of the line structure can be followed.

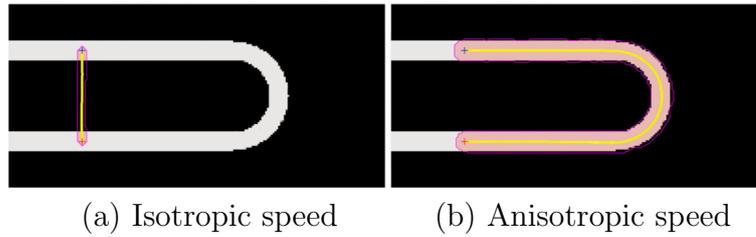


Figure 2.9: Comparison of the results using isotropic and anisotropic speed. (a) Result of classical minimal path using isotropic speed, i.e. at each pixel $\mathcal{F}_{\parallel} = \mathcal{F}_{\perp}$. (b) Result of [44] using anisotropic speed, i.e. \mathcal{F}_{\parallel} is much higher than \mathcal{F}_{\perp} (from [44]).

As anisotropic speed functions, usually Riemannian metrics (e.g., [42–44, 149, 150]) or Finsler metrics (e.g., [151, 152]) are used. However, often these improved metrics require more complex numerical schemes which are not accurate for high anisotropy [44] or cannot be extended in a straightforward manner to higher dimensional cases [151]. The computational complexity of these approaches is usually $\mathcal{O}(N \ln N)$, where N is the number of pixels or voxels of the image, compared to $\mathcal{O}(N)$ for minimal path approaches based on the fast marching method using isotropic speed [153].

2.4.2 Domain Lifting

Minimal path approaches based on *domain lifting* introduce additional dimensions to account for features such as radii and orientations (e.g., [154–156]), i.e., the dimensionality of the parameter space increases with the dimensionalities of both the image and the features. The main idea is illustrated in Fig. 2.10: In addition to the original (x, y) coordinates in the 2D image plane, a new dimension p of the parameter space is introduced to represent a new feature. For a point P on the vessel, the projection P_{xy} in the x, y plane corresponds to the 2D coordinates of the point, while the p coordinate corresponds to the value of the new feature for P . For example, Fig. 2.11 shows the domain lifting in [154]: The new feature is the radius of the line structure, so projecting the 3D curve in the lifted space onto the image space (Fig. 2.11a) results in the centerline of the line structure, while the projection onto the r axis (Fig. 2.11b) provides the radius of the line structure at each 2D position. The results of this method for retinal images are shown in Fig. 2.12. It turns out that the approach [154] not only determines the radius but also keeps the result well centered in the vessel, while the result of classical minimal path approaches does not adhere well to the center. The radius feature can also be used for 3D images, as shown in Fig. 2.13. In [155], the orientation

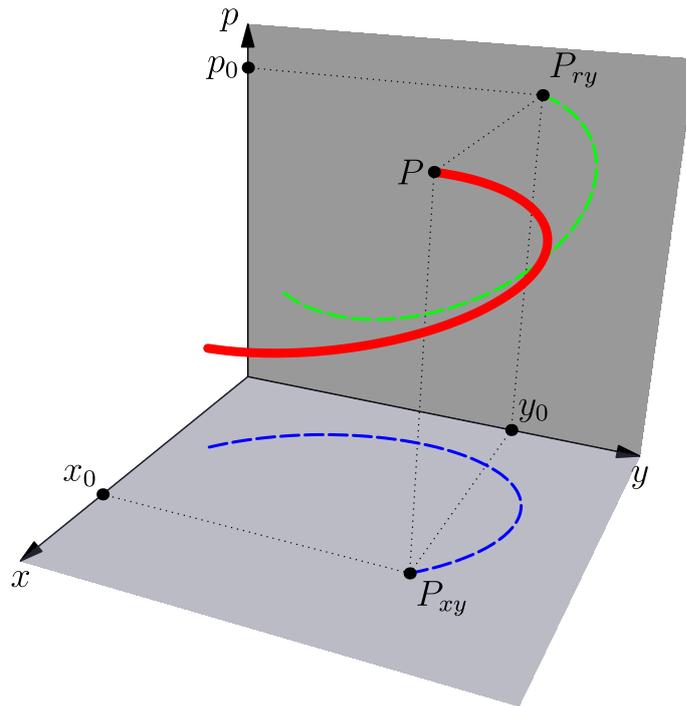


Figure 2.10: Domain lifting for 2D images. x and y are coordinates in the original 2D image space, and p is the additional dimension for a feature.

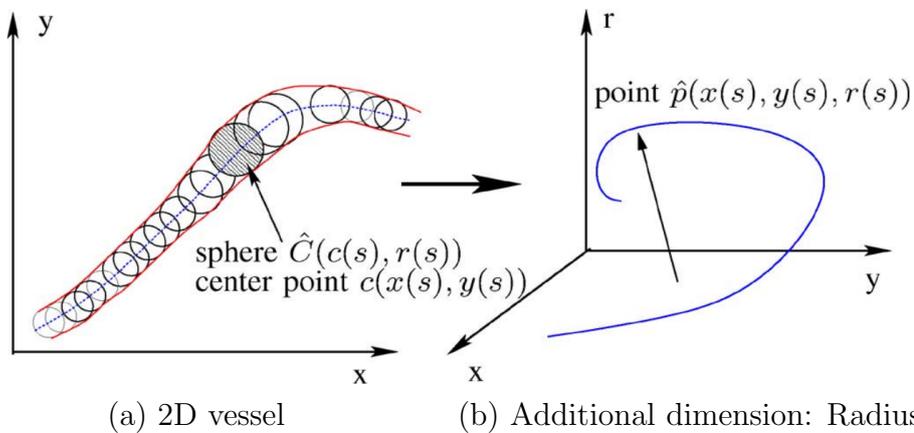


Figure 2.11: Minimal path method in with radius as an additional dimension of the parameter space. For each point on the centerline of a 2D vessel in (a), the (x, y) coordinates along with the radius is represented as a 3D point in the new parameter space in (b) (from [154]).

of the line structure is introduced as a further dimension in the parameter space. To do so, oriented filters are applied to the 2D image (Fig. 2.14a), resulting in the new parameter space (Fig. 2.14b). This parameter space is lifted again by the dimension for radius, in the same way as in [154], to achieve a 4D parameter space. Using the 4D parameter space, line structures with complex structures, such as

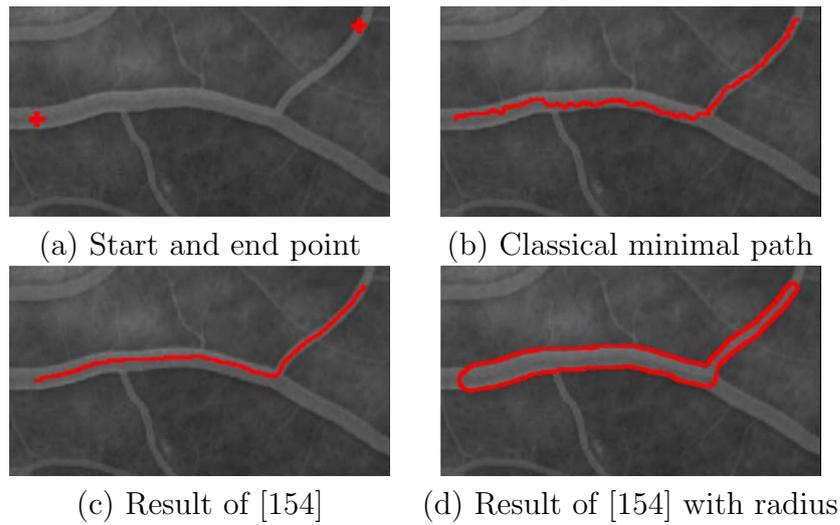


Figure 2.12: Segmentation of a retinal vessel and comparison with a classical minimal path approach. (a) Original retinal image and the start and end point. (b) Result (centerline) of the classical minimal path approach [41]. (c) Result (centerline) of the approach in [154]. (d) Result of the approach in [154], along with the vessel radius (from [154]).

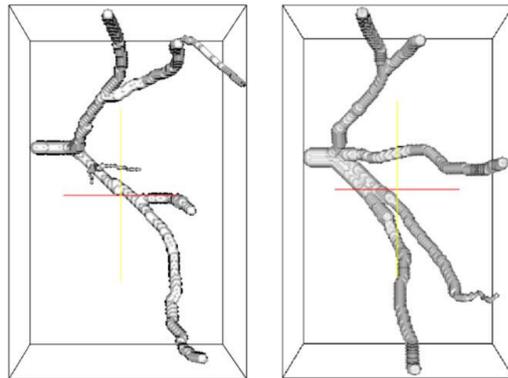


Figure 2.13: Segmentation result for vessels in two 3D CTA images using [154]. For 3D images with the additional dimension for the radius, the parameter space has 4 dimensions (from [154]).

crossings, can be better segmented, as shown in Fig. 2.15.

However, while by domain lifting features can be directly incorporated into the minimal path framework, increasing the dimensionality of the parameter space can cause several difficulties. First, high-dimensional problems are computationally much more demanding. To keep dimensionality reasonably low, often only 2D images or simple features which do not require many additional dimensions are used. For example, in [155,156] only orientations in 2D images were used. In [154], although 3D images are used, the additional feature is radius, which requires only one additional dimension. Similarly, [155] uses one additional dimension to

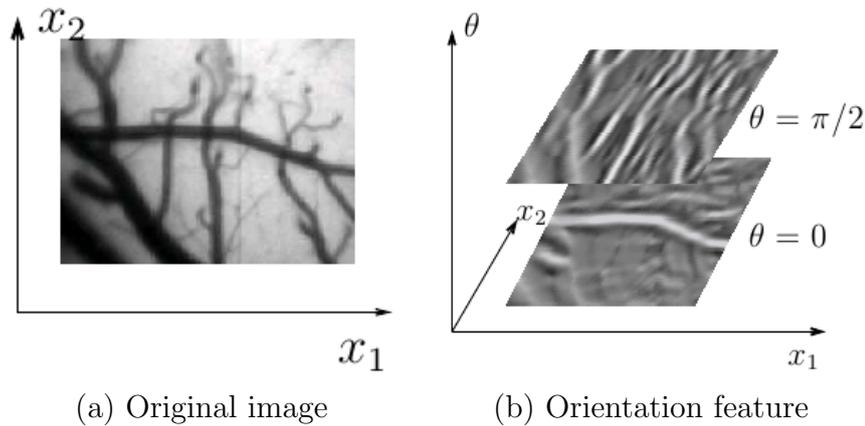


Figure 2.14: Minimal path method with radius and orientation as additional dimensions of the parameter space. (a) For each point on the centerline of a 2D vessel, the (x, y) coordinates along with the radius and the orientation of the vessel is represented as a 4D point in the new parameter space. (b) The parameter space with dimensions for the (x, y) coordinates and for the orientation is shown, without the dimension for radius (from [155]).

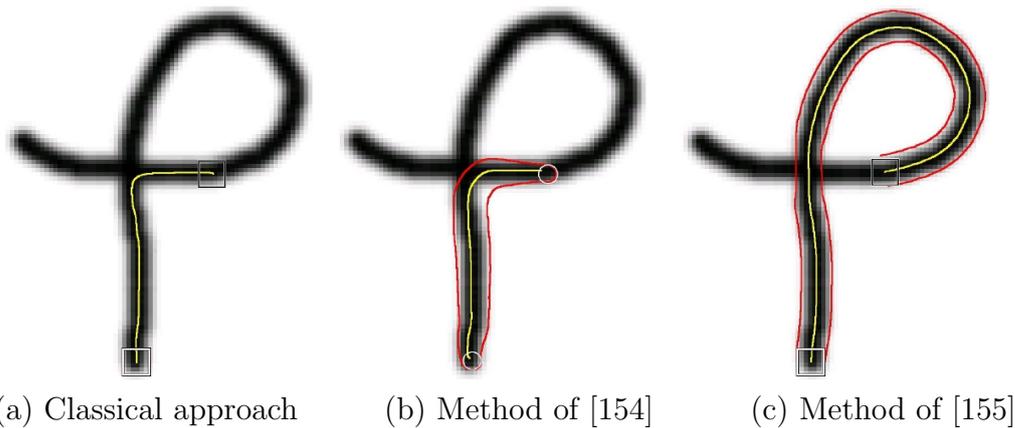


Figure 2.15: For line structures with crossing, both classical minimal approach (a) and the approach with radius as additional dimension (b) cannot follow the natural path. With the orientation feature [155], better result can be achieved (c) (from [155]).

represent 2D orientation, but this is not scalable to higher dimensional images since the number of additional dimensions necessary to represent orientation in higher dimensional parameter spaces increases exponentially. For 3D images, three Euler angles would be needed to specify the orientations, yielding three additional dimensions. Second, while the classical minimal path methods can be applied in arbitrary dimensions, some more complex numerical schemes (e.g., [156]) are inherently restricted to 2D or 3D parameter spaces, limiting their use in higher dimensional parameter spaces. Third, for additional dimensions, only discretized

features can be used, resulting in discretization errors. For example, in [155], only 12 values are used to represent orientations between 0 and π , thus the accuracy of these approaches is limited.

2.4.3 Line Graph Approaches

The minimal path methods reviewed in the previous sections mainly use a *continuous* formulation and employ the fast marching algorithm. Their *discrete* counterparts, i.e., minimal path approaches using a graph-theoretical formulation (e.g., [55,56]), have also been proposed. These approaches extend the original Dijkstra’s algorithm to incorporate more properties of the graphs. To do so, usually *line graphs* are used. Using line graphs, a large number of additional nodes and edges are inserted into the graph corresponding to the image grid to represent higher-order properties of paths. For example, Fig. 2.16 shows the three representations of a path according to [56]. In the original graph G , a path is represented as a sequence of four *nodes* (green, purple, orange, and blue dots) in G (Fig. 2.16a). The weight between two nodes describes the distance between them, and therefore this representation yields a length regularization. Alternatively, Fig. 2.16b shows that the same path can be represented as a sequence of three nodes in the line graph $L(G)$ which correspond to *edges* (green, purple, and orange line segments) in the original graph G . Using this representation, the weight between neighboring nodes in $L(G)$ represents the curvature between edges in G , which is of a higher order than length. The idea of using nodes in the line graph $L(G)$ to represent more complex structures in the original graph G can be further exploited. In Fig. 2.16c, a node in the line graph $L(G)$ corresponds to a pair of consecutive edges (green and purple bended line segments) in the original graph G . The weight between neighboring nodes in $L(G)$ represents the torsion in edge sequences in G , which is of an even higher order than the curvature.

However, methods using line graphs mainly suffer from two difficulties. First, the weights for all combinations of neighboring edge pairs need to be computed, and the number of nodes in the line graph increases exponentially with the number of points involved in computing the features. This results in relatively high computation time. Therefore, only features involving two points (curvature) or three points (torsion) have been exploited. In [55], only curvature for 2D images is used. In [56] curvature and torsion for 3D images are introduced, but this approach is only feasible for relatively small problems. For example, one of the graphs in [56] contained 280 billion edges. To incorporate features involving even more points,

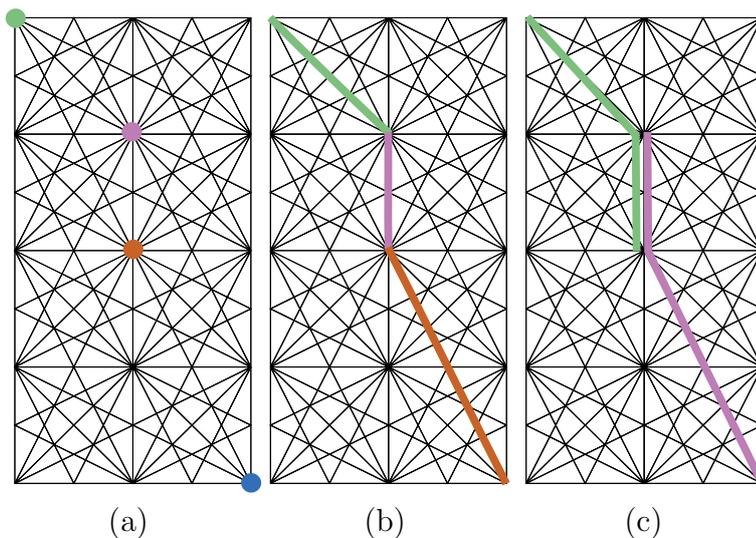


Figure 2.16: Three representations of the same path. Nodes of the path are shown in different colors. (a) Representation using nodes of the original graph G . (b) Representation using nodes of the line graph $L(G)$ which correspond to edges in the original graph G . (c) Representation using nodes of the line graph $L(G)$ which correspond to edge pairs in the original graph G (from [56]).

the graph size would become prohibitively large. Second, the used features are subject to *metrication errors* due to the discrete nature of the graphs. For example, these approaches have a finite resolution of orientations, limited by the number of edges connected to a node, i.e., the size of the neighborhood. Usually 16-neighborhood is used, and [56] employs a larger neighborhood of 32 neighbors, as displayed in Fig. 2.17a. Using the 32-neighborhood, 16 different orientations can be represented. This is not sufficient for certain applications which require high angular resolution (for example, the application of analyzing satellite images in Sect. 6.2.3). Furthermore, the distribution of the represented orientations is not homogeneous, i.e. in the region close to the horizontal and vertical axis, the angular resolution is coarser than in other regions. This problem remains even in the case of the much denser 72-neighborhood (Fig. 2.17b). In contrast, with continuous formulation using the fast marching method, arbitrarily high angular resolution of orientations can be represented at no extra computational cost.

2.4.4 Directional Information

In minimal path approaches using *directional information*, additional features and criteria are introduced to constrain the propagation of the wavefront. Usually these approaches rely on heuristics.

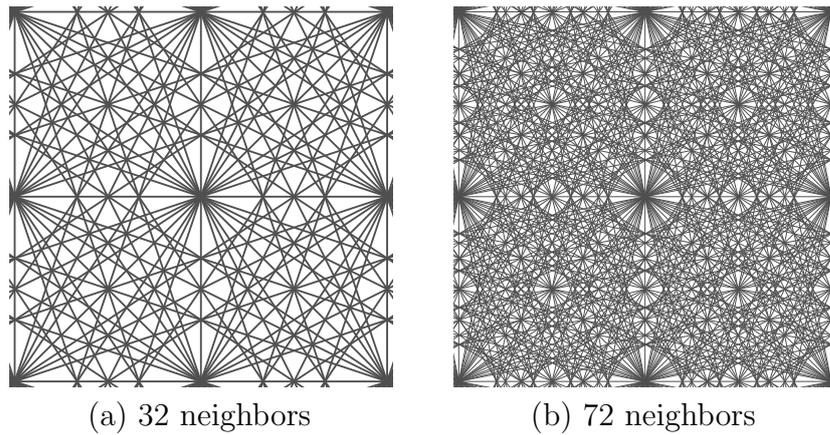


Figure 2.17: Neighborhoods in graphs for 2D images. (a) 32-neighborhood used in [56]. (b) 72-neighborhood (images generated using the software provided by [56], available at [157]).

In some of these approaches, the wave propagation can be stopped in undesired directions (e.g. [158, 159]). The main idea is based on the fact that the wavefront propagates faster inside line structures but slowly outside. In classical minimal path approaches, the wavefront propagates continuously in all directions, even in regions which are unlikely to be part of the line structure. This may cause unnecessary computation or short cuts. For example, Fig. 2.18 shows three stages of the propagation of a wavefront for the segmentation of branching blood vessels. The red voxels denotes regions in which the wavefront can propagate. This region is much larger than the actual blood vessels themselves. By ignoring regions which

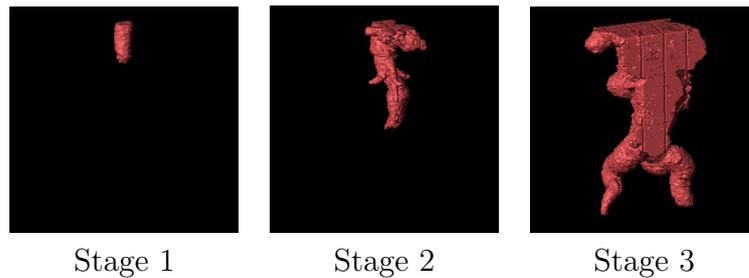


Figure 2.18: Three stages of the wave propagation for vessel segmentation (from [158]).

are unlikely to be part of the line structure, the wavefront can propagate faster towards the end point. Also, short cuts through the ignored regions can be avoided. At the time point t , by comparing the distance $\mathcal{U}_{\mathbf{x}_s}(\mathbf{v})$ of a voxel \mathbf{v} to the start point \mathbf{x}_s with the currently maximum distance $\mathcal{U}_{\max}(t)$ traveled by the wavefront, it can be heuristically determined if \mathbf{v} is inside or outside the line structure. In the latter case, the propagation through \mathbf{v} is *frozen*. This is illustrated in Fig. 2.19,

which shows the propagation at similar stages as in Fig. 2.18. The frozen region is shown using gray voxels. Apparently, only a small portion of the voxel remains unfrozen, and the wavefront propagates through a much smaller portion of the image than in Fig. 2.18.

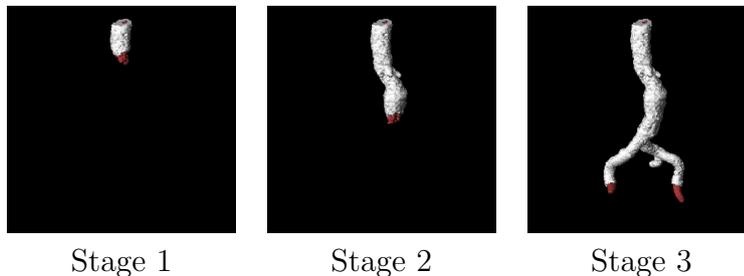


Figure 2.19: Three stages of the wave propagation for vessel segmentation. At the gray voxels the propagation is frozen, while at the red voxels the propagation continues (from [158]).

In some other approaches, the wave propagation is only allowed in certain directions. For example, in [45,160], an iterative approach based on locally optimal paths is used. In each iteration, a path must follow a specific direction d_0 , as illustrated in Fig. 2.20. At the position x_0 , a circular ROI of radius r_s is considered. Starting from x_0 , the circular sector around d_0 within an opening angle of 2α is the admissible region. Only paths which lie inside this region are allowed, while other directions are discarded. The path is optimal with respect to the local circular ROI, and the locally optimal paths from consecutive ROIs are concatenated to form the final result.

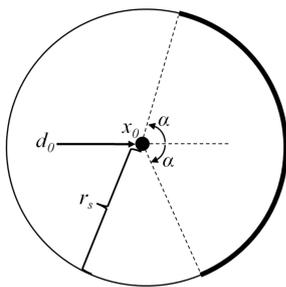


Figure 2.20: In each iteration, the locally optimal path starting from x_0 inside circular ROI is determined. Only paths within an opening angle of 2α are allowed. Other paths are discarded (from [160]).

In practice, these approach have shown better performance for certain applications. However, there are some drawbacks. From an optimization point of view, unlike the classical minimal path approaches, there does not exist an overall energy

function to measure the result in [45, 158, 159]. Furthermore, for some approaches (e.g., [45]), there is no guarantee that the end point can be always reached, since in the intermediate ROIs which do not contain the end point, the path can run into a direction which eventually does not lead to the end point.

2.4.5 Additional Keypoints

There also exist minimal path approaches which rely on *additional key points*. The key points can be inserted manually (e.g., [161]), which can greatly help to disambiguate. However, these approaches rely heavily on precise manual localization, so they are not suited for application which require automatic analysis. Furthermore, manually adding key points in 3D images is much more time consuming than in 2D images, and therefore these approaches are mainly limited to 2D images.

Key points can also be inserted automatically. For example, when the distance between the wavefront and the start point \mathbf{x}_s exceeds a certain threshold, a new key point \mathbf{v}_{key} can be inserted, which plays the same role as the original start point \mathbf{x}_s , i.e., the wavefront starts at both \mathbf{x}_s and \mathbf{v}_{key} (e.g., [162–164]). The main idea of [162] is illustrated in Fig. 2.21. For the original catheter tube image in Fig. 2.21a, the potential function is computed based on the Laplacian operator Fig. 2.21b. Starting from \mathbf{x}_s , which is shown as a magenta dot in the right lower corner in Fig. 2.21c, the wavefront propagates as in classical minimal path methods, until the distance from a certain pixel \mathbf{x}_T on the wavefront to \mathbf{x}_s reaches a threshold T . At \mathbf{x}_T , an additional key point is inserted, and the wavefront starts propagation there. Further key points are inserted in a similar manner, until the wavefront reaches the end point \mathbf{x}_e , which is shown as another magenta dot in the middle in Fig. 2.21a. By connecting each key point to the previous key point using gradient descent, the complete path can be obtained, as displayed in Fig. 2.21d.

This method is improved in [163] such that starting from only one single point, the complete tree structure can be segmented, i.e. the topological changes are taken into account, and the end points are determined automatically. For example, for the image in Fig. 2.22a, classical minimal path approaches require user-specified start and end points for each path, and yield a short cut in Fig. 2.22b. With the approach [162], the short cut can be avoided but all the start and end points still need to be specified (Fig. 2.22c). In contrast, in [163] only a single start point on the line structure is needed (Fig. 2.22d).

However, in these methods, only limited features have been used, such as the distance between the wavefront and the start point. More complex features in

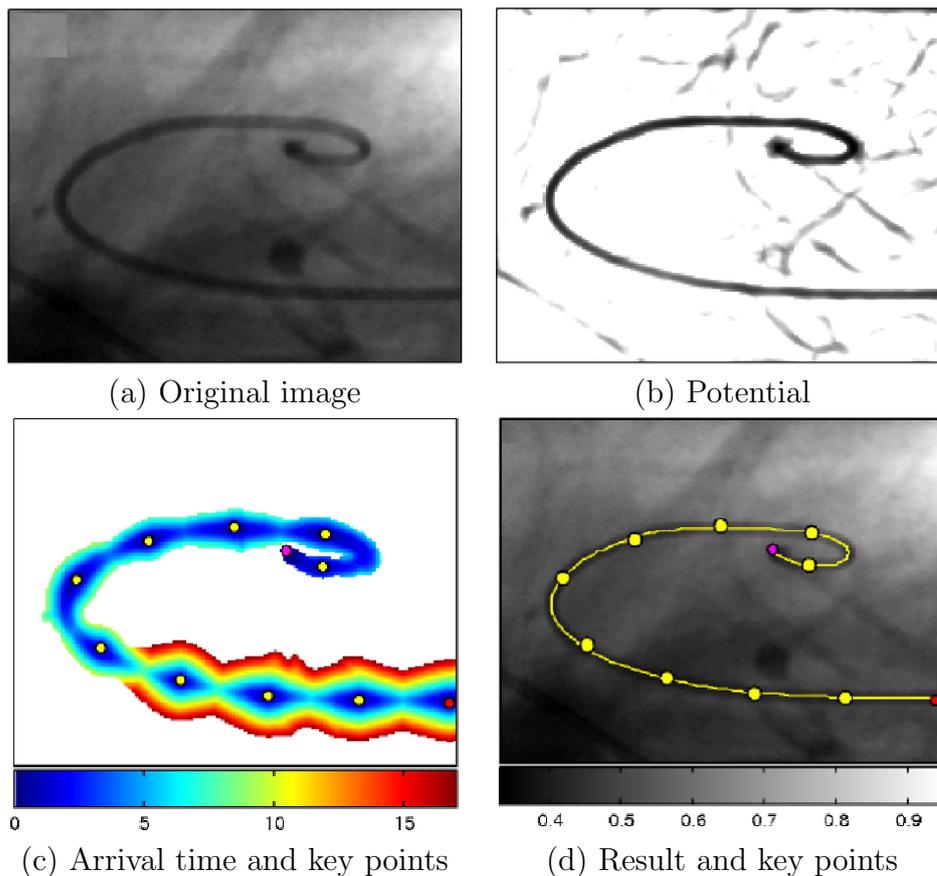


Figure 2.21: Minimal path approach with automatically inserted key points in [162]. (a) Original catheter tube image. (b) Potential based on the Laplacian. (c) Arrival time and the key points. The two magenta dots indicate the start and end points. (d) Final result, in which key points are connected to each other (from [162]).

terms of the path geometry, such as the curvature of the path, were not incorporated. Furthermore, most of these approaches (e.g., [161, 163, 164]) have been applied only to 2D images.

2.4.6 Dynamic Speed

The above described minimal path approaches use a *static* speed, i.e., the speed of the wavefront is computed *before* the propagation starts. In contrast, in [57–59, 165], the speed is updated *during* the wave propagation using position of the image points relative to the *normal* of the current wavefront.

An example is shown in Fig. 2.23. Suppose p is a point on the wavefront \mathcal{W} . When tracing back from \mathcal{W} to the start point, let w be the immediate predecessor of p . Since w is determined using gradient descent from p , the vector \vec{wp} points into

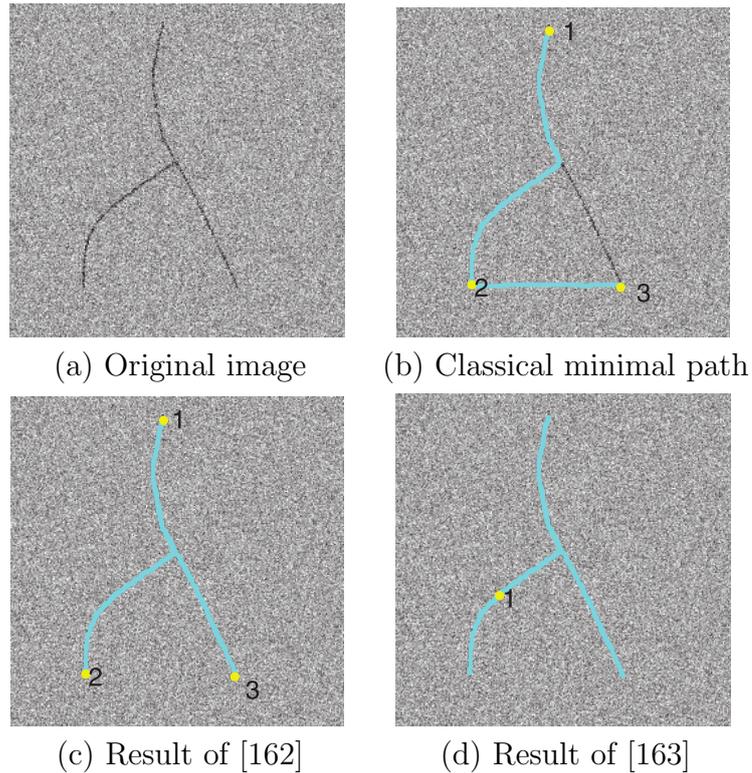


Figure 2.22: Minimal path approach with automatically inserted key points and only one user-specified point in [163]. (a) Original image. (b) Result of the classical minimal path approach. (c) Result of the approach with automatically inserted key points [162], where every start and end point must be specified by the user. (d) Result of the approach with automatically inserted key points [163], where the user only needs to specify one single point (from [163]).

the normal direction of \mathcal{W} . q and r are two points which have not yet been visited by \mathcal{W} . In [59], the speed at q and r depend on the angles γ_1 and γ_2 , respectively. In other words, the speed at an unvisited point depends on how strongly this point deviates from the normal direction of the wavefront. In contrast, in classical minimal path approaches, the speed at q and r is determined independently of \vec{w}_p . Fig. 2.24 shows the result of this approach for the segmentation of an actin filament in a microscopic image. To achieve a similar result with a classical minimal path approach [41], several additional key points need to be inserted by the user.

However, previous approaches using dynamic speed rely only on the normal of the wavefront. The normal is a property of a single point, but many important appearance and geometric properties of the path, which require several points to determine, cannot be computed using the normal alone. Furthermore, the approaches [58, 59, 165] use discrete formulation, so they suffer from discretization errors.

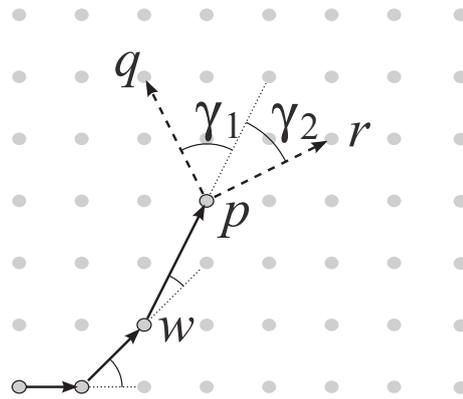
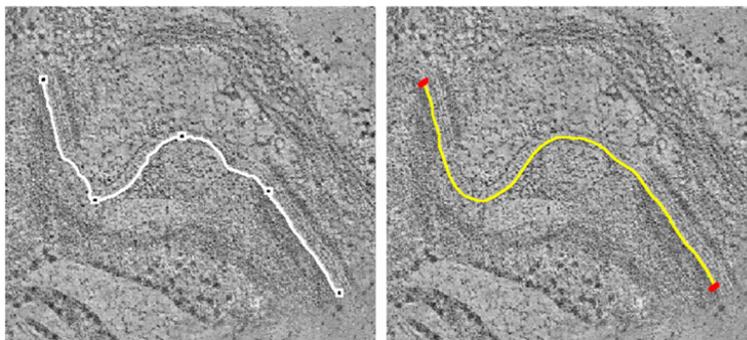


Figure 2.23: Dynamic speed (from [59].) The wavefront \mathcal{W} is propagating through the point p . The normal of \mathcal{W} shows in the direction $\vec{w}p$. The speed at the neighboring points q and r depends on the angles γ_1 and γ_2 , respectively.



(a) Classical minimal path with key points by user

(b) Result of [59]

Figure 2.24: Result of classical minimal path approach [41] with user interaction and an approach with dynamic speed [59]. An actin filament in a microscopic image is segmented. (a) Result of a classical minimal path approach [41]. In addition to the start and end points, several key points have been inserted by the user. (b) Result of the approach in [59] (from [59]).

2.4.7 Integer Programming

Some recent methods (e.g., [166–168]) assume that the segments of the optimal path form a subset of an *over-connected* graph, and *integer programming* is used to remove unnecessary segments from the graph, so that the final result consists of the remaining segments. Each segment is either completely included or completely excluded from the final result. Usually, the integer programming problems for these approaches are solved using commercial optimization solvers such as Cplex [169], Gurobi [170], or SCIP [171]. As an example, the main steps of the approach [167] are shown in Fig. 2.25. For the satellite image in Fig. 2.25a, segments with high probability to be part of the final result are identified using an appear-

ance model. These segments, shown as green lines in Fig. 2.25b, are connected with each other at nodes, displayed as red dots in Fig. 2.25b. The segments and nodes form an over-connected graph, on which further constraints are imposed to eliminate unnecessary segments. These constraints are formulated and optimized using integer programming to obtain the final result in Fig. 2.25c.

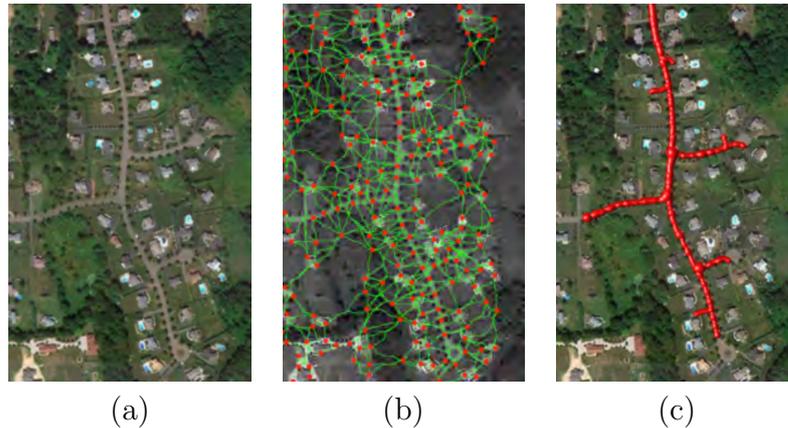


Figure 2.25: Segmentation result of [167]. (a) Input image. (b) Over-connected graph. (c) Final result (from [167]).

One disadvantage is that integer programming is used discriminatively, i.e. it is only used to decide which segments should be removed from the over-connected graph, but the graph itself is still constructed using standard methods. Since the segments cannot be altered once the graph is generated, these approaches are generally prone to errors in the construction of the over-connected graph. For example, in [166], thresholding followed by skeletonization is used, and in [167] a classical minimal path method is employed. In both cases, the result could be improved by more accurate methods. Furthermore, further features such as geometric properties *inside* individual segments were not exploited.

2.5 Hybrid Methods

There also exist approaches which combine different methods to segment line structures. Such approaches have mainly been used for the segmentation of blood vessels. By combining two methods with complementary properties, the topology and geometry of the line structures can be better determined than using a single method. These approaches can be roughly divided into two classes, which are reviewed below.

2.5.1 Geometry Refinement

In these approaches, both of the combined methods operate in the same image region. The first method creates a coarse segmentation, while the second method refines the geometry (e.g., [93, 172]).

For example, in [172], a minimal path method is combined with graph cuts. For the vessel segmentation in a 2D retinal image with given start and end points in Fig. 2.26a, the minimal path method is first used to obtain a path inside the vessel (not necessarily the centerline) (Fig. 2.26b). Without lifting the domain (described in Sect. 2.4.2), this method is computationally efficient but the geometry of the vessel is not determined. In contrast, with graph cuts the geometry can be obtained, but graph cuts alone are not suited for the segmentation of long and thin structures like retinal vessels due to the shrinking bias problem, as described above in Sect. 2.3. By using the result of the minimal path method as a hard constraint for graph cuts, the shrinking bias can be alleviated, yielding the final result in Fig. 2.26c.

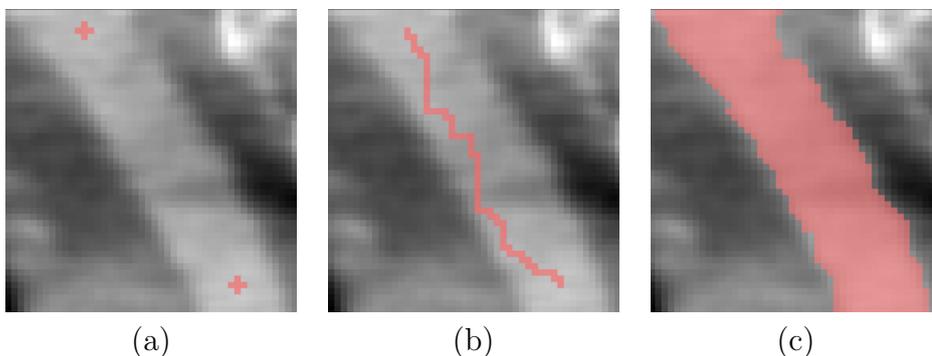


Figure 2.26: Segmentation of a vessel in a 2D retinal image [172]. (a) Retinal vessel to be segmented. The start and end points are shown as red crosses. (b) In the first step, a path inside the vessel is determined using a minimal path method. (c) In the second step, the geometry of the vessel is obtained using graph cuts (from [172]).

Similarly, the hybrid approach in [93] also employs another method to alleviate the shrinking bias of graph cuts. First, a parametric cylindrical model is used to obtain a coarse segmentation of the liver vessels in 3D CT images. After that, the coarse segmentation is used as hard constraint for graph cuts to get a more exact geometry, as shown in Fig. 2.27.

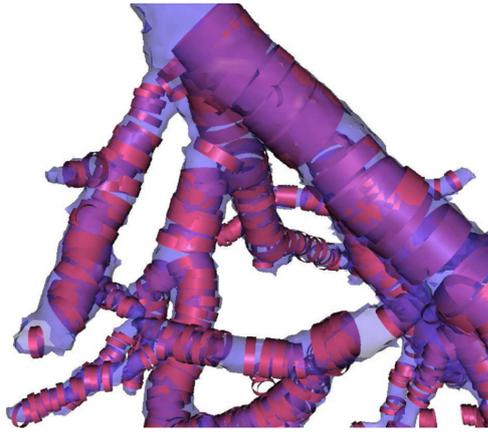


Figure 2.27: Segmentation of liver vessels in a 3D CT image [93]. The result of fitting a parametric cylindrical model (purple cylinders) are used as a hard constraint for a more exact segmentation based on graph cuts (blue shape) (from [93]).

2.5.2 Structure Completion

In these approaches, the combined methods operate in different image regions. The first method creates a segmentation of most parts of the structure, while the second method completes the missing parts which was not segmented by the first method (e.g., [58, 114, 173]).

In medical images, often thin structures are difficult to segment, while thick structures can be segmented with a much higher accuracy. Consequently, a common segmentation strategy for these images is to first segment the thick structures, and then use them to segment thin structures. For example, for the segmentation of vessel networks in 2D retinal images, the approach in [58] uses pixel classification to segment the thick structures (see the white segments in Fig. 2.28). However, thin structures cannot be segmented well using pixel classification, and consequently there are gaps between thick structures. These gaps are completed in a further step using a minimal path method (see the red segments in Fig. 2.28).

However, even though hybrid methods often achieve better results in terms of geometry refinement and structure completion, each step in these hybrid segmentation approaches still suffers from the disadvantages of the respective method reviewed earlier. For example, by combining graph cuts with other methods, the shrinking bias can be reduced but not methodologically avoided.

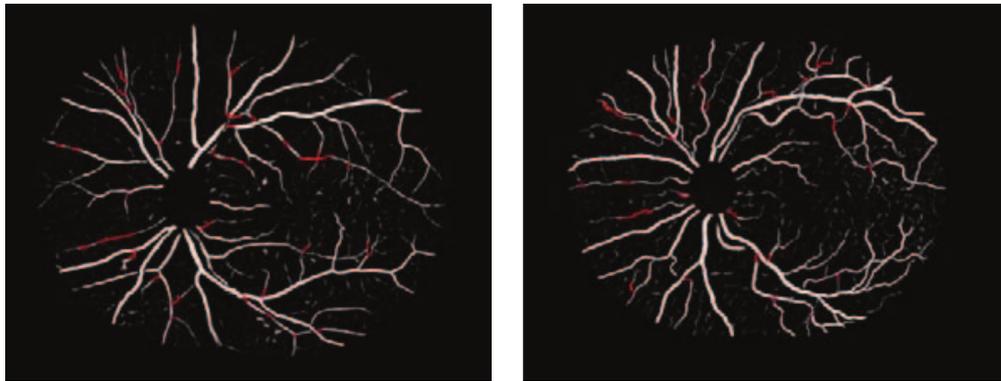


Figure 2.28: Segmentation of vessel networks in two 2D retinal images [58]. The white vessel parts are results using pixel classification, while the red vessel parts are results of a minimal path approach (from [58]).

Chapter 3

Background of Minimal Path Methods

As described in Chapters 1 and 2, the analysis of line structures plays a crucial role in many computer vision applications, and minimal path methods are often used. The aim of this thesis is the development of minimal path methods for analyzing line structures. In this chapter, the theoretical background of minimal path methods is reviewed.

3.1 Introduction to Minimal Path Methods

Minimal path methods formulate the segmentation of line structures as the determination of *minimal paths*, also referred to as *shortest paths* or *geodesics*. Using a given pair of start point \mathbf{x}_s and end point \mathbf{x}_e , the line structure is segmented in the form of a path between \mathbf{x}_s and \mathbf{x}_e , such that a certain *energy function* is minimized. Depending on the application, \mathbf{x}_s and \mathbf{x}_e can be specified manually or automatically.

3.1.1 Interpretation as Wave Propagation

Usually, minimal path methods can be intuitively interpreted as the composition of two consecutive steps: Wave propagation and backtracing. These steps are illustrated for the case of 2D images in Fig. 3.1. In the first step (i.e., wave propagation), a wavefront is initialized at the start point \mathbf{x}_s , and starts to propagate outwards (Fig. 3.1a). The speed $\mathcal{F}(\mathbf{x})$ of the wavefront at each pixel \mathbf{x} is computed using image features, which are application specific. Let $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ denote the arrival time of the wavefront at a pixel \mathbf{x} . Each pixel \mathbf{x} can have one out of three

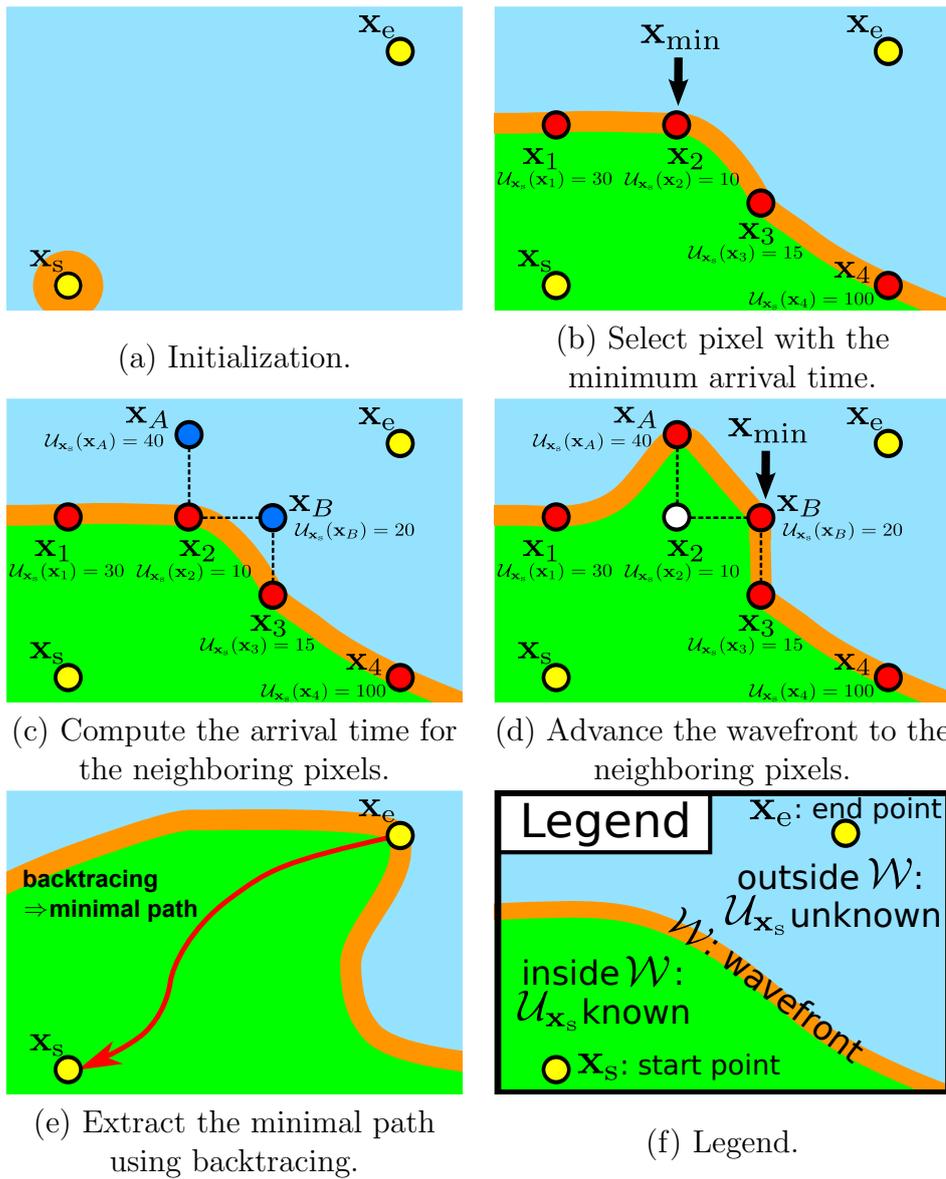


Figure 3.1: Steps of minimal path methods. The wavefront is initialized around the start point \mathbf{x}_s (a), and starts to propagate. After several steps, the wavefront has the shape as shown in (b), and the pixel on the wavefront with the minimum arrival time (i.e., \mathbf{x}_2) is selected as \mathbf{x}_{\min} . In (c), the arrival times of pixels \mathbf{x}_A and \mathbf{x}_B , which are neighbors of \mathbf{x}_2 , are computed. Then the wavefront advances to \mathbf{x}_A and \mathbf{x}_B (d). This is one step of wave propagation. The propagation is performed repeatedly until the end point \mathbf{x}_e is reached. The actual minimal path is extracted using backtracing, starting from \mathbf{x}_e (e).

labels: Alive (l_A), Trial (l_T), and Far (l_F). Then the image plane is partitioned by the wavefront into three regions, corresponding to the labels of the pixels: The region *inside the wavefront* is referred to as \mathcal{R}_A , and pixels inside this region are labeled with l_A . For every pixel in \mathcal{R}_A , $U_{\mathbf{x}_s}$ is finally determined, and it will not

change any more. The region *on the wavefront* is referred to as \mathcal{R}_T , and pixels inside this region are labeled with l_T . For every pixel in \mathcal{R}_T , $\mathcal{U}_{\mathbf{x}_s}$ is temporarily determined, and subject to changes due to the wave propagation in future. The region *outside the wavefront* is referred to as \mathcal{R}_F , and the label of pixels inside this region is l_F . For every pixel in this region, $\mathcal{U}_{\mathbf{x}_s}$ is not yet determined, the values in this region are initialized to infinity. In Fig. 3.1, \mathcal{R}_A , \mathcal{R}_T , and \mathcal{R}_F are illustrated as green, orange, and blue regions, respectively. At each time point, the pixel on the wavefront (i.e., in the Trial region \mathcal{R}_T) with the minimum arrival time \mathbf{x}_{\min} is selected, and the wavefront should advance at \mathbf{x}_{\min} (Fig. 3.1b). To determine the direction of advancement, first the arrival time at the neighbors of \mathbf{x}_{\min} in the Far region \mathcal{R}_F are computed (see \mathbf{x}_A and \mathbf{x}_B in Fig. 3.1c). After that, the label of \mathbf{x}_{\min} is changed from l_T to l_A , and the labels of \mathbf{x}_A and \mathbf{x}_B are changed from l_F to l_T . In other words, the wavefront advances to include \mathbf{x}_A and \mathbf{x}_B in the new Trial region \mathcal{R}_T , and \mathbf{x}_{\min} in the extended Alive region \mathcal{R}_A (Fig. 3.1d). Once $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_e)$ is computed for the end point \mathbf{x}_e , i.e., the wavefront reaches \mathbf{x}_e , the minimal path $\gamma_{\mathbf{x}_s, \mathbf{x}_e}^*$ can be extracted by starting from \mathbf{x}_e and iteratively determining the *predecessor* of the current position to trace back to \mathbf{x}_s (Fig. 3.1e). The computation of the predecessor depends on the specific minimal path methods, i.e. fast marching method or Dijkstra’s algorithm.

Definition 3.1 (Predecessor Operator). *For each image point \mathbf{x} , its predecessor $\text{pre}(\mathbf{x})$ is defined to be another image point \mathbf{x}_1 in the neighborhood of \mathbf{x} such that when starting from \mathbf{x} to trace back to \mathbf{x}_s , then \mathbf{x}_1 is the first image point after \mathbf{x} . The predecessor of \mathbf{x} can be computed using the operator $\mathbf{x}_{\text{pre}} = \text{pre}(\mathbf{x})$. For the fast marching method, $\text{pre}(\mathbf{x}) = \mathbf{x} - \tau \nabla \mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$, where τ is the step length, and ∇ is the gradient operator. For Dijkstra’s algorithm, the predecessor of each \mathbf{x} is stored explicitly.*

3.1.2 Advantages of Minimal Path Approaches

Compared to many segmentation approaches, for example, level sets or graph cuts, which were reviewed in Chapter 2, minimal path approaches have many desirable features, which make these approaches especially suitable for the segmentation and analysis of line structures.

From a theoretical point of view, the minimal paths are usually formulated as an energy minimization problem, for which the *global optimum* can be found efficiently. Unlike the level set method, in which often a pixel needs to be visited more than once during the deformation of the level set function, in minimal path

approaches each pixel is visited at most once, i.e. once the pixel \mathbf{x} is changed from the Trial region \mathcal{R}_T to the Alive region \mathcal{R}_A , thereby changing its label from l_T to l_A , the arrival time $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ is finally determined and the label and $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ will not be changed any more. Intuitively, the wavefront in minimal path approaches propagates only outwards and never shrinks.

From a practical point of view, the result of the minimal path approaches is always a line which connects the start point \mathbf{x}_s to the end point \mathbf{x}_e , and the coordinates of each point on the path is known explicitly. Consequently, other properties of the result, such as the curvature or torsion, can be computed easily. In contrast, in level sets and graph cuts, it is possible that the binary segmentation consists of more than one connected components, so that the start point \mathbf{x}_s is not connected to the end point \mathbf{x}_e , since in these approaches the topology of the result is usually subject to change. Moreover, in binary segmentation approaches such as level sets or graph cuts, the result needs to be further analyzed using, for example, thinning, to determine the coordinates of lines.

3.1.3 Algorithmic Taxonomy for Minimal Path Methods

The implementation of minimal path methods can be grouped into two major classes: *Discrete* methods are based on the Dijkstra's algorithm [174–176], which is a graph theoretical algorithm, while *continuous* methods are based on the fast marching method [41, 147, 177], which computes the numerical solution of partial differential equations (PDE). These two classes are very closely related, and they consist of similar steps. Both classes are widely used, and usually the global optimum of the energy function can be found in both cases.

In this thesis, the introduced methods are applicable to both the Dijkstra's algorithm and the fast marching method. The focus in our work is on the fast marching method, because it has several advantages compared to the Dijkstra's algorithm. First, due to the continuous formulation there is no metrication error, which is inherent for discrete algorithms using graphs. Consequently, the line structures can be segmented with subpixel accuracy. Second, since the predecessor is computed using the gradient operator as defined in Definition 3.1 above, there is no need to store the predecessor explicitly, resulting in less memory consumption than the Dijkstra's algorithm. Although the Dijkstra's algorithm can be adapted more easily to an anisotropic speed function, the algorithms developed in this thesis require only isotropic speed functions. Below, the theoretical background and the solution method of the fast marching method are reviewed.

3.2 Fast Marching Method

Given an n -dimensional image $g(\mathbf{x}) : \Omega \rightarrow \mathbb{R}^n$ as well as the start and end points $\mathbf{x}_s, \mathbf{x}_e \in \Omega$, the goal of *the fast marching method* is to find the path $\gamma_{\mathbf{x}_s, \mathbf{x}_e}^*$ such that the *energy* E is minimized:

Energy function of continuous minimal path approaches

$$\gamma_{\mathbf{x}_s, \mathbf{x}_e}^* := \arg \min_{\gamma} E(\gamma) = \arg \min_{\gamma} \int_{\gamma} (\mathcal{P}(\gamma(s)) + w) ds, \quad (3.1)$$

where γ is a *path*, \mathcal{P} is the *potential* derived from g , w is a regularization constant, and s is the arc length parameter. Let $\mathcal{A}_{\mathbf{x}_1, \mathbf{x}_2}$ denote the set of all paths γ connecting two given points \mathbf{x}_1 and \mathbf{x}_2 . The energy of the minimal path between \mathbf{x}_s and an arbitrary point \mathbf{x} in Ω is captured by $\mathcal{U}_{\mathbf{x}_s}$:

$$\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}) := \min_{\gamma \in \mathcal{A}_{\mathbf{x}_s, \mathbf{x}}} E(\gamma). \quad (3.2)$$

$\mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ can be interpreted as the arrival time of a wavefront \mathcal{W} emanating from \mathbf{x}_s and propagating outwards with the speed

$$\mathcal{F}(\mathbf{x}) = \frac{1}{\mathcal{P}(\mathbf{x})}. \quad (3.3)$$

To compute $\mathcal{U}_{\mathbf{x}_s}$, the numerically efficient upwind scheme [178] is employed in the fast marching method. Once $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_e)$ is computed, the minimal path $\gamma_{\mathbf{x}_s, \mathbf{x}_e}^*$ can be extracted by starting from \mathbf{x}_e and iteratively determining the *predecessor* of the current position to trace back to \mathbf{x}_s . Alternatively, $\mathcal{U}_{\mathbf{x}_s}$ can also be viewed as the solution of the Eikonal equation [41, 177], which is a non-linear partial differential equation:

$$\|\nabla \mathcal{U}_{\mathbf{x}_s}(\mathbf{x})\| = \mathcal{P}(\mathbf{x}) + w. \quad (3.4)$$

The fast marching method is usually implemented in two steps. In the first step, the wavefront propagates outwards to compute the arrival map $\mathcal{U}_{\mathbf{x}_s}$. In the second step, the line structure is computed based on $\mathcal{U}_{\mathbf{x}_s}$. These two steps are described in detail below.

3.2.1 Propagation of the Wavefront

The fast marching method using isotropic speed is given in Algorithm 3.1, in which a wavefront is propagated from the start point \mathbf{x}_s to the end point \mathbf{x}_e . For a point \mathbf{x} , $\mathcal{L}(\mathbf{x})$ denotes the label at \mathbf{x} , and $\mathcal{N}(\mathbf{x})$ denotes the set of direct neighbors of \mathbf{x} on a regular image grid.

Algorithm 3.1: Fast marching method

```

Input: Start point  $\mathbf{x}_s$ , end point  $\mathbf{x}_e$ , speed function  $\mathcal{F}$ 
Output: Action map  $\mathcal{U}_{\mathbf{x}_s}$ 
1 for each pixel coordinate  $\mathbf{x}$  do                                     // Initialization
2   if  $\mathbf{x} = \mathbf{x}_s$  then
3      $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}) \leftarrow 0$ ;  $\mathcal{L}(\mathbf{x}) \leftarrow l_T$ ;
4   else
5      $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}) \leftarrow \infty$ ;  $\mathcal{L}(\mathbf{x}) \leftarrow l_F$ ;
6 repeat                                                                // Main loop: Wave propagation
7    $\mathbf{x}_{\min} \leftarrow \arg \min_{\mathbf{x} \in \mathcal{R}_T} \mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ ;
8    $\mathcal{L}(\mathbf{x}_{\min}) \leftarrow l_A$ ;
9   for  $\mathbf{x}_n \in \mathcal{N}(\mathbf{x}_{\min}) \cap (\mathcal{R}_F \cup \mathcal{R}_T)$  do
10     $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_n) \leftarrow \text{UpwindScheme}(\mathbf{x}_n, \mathcal{F})$ ;
11    if  $\mathcal{L}(\mathbf{x}_n) = l_F$  then
12       $\mathcal{L}(\mathbf{x}_n) \leftarrow l_T$ ;
13 until  $\mathcal{L}(\mathbf{x}_e) = l_A$ ;          // End point reached  $\Rightarrow$  stop propagation
14 return  $\mathcal{U}_{\mathbf{x}_s}$ ;

```

First, the algorithm is initialized: The arrival time $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ is set to infinity and the label $\mathcal{L}(\mathbf{x})$ set to l_F for all image points \mathbf{x} , except for the start point \mathbf{x}_s , where $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_s) = 0$ and $\mathcal{L}(\mathbf{x}_s) = l_T$. Second, the wavefront propagates from \mathbf{x}_s towards \mathbf{x}_e by advancing one image point in each step. From the region \mathcal{R}_T , in which the arrival time for each image point is temporarily determined, the image point with minimum arrival time \mathbf{x}_{\min} selected as a candidate to be added to the \mathcal{R}_A and its label is changed to l_A (lines 7 and 8). After that, $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ is updated for all neighbors \mathbf{x}_n of \mathbf{x}_{\min} which are in \mathcal{R}_F or \mathcal{R}_T (line 10 to line 12): For the neighbor \mathbf{x}_n of the image point \mathbf{x}_{\min} , which has the currently minimum arrival time among all image points in the region \mathcal{R}_T , the function `UpwindScheme` determines the arrival time $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_n)$ using the *upwind scheme* (line 10). This is a finite difference numerical scheme used widely for the fast marching method [102]. Finally, since the arrival time of \mathbf{x}_n has been updated at least once, its label is changed to l_T (line 12).

A crucial data structure is a *priority queue* which keeps track of all image points in the region \mathcal{R}_T . It guarantees that \mathbf{x}_{\min} can be found in $\mathcal{O}(\log N_q)$ time

3.2. Fast Marching Method

(line 7), where N_q is the number of image points in \mathcal{R}_T .

3.2.2 Extracting the Line Structure

By the determination of $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ for all image points \mathbf{x} , the actual path is extracted using gradient descent, which starts from the end point \mathbf{x}_e , iteratively finds the predecessor of the current point, and terminates when the start point \mathbf{x}_s is reached. According to Definition 3.1, the predecessor is computed using the gradient operator, which can be implemented in a discrete or continuous manner. The discrete and faster implementation is given in Algorithm 3.2 below, and the continuous and more accurate implementation is given in Algorithm 3.3.

Algorithm 3.2: Backtracing of the fast marching method using discrete neighborhood.

Input: Minimal action map $\mathcal{U}_{\mathbf{x}_s}$, end point \mathbf{x}_e
Output: Path γ

```

1  $\gamma \leftarrow \text{CreateEmptyPath}();$ 
2  $\text{AppendPointToPath}(\mathbf{x}_e, \gamma);$ 
3  $\mathbf{x} \leftarrow \mathbf{x}_e;$ 
4 while  $\mathbf{x} \neq \mathbf{x}_s$  do           // Backtracing using discrete neighborhood
5      $t_{\min} \leftarrow \infty;$        // Temporary minimum arrival time in  $\mathcal{N}(\mathbf{x})$ 
6     for  $\mathbf{x}_1 \in \mathcal{N}(\mathbf{x})$  do
7         if  $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_1) < t_{\min}$  then
8              $\mathbf{x}_{\min} \leftarrow \mathbf{x}_1;$ 
9              $t_{\min} \leftarrow \mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_1);$ 
10     $\mathbf{x} \leftarrow \mathbf{x}_{\min};$ 
11     $\text{AppendPointToPath}(\mathbf{x}, \gamma);$ 
12 return  $\gamma;$ 

```

The Algorithm 3.2 starts with an empty path γ (line 1) and adds the end point \mathbf{x}_e to γ (line 2). Then, a neighbor of \mathbf{x}_e is selected as \mathbf{x}_{\min} such that the arrival time $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_{\min})$ is minimal among all neighbors of \mathbf{x}_e . \mathbf{x}_{\min} is appended to γ , and in turn its neighbor with the minimum arrival time is to be found. This procedure is repeated until the start point \mathbf{x}_s is reached. Since the points of the path are limited to image points on the regular grid, the path γ computed using this backtracing method has pixel accuracy.

To obtain subpixel accuracy at the cost of a higher computational effort, Algorithm 3.3 can be used. In general, the path γ is extracted by solving the ordinary differential equation:

$$\frac{d\gamma(t)}{dt} = -\nabla \mathcal{U}_{\mathbf{x}_s}(\gamma(t)). \quad (3.5)$$

Algorithm 3.3: Backtracing of the fast marching method using Runge-Kutta method.

Input: Minimal action map $\mathcal{U}_{\mathbf{x}_s}$, end point \mathbf{x}_e
Output: Path γ

```

1  $\gamma \leftarrow \text{CreateEmptyPath}();$ 
2  $\text{AppendPointToPath}(\mathbf{x}_e, \gamma);$ 
3  $\mathbf{x} \leftarrow \mathbf{x}_e;$ 
4 while  $\|\mathbf{x} - \mathbf{x}_s\| > \epsilon$  do           // Backtracing using Runge-Kutta method
5    $\Delta\mathbf{x}_1 \leftarrow \tau \cdot \nabla\mathcal{U}_{\mathbf{x}_s}(\mathbf{x});$ 
6    $\Delta\mathbf{x}_2 \leftarrow \tau \cdot \nabla\mathcal{U}_{\mathbf{x}_s}(\mathbf{x} - 0.5 \cdot \Delta\mathbf{x}_1);$ 
7    $\Delta\mathbf{x}_3 \leftarrow \tau \cdot \nabla\mathcal{U}_{\mathbf{x}_s}(\mathbf{x} - 0.5 \cdot \Delta\mathbf{x}_2);$ 
8    $\Delta\mathbf{x}_4 \leftarrow \tau \cdot \nabla\mathcal{U}_{\mathbf{x}_s}(\mathbf{x} - \Delta\mathbf{x}_3);$ 
9    $\Delta\mathbf{x} = \frac{1}{6} \cdot (\Delta\mathbf{x}_1 + 2 \cdot \Delta\mathbf{x}_2 + 2 \cdot \Delta\mathbf{x}_3 + \Delta\mathbf{x}_4);$ 
10   $\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x};$ 
11  $\text{AppendPointToPath}(\mathbf{x}_s, \gamma);$ 
12 return  $\gamma;$ 
```

Equation (3.5) is usually solved numerically using variants of the Runge-Kutta method, or its simplified versions, for example, the Heun's method. In Algorithm 3.3 the classical Runge-Kutta method using 4 points is implemented. Starting with an empty path γ , first the end point \mathbf{x}_e is appended to γ . Then repeatedly the next point of the path is found by adding an offset to the current point, computed as the weighted sum of four intermediate offsets (line 5 to line 9), until the distance between the current point \mathbf{x} and the start point \mathbf{x}_s is less than a small positive constant ϵ . $\Delta\mathbf{x}_1, \Delta\mathbf{x}_2, \Delta\mathbf{x}_3$ and $\Delta\mathbf{x}_4$ denote the four components used to compute the increment $\Delta\mathbf{x}$. In contrast to Algorithm 3.2, the points on the path are not limited to points on the regular image grid, i.e., the result γ has subpixel accuracy.

3.3 Problems of Classical Minimal Path Approaches

The Dijkstra's algorithm and the fast marching method are often referred to as *classical* minimal path approaches. These approaches have difficulties when dealing with complex structures. In the following, we demonstrate this problem exemplarily using the fast marching method.

Fig. 3.2 shows the several stages of the wave propagation using an image containing two line structures which cross each other twice. One of the structures is a vertical line, and the second one is a U-shaped curve open to the right. The start and end points are set both inside the U-shaped structure, which should be segmented. Both structures have the same radius and intensity values, so the speed in

3.3. Problems of Classical Minimal Path Approaches

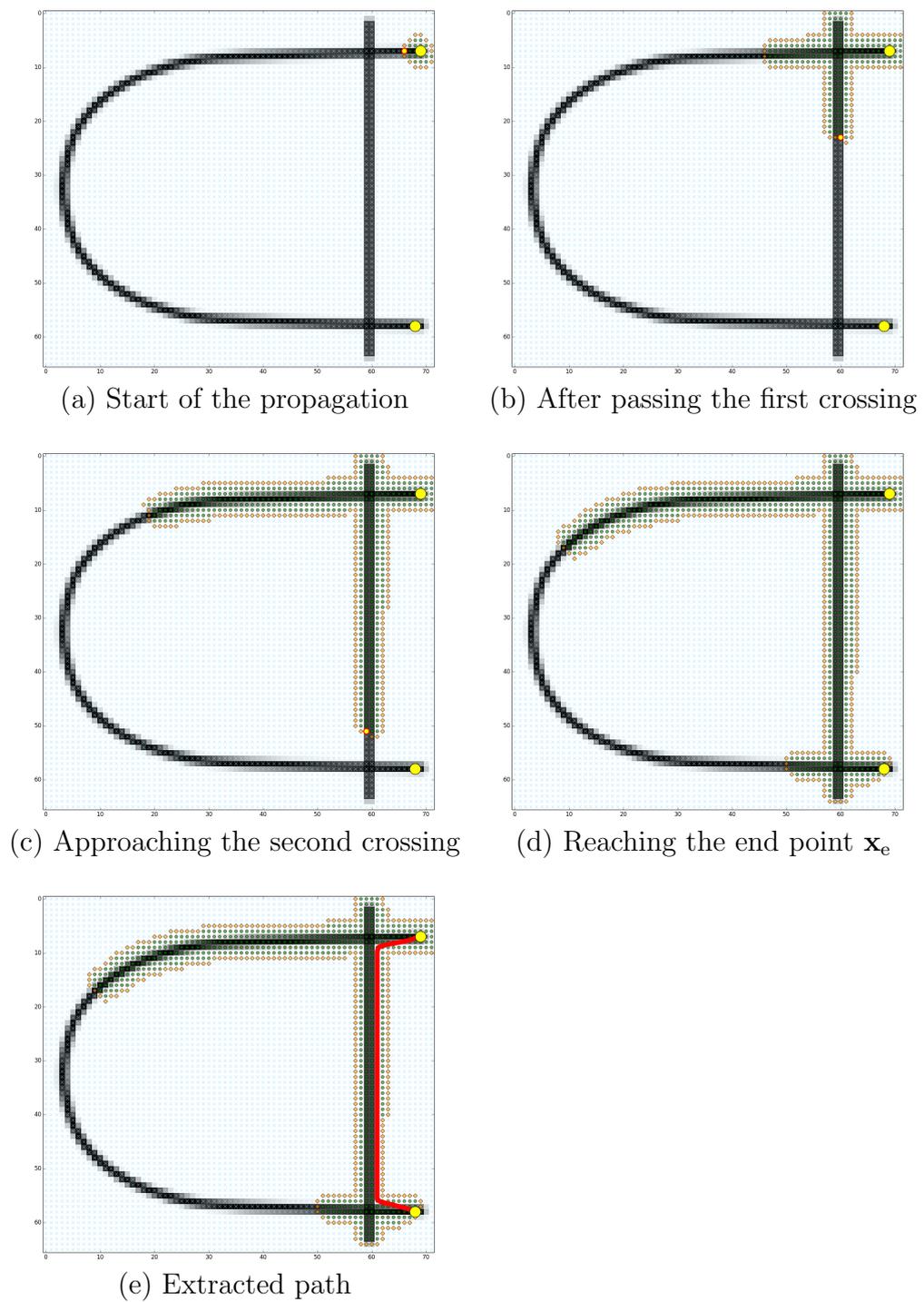


Figure 3.2: Problem of classical minimal path approaches. The Alive region \mathcal{R}_A is shown using green dots, the Trial region \mathcal{R}_T , i.e., the wavefront \mathcal{W} , is shown using orange diamonds, and the Far region \mathcal{R}_F is shown using blue crosses. The yellow circle with red boundary indicates the pixel \mathbf{x}_{\min} in \mathcal{R}_T with the minimum arrival time. (a) The wavefront starts at \mathbf{x}_s . (b) The wavefront passed the first crossing. (c) The wavefront approaches the second crossing. (d) The wavefront reaches the end point \mathbf{x}_e , the propagation terminates. (e) The minimal path (red line) is extracted using backtracing.

both line structures is roughly the same at each point along their centerlines. The Alive, Trial, and Far regions are shown as green dots, orange diamonds, and blue crosses, respectively. The pixel \mathbf{x}_{\min} in \mathcal{R}_T with the minimum arrival time is indicated as a yellow circle with red boundary. The wavefront \mathcal{W} starts to propagate at the start point \mathbf{x}_s (Fig. 3.2a). After passing the first crossing, the wavefront \mathcal{W} follows two different branches, propagates further at the same speed along both branches (Fig. 3.2b), and approaches the second crossing (Fig. 3.2c). Finally, after passing the second crossing, \mathcal{W} reaches the end point \mathbf{x}_e via the branch which is shorter than the other one, and the propagation terminates (Fig. 3.2d). Using backtracing starting from \mathbf{x}_e , the minimal path can be extracted (Fig. 3.2e).

Obviously, the path in (Fig. 3.2e) is a *short cut*, i.e., it does not correspond to the correct structure to segment, which contains the start point \mathbf{x}_s and end point \mathbf{x}_e . This is because in classical minimal path approaches, the speed is derived for each image point individually, and no *context information* is used. In the approaches presented in this thesis, we use different ways to employ context information so that short cut problems can be avoided, and the robustness of the minimal path approaches can be increased.

Chapter 4

Direction-Preserving Minimal Path Methods

In this chapter, we present a novel hybrid approach for automatic 3D segmentation and quantification of high-resolution 7 Tesla magnetic resonance angiography (MRA) images of the human cerebral vasculature. Our approach consists of two main steps. First, a 3D model-based approach is used to segment and quantify thick vessels and most parts of thin vessels. Second, remaining vessel gaps of the first step in low-contrast and noisy regions are completed using a 3D minimal path approach, which exploits directional information. We present two novel minimal path approaches: The first is an explicit approach based on energy minimization using probabilistic sampling, and the second is an implicit approach based on fast marching with anisotropic directional prior. We conducted an extensive evaluation with over 2300 3D synthetic images and 40 real 3D 7 Tesla MRA images. Quantitative and qualitative evaluation shows that our approach achieves superior results compared to a previous minimal path approach. Furthermore, our approach was successfully used in two clinical studies on stroke and vascular dementia. The work in this chapter was published in [37–40].

This chapter is organized as follows. In Sect. 4.1 we give an overview of our hybrid 3D vessel segmentation approach. Then, we describe the model-based approach (Sect. 4.2) and the two minimal path approaches (Sect. 4.3 and 4.4). A related iterative method for 2D vessel segmentation is presented in Sect. 4.5. Experimental results are presented in Chapter 5.

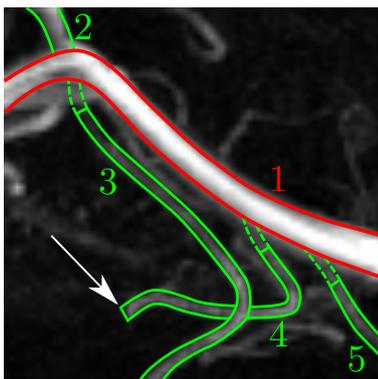


Figure 4.1: MIP of a region of a 3D 7T MRA image of the brain. For one thick vessel (red) and four thin vessels (green), the segmentation results of the model-based approach (first step, solid contours) and a minimal path approach (second step, dashed contours) are shown.

4.1 Overview of the Hybrid 3D Vessel Segmentation Approach

Our hybrid approach for 3D segmentation and quantification of vessels from 7T MRA images consists of two main steps. In the first step, thick vessels and most thin vessels are segmented using a 3D *model-based approach*: A parametric intensity model is fitted to the 3D image and vessel features such as the radius and the direction of vessel segments are determined (see Sect. 4.2 below). However, in low-contrast and highly noisy regions and at vessel branches, the parametric intensity model cannot always be fitted well, leading to gaps in these regions. Therefore, in the second step, the gaps are automatically completed using two novel 3D *minimal path approaches*. One minimal path approach is based on probabilistic sampling (described in Sect. 4.3 below) and the other is based on fast marching with anisotropic directional prior (Sect. 4.4). Both approaches incorporate direction information determined by the model-based approach.

Our hybrid approach is illustrated using a real 7T MRA image as shown in Fig. 4.1. Vessel 1 (red) is a thick vessel, with vessels 2-5 (green) as its branches. While vessels 1 and 2 are segmented correctly, vessels 3, 4, and 5 are not correctly connected to vessel 1 because of low image contrast. In our approach, we determine the centerlines for the gaps and assume that the radius of the vessels in the completed gaps is constant and equal to the radius of the last segment quantified by the model-based approach. Our goal is to find a gap completing path (dashed contours), while avoiding wrong connections. For example, in Fig. 4.1 one end of vessel 4 (marked with a white arrow) is not connected to any vessel, which is

correct since there is no thick vessel close to it.

4.2 3D Model-Based Segmentation and Quantification Approach

4.2.1 3D Parametric Intensity Model

For segmentation of cerebral vessels using a model-based approach, we use a *3D parametric intensity model* that represents the shape as well as the image intensities of vessels within a 3D region of interest (ROI), and which consists of an approximation g_{Cyl} of an ideal sharp 3D cylinder that is convolved with a 3D Gaussian [46]. The tubular model includes parameters for the width R of the tubular structure and the image blur σ , and is well-suited to model tubular structures of different radii. The complete 3D model g_M also incorporates intensity levels a_0 (surrounding tissue) and a_1 (vessel) as well as a 3D rigid transform \mathcal{R} with rotation $\boldsymbol{\alpha} = (\alpha, \beta, \gamma)^T$ and translation $\mathbf{x}_0 = (x_0, y_0, z_0)^T$, which yields

Parametric intensity model

$$g_M(\mathbf{x}, \mathbf{p}) = a_0 + (a_1 - a_0) g_{Cyl}(\mathcal{R}(\mathbf{x}, \boldsymbol{\alpha}, \mathbf{x}_0), R, \sigma) \quad (4.1)$$

with parameters $\mathbf{p} = (R, a_0, a_1, \sigma, \alpha, \beta, \gamma, x_0, y_0, z_0)^T$.

To segment a certain *vessel segment*, we use a model fitting approach based on least-squares fitting of the 3D cylindrical model g_M to the image intensities $g(\mathbf{x})$ within a spherical 3D ROI:

Segmentation of a vessel segment

$$\sum_{\mathbf{x} \in \text{ROI}} (g_M(\mathbf{x}, \mathbf{p}) - g(\mathbf{x}))^2 \rightarrow \min. \quad (4.2)$$

To segment a *complete vessel*, we apply an incremental estimation process based on a Kalman filter that starts at an initial position on the vessel and incrementally proceeds along a vessel (see [46] for details). The size and shape of the 3D ROI used for model fitting are automatically adapted to the shape of the vessel [179]. Using this approach, the vessels are segmented and quantified simultaneously, i.e., after segmentation no additional step for quantification is required, as opposed

to approaches which result in a binary segmentation (e.g., [33, 35]), and therefore require extra steps for quantification.

4.2.2 Automatic Initialization

Compared to previous work where the model was initialized manually for each vessel (e.g., [46]), we employ an *automatic* scheme. In this scheme, potential vessel structures are identified based on a vesselness map, which is obtained by applying a vesselness filter to the image data (e.g., [51], see Sect. 4.4.2 below). Initial vessel positions are chosen based on large vesselness values in this map, and the initial orientations are estimated based on the eigenvectors of the Hessian matrix at the chosen positions. After a vessel is segmented, it is masked out in the vesselness map to avoid a repeated segmentation. Since the image data comprises vessels of different radii, we apply this scheme twice for different values of the standard deviation of the vesselness filter, i.e., first using a large value σ_1 for the segmentation of thick vessels, and then using a small value σ_2 for the segmentation of thin vessels. We used values of $\sigma_1 = 3$ and $\sigma_2 = 1$ in all experiments.

4.2.3 Partly Missing Connections

Using the model-based approach described above, thick vessels and most of the thin vessels are usually segmented successfully. However, it is often very difficult to segment the connecting parts between thick vessels and their branches, since these parts typically lie within regions with a low image contrast and a high noise level, and consequently the parametric intensity model cannot be fitted well. Thus, gaps exist after model-based segmentation, which must be completed (see Fig. 4.1). In our approach, gaps are detected automatically by identifying vessel endpoints that are close to another vessel but not yet connected to it.

In Fig. 4.2, the problem of using previous minimal path approaches for gap completion is illustrated. Usually, these approaches yield short cuts, and the resulting vessel has an abrupt change of direction in the gap. The reason is that these approaches do not incorporate initial directional information, such as the direction of the last segmented part of the vessel branch before the gap (which can be well determined using a model-based approach). In contrast, we introduce minimal path approaches which can effectively incorporate the initial direction \mathbf{d}_s and yield a smooth transition from the thin vessel (e.g., the solid green vessel in Fig. 4.2), which is segmented by the model-based approach, to the thick vessel (e.g., see the *desired* dashed green vessel in Fig. 4.2). Two different approaches

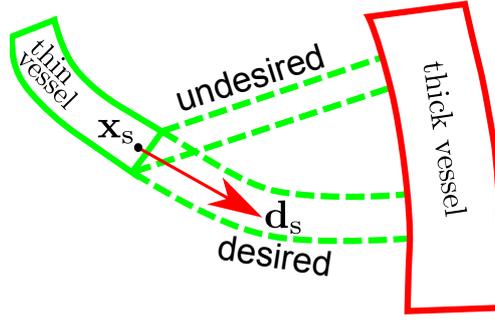


Figure 4.2: Comparison of our approach with previous minimal path approaches. Previous minimal path approaches usually yield a short cut in image regions with high noise level and low image contrast, while with our approach the initial direction \mathbf{d}_s from the model-based approach can be incorporated and therefore the transition is smooth.

are proposed, which allow preserving the smooth transition either *explicitly* or *implicitly*. The explicit approach (Sect. 4.3 below) is based on probabilistic sampling, and the implicit approach (Sect. 4.4) is based on fast marching with anisotropic directional prior.

4.3 Explicit Initial Direction Preservation: 3D Probabilistic Minimal Path

In the 3D probabilistic minimal path approach, we denote the vessel in a gap as γ . An energy function is used to measure γ by *explicitly* imposing the smoothness of the vessel as a soft constraint.

4.3.1 Energy Function

For each γ , we denote \mathcal{V}_f as the sequence of voxels on γ . \mathbf{x}_s and \mathbf{x}_e are the start and end voxels in \mathcal{V}_f , respectively. $\mathcal{N}_{\mathbf{x}_i}$ denotes the set of neighboring centerline voxels of \mathbf{x}_i . Furthermore, \mathcal{M} is the set of voxels which are segmented as vessel by the model-based approach and $\mathcal{M}_e \subset \mathcal{M}$ is the set of the vessel end points from that step. Then, the energy function can be defined as:

Energy function of probabilistic minimal path

$$E(\gamma) = \sum_{\mathbf{x}_i \in \mathcal{V}_f} D(\mathbf{x}_i) + \lambda \cdot \sum_{\mathbf{x}_i \in \mathcal{V}_f} S(\mathbf{x}_i), \quad (4.3)$$

$$\text{such that } \mathbf{x}_s \in \mathcal{M}_e, \quad (4.4)$$

$$\text{and } \mathbf{x}_e \in \mathcal{M}. \quad (4.5)$$

$D(\mathbf{x}_i)$ is the *data term* which describes the probability that a voxel \mathbf{x}_i is located within a vessel. The more the intensity of a voxel deviates from the maximum intensity in its neighborhood, the lower is the probability that this voxel is located within a vessel and thus the higher is the corresponding energy. Accordingly, $D(\mathbf{x}_i)$ is defined as:

$$D(\mathbf{x}_i) = |g(\mathbf{x}_i) - \max_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}} g(\mathbf{x}_j)|. \quad (4.6)$$

$S(\mathbf{x}_i)$ encodes the *smoothness* of the vessel at \mathbf{x}_i , which is independent of the image data and weighted by the scalar λ . Higher curvatures of the centerline are assigned a higher energy. Let $\theta(\mathbf{d}_1, \mathbf{d}_2)$ denote the angle between two directions \mathbf{d}_1 and \mathbf{d}_2 :

$$\theta(\mathbf{d}_1, \mathbf{d}_2) = \arccos \left(\frac{\mathbf{d}_1^T \cdot \mathbf{d}_2}{\|\mathbf{d}_1\| \cdot \|\mathbf{d}_2\|} \right). \quad (4.7)$$

Then $S(\mathbf{x}_i)$ is defined as:

$$S(\mathbf{x}_i) = \theta(\mathbf{d}_{\mathbf{x}_i}, \mathbf{d}_{\mathbf{x}_{i+1}}), \quad (4.8)$$

where

$$\mathbf{d}_{\mathbf{x}_i} = \mathbf{x}_i - \mathbf{x}_{i-1}, \quad (4.9)$$

$$\mathbf{d}_{\mathbf{x}_{i+1}} = \mathbf{x}_{i+1} - \mathbf{x}_i. \quad (4.10)$$

The curvature at \mathbf{x}_i is measured by the angle between the two consecutive vessel segments at \mathbf{x}_i . $\sum D$ and $\sum S$ in (4.3) can be interpreted as the energy functions for the likelihood and the prior in Bayes' theorem, respectively. The sum of $\sum D$ and $\sum S$ corresponds to the energy of the posterior. The hard constraints (4.4) and (4.5) state that the start and end points \mathbf{x}_s and \mathbf{x}_e are voxels that have been segmented as vessel in the previous model-based segmentation step.

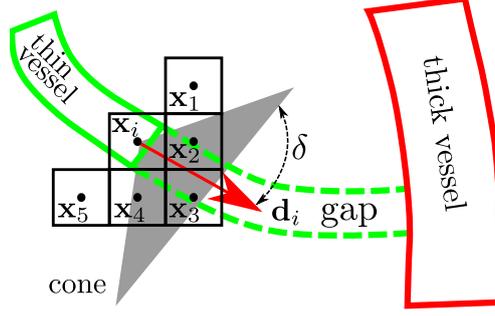


Figure 4.3: 2D sketch of 3D sampling within a cone (gray triangle). $\mathcal{N}_{\mathbf{x}_i}^{\text{cone}} = \{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$. $\mathbf{x}_1, \mathbf{x}_5 \notin \mathcal{N}_{\mathbf{x}_i}^{\text{cone}}$ since these points are outside of the cone.

4.3.2 Sampling Scheme

The start point $\mathbf{x}_i = \mathbf{x}_s$ and tangent $\mathbf{d}_i = \mathbf{d}_s$ are given by the end point and tangent of the last vessel segment from model-based segmentation. Then, a cone which opens in the direction of \mathbf{d}_i with an angle of 2δ and the apex at \mathbf{x}_i is constructed, indicated as the gray triangle in Fig. 4.3. The set of all neighboring voxels within the $3 \times 3 \times 3$ neighborhood of \mathbf{x}_i inside this cone is denoted by $\mathcal{N}_{\mathbf{x}_i}^{\text{cone}}$. Let $g(\mathbf{x})$ denote the intensity value of voxel \mathbf{x} , then the next voxel \mathbf{x}_{i+1} of the candidate vessel is selected randomly from $\mathcal{N}_{\mathbf{x}_i}^{\text{cone}}$, with a probability given by:

$$P(\mathbf{x}) = \frac{g(\mathbf{x})}{\sum_{\mathbf{x}_j \in \mathcal{N}_{\mathbf{x}_i}^{\text{cone}}} g(\mathbf{x}_j)}. \quad (4.11)$$

The closer $g(\mathbf{x})$ is to the maximum intensity within $\mathcal{N}_{\mathbf{x}_i}^{\text{cone}}$, the higher is the probability that \mathbf{x} is selected as the next voxel of the candidate vessel. Once the next voxel has been found, the tangent can be updated and used to find further voxels. This process is repeated until a voxel is reached which is in \mathcal{M} , or until a maximum length is reached.

The vessel centerline candidate is a sample which is then evaluated by (4.3). Note that the hard constraints (4.4) and (4.5) are satisfied automatically, because for each successfully completed gap, the sampling always starts at the last point of the vessel segmented by model-based approach, and ends at a voxel in a vessel. The sample with the lowest energy value is selected as the final result of the probabilistic optimization. If the optimal sample does not end in \mathcal{M} , we conclude that this vessel end should not be connected to a thick vessel (cf. one end of vessel 4 in Fig. 4.1 indicated by a white arrow, which does not have a thick vessel close to it).

4.4 Implicit Initial Direction Preservation: 3D Fast Marching with Anisotropic Directional Prior

In contrast to the explicit approach described in the previous section, the information of the initial direction can also be incorporated *implicitly* using the fast marching framework. First, we briefly describe the standard fast marching approach (Sect. 4.4.1). Then, we present an approach to incorporate an anisotropic directional prior into the fast marching approach (Sect. 4.4.2).

4.4.1 Standard Fast Marching Approach

Given a start point \mathbf{x}_s and an end point \mathbf{x}_e , fast marching aims at finding the path γ minimizing the energy function

$$E(\gamma) = \int_{\gamma} (\mathcal{P}(s) + w) ds, \quad (4.12)$$

where \mathcal{P} is a *potential function* derived from an image, w is a constant regularization term which controls the smoothness of the path, and s is the arc length parameter. \mathcal{P} can be interpreted as the inverse of the propagation speed \mathcal{F} of a wavefront, which emanates from the start point \mathbf{x}_s :

$$\mathcal{P}(s) = \frac{1}{\mathcal{F}(s)}. \quad (4.13)$$

This wavefront keeps propagating outwards, until an end point \mathbf{x}_e is reached. Let $\mathcal{A}_{\mathbf{x}_1, \mathbf{x}_2}$ denote the set of all paths γ connecting two given points \mathbf{x}_1 and \mathbf{x}_2 , then the arrival time of the wavefront at each voxel \mathbf{x} can be represented by the *minimal action map* $\mathcal{U}_{\mathbf{x}_s}$:

$$\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}) := \min_{\gamma \in \mathcal{A}_{\mathbf{x}_s, \mathbf{x}}} E(\gamma). \quad (4.14)$$

Usually, (4.14) is computed by solving the *Eikonal equation*

$$\|\nabla \mathcal{U}_{\mathbf{x}_s}(\mathbf{x})\| = \mathcal{P}(s) + w \quad (4.15)$$

using efficient numerical schemes such as the *upwind scheme* (e.g., [178]). Note that this equation employs the Euclidean norm of the gradient $\|\nabla \mathcal{U}_{\mathbf{x}_s}(\mathbf{x})\|$, which is the same for all directions, i.e., the speed function \mathcal{F} must be *isotropic*. Once $\mathcal{U}_{\mathbf{x}_s}$ is

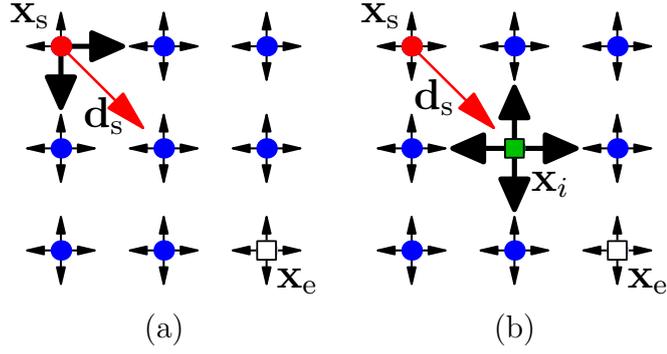


Figure 4.4: 2D sketch of 3D anisotropic effect. (a) Anisotropic speed: The two large arrows at \mathbf{x}_s indicate higher speed in two specific *directions*. (b) Isotropic speed: The four large arrows at the voxel \mathbf{x}_i indicate higher speed in the specific *area*.

computed, the minimal path can be extracted: Starting from \mathbf{x}_e , the *predecessor* of the current position is determined using gradient descent. This process is repeated until \mathbf{x}_s is reached. The final result of fast marching approaches has *subvoxel* accuracy, i.e., *metrication errors* caused by the discrete grid structure, which are an inherent problem of discrete approaches such as Dijkstra’s algorithm, are avoided.

There are several differences between this standard fast marching approach and the probabilistic minimal path approach described in Sect. 4.3. First, the actual path is only computed after the wavefront reaches \mathbf{x}_e . During the propagation of the wavefront, there is no explicit representation of a path. Instead, the necessary information to compute the path is embedded *implicitly* in $\mathcal{U}_{\mathbf{x}_s}$. Second, in the probabilistic approach, the smoothness of the path is regularized by assigning a high energy to paths with high curvature, while here the smoothness regularization is achieved indirectly via length regularization. This is because the energy function can be decomposed as:

$$E(\gamma) = \int_{\gamma} (\mathcal{P}(s) + w) ds = \int_{\gamma} \mathcal{P}(s) ds + w \int_{\gamma} 1 ds, \quad (4.16)$$

where $\int_{\gamma} 1 ds$ is the Euclidean length of γ . Therefore, the energy function (4.16) prefers shorter paths, and smoothness is thus a consequence of this preference, since usually it is assumed that shorter paths are also smoother. However, shorter paths are not always the better results (cf. Fig. 4.2). To address this problem, below we introduce a fast marching approach with an anisotropic directional prior.

4.4.2 Anisotropic Directional Prior

As mentioned above in Sect. 4.2.3, the model-based approach provides an estimate of the direction \mathbf{d}_s of a vessel at each start point \mathbf{x}_s , which can be used to segment vessels in low-contrast and noisy regions. Obviously, \mathbf{d}_s involves anisotropic information, which may suggest using an *anisotropic* fast marching approach. However, current anisotropic fast marching approaches (e.g., [43, 44]) can only cope with speed functions defined by symmetric positive definite metric tensors, which means that the speed function at each pixel or voxel \mathbf{x} has an *elliptical* profile centered at \mathbf{x} . Furthermore, for high anisotropy, the numeric solution is either not accurate or very time-consuming [44]. In comparison, our approach does not have these limitations. First, the direction \mathbf{d}_s of a vessel is incorporated in a principled manner. Second, we use *isotropic* speed functions to create an anisotropic behavior which is not limited to be elliptic but can be more general (irregular). Third, since we always use the numeric solution for *isotropic* fast marching, our approach is accurate and computationally efficient, even for high anisotropy.

The main idea is to modify the potential function \mathcal{P} , or equivalently the propagation speed \mathcal{F} , so that it is possible to incorporate the prior directional information \mathbf{d}_s into the *isotropic* Eikonal equation. This is illustrated in Fig. 4.4. There, each dot or square represents a voxel, and the start point \mathbf{x}_s and end point \mathbf{x}_e are highlighted as red dots and white squares, respectively. The initial direction \mathbf{d}_s is indicated as a red arrow. The black arrows show the speed at voxels in different directions. Suppose that we prefer paths such that the initial tangents of the paths point in directions similar to \mathbf{d}_s . Using an anisotropic speed (Fig. 4.4, left), this would be achieved by increasing the speeds in specific directions at \mathbf{x}_s (large black arrows). However, such an anisotropic speed cannot be handled using the isotropic Eikonal equation. Moreover, since this speed function does not fit into an ellipse centered at \mathbf{x}_s , it cannot be handled by anisotropic fast marching either (e.g., [43, 44]). The idea in our approach is that a similar anisotropic effect can be achieved using an *isotropic* speed. Instead of increasing the speed in preferred *directions*, we increase the *isotropic* speed at preferred *image positions*. For example, in Fig. 4.4 (right), all speeds in the different directions at the green voxel are increased to the same large value. On the scale of the whole image, an anisotropy is achieved, i.e., the paths passing through the green voxel are preferred, but on the scale of individual voxels, this speed function is still isotropic, and consequently the *isotropic* Eikonal equation can be applied. To incorporate the *anisotropy*, we propose the following energy function as an extension of (4.12):

Energy function with anisotropic directional prior

$$E(\gamma, \mathbf{x}_s, \mathbf{d}_s) = \int_{\gamma} \left(\mathcal{P}_{\text{comp}}(s, \mathbf{x}_s, \mathbf{d}_s) + w \right) ds. \quad (4.17)$$

Similar to (4.13), the composite potential function can be interpreted as the inverse of the composite speed:

$$\mathcal{P}_{\text{comp}}(s, \mathbf{x}_s, \mathbf{d}_s) = \frac{1}{\mathcal{F}_{\text{comp}}(s, \mathbf{x}_s, \mathbf{d}_s)}, \quad (4.18)$$

where $\mathcal{F}_{\text{comp}}$ is defined as:

$$\begin{aligned} & \mathcal{F}_{\text{comp}}(s, \mathbf{x}_s, \mathbf{d}_s) \\ &= \mathcal{F}_v(s) + w \cdot \mathcal{F}_{\text{dir}}(\mathbf{x}(s) - \mathbf{x}_s, \mathbf{d}_s) + \mathcal{F}_0. \end{aligned} \quad (4.19)$$

$\mathcal{F}_{\text{comp}}$ includes a speed function \mathcal{F}_v based on a multiscale vesselness filter, a speed function \mathcal{F}_{dir} based on an anisotropic (irregular) directional prior, as well as a constant speed \mathcal{F}_0 (w is a scalar weight). Note that our approach can cope with more general anisotropic behavior compared to previous anisotropic fast marching approaches with elliptical speed functions (e.g., [43, 44]). Our speed functions are detailed below:

4.4.2.1 Vesselness-based Speed Function \mathcal{F}_v

The speed function \mathcal{F}_v is defined as:

$$\mathcal{F}_v(s) = \max_{\sigma_{\min} \leq \sigma \leq \sigma_{\max}} \mathcal{V}_s(\mathbf{x}). \quad (4.20)$$

In our approach, the filter from [51] is used, i.e.:

$$\mathcal{V}_s(\mathbf{x}) = \begin{cases} |\lambda_3| \left(\frac{\lambda_2}{\lambda_3} \right)^{\gamma_{23}} \left(1 + \frac{\lambda_1}{|\lambda_2|} \right)^{\gamma_{12}}, & \text{if } \lambda_1 \leq 0, \lambda_2, \lambda_3 < 0 \\ |\lambda_3| \left(\frac{\lambda_2}{\lambda_3} \right)^{\gamma_{23}} \left(1 - \alpha \frac{\lambda_1}{|\lambda_2|} \right)^{\gamma_{12}}, & \text{if } \frac{|\lambda_2|}{\alpha} > \lambda_1 > 0 > \lambda_2, \lambda_3 \\ 0, & \text{otherwise.} \end{cases} \quad (4.21)$$

where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of the Hessian matrix at \mathbf{x} , while γ_{12}, γ_{23} control the sensitivity of the filter.

4.4.2.2 Anisotropic Directional Prior Speed Function \mathcal{F}_{dir}

We introduce a new function \mathcal{F}_{dir} which depends only on the spatial relationship between voxel coordinates $\mathbf{x}(s)$, \mathbf{x}_s and direction \mathbf{d}_s , but not on the intensities of an image, i.e., \mathcal{F}_{dir} represents our spatial *a priori* knowledge. Let $\mathbf{d} := \mathbf{x}(s) - \mathbf{x}_s$, then \mathcal{F}_{dir} can be written as:

$$\mathcal{F}_{\text{dir}}(\mathbf{d}, \mathbf{d}_s) = f_{\text{dir}}(\mathbf{d}, \mathbf{d}_s) \cdot f_{\text{att}}(\mathbf{d}), \quad (4.22)$$

where f_{dir} is the speed depending on the deviation of \mathbf{d} from \mathbf{d}_s , and f_{att} is an attenuation function. For f_{dir} , we propose the following Gaussian speed function:

$$f_{\text{dir}}(\mathbf{d}, \mathbf{d}_s) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\theta(\mathbf{d}, \mathbf{d}_s)^2}{2\sigma^2}\right), \quad (4.23)$$

where $\theta(\mathbf{d}, \mathbf{d}_s)$ is the angle between \mathbf{d} and \mathbf{d}_s , as defined in (4.7). With increasing distance from \mathbf{x}_s , the influence of the prior is reduced to avoid unduly straight results. This is modeled using the following attenuation function f_{att} :

$$f_{\text{att}}(\mathbf{d}) = \exp\left(-\frac{\|\mathbf{d}\|}{\alpha}\right). \quad (4.24)$$

4.4.2.3 Constant Speed Function \mathcal{F}_0

At some voxels, the sum $\mathcal{F}_v + \mathcal{F}_{\text{dir}}$ may be zero. However, the speed $\mathcal{F}_{\text{comp}}$ cannot be zero because this would cause a division by zero in (4.18). Therefore, an additional small positive constant speed \mathcal{F}_0 is added to each voxel so that the wavefront always propagates outwards and never stops before reaching \mathbf{x}_e .

Since \mathcal{F}_v , \mathcal{F}_{dir} , and \mathcal{F}_0 are all isotropic, $\mathcal{F}_{\text{comp}}$ and $\mathcal{P}_{\text{comp}}$ are also isotropic. Thus, to find the minimum of (4.17), we can use the same numerical scheme as for the isotropic Eikonal equation (4.15), i.e., the upwind scheme [178], to solve the following *extended* Eikonal equation:

Extended Eikonal equation

$$\|\nabla \mathcal{U}_{\mathbf{x}_s}(\mathbf{x})\| = \mathcal{P}_{\text{comp}}(s, \mathbf{x}_s, \mathbf{d}_s) + w. \quad (4.25)$$

For each endpoint \mathbf{x}_s of a segmented vessel we check whether there exists another segmented vessel within a distance of 8 voxels. If there is no such vessel, then we conclude that \mathbf{x}_s should not be connected to another vessel (cf. one end of vessel 4 in Fig. 4.1 indicated by a white arrow).

4.4.3 Interpretation of Anisotropic Directional Prior

The effects of the anisotropic directional prior can be illustrated by an example shown in Fig. 4.5. The thin vessel (green) should be connected to the thick vessel (red). For simplicity and clarity, we assume that $\mathcal{F}_v = 0$ for each voxel outside these two vessels. In other words, only the constant speed \mathcal{F}_0 and the anisotropic directional prior \mathcal{F}_{dir} influence the propagation of the wavefront, which starts at the voxel \mathbf{x}_s and ends when the thick vessel is reached at the blue point. The different values of $\mathcal{U}_{\mathbf{x}_s}$ are displayed using different grayscales: Darker regions are reached by the wavefront before brighter regions. The boundaries (dashed lines) between regions with different $\mathcal{U}_{\mathbf{x}_s}$ values are called *iso-arrival lines*.

Fig. 4.5a shows the case without the anisotropic directional prior, i.e., $\mathcal{F}_{\text{dir}} = 0$. In this case, the speed is the same for all voxels, so the wavefront propagates isotropically with the constant speed \mathcal{F}_0 , and therefore the iso-arrival lines are circular. Consequently, these two vessels are connected with each other via a straight line. In this case, there is a sharp turn at \mathbf{x}_s , which does not correspond to the general assumption that vessels should be smooth.

In Fig. 4.5b, the anisotropic directional prior is incorporated. The initial direction \mathbf{d}_s is considered reliable since it is computed using the model-based approach in image regions with higher contrast. This directional information can be used to improve the result by adding \mathcal{F}_{dir} to each voxel according to (4.23). Therefore, the speed is higher at the voxels in preferred regions, and there the wavefront propagation is faster. Consequently, the iso-arrival lines are not circular any more. The vessels are connected with a curve such that the transition between this curve and the green vessel is smooth.

4.5 Implicit Initial Direction Preservation: Iterative 2D Fast Marching

Based on the implicit approach described above in Sect. 4.4, we introduce a new approach to segment the vessel centerline and determine the radius in an iterative manner. The implicit approach is based on the fast marching method and uses a directional prior. This prior promotes anisotropic propagation of the wavefront and is combined with speed functions based on a vesselness measure and the vessel radius. In contrast to previous approaches based on fast marching (e.g., [41, 154, 162]), our approach can deal with short cuts and crossings in long vessels. An important difference to [45] is that we use the prior as a *soft* constraint.

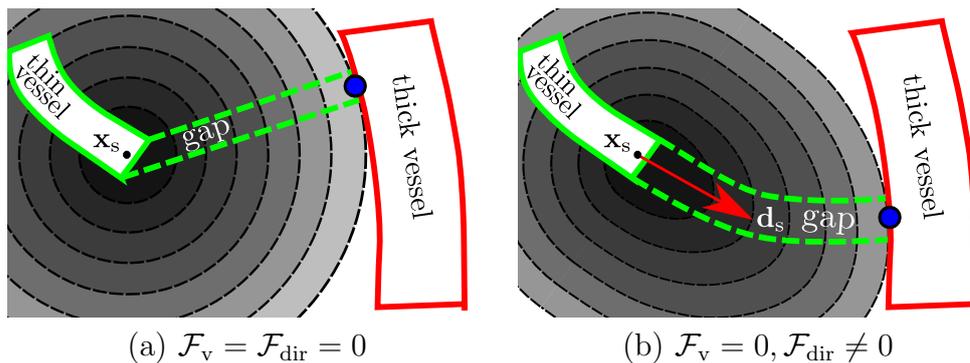


Figure 4.5: Effect of the directional prior. (a) $\mathcal{F}_v = \mathcal{F}_{\text{dir}} = 0$. The wavefront propagates with the same constant speed \mathcal{F}_0 overall, resulting in circular iso-arrival lines and a sharp turn at \mathbf{x}_i . (b) $\mathcal{F}_v = 0, \mathcal{F}_{\text{dir}} \neq 0$. The wavefront propagation speed $\mathcal{F}_0 + \mathcal{F}_{\text{dir}}$ is faster in preferred regions, resulting in non-circular iso-arrival lines and a smooth transition at \mathbf{x}_i , and gradual change of direction.

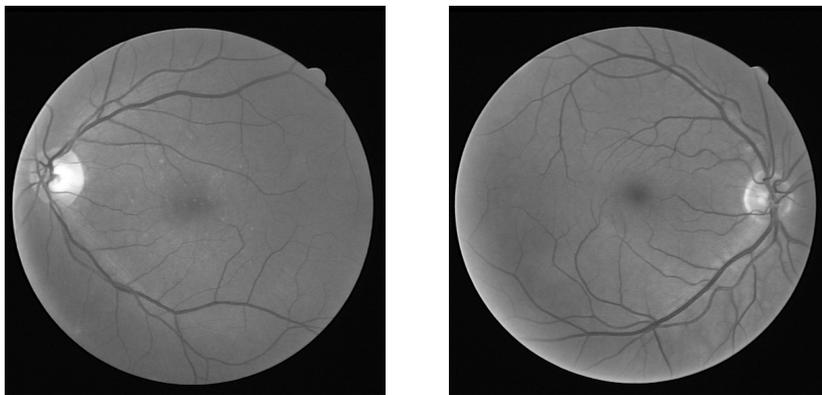


Figure 4.6: Two retinal images of the DRIVE dataset [24].

Moreover, all speed functions are constructed in such a way that isotropic fast marching can be used for the optimization instead of the computationally more expensive anisotropic fast marching. Furthermore, our approach is more efficient than previous approaches (e.g., [155]), since it uses only a 3D parameter space and restricts the computation to a relatively small image region.

4.5.1 Iterative Framework

Our approach mimics the way a human observer segments a vessel. Usually, an observer does not segment the complete vessel at once but starts from some part of the vessel and follows it. Fig. 4.7 (right) illustrates this procedure. For a point \mathbf{x}_k within the vessel and a direction \mathbf{d}_k , we try to find a small part of the vessel within a circular ROI centered around \mathbf{x}_k . To do so, we compute the composite speed function with three components (see Sect. 4.5.2) for the ROI and find the

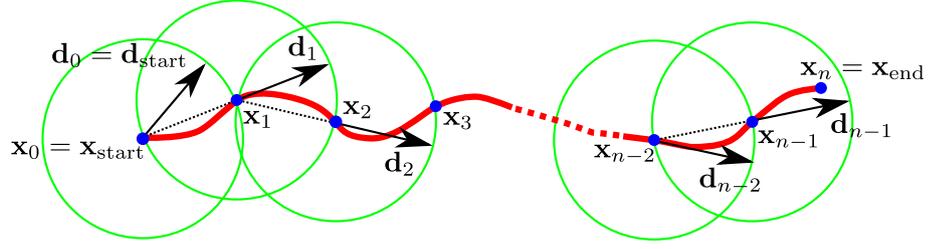


Figure 4.7: Sketch of the iterative vessel segmentation approach.

pixel \mathbf{x}_{\min} with the minimum arrival time on the ROI boundary. The optimal path is found by back-tracing from \mathbf{x}_{\min} to \mathbf{x}_k using the gradient descent method. After that, a new direction \mathbf{d}_{k+1} is predicted as $\mathbf{d}_{k+1} = (\mathbf{x}_{\min} - \mathbf{x}_k)$. We set $\mathbf{x}_{k+1} := \mathbf{x}_{\min}$ and the next ROI centered around \mathbf{x}_{k+1} is analyzed. The initial position \mathbf{x}_0 and direction \mathbf{d}_0 are provided by the user. This procedure is repeated until certain termination criteria (see Sect. 4.5.3) are satisfied. Upon termination, the sequence of paths from individual ROIs are concatenated to form the final result.

4.5.2 Speed Functions

Our speed function is defined as:

Speed function of iterative fast marching

$$\mathcal{F}_{\text{comp}}(\mathbf{x}, r) = \mathcal{F}_v(\mathbf{x}, r) + w_r \mathcal{F}_{\text{radius}}(\mathbf{x}, r) + w_d \mathcal{F}_{\text{dir}}(\mathbf{x}, r) \quad (4.26)$$

where \mathcal{F}_v , $\mathcal{F}_{\text{radius}}$, and \mathcal{F}_{dir} are speed functions derived from a vesselness measure, the vessel radius, and the vessel direction, respectively. These speed functions are weighted by the factors w_r and w_d . \mathbf{x} is the pixel position, and r is the estimate of the radius. \mathcal{F}_v and \mathcal{F}_{dir} are independent of r , i.e., $\mathcal{F}_v(\mathbf{x}, r) = \mathcal{F}_v(\mathbf{x})$ and $\mathcal{F}_{\text{dir}}(\mathbf{x}, r) = \mathcal{F}_{\text{dir}}(\mathbf{x})$ for $r = r_{\min}, \dots, r_{\max}$. The definitions of these speed functions are described in the following.

4.5.2.1 Speed Function Based on Vesselness

We use a multiscale vesselness filter [50] to enhance tubular structures, i.e.:

$$\mathcal{F}_v(\mathbf{x}) = \max_{s_{\min} \leq s \leq s_{\max}} \mathcal{V}_s \mathbf{x}, \quad \text{with} \quad (4.27)$$

$$\mathcal{V}_s \mathbf{x} = \begin{cases} 0, & \text{if } \lambda_2 > 0 \\ \exp\left(-\frac{1}{\beta^2} \cdot \frac{\lambda_1^2}{\lambda_2^2}\right) \left(1 - \exp\left(-\frac{1}{2c^2} \cdot (\lambda_1^2 + \lambda_2^2)\right)\right), & \text{otherwise} \end{cases} \quad (4.28)$$

where the scale s ranges from s_{\min} to s_{\max} . λ_1 and λ_2 are the eigenvalues of the Hessian matrix at \mathbf{x} . β and c control the sensitivity of the filter.

4.5.2.2 Speed Function Based on Radius

To segment not only the vessel path but also to quantify its radius, we apply the speed function:

$$\mathcal{F}_{\text{radius}}(\mathbf{x}, r) = \frac{1}{\mathcal{P}_{\text{radius}}(\mathbf{x}, r)}, \quad \text{with} \quad (4.29)$$

$$\mathcal{P}_{\text{radius}}(\mathbf{x}, r) = w_{\mu} \left(\frac{\mu_{\text{int}}(s(\mathbf{x}, r))}{r} - \frac{\mu_{\text{int}}(s(\mathbf{x}_k, r))}{r} \right)^2 + w_{\sigma} \left(\frac{\sigma_{\text{int}}^2(s(\mathbf{x}, r))}{r} - \frac{\sigma_{\text{int}}^2(s(\mathbf{x}_k, r))}{r} \right)^2, \quad (4.30)$$

where $s(\mathbf{x}, r)$ is a sphere of radius r centered at \mathbf{x} , μ_{int} and σ_{int}^2 denote the mean and variance of the pixel intensities within this sphere, respectively, and w_{μ} as well as w_{σ} are two factors [154]. Note that in contrast to $\mathcal{F}_v(\mathbf{x})$, $\mathcal{F}_{\text{radius}}(\mathbf{x}, r)$ has an additional parameter r for the radius information, thus $\mathcal{F}_{\text{radius}}(\mathbf{x}, r)$ has a 3D parameter space where r ranges from r_{\min} to r_{\max} .

4.5.2.3 Speed Function Based on Directional Prior

The predicted direction \mathbf{d}_k from the last ROI is important a priori information because it can help to reduce the ambiguities caused by low image contrast, noise, or crossings, and can prevent the algorithm from returning to already segmented vessel parts. To incorporate \mathbf{d}_k into our framework, we suggest a directional prior which emphasizes directions close to \mathbf{d}_k as increased speed in these directions. Note that in single-pass fast marching algorithms, such directional information is not available. We propose the following Gaussian speed function:

$$\mathcal{F}_{\text{dir}}(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\theta^2}{2\sigma^2}\right), \quad \text{with } \theta = \arccos\left(\frac{\mathbf{d}_k^T \cdot (\mathbf{x} - \mathbf{x}_k)}{|\mathbf{d}_k| \cdot |\mathbf{x} - \mathbf{x}_k|}\right). \quad (4.31)$$

Although a higher speed is allowed if the direction $(\mathbf{x} - \mathbf{x}_k)$ is closer to the predicted direction \mathbf{d}_k , other directions are not excluded, i.e., this is a *soft* constraint, in contrast to the *hard* constraint in [45]. Note that although this prior achieves an anisotropic effect at the scale of the ROIs, it is isotropic at the scale of pixels, thus it can be optimized using the isotropic fast marching method.

4.5.3 Termination Criteria

We employ two termination criteria which are examined in each iteration. The algorithm terminates if one of these criteria is satisfied. First, we test whether the end point \mathbf{x}_{end} is within the current ROI. If this is the case, then we start back-tracing from the end point instead of back-tracing from a pixel at the ROI boundary, and after that the iteration is terminated. Otherwise, we proceed as described in Sect. 4.5.1 above. Second, upon finding a vessel in an ROI, we compute the percentage of vessel pixels with reasonably high vesselness values. More precisely, we check whether the following inequality is satisfied:

$$\frac{1}{|V|} \sum_{\mathbf{x} \in V} \chi(\mathcal{F}_v(\mathbf{x}), T_v) > T_\chi, \text{ with } \chi(\mathcal{F}_v(\mathbf{x}), T_v) = \begin{cases} 1, & \text{if } \mathcal{F}_v(\mathbf{x}) > T_v \\ 0, & \text{otherwise} \end{cases} \quad (4.32)$$

where $|V|$ denotes the number pixels belonging to vessel V in the current ROI, $\mathcal{F}_v(\mathbf{x})$ is the vesselness at the pixel \mathbf{x} . T_v is the vesselness threshold, and T_χ is a threshold in percentage. If T_χ is not reached, then we conclude that the current vessel candidate does not have the appearance of a vessel and thus the segmentation is terminated.

Chapter 5

Experimental Results of Direction-Preserving Minimal Path Methods

In this chapter, experimental results for the direction-preserving minimal path methods introduced in Chapter 4 are presented. Results for 3D synthetic images are provided in Sect. 5.1, and results for real 3D 7T MRA images are shown in Sect. 5.2. Finally, results for the iterative 2D fast marching method are given in Sect. 5.3

5.1 Synthetic 3D Images

We have generated 2370 3D synthetic images using different 3D analytic intensity models to quantitatively analyze the performance of our approach. In this section, we present experiments using two types of 3D synthetic images containing structures (branching vessels and parallel vessels) which resemble typical configurations of vascular structures in real 3D 7T MRA images. Two examples of such configurations are shown in Figures 5.10a and 5.11a. Sketches of corresponding synthetic images are shown in Figures 5.1 and 5.6, respectively.

5.1.1 Branching Vessels

In this experiment, 3D synthetic images of branching vessels are considered. For an example of a branching vessel in a 3D 7T MRA image, see Fig. 5.10a. A sketch of this configuration is provided in Fig. 5.1: Each image contains a horizontal thick vessel V_A and an inclined thin vessel V_B . Both V_A and V_B are modeled

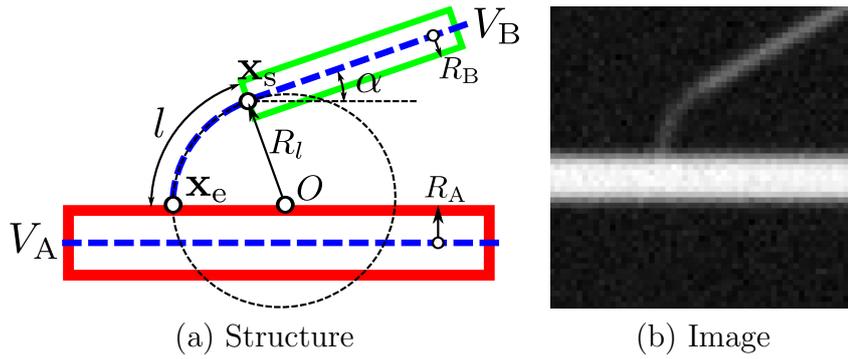


Figure 5.1: Synthetic 3D branching vessel. (a) Sketch of a branching vessel used for quantitative evaluation. Between the horizontal vessel V_A with radius R_A and the inclined vessel V_B with radius R_B and angle α to V_A , there is a low-contrast part of length l . This small part is an arc of a circle with radius R_l and centered at O . The starting and end points are \mathbf{x}_s and \mathbf{x}_e , respectively. Gaussian noise is added. (b) 2D section of an example of a 3D synthetic image based on (a).

as smoothed straight cylinders, with radius R_A and R_B , respectively. The angle between V_A and V_B is denoted by α . Furthermore, there is a curved part of length l between V_B and V_A , for which the voxel intensities are significantly lower than the intensities in V_A or V_B . The model-based approach generally leads to a gap in such parts, therefore we denote this part as gap. This gap starts at \mathbf{x}_s , and terminates at a point \mathbf{x}_e on the surface of V_A . The shape of this vessel part is an arc of a circle with a radius R_l . The center of the circle is O and lies on the surface of V_A . Furthermore, Gaussian image noise with standard deviation σ_n is added. All images have the same size of $128 \times 128 \times 32$ voxels. The parameters R_B , R_A , as well as the intensities in V_B and V_A are all determined based on observed values for vessels in the region around the lenticulostriate arteries (LSA) in real 3D 7T MRA images. The values of these parameters remain fixed throughout all our experiments. Specifically, we used the following parameter values: The radii of V_A and V_B are $R_A = 4$ voxels and $R_B = 1$ voxel, respectively. The mean intensities of V_A , V_B , and the low-contrast region are $g_A = 450$, $g_B = 200$, and $g_{\text{gap}} = 100$, respectively. Furthermore, the mean background intensity is $g_{\text{bg}} = 50$. We study the performance of our approaches for varying values of the parameters α , l , and σ_n .

The performance of the model-based approach was previously evaluated in [46], therefore we concentrate on the evaluation of the minimal path approaches described in Sect. 4.3 and Sect. 4.4. In our experiments we used a termination point at \mathbf{x}_s for the model-based approach, so that the minimal path approaches always start at the same position \mathbf{x}_s . To measure the performance, we use the

5.1. Synthetic 3D Images

mean error and standard deviation between the segmented centerline γ and the corresponding ground truth (in voxels)

Mean error

$$\bar{e}_\gamma = \frac{1}{|\gamma|} \sum_{\mathbf{x}_i \in \gamma} |\mathbf{x}_i - \mathbf{x}_{\text{gt}(i)}|, \quad (5.1)$$

where \mathbf{x}_i are sample points on γ , and $\mathbf{x}_{\text{gt}(i)}$ is the closest point of the ground truth to \mathbf{x}_i . $|\gamma|$ denotes the number of points on γ . In Fig. 5.2, the error measure is illustrated using a path with five sample points $\mathbf{x}_1, \dots, \mathbf{x}_5$, i.e., $|\gamma| = 5$. Note that the closest points $\mathbf{x}_{\text{gt}(1)}, \dots, \mathbf{x}_{\text{gt}(5)}$ are arbitrarily located on the curve of ground truth. In other words, they are *not* necessarily control points of the ground truth.

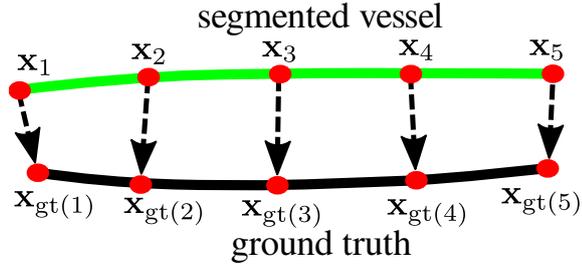


Figure 5.2: The mean error is determined based on the mean distance between the sample points \mathbf{x}_i of a segmented vessel and the closest point $\mathbf{x}_{\text{gt}(i)}$ on the corresponding ground truth.

In the following, we study the performance of our approaches for varying angle α (Sect. 5.1.1.1), varying gap length l (Sect. 5.1.1.2), and varying image noise level σ_n (Sect. 5.1.1.3). In each case, the results of our explicit approach (PROB, described in Sect. 4.3) and of our implicit approach (FM-ADP, described in Sect. 4.4) are compared with a previous (classical) fast marching approach (FM-V) [41]. The difference between FM-ADP and FM-V is that for FM-ADP the speed function is based both on the directional prior and the vesselness measure, while for FM-V the speed is based only on the vesselness measure.

5.1.1.1 Varying Angle

To analyze the performance of our approaches with respect to the angle α between vessels V_A and V_B , we used a large range of 20 different values of angles: $\alpha = 0.1\pi, 0.12\pi, 0.14\pi, \dots, 0.48\pi$. The gap length and image noise level are held

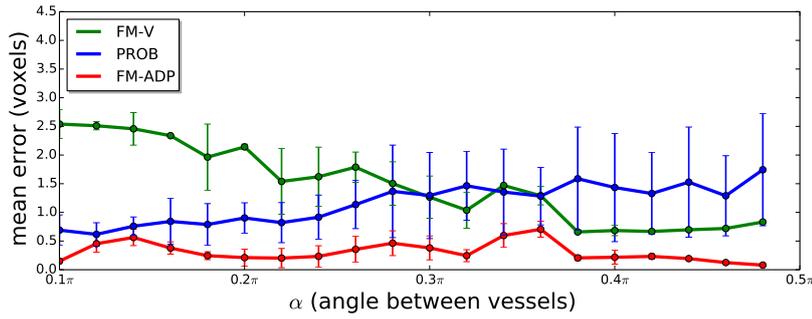


Figure 5.3: Mean error and standard deviation of the centerline positions for different *angles* for branching vessels in 3D synthetic image data: the angle α between vessels V_A and V_B varies from 0.1π to 0.48π , with an increment of 0.02π . The other parameters are constant: $l = 6$ and $\sigma_n = 10$. The lines indicate the mean errors, and the error bars show the standard deviation of 30 repetitions for each image.

constant at $l = 6$ and $\sigma_n = 10$. To account for the randomness of noise, the addition of image noise was repeated 30 times, yielding $20 \times 30 = 600$ 3D images in total. For each value of α , the mean error and standard deviation of the centerline positions are computed, see Fig. 5.3. It turns out that FM-ADP achieves consistently the lowest mean error, which is well within the subvoxel range, compared to the other two approaches. For small or medium angles ($\alpha < 0.36\pi$), PROB outperforms FM-V, while for larger angles FM-V is better than PROB. Moreover, FM-V and FM-ADP both yield low standard deviations, while the standard deviation of PROB increases with increasing angle.

5.1.1.2 Varying Gap Length

In this experiment, we varied the gap length l from 2.5 voxels to 16 voxels with an increment of 0.5 voxels (28 different values). Based on our experience, the *typical length* of a gap is at most 8 voxels for vessels in LSA regions in real 3D 7T MRA images. For the other two parameters we used $\alpha = \frac{\pi}{6}$ and $\sigma_n = 10$. As in the experiment above, addition of image noise was repeated 30 times, yielding $28 \times 30 = 840$ 3D images in total. The mean errors and standard deviations are shown in Fig. 5.4. For typical gap lengths ($l < 8$) voxels, FM-ADP and PROB perform better than FM-V. The mean error of PROB is larger than that of FM-V for $l \geq 11$ voxels. Furthermore, PROB performs somewhat better than FM-ADP for $l \leq 5$ voxels. However, for larger gaps FM-ADP achieves significantly better results. Overall, FM-ADP achieves the best result and a small mean error, even for gaps as long as 16 voxels. For gaps longer than 5 voxels, the results of PROB

5.1. Synthetic 3D Images

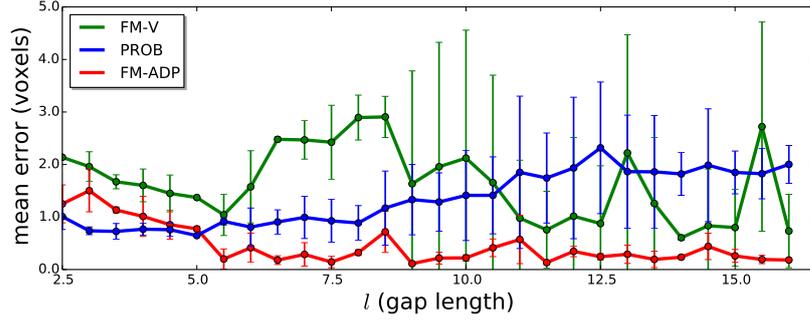


Figure 5.4: Same as Fig. 5.3, but using different gap lengths $l = 2.5, 3.0, 3.5, \dots, 16$, and fixed values $\alpha = \frac{\pi}{6}$ and $\sigma_n = 10$.

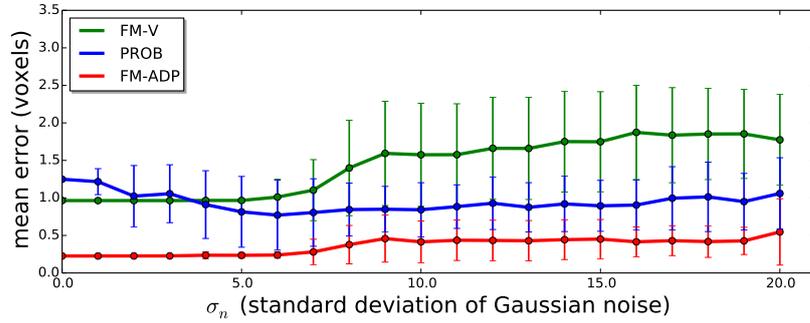


Figure 5.5: Same as Fig. 5.3, but using different noise levels $\sigma_n = 0, 1, 2, \dots, 20$, and fixed values $\alpha = \frac{\pi}{6}$ and $l = 6$.

and FM-V typically exhibit a higher standard deviation. In contrast, for all gap lengths, FM-ADP yields consistently a low standard deviation.

5.1.1.3 Varying Noise

In this experiment, the robustness of our approaches against image noise is investigated. In real 3D 7T MRA images of the LSA region, the noise can be described by a Gaussian distribution with a standard deviation of $\sigma_n = 10$. We used a large range for the noise level which is twice as large as the observed standard deviation. Specifically, we chose $\sigma_n = 0, 1, 2, \dots, 20$ (21 different values). The other two parameters are held constant at $\alpha = \frac{\pi}{6}$ and $l = 6$. The addition of image noise was repeated 30 times, yielding $21 \times 30 = 630$ 3D images in total. Fig. 5.5 shows the results.

For higher noise levels ($\sigma_n \geq 5$), both PROB and FM-ADP outperform FM-V in terms of mean error. The lowest mean error is obtained by FM-ADP which remains consistently lower than 0.5 voxels. Generally, the standard deviation increases with higher noise levels. For noise levels around the typical value in the

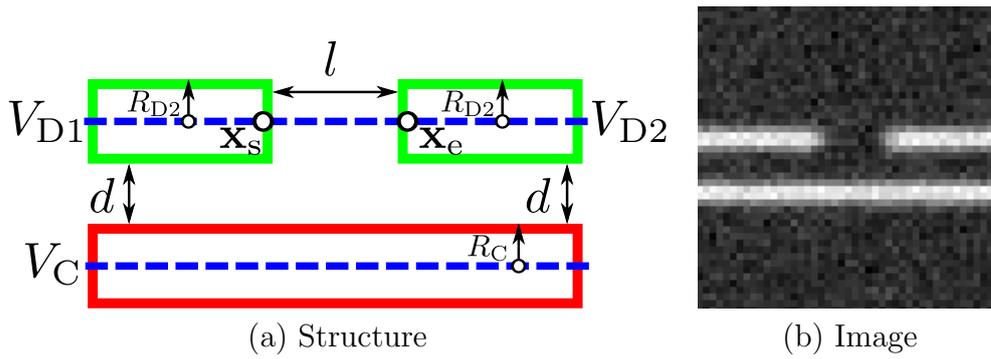


Figure 5.6: Synthetic 3D parallel vessels. (a) Sketch of parallel vessels with a gap used for quantitative evaluation. The upper vessel consists of two segments V_{D1} and V_{D2} , each at the same distance d to the lower vessel V_C . There is a gap of length l between them, which starts at \mathbf{x}_s and ends at \mathbf{x}_e . Each vessel has the same radius: $R_C = R_{D1} = R_{D2}$. The image contains Gaussian noise with standard deviation of σ_n . (b) 2D section of an example of a 3D synthetic image based on (a).

real 3D 7T MRA images ($\sigma_n = 10$), both PROB and FM-ADP achieve significantly lower mean error and standard deviations than FM-V. Furthermore, FM-ADP yields consistently the smallest mean error and standard deviation among the three approaches.

5.1.2 Parallel Vessels

In this experiment, 3D synthetic images are used which contain two parallel vessels, one of which has a gap due to low contrast. For an example of parallel vessels in 3D 7T MRA images, see Fig. 5.11a. A sketch of this configuration is provided in Fig. 5.6. All images have the same size of $128 \times 128 \times 32$ voxels, and all vessels have the same small radius $R_C = R_{D1} = R_{D2} = 1$ voxel as well as the same mean intensity $g_C = g_{D1} = g_{D2} = 200$. The vessel parts V_{D1} and V_{D2} are both parallel to V_C with a distance $d = 5$ voxels. The intensity in the gap is $g_{bg} = 50$, which is the same as the background intensity. The gap length l varies between 1 and 10 voxels (10 different values), and Gaussian noise with $\sigma_n = 10$ is added 30 times, yielding $10 \times 30 = 300$ 3D images in total. The mean error and standard deviation for each experiment are shown in Fig. 5.7.

For smaller gaps with $l \leq 6$, all three approaches have very similar performance. However, for larger gaps, FM-ADP performs significantly better than the other two approaches. The mean error and standard deviation of both PROB and FM-V increase quickly with growing l , while FM-ADP consistently achieves a low mean error and small standard deviation.

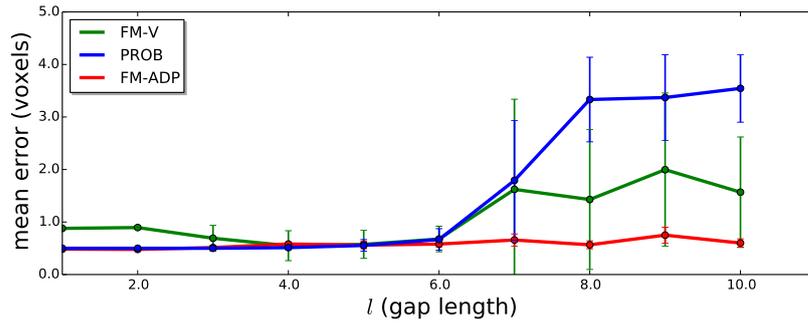


Figure 5.7: Mean error and standard deviation for different *gap lengths* for parallel vessels: The gap of length l between vessel parts V_{D1} and V_{D2} varies from 1 to 10, with an increment of 1.

5.2 Real 3D 7T MRA Images

We have also conducted extensive experiments using 40 high-resolution 3D 7T MRA images of the region around the human LSAs. All images have been assessed qualitatively and the results were used for clinical studies. For 10 of the 40 images, we carried out a quantitative evaluation by comparing the results with ground truth. Furthermore, the results of our explicit approach (PROB) and of our implicit approach (FM-ADP) are compared with a previous fast marching approach (FM-V) in [41].

5.2.1 Materials

The 3D 7T MRA images of the LSA regions were acquired without contrast agent using a 3D gradient echo time-of-flight sequence with a repetition time of 15 ms, echo times of 4.84 ms or 4.85 ms, a field of view of $135 \text{ mm} \times 180 \text{ mm}$, and flip angles of 25 or 30 degrees. We applied isotropic interpolation, yielding 128-162 slices for each 3D image. The size of the slices varies from 128×270 to 162×417 voxels. The 3D images have an isotropic resolution of 0.23 mm. The images are acquired from healthy subjects and patients with stroke or vascular dementia.

5.2.2 Segmentation Results

We have successfully applied our hybrid approach to all 40 3D 7T MRA images. For example, Fig. 5.8 shows the result for one 3D image of the LSA region. It can be seen that thick and thin vessels have been well segmented. In addition, local vascular parameters such as the vessel radius are automatically determined. In Fig. 5.8, the vessel radius is visualized by a color map, with radii ranging between

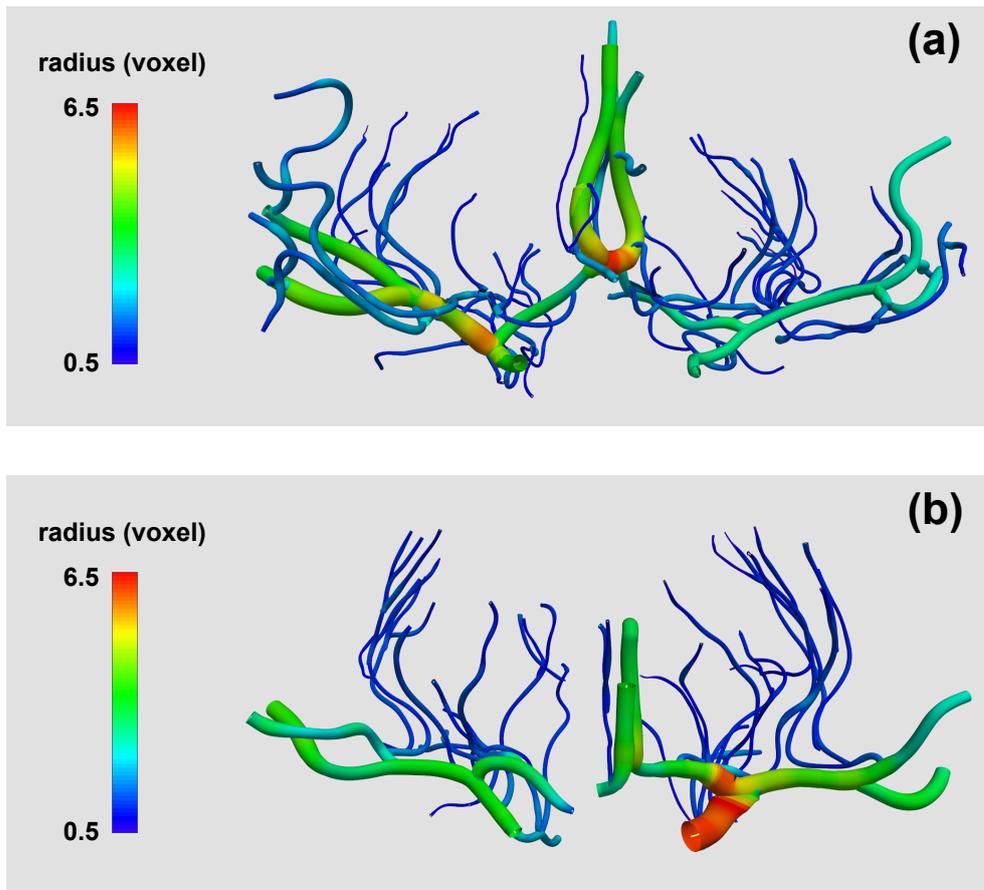
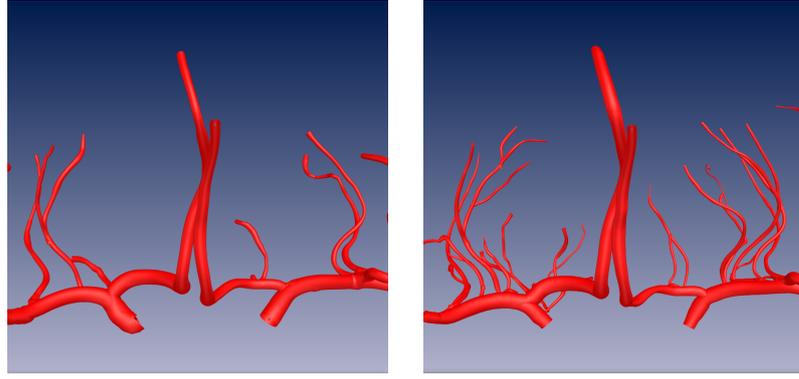


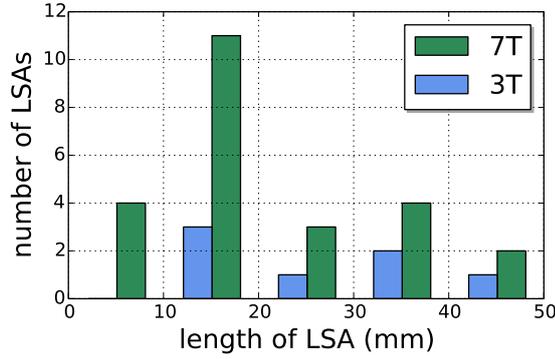
Figure 5.8: Segmentation and quantification result of vessels in the LSA region from a real 3D 7T MRA image using the proposed hybrid approach with FM-ADP. The radius is color-coded and ranges from 0.5 to 6.5 voxels. Red color corresponds to a larger radius, blue color corresponds to a smaller radius. (a) and (b) show the LSAs in the same region of two patients.

0.5 (blue) and 6.5 (red) voxels.

To demonstrate the benefit of 7T MRA in comparison to lower field MRA, we segmented the LSAs from 3T and 7T images, both acquired from the same subject. The original images are shown in Fig. 1.4 and the results are displayed in Fig. 5.9. A qualitative and quantitative comparison shows that using 7T images significantly more vessels can be segmented than using 3T images. For example, using the 7T image (Fig. 5.9b) at least twice as many vessels are segmented compared to the 3T image (Fig. 5.9a) for each interval of considered vessel lengths in Fig. 5.9c (0 to 10 mm, 10 mm to 20 mm, etc.). In total, 24 vessels are segmented using the 7T image, compared to 7 vessels using the 3T image. In particular, with 7T much more short vessels were successfully segmented. For example, in the 7T image, 11 vessels between 10 and 20 mm were segmented, but in the 3T image there were only



(a) Result for 3T MRA image (b) Result for 7T MRA image



(c) Quantitative comparison of the results in (a) and (b)

Figure 5.9: Comparison of the segmented LSAs using (a) 3T MRA and (b) 7T MRA images of the same subject, as well as (c) the number of vessels and their lengths for (a) and (b), shown as a histogram with a bin size of 10 mm.

3. Furthermore, in the 3T image no vessels shorter than 10 mm were segmented. In Figures 5.10 and 5.11 we provide more detailed examples of segmentation results and gap completion. Fig. 5.10a shows the original image (volume rendering) of a branching vessel. This configuration corresponds to the synthetic images used for evaluation in Sect. 5.1.1 above. Using the model-based approach, the branch (red) and the horizontal thick vessel (yellow) can be well segmented (Fig. 5.10b). The remaining gap between them is completed using minimal path approaches (magenta). FM-V does not take into account the initial direction, and therefore

Table 5.1: Quantitative results for 508 gaps (errors in mm).

Approach	Total	Centerline error \bar{e}_γ					Avg. run time per 3D image
		$\bar{e}_\gamma < 0.3$	$\bar{e}_\gamma < 0.4$	$\bar{e}_\gamma < 0.5$	$\bar{e}_\gamma < 0.6$	Mean \bar{e}_γ	
FM-V	99.61%	34.45%	61.22%	83.46%	92.72%	0.406	0.92 s
PROB	96.65%	56.69%	76.18%	86.61%	90.35%	0.313	51.68 s
FM-ADP	99.61%	61.22%	86.42%	94.29%	97.64%	0.288	1.63 s

it yields a short cut so that the completed gap lies almost completely outside the true vessel, and the transition between both parts is not smooth (Fig. 5.10c). In contrast, both PROB and FM-ADP incorporate the initial direction determined by the model-based approach, and consequently the results follow the true vessel, and the transition is smooth (Fig. 5.10d, e). Fig. 5.11 in addition shows that our approaches deal better with parallel vessels than FM-V. This configuration corresponds to the synthetic images used for evaluation in Sect. 5.1.2 above. Without employing the directional prior, FM-V connects the red part incorrectly to a parallel vessel since this short cut is much shorter than the correct result. In contrast, both PROB and FM-ADP connect the red part to the correct thick vessel, while keeping the transition between both parts smooth. Note that in both cases shown in Fig. 5.10 and Fig. 5.11, the centerline in the gap is not simply a straight line. Instead, the centerline follows the desired vessel, which is typically curved.

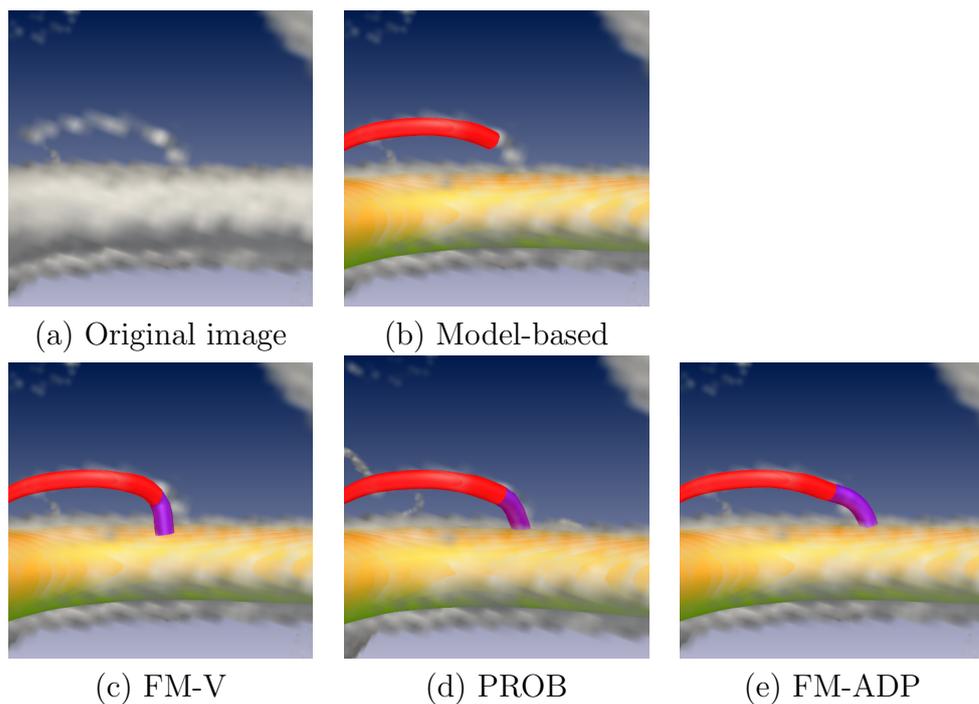


Figure 5.10: Segmentation results for a branching vessel in a real 3D 7T MRA image. Red parts are results of the model-based approach, magenta parts are results of gap completion. (a) Volume rendering of original image. (b) Results of model-based approach. (c) Result of previous FM-V. (d) Result of PROB. (e) Result of FM-ADP.

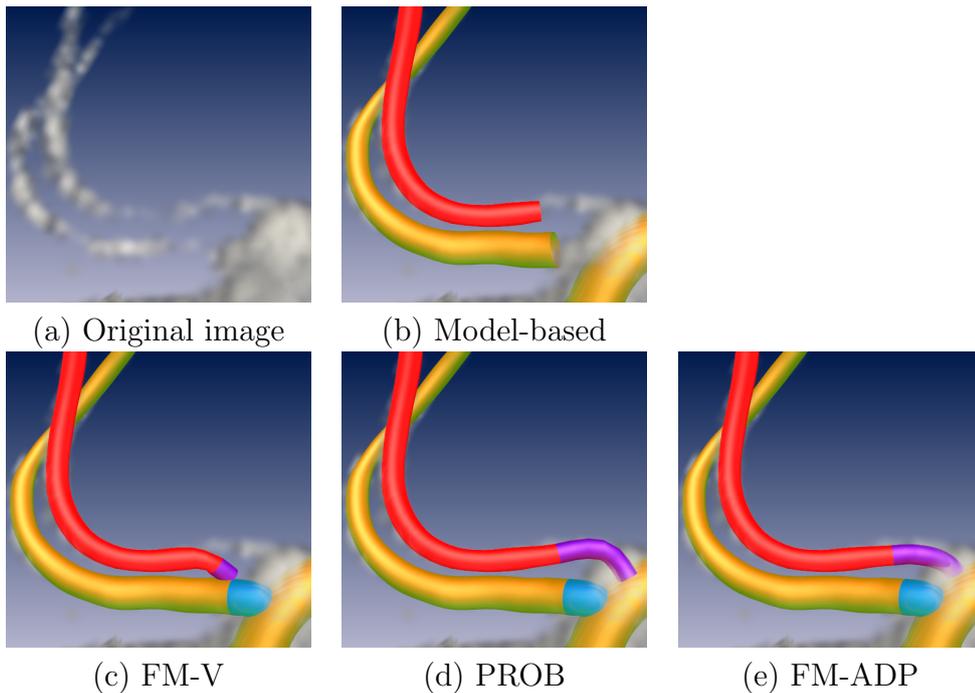


Figure 5.11: Same as Fig. 5.10 but for two parallel thin vessels and completion of two gaps (magenta and cyan).

5.2.3 Quantitative Evaluation

We have also quantitatively evaluated the segmentation results using manually created ground truth for 508 gaps in 10 3D 7T MRA images. The results of our minimal path based approaches for gap completion, i.e., PROB and FM-ADP, are compared with a previous fast marching approach (FM-V) [41] using the error measure defined in (5.1). Table 5.1 shows a summary of this comparison as percentages of the total numbers of completed gaps, percentages of the numbers of completed gaps with an error smaller than 4 thresholds, i.e., $\bar{e}_\gamma < 0.3, 0.4, 0.5$, and 0.6 mm, the mean error of all completed gaps, and the average run time per 3D image. It turned out that using fast marching-based approaches, i.e., FM-V and FM-ADP, 99.61% of the gaps are completed, which is slightly more than the results using PROB (i.e., 96.65%). For all considered error thresholds, FM-ADP leads to a higher number of completed gap centerlines closer to the ground truth (e.g., for $\bar{e}_\gamma < 0.3$ mm we have 61.22%, compared to 34.45% for FM-V and 56.69% for PROB), while the result of PROB is better than that of FM-V for the cases $\bar{e}_\gamma < 0.3$, $\bar{e}_\gamma < 0.4$, and $\bar{e}_\gamma < 0.5$. Furthermore, for FM-ADP a lower mean error of 0.288 mm compared to 0.406 mm (FM-V) and 0.313 mm (PROB) is achieved. In addition, a paired t-test was performed. It turned out that both differences of FM-ADP with respect to FM-V and PROB are significant, as the corresponding

Table 5.2: Overview of the six subjects in the stroke study.

Subject type	Number	Infarct	Recovered
A	2	Yes	Yes
B	2	Yes	No
C	2	No	–

p-values are lower than 0.00001 in both cases. Another advantage of FM-ADP is that it is much more efficient than PROB, and only slightly slower than FM-V. The average run time for completing all gaps in one 3D image is 1.63 seconds (single-thread, 2.67 GHz CPU), while PROB needs 51.68 seconds.

5.2.4 Clinical Studies

Furthermore, two clinical studies have been carried out using quantitative parameters determined by our approach. In these studies, the morphological properties of the LSAs were investigated, and the relationship between these properties and serious diseases, specifically stroke and vascular dementia, was examined. In the following, we briefly describe these studies.

5.2.5 Stroke

In this study, vessels related to an infarct area were investigated. Six subjects were examined, which belong to three classes: Subjects of class A had an infarct but are fully recovered, while subjects of class B also had an infarct but are not fully recovered. Subjects of class C had no diseases in cerebral vessels at all (see Table 5.2).

The quantitative parameters used in this study include mean radius, length, distance between the start and end point of the vessels, and 3D tortuosity. The study revealed that class B subjects exhibit overall degradation or loss of thin vessels, compared to subjects of class A or C. It also turned out that the number of the LSAs alone is not sufficient to characterize the relationship between the infarct and vessel properties. For example, class A subjects have a similar number of LSAs as class C subjects. However, by comparing other quantitative morphological properties (e.g., length, radius), it was found that the infarct-related LSAs of class A subjects are not only different from the LSAs of class C subjects, who never had an infarct, but are also different from their own LSAs which were not affected

by the infarct. The study demonstrates the importance of the analysis of the morphological properties of LSAs (see [180] for more details).

5.2.6 Vascular Dementia

In this study, the status of LSAs of patients with subcortical vascular dementia was investigated for the first time. In total, 24 subjects were examined. 12 subjects had this disease, while the other 12 subjects were age- and sex-matched normal controls. The number of vessels as well as morphological properties (e.g., length, tortuosity, and radius of individual vessels) were used to assess the abnormalities of the LSAs. The study showed that the LSAs of the patients with subcortical vascular dementia have larger radii and higher tortuosity than normal controls, while the number of LSAs is smaller. Also, this study demonstrates that segmentation and quantification of vessels are crucial for distinguishing between normal and pathological cases (see [181] for more details).

5.2.7 Aging Effect

In this study, we analyzed 112 images in order to study the morphological change of LSAs due to aging.

5.3 Synthetic 2D Images

We carried out experiments using synthetic images and real retinal images from the DRIVE dataset [24]. We compared our approach with the traditional fast marching approach (traditional FM) [41], which uses gradient information, and the recent method of Li and Yezzi [154] (Li-Yezzi). For our approach, the user provides two start points to specify the start position and initial direction. For the other approaches, the user provides one start point. In all cases, an end point is also provided by the user. In Fig. 5.12 and Fig. 5.13, the segmentation results of the three approaches are highlighted by blue stripes and the centerlines are shown by yellow lines. The red and green circles represent the radii at the start and end points, respectively. User-specified points are marked with a yellow cross.

We have used different synthetic images. For example, Fig. 5.12 (top) shows an image of an ideal dark curve with significant salt and pepper noise as well as Gaussian noise. In this case, both the traditional FM and Li-Yezzi yield a short cut, while our approach is able to segment the entire structure. A second example is shown in Fig. 5.12 (bottom), which represents a loop. Here, the traditional FM

and Li-Yezzi turn directly to the end point at the crossing, while our approach correctly follows the natural path and passes the crossing twice. A very important difference to previous minimal path approaches is the correct segmentation of the crossing. In these approaches, each pixel in the image is visited at most once by the wavefront. Consequently, a vessel crossing itself cannot be segmented correctly since it would require the pixels at the crossing to be visited twice by the wavefront. In contrast, we use a ROI in each iteration, therefore in each *iteration* the pixels in the ROI can be visited at most once, but in different iterations the same pixels can be visited repeatedly. As a result, the crossing can be segmented correctly, as shown in Fig. 5.12, second row.

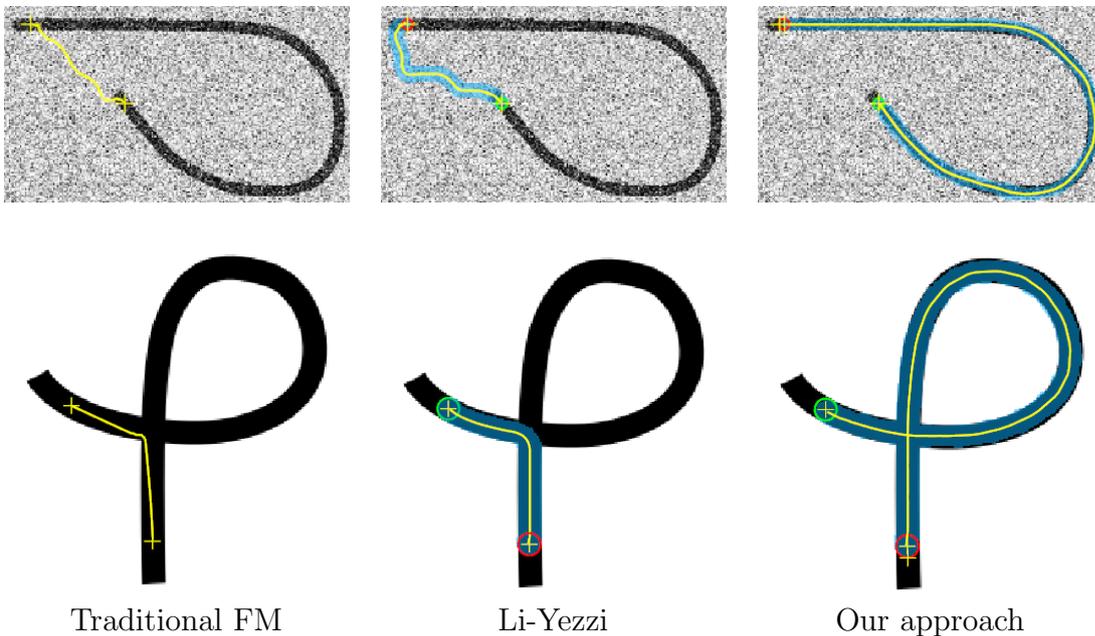


Figure 5.12: Comparison of different fast marching methods for a synthetic curve with noise (top), and a loop (bottom).

5.4 Real 2D Retinal Images

To demonstrate that our approach copes well with long vessels, for which one-pass fast marching approaches have difficulties (short cuts, problems with crossings), we have successfully applied our approach to 91 long vessels from 10 images of the DRIVE dataset. For example, Fig. 5.13 shows one of the segmented vessels. The approaches were applied to the full size images (565×584 pixels), and the results have been cropped for better visibility. It can be seen that our approach correctly deals with the three crossings (one in the red box 1, two in the red box

2), while both the traditional FM and the Li-Yezzi follow a wrong path at some of the crossings.

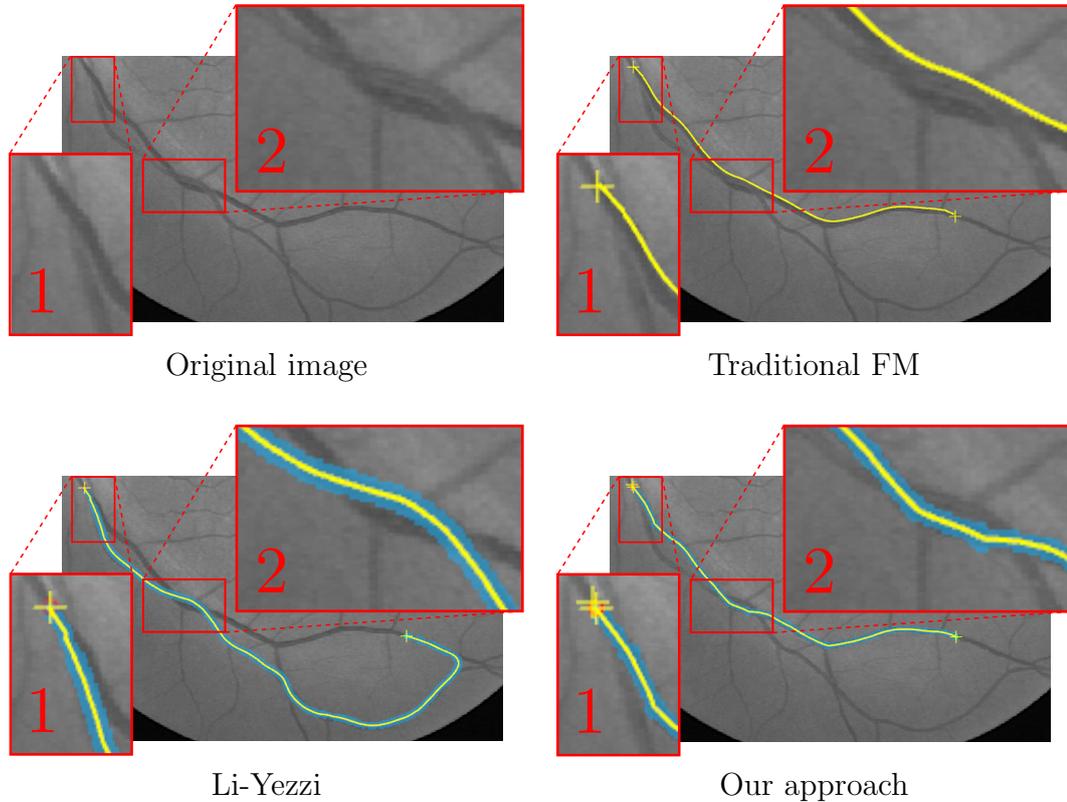


Figure 5.13: Comparison of different fast marching methods for an image from the DRIVE dataset.

We also conducted a quantitative evaluation. The segmentation accuracy has been measured by counting the number of overlapping pixels of the segmentation result with ground truth provided by the DRIVE dataset. In the ground truth, vessel pixels are labeled as foreground and all other pixels as background. Note that the ground truth does not contain information about the centerlines, therefore the traditional FM, which only segments centerlines, could not be evaluated. In addition to the accuracy in terms of true positive rate (TP) and false positive rate (FP), the computational efficiency has been measured by the ratio of the run time of our approach and that of Li-Yezzi (run time percentage). The results for accuracy are shown in Fig. 5.14a, b. In Fig. 5.14c, a comparison of run time of our approach and Li-Yezzi is shown. It turns out that our approach successfully segmented most vessels with a higher accuracy than Li-Yezzi. In terms of mean true positives (\overline{TP}) and mean false positives (\overline{FP}), our approach achieved $\overline{TP} = 0.82$ and $\overline{FP} = 0.21$, in contrast to $\overline{TP} = 0.51$ and $\overline{FP} = 0.75$ achieved by Li-Yezzi. In addition to the higher \overline{TP} and lower \overline{FP} , in both cases

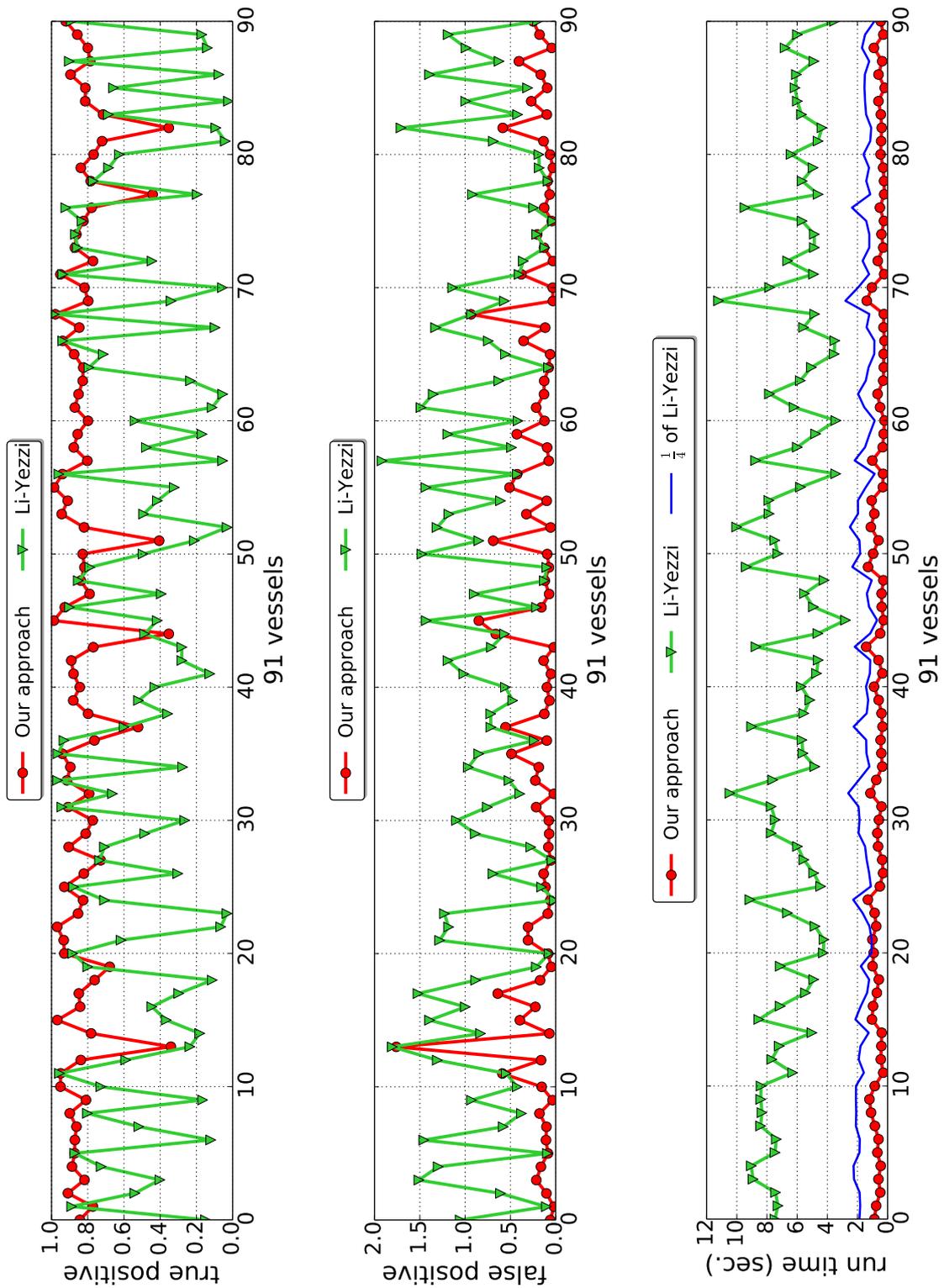


Figure 5.14: Quantitative evaluation using retinal images from the DRIVE dataset.

we achieved a smaller standard deviation. In our approach, the low true positive and high false positive rates for some vessels (e.g., vessels 14 and 45) are caused

5.4. Real 2D Retinal Images

	\overline{TP}	\overline{FP}	Avg. run time
Li-Yezzi	0.51±0.31	0.75±0.48	6.33±1.78
Our approach	0.82±0.13	0.21±0.25	0.59±0.32

Table 5.3: Evaluation of the iterative fast marching approach and comparison with Li-Yezzi: The mean values and standard deviations of \overline{TP} , \overline{FP} , and run time.

by the very low image contrast and high noise. For segmenting all vessels, our approach requires in the worst case less than 25%, and on average 9.32% of the computation time of Li-Yezzi. To better illustrate this, the curve representing 25% run time of Li-Yezzi is displayed as a blue curve in Fig. 5.14c. A summary of the quantitative results is provided in Table 5.3.

In Fig. 5.15, our approach is compared with Li-Yezzi exemplarily using four relatively long vessels, which demonstrate the challenges of retinal vessel segmentation. The true positives, false positives and false negatives are displayed using green, red, and blue colors, respectively. It turns out that for images with high contrast and low noise, and for vessels with a relatively simple structure, i.e., without bifurcations, both Li-Yezzi and our approach achieve similar results (Fig. 5.15, first row). For vessels with bifurcations, it is crucial to select the correct path. For the image in Fig. 5.15, second row, our approach is able to follow the path with less curvature, which is the correct one. In contrast, Li-Yezzi follows a wrong path. Also for vessels with a cluttered background in an image with low contrast, our approach using a directional prior proves to be effective. For example, in Fig. 5.15, third row, the contrast is low, and the background contains many blurry line structures which have similar intensities and shapes as the vessels. In spite of the difficulties, our approach yields the correct vessel, while Li-Yezzi results in a wrong path. This is partly due to the fact that Li-Yezzi relies on the assumption that vessels have homogeneous intensity, and the intensity distribution inside the vessels is very different from that outside the vessels. With a cluttered background, this assumption is not valid, and consequently the vessels cannot be well distinguished from the background. A further example shows that even without the background clutter, the homogeneity assumption of Li-Yezzi does not always hold. In Fig. 5.15, fourth row, there is little clutter in the background, but inside the vessel to be segmented, the intensities vary strongly. For example, at one end (close to the upper right corner of the image), the intensity is high, while at the other end (close the lower left corner of the image), the intensity is low. Consequently, Li-Yezzi follows a path similar to the end with high intensity, which is

not correct. In contrast, our approach finds the correct vessel.

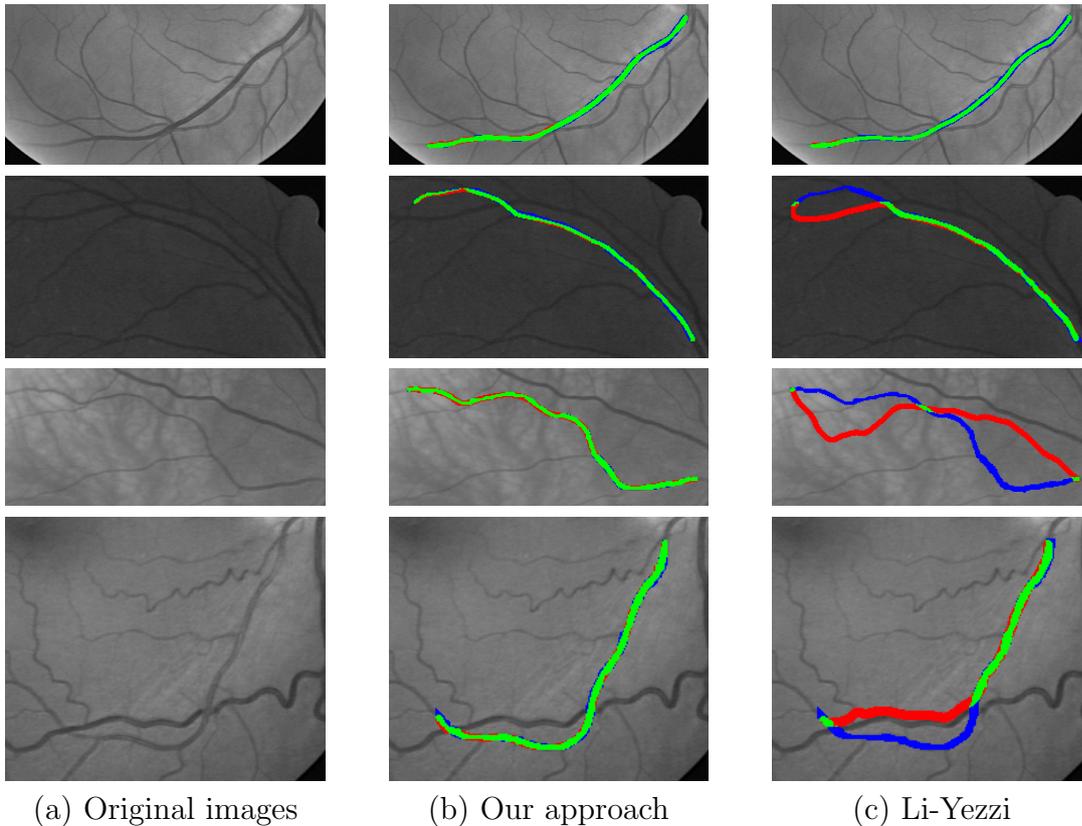


Figure 5.15: Example results for our approach and Li-Yezzi using retinal images from the DRIVE dataset. True positives, false positives and false negatives are displayed using green, red, and blue colors, respectively. (a) Original images. (b) Second row: Results of our approach. (c) Results using the approach Li-Yezzi [154]. First column: For high image contrast and low noise, and a relatively simple vessel structure, both methods achieve similar results. Second column: At a bifurcation, Li-Yezzi follows a wrong path. Third column: For low image contrast and strongly cluttered background, Li-Yezzi fails to segment the true vessel. Fourth column: If the intensity distribution is not homogeneous along a vessel, Li-Yezzi cannot segment the vessel correctly.

Chapter 6

Progressive Minimal Path Method

In this chapter, we propose a novel minimal path method for the segmentation of 2D and 3D line structures. Minimal path methods perform propagation of a wavefront emanating from a start point at a speed derived from image features, followed by path extraction using backtracing. Usually, the computation of the speed and the propagation of the wave are two *separate* steps, and *point features* are used to compute a *static* speed. We introduce a new continuous minimal path method which steers the wave propagation *progressively* using *dynamic* speed based on *path features*. We present three instances of our method, using an appearance feature of the path, a geometric feature based on the curvature of the path, and a joint appearance and geometric feature based on the tangent of the wavefront. These features have not been used in previous continuous minimal path methods. We compute the features *dynamically* during the wave propagation, and also *efficiently* using a fast numerical scheme and a low-dimensional parameter space. Our method does not suffer from discretization or metrication errors. The work in this chapter was published in [47–49].

This chapter is organized as follows. We introduce our new progressive minimal path method in Sect. 6.1. Three instances of our framework are presented in Sect. 6.2. Experimental results are described in Chapter 7.

6.1 Minimal Path Method with Dynamic Speed

In this section, we introduce a new progressive minimal path framework using dynamic speed.

6.1.1 Dynamic Speed

In most previous minimal path approaches, the computation of the speed and the wave propagation are two consecutive steps. First, a *static* speed $\mathcal{F}_{\text{stat}}$ is computed, and then it is used for wave propagation without update. Note that $\mathcal{F}_{\text{stat}}$ does not allow incorporating information about regions which were already visited by the wavefront. Such information is only available *during* wave propagation and changes as soon as the wavefront changes. To exploit such information, we introduce a *dynamic* speed function \mathcal{F}_{dyn} . Our new energy function is formulated as:

Energy function of the progressive minimal path method

$$E(\gamma) := \int_{\gamma} \left(\mathcal{P}_{\text{stat}}(\gamma(s)) + \mathcal{P}_{\text{dyn}}(\gamma(s)) + w \right) ds, \quad (6.1)$$

$$\mathcal{P}_{\text{dyn}}(\mathbf{x}) = \begin{cases} 0, & \text{if } \gamma(\mathbf{x}) \text{ satisfies constraint } \mathcal{T} \\ \mathcal{C}_0(\mathbf{x}), & \text{otherwise,} \end{cases} \quad (6.2)$$

$$\mathcal{C}_0(\mathbf{x}) = \mathcal{C}_{\text{max}} - \mathcal{P}_{\text{stat}}(\mathbf{x}), \quad (6.3)$$

where $\mathcal{P}_{\text{stat}}$ is the static potential computed before the propagation starts, and \mathcal{P}_{dyn} is the dynamic potential computed during the propagation. \mathcal{T} is an application-specific constraint for paths, and \mathcal{C}_{max} is a large positive constant (e.g., $\mathcal{C}_{\text{max}} = 10^7$). Our goal is to compute the minimal action map of the composite speed $\mathcal{F}_{\text{comp}}$, which consists of the static speed $\mathcal{F}_{\text{stat}}$ and the dynamic speed \mathcal{F}_{dyn} :

$$u_{\mathbf{x}_s}(\mathbf{x}) := \min_{\gamma \in \mathcal{A}_{\mathbf{x}_s, \mathbf{x}}} \int_{\gamma} \left(\frac{1}{\mathcal{F}_{\text{comp}}(\gamma(s))} + w \right) ds, \quad (6.4)$$

$$\mathcal{F}_{\text{comp}}(\mathbf{x}) = \mathcal{F}_{\text{stat}}(\mathbf{x}) + \mathcal{F}_{\text{dyn}}(\mathbf{x}), \quad (6.5)$$

$$\mathcal{F}_{\text{dyn}}(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathcal{P}_{\text{dyn}}(\mathbf{x}) = 0, \\ -\mathcal{F}_{\text{stat}}(\mathbf{x}) + \frac{1}{\mathcal{C}_{\text{max}}}, & \text{if } \mathcal{P}_{\text{dyn}}(\mathbf{x}) = \mathcal{C}_0(\mathbf{x}). \end{cases} \quad (6.6)$$

Note that the formulation of the energy using the potential in (6.1) is equivalent to the one using the speed in (6.4), i.e., for $\mathcal{P}_{\text{stat}} = \frac{1}{\mathcal{F}_{\text{stat}}}$ and $\mathcal{P}_{\text{dyn}} = \frac{1}{\mathcal{F}_{\text{dyn}}}$ we have

$$\mathcal{P}_{\text{stat}} + \mathcal{P}_{\text{dyn}} = \frac{1}{\mathcal{F}_{\text{stat}} + \mathcal{F}_{\text{dyn}}}. \quad (6.7)$$

6.1. Minimal Path Method with Dynamic Speed

This can be seen as follows. Since \mathcal{P}_{dyn} is either 0 or \mathcal{C}_0 , we only have to consider two cases. In the first case, $\mathcal{P}_{\text{dyn}} = 0$. According to (6.6), \mathcal{F}_{dyn} is also 0, so $\mathcal{P}_{\text{stat}} + \mathcal{P}_{\text{dyn}} = \mathcal{P}_{\text{stat}} = \frac{1}{\mathcal{F}_{\text{stat}}} = \frac{1}{\mathcal{F}_{\text{stat}} + \mathcal{F}_{\text{dyn}}}$. In the second case, $\mathcal{P}_{\text{dyn}} = \mathcal{C}_0$. Using (6.3), we have $\mathcal{P}_{\text{stat}} + \mathcal{P}_{\text{dyn}} = \mathcal{C}_{\text{max}}$. Furthermore, (6.6) implies $\mathcal{F}_{\text{stat}} + \mathcal{F}_{\text{dyn}} = \frac{1}{\mathcal{C}_{\text{max}}}$, so also in this case, $\mathcal{P}_{\text{stat}} + \mathcal{P}_{\text{dyn}} = \frac{1}{\mathcal{F}_{\text{stat}} + \mathcal{F}_{\text{dyn}}}$.

6.1.2 Progressive Minimal Path Framework

The main idea of our progressive minimal path method is to steer the wave propagation using *dynamically* computed features. In classical fast marching approaches, the point \mathbf{x}_{min} on the wavefront \mathcal{W} with *minimum* arrival time $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ is the position where the wave propagates in the current step, i.e., \mathcal{W} advances to the neighbors of \mathbf{x}_{min} outside \mathcal{W} . In our method, we decide during *each step* of the propagation whether to advance \mathcal{W} or not. At \mathbf{x}_{min} , we compute a feature $f(\mathbf{x}_{\text{min}}, \mathcal{R}_A)$ within the region \mathcal{R}_A (Alive) and check if it satisfies a certain *constraint*. The features and their constraints depend on the instance of our framework (see Sect. 6.2 below). We use features that are determined from a *local path* γ_{local} , which is extracted using gradient descent of $\mathcal{U}_{\mathbf{x}_s}$ inside \mathcal{W} . If γ_{local} *violates* the constraint, then the composite speed $\mathcal{F}_{\text{comp}}(\mathbf{x}_{\text{min}})$ is reduced, which corresponds to the second case in (6.6) above, and the propagation through \mathbf{x}_{min} is slowed down. If γ_{local} satisfies the constraint, then the composite speed $\mathcal{F}_{\text{comp}}$ remains unchanged ($\mathcal{F}_{\text{dyn}} = 0$), and \mathcal{W} advances as in classical fast marching approaches, which corresponds to the first case in (6.6). The update *during* the propagation of \mathcal{W} is the *dynamic* aspect of our speed $\mathcal{F}_{\text{comp}}$.

One step of the wave propagation incorporating dynamic speed is illustrated in Fig. 6.1. The wavefront \mathcal{W} , i.e., the region \mathcal{R}_T , is shown as an orange stripe. Inside \mathcal{W} (green region \mathcal{R}_A), $\mathcal{U}_{\mathbf{x}_s}$ is known since \mathcal{W} has already visited all image points there, but outside \mathcal{W} (blue region \mathcal{R}_F), $\mathcal{U}_{\mathbf{x}_s}$ is unknown. Let $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ be points on \mathcal{W} (Fig. 6.1a). The point \mathbf{x}_{min} with minimum $\mathcal{U}_{\mathbf{x}_s}$, i.e. \mathbf{x}_2 , is considered. The substeps illustrated in Fig. 6.1b and 6.1c are the main difference to classical fast marching approaches. Starting at \mathbf{x}_2 , a local path $\gamma_{\text{local}}(\mathbf{x}_2)$ with a fixed length Γ is extracted (Fig. 6.1b). This is possible since $\mathcal{U}_{\mathbf{x}_s}$ is known inside \mathcal{W} . After that, it is examined whether γ_{local} satisfies the constraint (Fig. 6.1b). If the constraint is *violated*, then $\mathcal{F}_{\text{comp}}(\mathbf{x}_2)$ should be reduced, and $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_2)$ re-computed, while \mathcal{W} does *not* change (Fig. 6.1c). Obviously, $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_2)$ increases, so the propagation through \mathbf{x}_2 (with $\gamma_{\text{local}}(\mathbf{x}_2)$ violating the constraint) is slowed down. If the constraint is satisfied, then we proceed as in classical fast marching approaches:

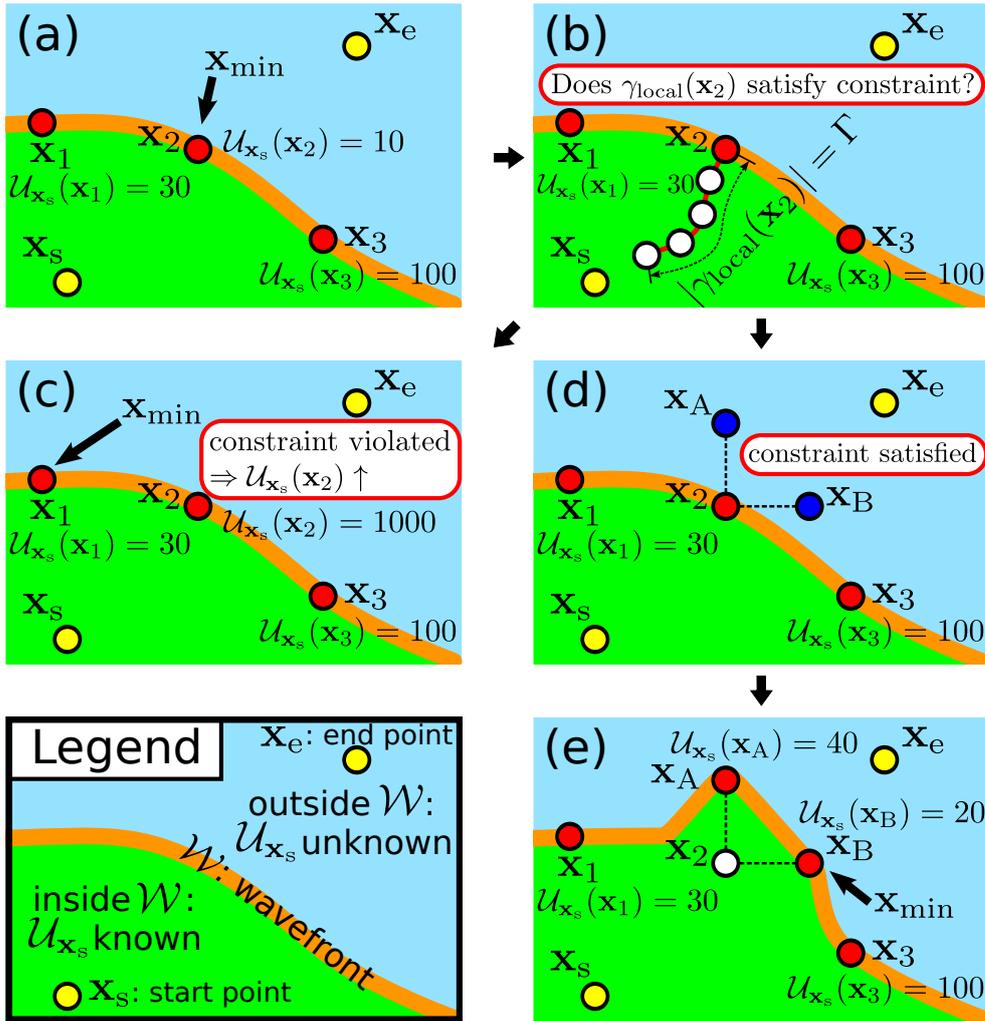


Figure 6.1: One step of the wave propagation with dynamic speed. (a) Select the point \mathbf{x}_{\min} on \mathcal{W} with minimum arrival time. (b) For $\gamma(s) = \mathbf{x}_2$, extract $\gamma_{\text{local}}(\mathbf{x}_2)$ with length Γ , test if γ_{local} satisfies the constraint. (c) γ_{local} does not satisfy the constraint: $\mathcal{F}_{\text{comp}}(\mathbf{x}_2)$ is reduced and therefore $U_{x_s}(\mathbf{x}_2)$ increases. Select another point on \mathcal{W} . (d) γ_{local} satisfies the constraint: Compute arrival time for neighbors (same as in previous minimal path approaches). (e) \mathcal{W} advances. Green, blue, and orange regions represent \mathcal{R}_A (Alive), \mathcal{R}_F (Far), and \mathcal{R}_T (Trial), respectively.

The neighbors of \mathbf{x}_2 outside \mathcal{W} , i.e., \mathbf{x}_A and \mathbf{x}_B , are found (Fig. 6.1d) and \mathcal{W} advances to \mathbf{x}_A and \mathbf{x}_B , while \mathbf{x}_2 is moved into the inside of \mathcal{W} (Fig. 6.1e). In either case, the point on \mathcal{W} with minimum U_{x_s} value, i.e., \mathbf{x}_{\min} , will be considered in the next iteration.

6.1.2.1 Generic Functions of the Progressive Minimal Path Method

We introduce four generic functions for our progressive minimal path method, which can be tailored to a specific task: `NeedDynamicSpeed`, `ComputeFeature`,

`IsSatisfied`, and `ComputeDynamicSpeed`. The first three of these functions are *application specific*, and instances of them for three applications will be described below in Sections 6.2.1, 6.2.2, and 6.2.3, respectively. Generally speaking, `NeedDynamicSpeed` checks if the speed at a point \mathbf{x} needs to be updated, `ComputeFeature` computes a feature f of the region \mathcal{R}_A when the wavefront reaches \mathbf{x} , and `IsSatisfied` checks if f satisfies a constraint. Depending on the constraint, `ComputeDynamicSpeed` uses (6.2) and (6.6) to compute the dynamic speed \mathcal{F}_{dyn} (see Sect. 6.1.2.2 below). Note that there exists a significant difference to previous minimal path approaches based on a static speed: $\mathcal{F}_{\text{stat}}$ cannot incorporate f since it is computed *before* the wave propagation starts, and at this time \mathcal{R}_A is still unknown. The only way to compute f is to derive it dynamically when \mathcal{R}_A is known, i.e., *during* the wavefront propagation.

6.1.2.2 Computing the Dynamic Speed

Our framework can be applied using a continuous formulation based on the fast marching method, and using a discrete formulation based on Dijkstra’s algorithm. A unified algorithm for both continuous and discrete formulations is given in Algorithm 6.1. Let $\mathcal{N}(\mathbf{x})$ denote the set of direct neighbors of \mathbf{x} on a regular image grid. The difference between the fast marching method and Dijkstra’s algorithm is the function `ComputeAction` in lines 15 and 20 of Algorithm 6.1, which updates the temporary action map $\mathcal{U}_{\mathbf{x}_s}$ at the neighbors of \mathbf{x}_{min} (point with current minimum action). The fast marching method employs the *upwind scheme* [178] to compute a continuous action, while in Dijkstra’s algorithm, the action is computed in a discrete manner using the neighbor with minimum action. For each point $\mathbf{x} \in \Omega$, the dynamic speed \mathcal{F}_{dyn} should be computed at most once. This is ensured by the set \mathcal{M}_{dyn} initialized in line 7.

As the wavefront propagates, the arrival time for each \mathbf{x} is computed. When the end point \mathbf{x}_e is finally reached, we obtain the path using backtracing from \mathbf{x}_e with the predecessor operator. Under the assumption that the points involved in the backtracing do not depend on points with dynamic speed \mathcal{F}_{dyn} , one can straightforwardly show (based on the proof of the original Dijkstra’s algorithm) that the result $\gamma_{\mathbf{x}_s, \mathbf{x}_e}^*$ is globally optimal.

Algorithm 6.1: Progressive minimal path framework

Input: Start point \mathbf{x}_s , end point \mathbf{x}_e , speed function $\mathcal{F}_{\text{stat}}$
Output: Action map $\mathcal{U}_{\mathbf{x}_s}$

```

1 for each pixel coordinate  $\mathbf{x}$  do // Initialization
2   if  $\mathbf{x} = \mathbf{x}_s$  then
3      $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}) \leftarrow 0$ ;  $\mathcal{L}(\mathbf{x}) \leftarrow l_T$ ;
4   else
5      $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}) \leftarrow \infty$ ;  $\mathcal{L}(\mathbf{x}) \leftarrow l_F$ ;
6  $\mathcal{F}_{\text{dyn}} \leftarrow \mathcal{F}_{\text{stat}}$ ;
7  $\mathcal{M}_{\text{dyn}} \leftarrow \emptyset$ ;
8 repeat // Main loop: progressive propagation
9    $\mathbf{x}_{\min} \leftarrow \arg \min_{\mathbf{x} \in \mathcal{R}_T} \mathcal{U}_{\mathbf{x}_s}(\mathbf{x})$ ;
10  if NeedDynamicSpeed( $\mathbf{x}_{\min}$ )
11  and  $\mathbf{x}_{\min} \notin \mathcal{M}_{\text{dyn}}$ 
12  and  $\mathbf{x}_{\min} \neq \mathbf{x}_s$  then // Update  $\mathcal{F}_{\text{dyn}}$ 
13     $f \leftarrow \text{ComputeFeature}(\mathbf{x}_{\min}, \mathcal{R}_A)$ ;
14     $\mathcal{F}_{\text{dyn}}(\mathbf{x}_{\min}) \leftarrow \text{ComputeDynamicSpeed}(f)$ ;
15     $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_{\min}) \leftarrow \text{ComputeAction}(\mathbf{x}_{\min}, \mathcal{F}_{\text{dyn}})$ ;
16     $\mathcal{M}_{\text{dyn}} \leftarrow \mathcal{M}_{\text{dyn}} \cup \mathbf{x}_{\min}$ ;
17  else // Same as standard minimal path
18     $\mathcal{L}(\mathbf{x}_{\min}) \leftarrow l_A$ ;
19    for  $\mathbf{x}_n \in \mathcal{N}(\mathbf{x}_{\min}) \cap (\mathcal{R}_F \cup \mathcal{R}_T)$  do
20       $\mathcal{U}_{\mathbf{x}_s}(\mathbf{x}_n) \leftarrow \text{ComputeAction}(\mathbf{x}_n, \mathcal{F}_{\text{comp}})$ ;
21      if  $\mathcal{L}(\mathbf{x}_n) = l_F$  then
22         $\mathcal{L}(\mathbf{x}_n) \leftarrow l_T$ ;
23 until  $\mathcal{L}(\mathbf{x}_e) = l_A$ ;
```

Theorem 6.1.2.1 (Run time of Algorithm 6.1). *If the run time of the classical fast marching approach is $\mathcal{O}(f_{\text{FM}}(N))$, then the worst case run time of our approach is $\mathcal{O}(f_{\text{FM}}(N) + m \cdot N)$, where N is the number of pixels/voxels and m is the time needed for the extraction and evaluation of one local path.*

Proof. Since the local paths have the same length, the time m needed to extract and evaluate one path is constant. In the worst case, the local path is computed for each image point, therefore the overall additional computational overhead is $m \cdot N$. \square

In the next section, we present three novel path features for our progressive minimal path method which cannot be incorporated into previous minimal path approaches based on static speed. With these features, segmentation results can be enhanced significantly. To segment line structures with our approach, we use

the fast marching method since it achieves subpixel accuracy, and consequently discretization and metrication errors are avoided.

6.2 Instances of the Progressive Minimal Path Method

We present three instances of our progressive minimal path method. For each instance, we describe the motivation and objective, followed by the used path features and constraints, and the corresponding functions `ComputeDynamicSpeed`, `ComputeFeature`, and `IsSatisfied`.

6.2.1 Appearance Feature

First, we introduce an *appearance* feature of local paths to cope with paths involving gaps due to artifacts.

6.2.1.1 Motivation and Objective

For line structures like the one in Fig. 1.8a, the start and end points are close to each other, and the structures contain gaps due to artifacts. For these cases, most previous minimal path approaches yield short cuts, which are much shorter than the true path. In contrast, we use an appearance path feature to avoid short cuts.

6.2.1.2 Feature and Constraint

Our static speed is based on a multiscale vesselness measure $\mathcal{V}(\mathbf{x})$ [50], which enhances line structures in images. For 2D images, we have:

Point feature: Vesselness of a pixel

$$\mathcal{F}_{\text{stat}}(\mathbf{x}) = \mathcal{V}(\mathbf{x}) + \epsilon = \max_{s_{\min} \leq s \leq s_{\max}} \mathcal{V}_s(\mathbf{x}) + \epsilon, \quad (6.8)$$

$$\mathcal{V}_s(\mathbf{x}) = \begin{cases} 0, & \text{if } \lambda_2 > 0 \\ \exp(A) (1 - \exp(B)), & \text{otherwise,} \end{cases} \quad (6.9)$$

$$A = -\frac{1}{\beta^2} \cdot \frac{\lambda_1^2}{\lambda_2^2}, \quad B = -\frac{1}{2c^2} \cdot (\lambda_1^2 + \lambda_2^2), \quad (6.10)$$

where ϵ is a small constant to avoid zero speed so that the potential is always well defined. The scale s ranges from s_{\min} to s_{\max} . λ_1 and λ_2 are the eigenvalues of the Hessian matrix at \mathbf{x} . β and c control the sensitivity of the filter. The value of \mathcal{V}_s is normalized to the interval $[0, 1]$.

The functions introduced in Sect. 6.1.2.1 above and used in Algorithm 6.1 are defined as follows. `NeedDynamicSpeed`(\mathbf{x}) returns true if $\mathcal{V}(\mathbf{x}) < T_v$, where T_v is a threshold for the vesselness, and false otherwise. `ComputeFeature` computes the appearance feature, i.e., *mean vesselness* $\bar{\mathcal{V}}(\gamma)$ for a path γ , which is defined as

Path feature: Mean vesselness of a path

$$\bar{\mathcal{V}}(\gamma) = \frac{1}{|\gamma|} \int_{\gamma} \mathcal{V}(\gamma(s)) ds, \quad (6.11)$$

where $|\gamma|$ denotes the length of γ . Our feature is derived from a *local path* $\gamma_{\text{local}}(\mathbf{x})$ of fixed length Γ (Fig. 6.1b), which is the last part of the path from \mathbf{x}_{\min} to \mathbf{x}_s . γ_{local} is computed by backtracing from \mathbf{x} until $|\gamma_{\text{local}}(\mathbf{x})| = \Gamma$. We use $\bar{\mathcal{V}}(\gamma_{\text{local}}(\mathbf{x}))$ in `IsSatisfied`, and γ_{local} *satisfies* the constraint if and only if $\bar{\mathcal{V}}(\gamma_{\text{local}}(\mathbf{x})) > T_v$:

IsSatisfied for appearance feature

$$\text{IsSatisfied}(\mathbf{x}) = \begin{cases} \text{true,} & \text{if } \bar{\mathcal{V}}(\gamma_{\text{local}}(\mathbf{x})) > T_v \\ \text{false,} & \text{otherwise,} \end{cases} \quad (6.12)$$

For example, suppose $\gamma_{\text{local}}(\mathbf{x})$ lies in a large gap between two really separated points, like \mathbf{x}_s and \mathbf{x}_e in Fig. 1.8a. There, $\bar{\mathcal{V}}$ is low, so γ_{local} violates the constraint at \mathbf{x} , and the speed $\mathcal{F}_{\text{comp}}$ is low according to the second case in (6.6). Therefore, the wavefront hardly propagates at \mathbf{x} , and short cuts can be avoided. In contrast, if γ_{local} covers a part of the line structure with small gaps due to artifacts, $\bar{\mathcal{V}}$ is only slightly lower compared to the case when γ_{local} lies completely inside the line structure, so the constraint is still satisfied, which corresponds to the first case in (6.6), where $\mathcal{F}_{\text{dyn}} = 0$, and $\mathcal{F}_{\text{comp}}$ is not changed. In our experiments, we used local paths of about 10 pixel length to determine the feature in (6.11). Note that, for example, in the discrete minimal path approach based on line graphs and static speed [56], only paths of length 3 can be handled.

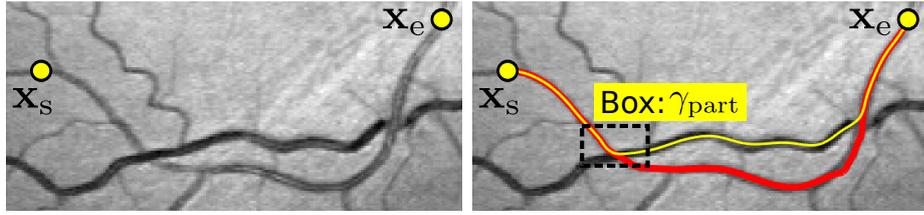


Figure 6.2: Left: Retinal vessels crossing each other (image from DRIVE dataset [24]). Right: Segmentation results. Yellow and red colors indicate the results of an approach with length regularization [41] and our curvature-based approach, respectively.

6.2.2 Geometric Feature

Second, we introduce a *geometric* feature, i.e., the *curvature* of local paths, to avoid paths with high curvature.

6.2.2.1 Motivation and Objective

The energy function of the classical fast marching approach in (4.12) can also be written as:

$$E(\gamma) = \int_{\gamma} \mathcal{P}(\gamma(s)) ds + w \cdot \int_{\gamma} 1 ds, \quad (6.13)$$

Since $\int_{\gamma} 1 ds$ is the length of the path γ [41], the regularization term in (6.13) assigns a lower energy to shorter paths. Usually, shorter paths are assumed to have a lower curvature, so previous approaches employ *length regularization* to indirectly limit the curvature of the path. However, this assumption is not always fulfilled. For example, retinal vessels in 2D images often cross each other, and the shortest path found using length regularization may belong to different vessels. This problem is illustrated in Fig. 6.2: Although the red path is correct, an approach with length regularization [41] yields the yellow path, since it is shorter. In 3D medical images, similar problems exist in regions with high noise level and low contrast. With a curvature feature, the curvature (smoothness) of vessels can be exploited *independently* of the length regularization, so shorter paths with wrong sharp turns can be avoided. We incorporate the curvature information by extracting local paths *during* the wave propagation.

6.2.2.2 Feature and Constraint

Compared to the energy function of the classical fast marching approach (6.13), in our approach the dynamic potential \mathcal{P}_{dyn} in (6.2) represents the curvature feature, independently of the length regularization. The static speed $\mathcal{F}_{\text{stat}}$ and the

local path γ_{local} are computed in the same way as in Sect. 6.2.1.2 above, but `NeedDynamicSpeed(x)` returns true for all points \mathbf{x} . `ComputeFeature` computes the curvature κ of γ_{local} . This is possible since the geometry of γ_{local} is determined explicitly. Suppose \mathbf{x}_f , \mathbf{x}_m , and \mathbf{x}_l are the first, middle, and last point of γ_{local} , respectively. Then, similar to [55], we used a measure for the curvature based on the angle between the vectors $\mathbf{d}_1 := \mathbf{x}_m - \mathbf{x}_l$ and $\mathbf{d}_2 := \mathbf{x}_f - \mathbf{x}_m$:

$$\kappa(\mathbf{d}_1, \mathbf{d}_2) = \arccos \left(\frac{\mathbf{d}_1^T \cdot \mathbf{d}_2}{\|\mathbf{d}_1\| \cdot \|\mathbf{d}_2\|} \right). \quad (6.14)$$

`IsSatisfied` returns true if $\kappa(\gamma_{\text{local}}(\mathbf{x})) < T_\kappa$, where T_κ is a threshold for the curvature:

IsSatisfied for geometric feature

$$\text{IsSatisfied}(\mathbf{x}) = \begin{cases} \text{true}, & \text{if } \kappa(\gamma_{\text{local}}(\mathbf{x})) < T_\kappa \\ \text{false}, & \text{otherwise,} \end{cases} \quad (6.15)$$

Using our approach, paths containing parts with high curvature can be avoided, even if such paths may be shorter than the correct path. For example, in Fig. 6.2, our approach avoids the yellow path, because it contains a part γ_{part} with high curvature (dashed box). On the other hand, the red path does *not* contain highly curved parts. Consequently, our approach correctly finds the red path.

To better illustrate the difference between our approach and classical minimal path methods, a further example is provided in Fig. 6.3. In Fig. 6.3a, the same structure as the one in Fig. 3.2 is shown. The wavefront starts to propagate at the start point \mathbf{x}_s . When passing the first crossing, a local path γ_{local} with high curvature is detected (Fig. 6.3b). Consequently, the speed at the corresponding pixel on the wavefront is reduced dynamically, so that the wavefront propagates along the correct path, while the propagation along the short cut is slowed down significantly. The pixels with dynamic speed are highlighted with magenta boxes (Fig. 6.3c). Due to the dynamic speed, the wavefront does not proceed into the short cut (Fig. 6.3d). When the wavefront reaches the end point \mathbf{x}_e , the propagation terminates (Fig. 6.3e). The minimal path (red line) is extracted using backtracing (Fig. 6.3f). In contrast to the classical minimal path approach in Fig. 3.2, our method finds the correct path.

Since with our approach, the actual *geometry* of the path is available, we

6.2. Instances of the Progressive Minimal Path Method

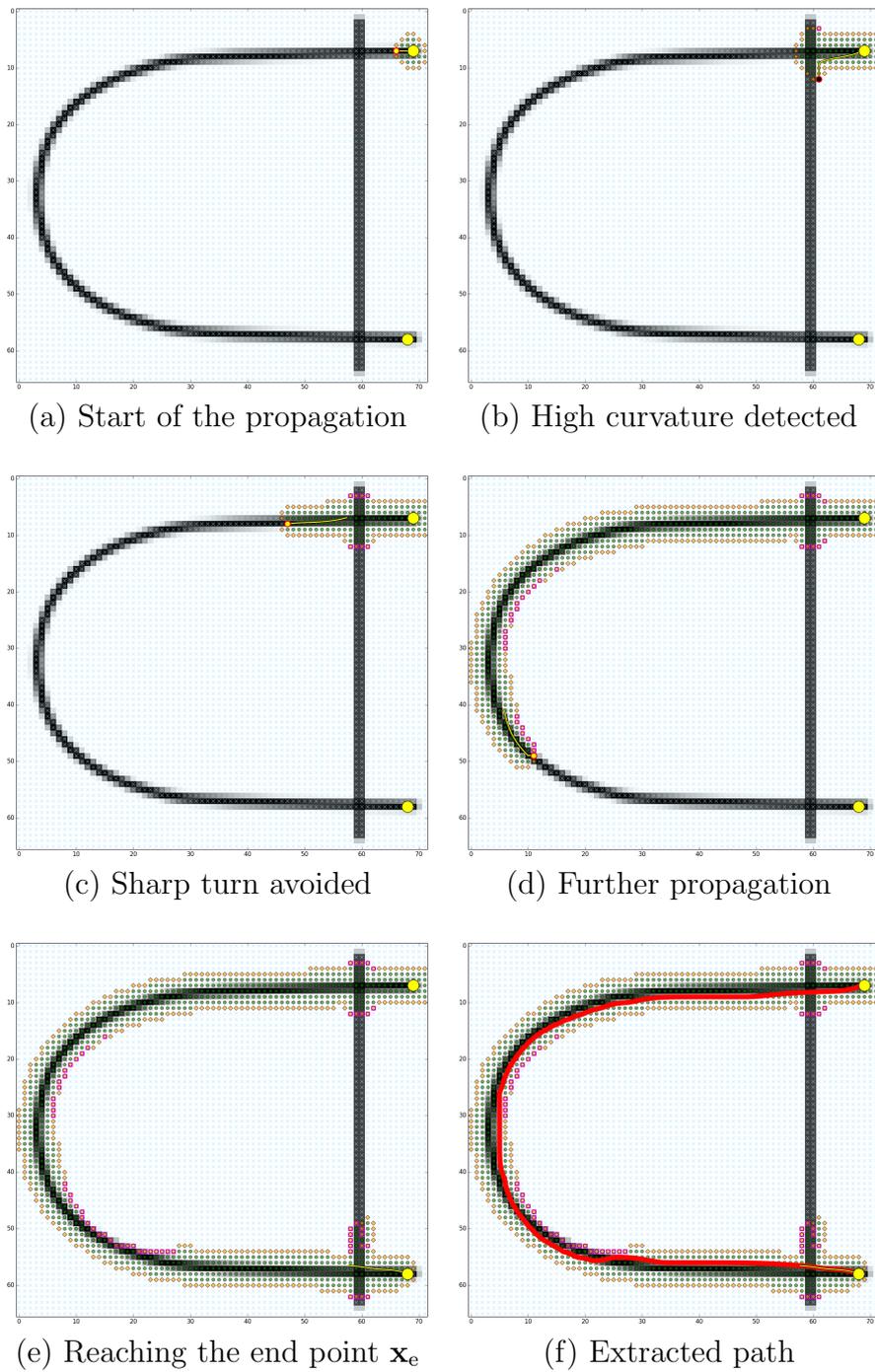


Figure 6.3: Wave propagation of our progressive minimal path approach. The Alive region \mathcal{R}_A is shown using green dots, the Trial region \mathcal{R}_T , i.e., the wavefront \mathcal{W} , is shown using orange diamonds, and the Far region \mathcal{R}_F is shown using blue crosses. The yellow circle with red boundary indicates the pixel \mathbf{x}_{\min} in \mathcal{R}_T with the minimum arrival time. (a) The wavefront starts at \mathbf{x}_s . (b) At the first crossing, high curvature is detected. (c) The wavefront propagates along the correct path, while the propagation along the short cut is slowed down. The pixels with dynamic speed are highlighted with magenta boxes. (d) The wavefront does not proceed into the short cut. (e) The wavefront reaches the end point \mathbf{x}_e , the propagation terminates. (f) The minimal path (red line) is extracted using backtracing.

can not only compute the curvature but also features of higher order (in principle of arbitrary order). This is a fundamental difference compared to recent approaches [55,56], where the computation of the curvature (involving 3 points) is quite expensive, and computation of the torsion (involving 4 points) is even more expensive, since the computation time increases exponentially with the number of involved points. Features beyond torsion (involving 5 points or more) were not used. But in our approach we can use features based on a much higher number of points (e.g., 10), since the computation time only increases linearly with the number of involved points. Also, using our continuous formulation, we do not have discretization (due to discretization of orientations) or metrication errors (due to the grid structure).

6.2.3 Joint Appearance and Geometric Feature

Third, we introduce a *joint appearance and geometric* feature to segment rivers and detect bridges in satellite images.

6.2.3.1 Motivation and Objective

Previous minimal path approaches are concerned with the segmentation of the paths. We show that it is also possible to detect other structures which are related to the paths. In particular, we introduce a novel path feature to *simultaneously* segment rivers and detect bridges in satellite images. To the best of our knowledge, this is the first attempt to combine segmentation based on minimal paths with object detection in a principled way.

6.2.3.2 Feature and Constraint

Our goal is to segment rivers as minimal paths, and detect bridges as structures related to rivers. When using gradient descent for backtracing, the extracted path at an image point \mathbf{x} is always perpendicular to the wavefront at \mathbf{x} , because the gradient shows in the direction of the steepest descent, which is the negative normal direction of the wavefront. Since in satellite images, bridges are typically perpendicular to the river, they correspond to *tangents* of the wavefront at \mathbf{x} . Consequently, bridges can be detected by examining the appearance of the tangent of the wavefront. Since rivers and bridges are both line structures but with different radii, we use two vesselness maps \mathcal{V}_R and \mathcal{V}_B , which are computed using different ranges of scale parameters (e.g., $s = 4$ and 6 for \mathcal{V}_R , and $s = 1$ and 2 for \mathcal{V}_B).

6.2. Instances of the Progressive Minimal Path Method

Furthermore, rivers typically have a homogeneous intensity distribution, thus the variance of the intensity structure is usually low. Therefore, we exploit the *homogeneity* \mathcal{H} of the image intensities, defined as:

Homogeneity of image intensities

$$\mathcal{H} = T(G_s * M_V), M_V(\mathbf{x}) = e^{-2 \cdot V_n(R_{\mathcal{H}}(\mathbf{x}))}. \quad (6.16)$$

where $R_{\mathcal{H}}(\mathbf{x})$ denotes a local image region centered at \mathbf{x} , and V_n is the intensity variance normalized to the interval $[0, 1]$. The exponential function enhances the resulting values, which are then smoothed with a Gaussian G_s and thresholded by T . For the static speed we use $\mathcal{F}_{\text{stat}} = \mathcal{V}_R \cdot \mathcal{H}$. `NeedDynamicSpeed`(\mathbf{x}) returns true for all pixels \mathbf{x} . `ComputeFeature` computes two features, i.e., the mean vesselness $\bar{\mathcal{V}}_R(\gamma_{\text{local}}(\mathbf{x}))$ over the local path, and the mean vesselness $\bar{\mathcal{V}}_B(t_{\text{local}}(\mathbf{x}))$ over the tangent $t_{\text{local}}(\mathbf{x})$ of the wavefront at \mathbf{x} , where $\bar{\mathcal{V}}_R$ and $\bar{\mathcal{V}}_B$ are defined as in (6.11). In our approach, we determine γ_{local} and t_{local} (*geometric* features) and then compute their *appearance* $\bar{\mathcal{V}}_R$ and $\bar{\mathcal{V}}_B$. If $\bar{\mathcal{V}}_R(\gamma_{\text{local}}(\mathbf{x})) < f_0 \cdot T_R$ and $\bar{\mathcal{V}}_B(t_{\text{local}}(\mathbf{x})) < f_0 \cdot T_B$, where $f_0 = 0.05$, and T_R and T_B are thresholds, then \mathbf{x} violates the constraint. Otherwise, the constraint is satisfied. In addition, if $\bar{\mathcal{V}}_R(\gamma_{\text{local}}) > T_R$ and $\bar{\mathcal{V}}_B(t_{\text{local}}(\mathbf{x})) > T_B$, then \mathbf{x} is labeled l_{obj} in an object map \mathcal{M}_{obj} , while other pixels corresponding to the background are labeled l_{bg} . The detection of bridges is summarized in the following Algorithm 6.2, where the Boolean variable `sat`(\mathbf{x}) is true if and only if the constraint is satisfied:

Algorithm 6.2: IsSatisfied for joint appearance and geometric feature

Input: \mathbf{x} , $\bar{\mathcal{V}}_R(\gamma_{\text{local}}(\mathbf{x}))$, $\bar{\mathcal{V}}_B(t_{\text{local}}(\mathbf{x}))$, object map \mathcal{M}_{obj} , thresholds for river T_R and for bridge T_B

Output: `sat`(\mathbf{x}), updated object map \mathcal{M}_{obj}

```

1 sat( $\mathbf{x}$ )  $\leftarrow$  true;
2  $\mathcal{M}_{\text{obj}}(\mathbf{x}) \leftarrow l_{\text{bg}}$ ;
3  $f_0 \leftarrow 0.05$ ;
4 if  $\bar{\mathcal{V}}_R(\gamma_{\text{local}}) > T_R$  and  $\bar{\mathcal{V}}_B(t_{\text{local}}) > T_B$  then
5   |  $\mathcal{M}_{\text{obj}}(\mathbf{x}) \leftarrow l_{\text{obj}}$ ; // Bridge detected
6 else if  $\bar{\mathcal{V}}_R(\gamma_{\text{local}}) < f_0 \cdot T_R$  and  $\bar{\mathcal{V}}_B(t_{\text{local}}) < f_0 \cdot T_B$  then
7   | sat( $\mathbf{x}$ )  $\leftarrow$  false; //  $\mathbf{x}$  does not correspond to a river
8 return sat( $\mathbf{x}$ ),  $\mathcal{M}_{\text{obj}}$ 

```

In Algorithm 6.2, the tangent $t_{\text{local}}(\mathbf{x})$ is used both for object detection and determination of the dynamic speed. `IsSatisfied` returns true if $\gamma_{\text{local}}(\mathbf{x})$ corresponds to a river and $t_{\text{local}}(\mathbf{x})$ corresponds to a bridge, i.e., if $\bar{\mathcal{V}}_R(\mathbf{x})$ and $\bar{\mathcal{V}}_B(\mathbf{x})$

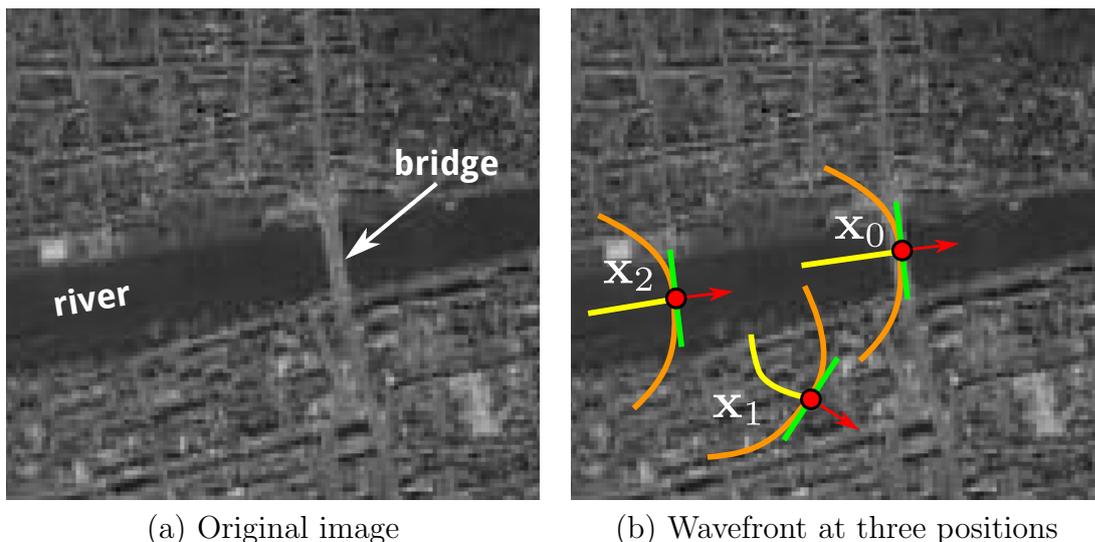


Figure 6.4: Principle of tangent-based bridge detection. (a) Original satellite image with a river and a bridge. The bridge is roughly perpendicular to the river. (b) Sketch of wavefront (orange) at three positions \mathbf{x}_0 , \mathbf{x}_1 , and \mathbf{x}_2 (red circles), along with the local paths $\gamma_{\text{local}}(\mathbf{x}_0)$, $\gamma_{\text{local}}(\mathbf{x}_1)$, and $\gamma_{\text{local}}(\mathbf{x}_2)$ (yellow lines), the normals $\mathbf{n}_{\text{local}}(\mathbf{x}_0)$, $\mathbf{n}_{\text{local}}(\mathbf{x}_1)$, and $\mathbf{n}_{\text{local}}(\mathbf{x}_2)$ (red arrows), and the tangents $t_{\text{local}}(\mathbf{x}_0)$, $t_{\text{local}}(\mathbf{x}_1)$, and $t_{\text{local}}(\mathbf{x}_2)$ (green lines) at each position. At \mathbf{x}_0 , a bridge is detected. At \mathbf{x}_1 , no bridge is detected, and the wave propagation is slowed down. At \mathbf{x}_2 , no bridge is detected, but the wave propagation is *not* slowed down.

are both high. The points \mathbf{x} on the optimal path with $\mathcal{M}_{\text{obj}}(\mathbf{x}) = l_{\text{obj}}$ are clustered using nearest neighbors, and each cluster center corresponds to one bridge. Note that with the fast marching method, tangents of *arbitrary* orientations can be computed in *constant* time, while with Dijkstra’s algorithm, the number of possible tangent orientations depends on the number of edges connected to each node, which is typically 4 or 8, and at most 32 in previous approaches [56], and the run time increases with the number of edges per node.

The main idea of our approach is illustrated in Fig. 6.4. In the original satellite image, there is a bridge which is roughly perpendicular to a river (Fig. 6.4a). For three positions \mathbf{x}_0 , \mathbf{x}_1 , and \mathbf{x}_2 , the corresponding local paths $\gamma_{\text{local}}(\mathbf{x}_0)$, $\gamma_{\text{local}}(\mathbf{x}_1)$, and $\gamma_{\text{local}}(\mathbf{x}_2)$, normals $\mathbf{n}_{\text{local}}(\mathbf{x}_0)$, $\mathbf{n}_{\text{local}}(\mathbf{x}_1)$, and $\mathbf{n}_{\text{local}}(\mathbf{x}_2)$ and tangents $t_{\text{local}}(\mathbf{x}_0)$, $t_{\text{local}}(\mathbf{x}_1)$, and $t_{\text{local}}(\mathbf{x}_2)$ are sketched (Fig. 6.4b). At position \mathbf{x}_0 , the mean vesselness of the river and the bridge $\bar{\mathcal{V}}_{\text{R}}(\gamma_{\text{local}}(\mathbf{x}_0))$ and $\bar{\mathcal{V}}_{\text{B}}(t_{\text{local}}(\mathbf{x}_0))$, respectively, are both relatively high, since $\gamma_{\text{local}}(\mathbf{x}_0)$ lies inside the river region, where \mathcal{V}_{R} is high, and $t_{\text{local}}(\mathbf{x}_0)$ lies inside the bridge region, where \mathcal{V}_{B} is high. Thus, a bridge is detected at \mathbf{x}_0 . In contrast, at position \mathbf{x}_1 , $\bar{\mathcal{V}}_{\text{R}}(\gamma_{\text{local}}(\mathbf{x}_1))$ is low, since $\gamma_{\text{local}}(\mathbf{x}_1)$ lies *outside* the river region, where \mathcal{V}_{R} is low, and $t_{\text{local}}(\mathbf{x}_1)$ lies *outside* the bridge

6.2. Instances of the Progressive Minimal Path Method

region, where \mathcal{V}_B is low. Consequently, no bridge is detected at \mathbf{x}_1 , and the wave propagation through \mathbf{x}_1 is slowed down, since the wavefront at \mathbf{x}_1 has left the river region. At position \mathbf{x}_2 , $\bar{\mathcal{V}}_R(\gamma_{\text{local}}(\mathbf{x}_2))$ is high, since $\gamma_{\text{local}}(\mathbf{x}_2)$ lies inside the river region, where \mathcal{V}_R is high, but $\bar{\mathcal{V}}_B(t_{\text{local}}(\mathbf{x}_2))$ is low, since $t_{\text{local}}(\mathbf{x}_2)$ lies *outside* the bridge region, where \mathcal{V}_B is low. Therefore, no bridge is detected at \mathbf{x}_2 , but in contrast to the case of \mathbf{x}_1 , the wave propagation through \mathbf{x}_2 is *not* slowed down, since the wavefront at \mathbf{x}_2 still remains in the river region.

Chapter 7

Experimental Results of Progressive Minimal Paths

In this chapter, experimental results for our progressive minimal path method introduced in Chapter 6 are presented. We conducted quantitative and qualitative evaluations using 2D and 3D images from different application areas, including synthetic images, retinal images, satellite images of streets, rivers, and bridges, and 3D 7T MRA images of human brain vessels. We also compared our method with previous approaches.

7.1 Appearance Feature

We have evaluated our method in conjunction with the appearance feature described in Sect. 6.2.1 above using 2D synthetic and real images, including images of retinal vessels and satellite images of rivers and streets. We compared the results of our method with two previous fast marching approaches: A minimal path approach based on the classical fast marching method and a vesselness measure (FM-V) [41], and a domain-lifting approach using an additional dimension for the radius (LY) [154].

7.1.1 2D Synthetic Images

Fig. 7.1 shows experimental results for a curved line structure with many gaps due to artifacts (left column) and for a spiral with a high noise level (right column). For the curved line structure, the total length of all gaps due to artifacts is 80 pixel, while the length of the short cut between the start point \mathbf{x}_s and the end point \mathbf{x}_e is only 17 pixel. For the image of the spiral, Gaussian noise with standard

deviation $\sigma_n = 40$ was added. For both types of images, FM-V and LY yield short cuts, while our method yields the correct path. To quantitatively evaluate the performance, we generated images of size 72×140 pixels with four types of 2D line structures, ranging from a structure with high curvature (Fig. 7.4a, type A) to a straight line (Fig. 7.4d, type D), each with three different radii: $r = 1.5, 2.5,$ and 3.5 pixel. Furthermore, we added Gaussian noise with 21 levels between $\sigma_n = 0$ and $\sigma_n = 100$. The noise was added independently 30 times. In total, 7560 synthetic images ($4 \times 3 \times 21 \times 30$) were used.

We compared the results of our method and those of FM-V and LY with the ground truth. The ground truth is a binary segmentation of all pixels corresponding to the line structures, while the results from fast marching approaches are centerlines with subpixel accuracy. For a comparison, we determined the set \mathcal{S}_0 of pixels which lie on the path of the segmented centerline, and the set $\mathcal{S}_1 \subset \mathcal{S}_0$, which consists of pixels of \mathcal{S}_0 inside the binary segmentation of the line structures. To measure the accuracy of the segmented centerlines, the ratio between the numbers of pixels of \mathcal{S}_1 and \mathcal{S}_0 is used.

Accuracy measure: Q_{inside}

$$Q_{\text{inside}} = \frac{|\mathcal{S}_1|}{|\mathcal{S}_0|} \quad (7.1)$$

We studied the sensitivity of the results with respect to the crucial parameters Γ (length of the local path γ_{local}), and T_v (threshold of the mean vesselness $\bar{V}(\gamma_{\text{local}})$). It turns out that using higher values of Γ , higher mean values and lower variances of Q_{inside} are obtained (see Fig. 7.2). We consider $\Gamma = 6$ a reasonable choice, since longer paths require more computation time but yield only limited improvement. Concerning T_v , the results in Fig. 7.3 suggest that it should be high enough to avoid short cuts, for example, $T_v = 0.6$. But if T_v is too high, for example, $T_v = 0.9$, then correct paths are often not found. In our experiments, we use an intermediate value $T_v = 0.8$. We also computed the corresponding standard deviation of the results over the 30 runs of adding noise.

Using $\Gamma = 6$ and $T_v = 0.8$, we conducted experiments to study the robustness of our method. For strongly curved lines (type A with $r = 1.5$), the mean values and the standard deviations of Q_{inside} for 30 runs with different noise levels are shown exemplarily in Fig. 7.5, first row. FM-V yields increasingly short cuts with increasing standard deviation of the noise $\sigma_n > 10$, resulting in a significant decrease of Q_{inside} , and LY also yields short cuts, leading to relatively low values of

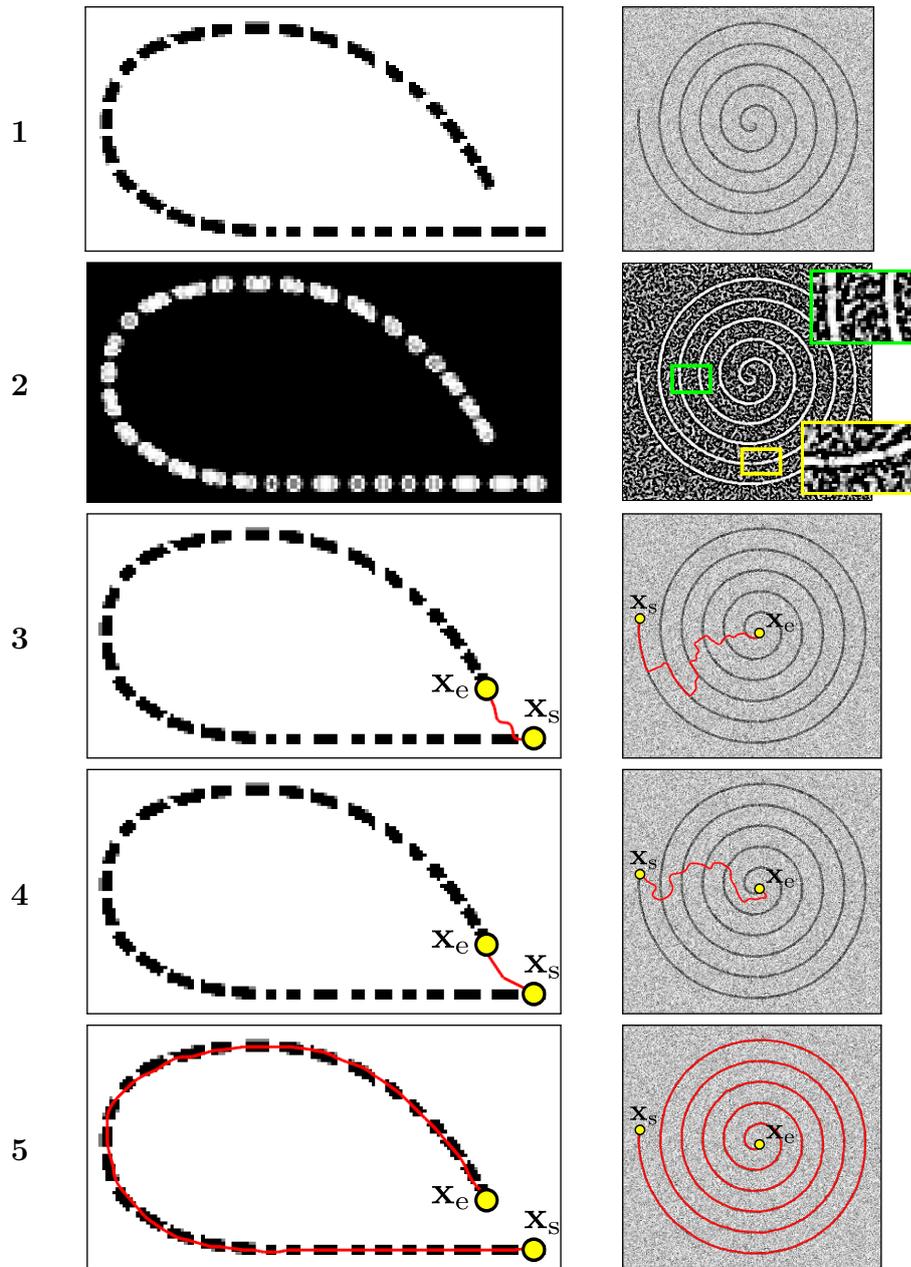


Figure 7.1: Segmentation of a curved line structure with many gaps due to artifacts (left column), and a spiral with a high noise level (right column). x_s and x_e are indicated as yellow circles, and the centerlines are shown as red lines. Row 1: Original image. Row 2: Vesselness map, high values are represented by bright intensities. For the spiral, two regions marked by boxes have been enlarged. Green: high vesselness in the area between two spiral lines, yellow: low vesselness along the spiral line and high vesselness nearby). Row 3: Result of FM-V. Row 4: Result of LY. Row 5: Result of our progressive minimal path method.

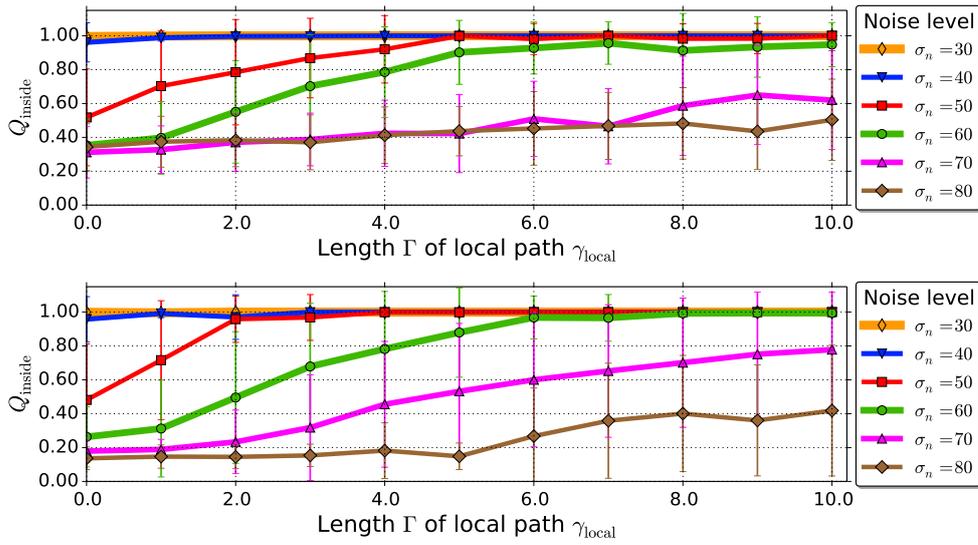


Figure 7.2: Dependency of Q_{inside} on the length Γ of the local paths γ_{local} . First row: Type A, $r = 1.5$. Second row: Type B, $r = 1.5$.

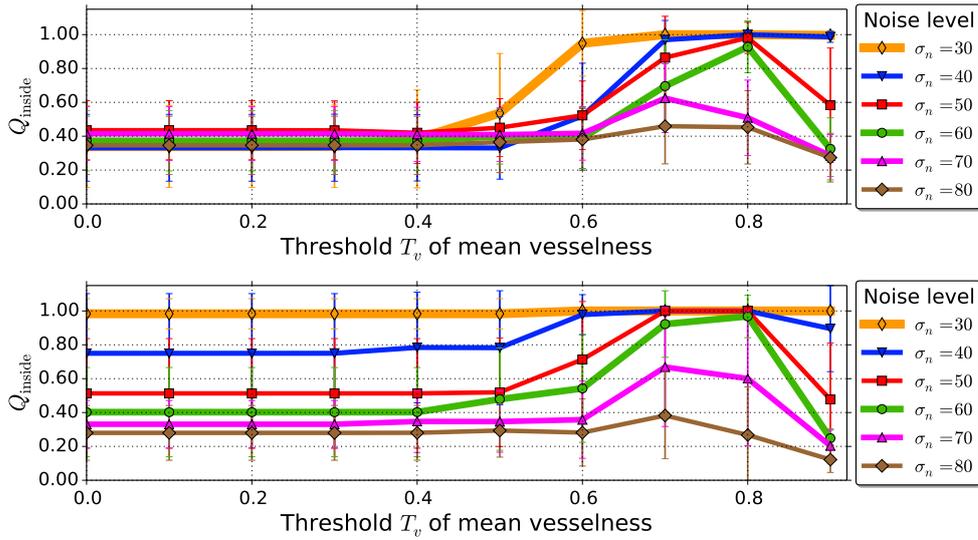


Figure 7.3: Dependency of Q_{inside} on the threshold T_v of the mean vesselness of the local paths γ_{local} . First row: Type A, $r = 1.5$. Second row: Type B, $r = 1.5$.

Q_{inside} . In comparison, our approach finds the correct path, even in the case of a high level of image noise of $\sigma_n = 60$ (see Fig. 7.4a), where still $Q_{\text{inside}} = 92.77\%$ can be achieved. The images of type B ($r = 1.5$) contain a less strongly curved line. Also here, LY often yields short cuts, while FM-V achieves better results than for type A. Our approach yields a similarly good result as for type A (see Fig. 7.5, second row). The results for images of type C ($r = 2.5$) are shown in the third row of Fig. 7.5. For $\sigma_n < 20$, all three approaches have a similar performance.

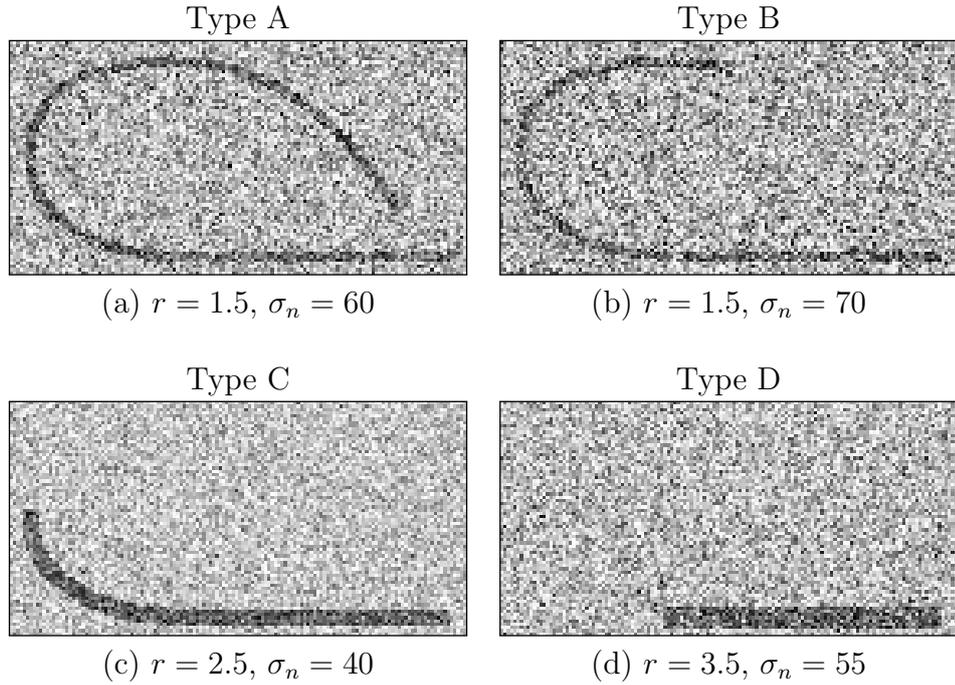


Figure 7.4: Examples of synthetic images with four different types of 2D line structures. Each image contains a line structure with radius r and is distorted by Gaussian noise with standard deviation σ_n .

With increasing noise level, the performance of LY decreases strongly, while that of FM-V and our approach remains relatively high. Finally, for the straight line structure of type D ($r = 3.5$), the results are similar as for type C, but the decrease of the performance of LY starts at a higher noise level of $\sigma_n = 30$ (Fig. 7.5 fourth row). For a high noise level of $\sigma_n = 80$, FM-V and our approach still achieve a value of Q_{inside} over 95%.

7.1.2 2D Real Images

We also successfully applied our progressive minimal path to 10 medical images of retinal vessels from the STARE dataset [182], as well as 10 satellite images of rivers and 10 satellite images of streets from Google Maps (maps.google.com). For example, in Fig. 7.6, first column, the correct path of the retinal vessel is very long, the image intensities within the vessels vary strongly, and the middle part is much darker than the two ends. Consequently, LY does not yield a good result because it assumes a relatively homogeneous intensity distribution within the vessel. FM-V follows another vessel and yields a short cut, while our approach finds the correct vessel. In Fig. 7.6, second column, the river has a high curvature and similar intensities as the riverbank, which causes LY to yield some wrong

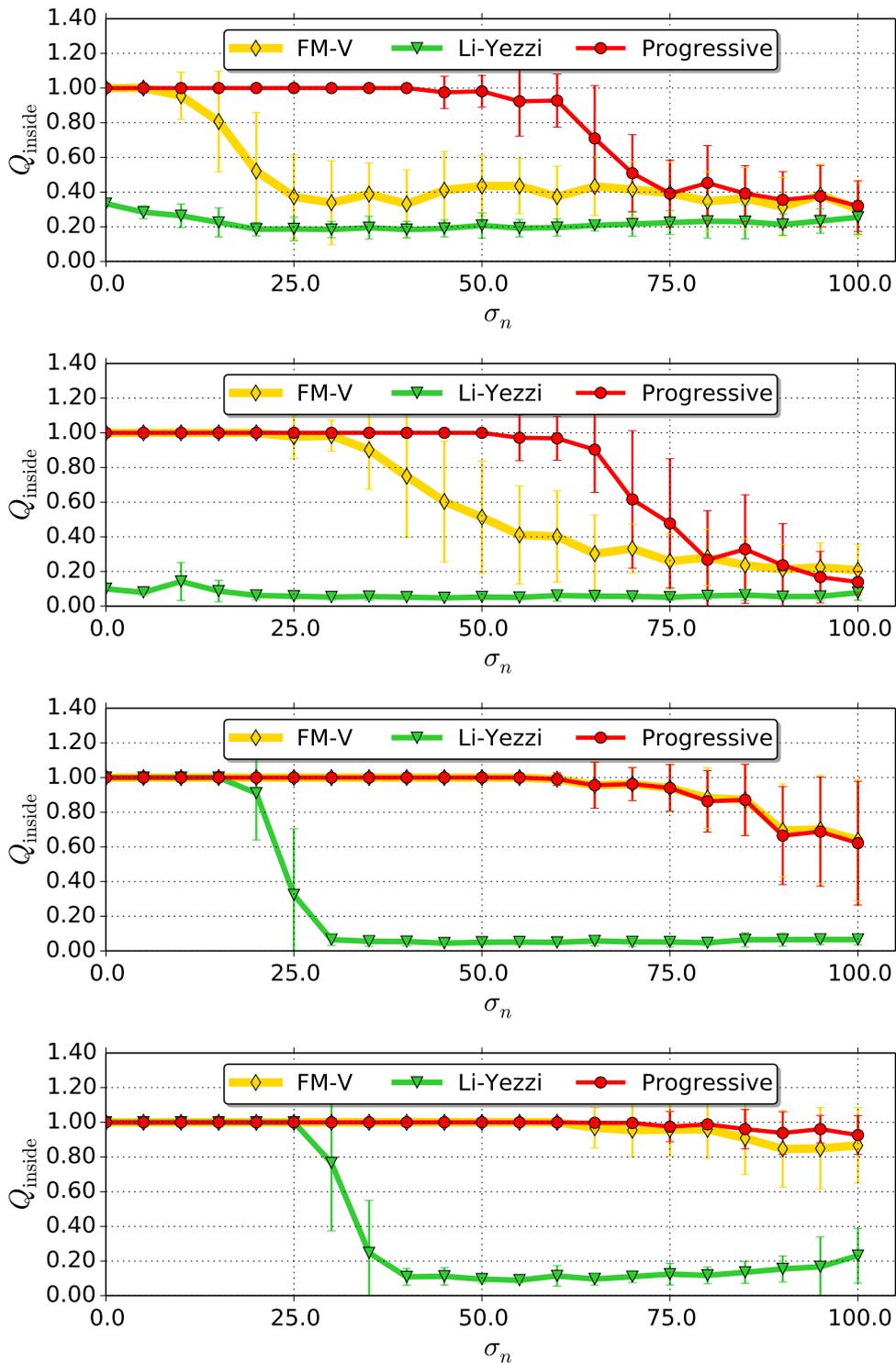


Figure 7.5: Dependency of Q_{inside} on the noise level for four different types of 2D line structures. First row: Type A, $r = 1.5$. Second row: Type B, $r = 1.5$. Third row: Type C, $r = 2.5$. Fourth row: Type D, $r = 3.5$.

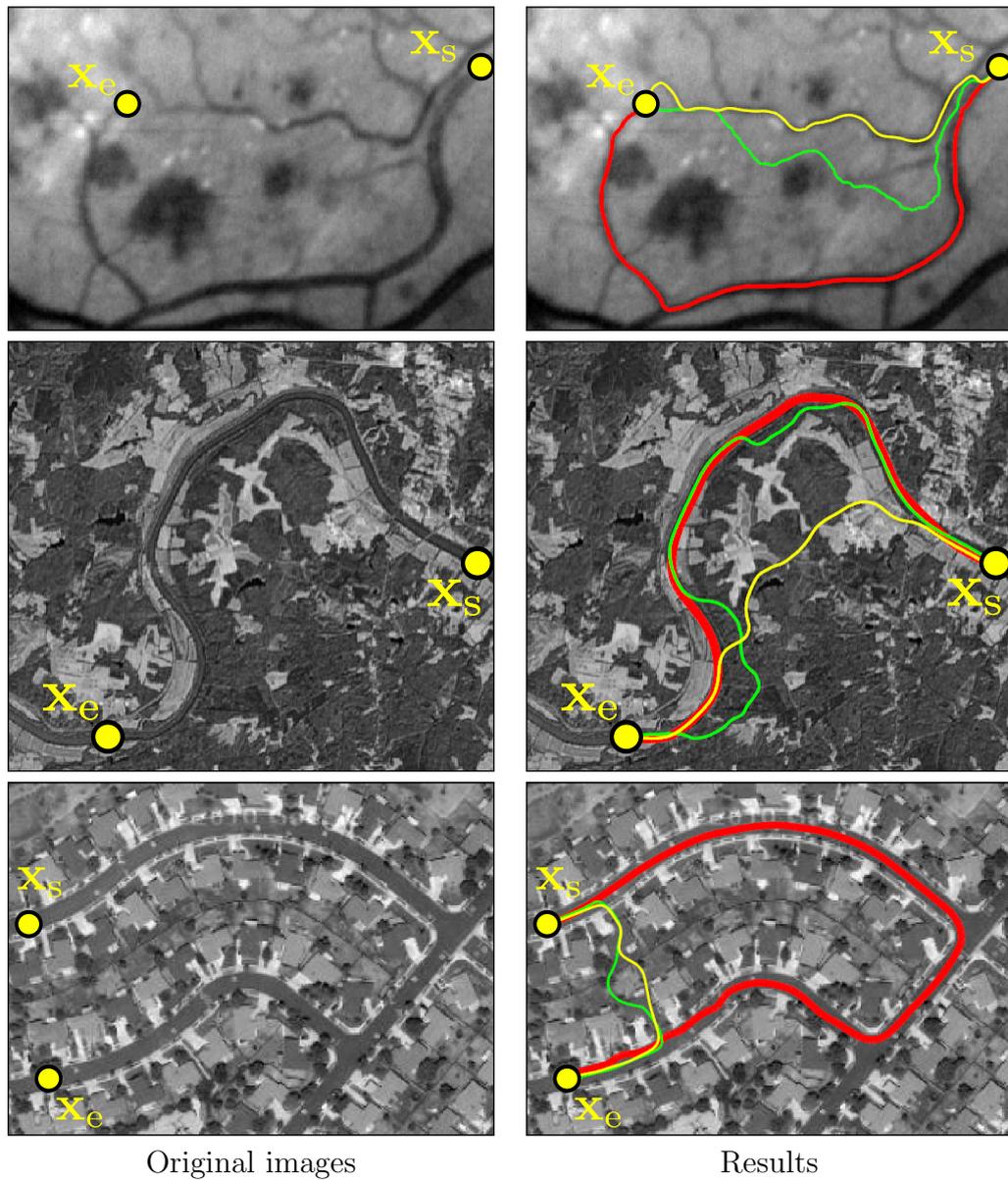


Figure 7.6: Segmentation of real images. Row 1: Retinal image. Row 2: Satellite image of a river. Row 3: Satellite image of a street. Results of FM-V (yellow), LY (green), and our progressive minimal path method (red).

turns. Additionally, the region enclosed by the river is very cluttered, so FM-V yields a short cut. Again, our approach finds the correct path. In Fig. 7.6, third column, several houses have similar intensities as the street. Furthermore, the markings on the street also pose difficulties. Both LY and FM-V find a short cut, while our approach is able to find the correct path. For the other images, we obtained similar results.

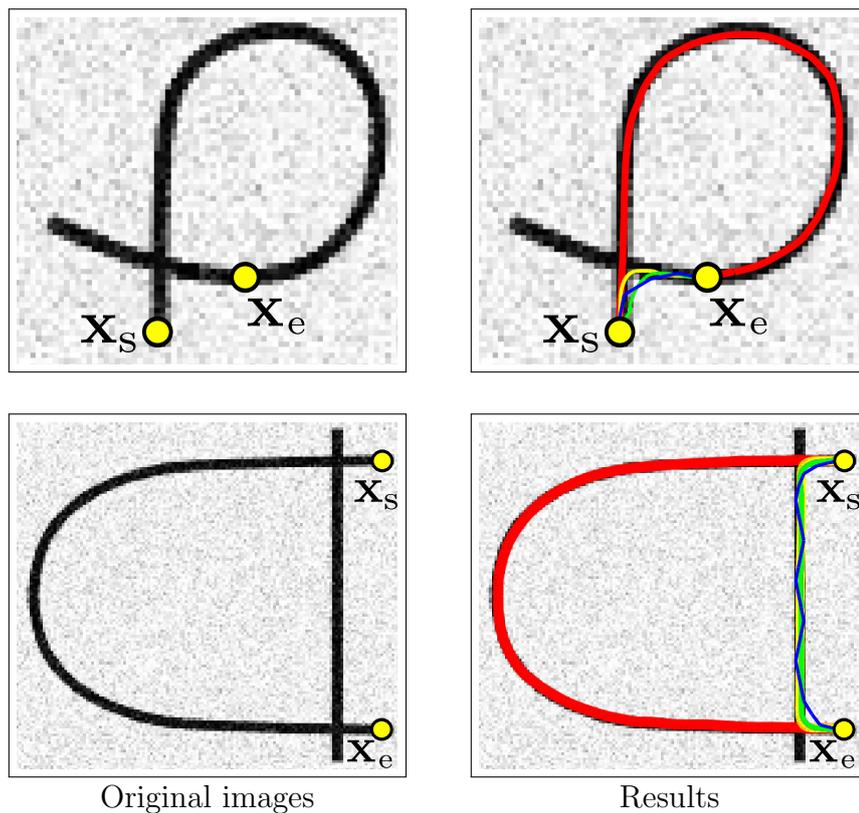


Figure 7.7: Synthetic images of a loop (left) and an open curve which is crossed by a straight line (right). Results of FM-V (yellow), LY (green), USK (blue), and our progressive minimal path method (red).

7.2 Geometric Feature

We also evaluated our progressive minimal path method using the geometric feature (curvature) described in Sect. 6.2.2 above based on 2D synthetic data and real retinal images, and compared the results with FM-V [41], LY [154], and USK [56]. Furthermore, we considered vessel segmentation in 3D synthetic data and 3D 7T high-resolution MRA images, and compared the results of our method with three previous approaches.

7.2.1 2D Synthetic Images

We tested our method using different 2D synthetic images. For example, Fig. 7.7, first row, shows a loop, and Fig. 7.7, second row, displays an open curve which is crossed by a straight line. In both cases, our method yields the paths with low curvature (red lines), even though these paths are much longer than the short cuts. In contrast, FM-V (yellow), LY (green), and USK (blue) result in short cuts.

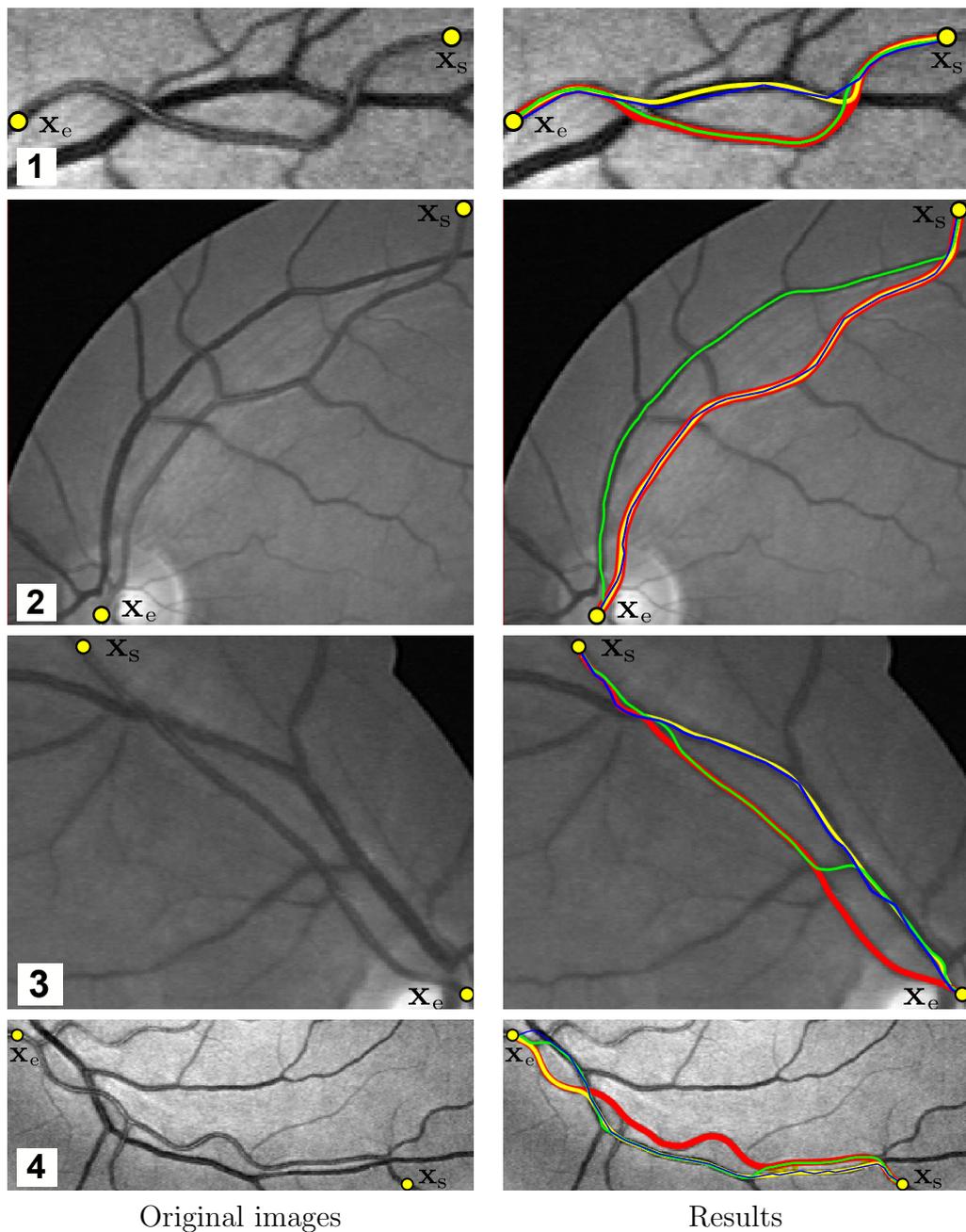


Figure 7.8: Four different examples of retinal vessels from the DRIVE dataset. Results of FM-V (yellow), LY (green), USK (blue), and our progressive minimal path method (red).

7.2.2 2D Retinal Images

For evaluating our method based on real data we have used all 40 retinal images of the DRIVE dataset [24]. The images have a size of 565×584 pixels. For each image, one to three vessels which cross other vessels were selected, resulting in 81 vessels in total. Such vessels are very common in retinal images and difficult to

Table 7.1: Comparison of quantitative results for retinal images.

Vessel	FM-V		LY		USK		Progressive	
	Q_{inside}	Time	Q_{inside}	Time	Q_{inside}	Time	Q_{inside}	Time
1	54.90%	0.10 s	94.04%	0.16 s	60.85%	6.52 s	94.55%	0.26 s
2	100.00%	0.42 s	11.49%	2.36 s	100.00%	39.70 s	100.00%	0.83 s
3	26.42%	0.21 s	67.49%	0.71 s	29.89%	18.16 s	99.61%	0.60 s
4	31.96%	0.34 s	42.78%	1.27 s	15.84%	29.51 s	99.42%	0.98 s
Mean of 81	45.27%	0.25 s	48.18%	0.74 s	47.47%	20.26 s	99.13%	0.64 s

cope with. Example results for four vessels are shown in Fig. 7.8. In the first row, a vessel crosses another vessel which has a similar radius. There, FM-V and USK result in short cuts, while our approach finds the correct path. LY also finds the correct path, because the intensity of the vessel is quite different from the other vessel. However, for the example in the second row, the intensity is not always a good feature, since the intensities inside the vessel vary significantly. As a result, LY leaks into a neighboring vessel, while FM-V, USK, and our approach yield the correct path. For the vessel in the third row, FM-V, LY, and USK do not yield the correct path. In contrast, our approach yields the correct path, as well as for the example in the fourth row, which shows an even more difficult case, where the correct vessel is relatively long, with several crossings with other vessels.

For a quantitative evaluation, we compared our results with ground truth and computed the ratio Q_{inside} defined in Sect. 7.1.1 above. The results and run time for the four vessels in Fig. 7.8 as well as the mean values over all 81 vessels are summarized in Table 7.1. It can be seen that our approach yields significantly higher values for Q_{inside} compared to FM-V, LY, and USK (99.13% on average, compared to 45.27%, 48.18%, and 47.47% for the previous approaches). Additionally, our approach requires a lower mean run time of 0.64s compared to 0.74s of LY and 20.26s of USK, using a single-threaded implementation on a PC with a 2.67 GHz CPU and 48 GB memory.

7.2.3 3D Synthetic Images of Vessels

For segmentation of 3D synthetic images of vessels we integrated our progressive minimal path method with a model-based approach. The segmentation consists of two steps. In the first step, vessels in regions with low noise and high image contrast are segmented by fitting a parametric intensity model g_M to the image intensities $g(\mathbf{x})$ within a spherical 3D ROI [46]

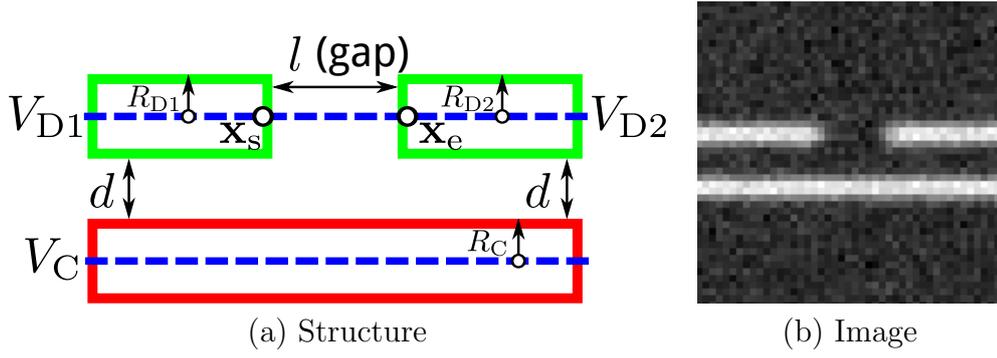


Figure 7.9: (a) Sketch of *parallel vessels* with a gap. The upper vessel consists of two segments V_{D1} and V_{D2} , each at the same distance d to the lower vessel V_C . There is a gap of length l between them, which starts at \mathbf{x}_s and ends at \mathbf{x}_e . Each vessel has the same radius: $R_C = R_{D1} = R_{D2}$. The image contains Gaussian noise with standard deviation of σ_n . (b) 2D section of an example of a 3D synthetic image based on (a).

Parametric intensity model

$$g_M(\mathbf{x}, \mathbf{p}) = a_0 + (a_1 - a_0) g_{Cyl}(\mathcal{R}(\mathbf{x}, \boldsymbol{\alpha}, \mathbf{x}_0), R, \sigma) \quad (7.2)$$

$$\sum_{\mathbf{x} \in \text{ROI}} (g_M(\mathbf{x}, \mathbf{p}) - g(\mathbf{x}))^2 \rightarrow \min. \quad (7.3)$$

with parameters $\mathbf{p} = (R, a_0, a_1, \sigma, \alpha, \beta, \gamma, x_0, y_0, z_0)^T$. R is the radius of a vessel, a_0 and a_1 are the intensity levels of the vessel and the surrounding tissue, σ is the image blur, $(\alpha, \beta, \gamma)^T$ and $(x_0, y_0, z_0)^T$ describe a rotation and a translation, respectively. g_{Cyl} is a 3D cylindrical model obtained by convolving an ideal sharp 3D cylinder with a 3D Gaussian [46]. In regions with high noise or low image contrast, some vessel parts are difficult to segment, leading to gaps between vessels. In the second step, these gaps are completed using our progressive minimal path method. Fig. 7.9 shows a sketch and a synthetic image for a typical situation, i.e. two parallel vessels, where one of them contains a gap. The parameters for generating the 3D synthetic images (e.g., vessel radius, intensities, noise level) have been chosen in accordance with real 3D 7T MRA images.

The images have a size of $128 \times 128 \times 32$ voxels. We applied our progressive method in conjunction with the curvature feature. To evaluate the performance of gap completion against ground truth we used the following error measure

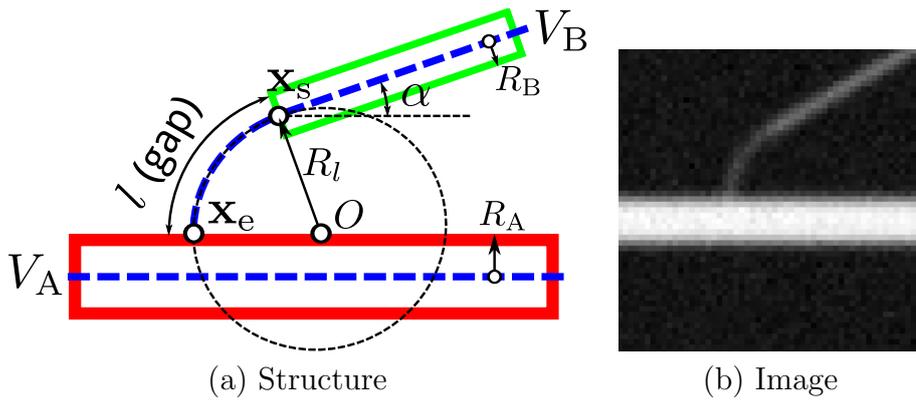


Figure 7.10: (a) Sketch of *branching vessel*. Between the horizontal vessel V_A with radius R_A and the inclined vessel V_B with radius R_B and angle α to V_A , there is a low-contrast part of length l . This small part is an arc of a circle with radius R_l and centered at O . The starting and end points are \mathbf{x}_s and \mathbf{x}_e , respectively. Gaussian noise is added. (b) 2D section of an example of a 3D synthetic image based on (a).

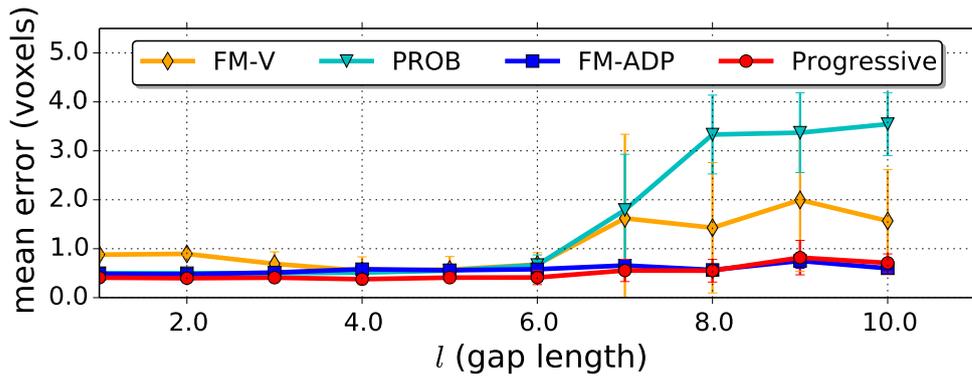


Figure 7.11: Mean error and standard deviation of the centerline positions for different gap lengths for *parallel vessels* in 3D synthetic images using different minimal path methods: The gap of length l between vessel parts V_{D1} and V_{D2} (see Fig. 7.9) varies from 1 to 10 voxels, with an increment of 1 voxel. The standard deviation of noise is $\sigma_n = 10$.

Error measure

$$\bar{e}_\gamma = \frac{1}{|\gamma|} \sum_{\mathbf{x}_i \in \gamma} |\mathbf{x}_i - \mathbf{x}_{\text{gt}(i)}|, \quad (7.4)$$

where \mathbf{x}_i are sample points on the path γ and $\mathbf{x}_{\text{gt}(i)}$ is the closest point of the ground truth to \mathbf{x}_i . $|\gamma|$ denotes the number of points of γ . We compared our progressive minimal path method with three previous approaches, i.e. the classical

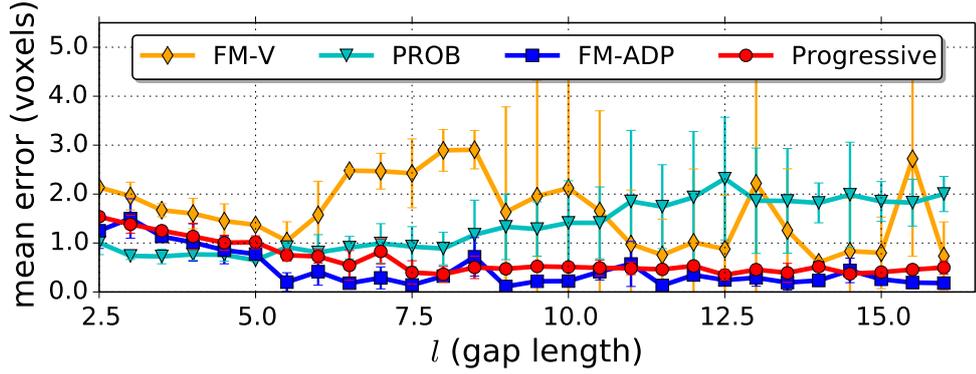


Figure 7.12: Mean error and standard deviation of the centerline positions for different gap lengths for *branching vessels* in 3D synthetic images using different minimal path methods: The gap length l between vessels V_A and V_B (see Fig. 7.10) varies from 2.5 to 16, with an increment of 0.5. The other parameters are constant for all experiments: $\alpha = \frac{\pi}{6}$ and $\sigma_n = 10$.

Table 7.2: Mean errors and standard deviations (in voxels) of the centerline positions for all experiments with 3D synthetic images.

Image type	FM-V	PROB	FM-ADP	Progressive
Parallel vessels	1.09±1.03	1.53±1.41	0.58±0.11	0.50±0.23
Branching vessels	1.65±1.39	1.35±0.86	0.46±0.43	0.66±0.36

fast marching approach (FM-V) [41], a probabilistic approach (PROB) [40], and a fast marching approach with directional prior (FM-ADP) [40]. For the parallel vessels illustrated in Fig. 7.9, we varied the gap length l from 1 to 10 voxels, with an increment of 1 voxel. For the parallel vessels illustrated in Fig. 7.10, we varied the gap length l from 2.5 to 16 voxels, with an increment of 0.5 voxel. The images were distorted by additive Gaussian noise with standard deviation $\sigma_n = 10$. The noise was added independently 30 times, yielding $10 \times 30 = 300$ 3D images. In Fig. 7.11 it can be seen that for parallel vessels, our method achieves the lowest mean error for most values of l . By comparison of the mean values and standard deviations for all 3D images (Table 7.2, first row), it turns out that our approach outperforms the three previous approaches. Fig. 7.12 shows that for branching vessels, the mean error of our method is significantly lower than that of FM-V and PROB, and is similar to the result of FM-ADP. Furthermore, our approach achieves a lower standard deviation than previous approaches (Table 7.2, second row).

Table 7.3: Quantitative results for 508 gaps in 3D 7T MRA images (errors in mm).

Approach	Total	Centerline error \bar{e}_γ				Mean \bar{e}_γ	Mean run time per 3D image
		$\bar{e}_\gamma < 0.3$	$\bar{e}_\gamma < 0.4$	$\bar{e}_\gamma < 0.5$	$\bar{e}_\gamma < 0.6$		
FM-V	99.61%	34.45%	61.22%	83.46%	92.72%	0.406	0.92 s
PROB	96.65%	56.69%	76.18%	86.61%	90.35%	0.313	51.68 s
FM-ADP	99.61%	61.22%	86.42%	94.29%	97.64%	0.288	1.63 s
Progressive	99.61%	86.61%	92.32%	95.47%	97.64%	0.222	19.36 s

7.2.4 3D 7T MRA Images of Brain Vessels

We also applied our progressive minimal path method to 3D 7T MRA images of human brain vessels. Compared to 1.5T or 3T MRA, 7T MRA images display significantly more thin vessels. In our evaluation we considered lenticulostriate arteries (LSAs), which are microvessels in the brain and are susceptible to diseases like stroke. The 3D 7T MRA images were acquired without contrast agent using a 3D gradient echo time-of-flight sequence with a repetition time of 15 ms, echo times of 4.84 ms or 4.85 ms, a field of view of 135 mm \times 180 mm, and flip angles of 25 or 30 degrees. We applied isotropic interpolation, yielding 128-162 slices for the 3D images and an isotropic resolution of 0.23 mm. The size of the slices varies from 128 \times 270 to 162 \times 417 voxels. We analyzed the performance of our method for gap completion using the error measure in (7.4) with manually determined ground truth, and performed a comparison with the previous approaches FM-V [41], PROB [40], and FM-ADP [40]. We considered 508 gaps in the 3D 7T MRA images. Table 7.3 provides the percentage of the total number of completed gaps, the percentage of the total number of completed gaps within error ranges of $\bar{e}_\gamma < 0.3, 0.4, 0.5$, and 0.6 mm, the mean error over all completed gaps, and the mean run time per 3D image (using a single-threaded implementation on a PC with a 2.67 GHz CPU and 48 GB memory) for all four approaches.

Results for the 10 3D MRA images and the average over all 10 3D 7T MRA images are provided in Fig. 7.13 and Fig. 7.14, respectively. It can be seen that our method yields the lowest mean error and a much higher percentage of completed gaps within small error ranges ($\bar{e}_\gamma < 0.3$ mm and $\bar{e}_\gamma < 0.4$ mm). We also applied our progressive minimal path method in conjunction with the model-based approach described in Sect. 7.2.3 above to segment vessels within the whole brain in three 3D 7T MRA images. Fig. 7.15 shows the result for one 3D image. It can be seen that a large number of thin vessels have been successfully segmented.

7.2. Geometric Feature

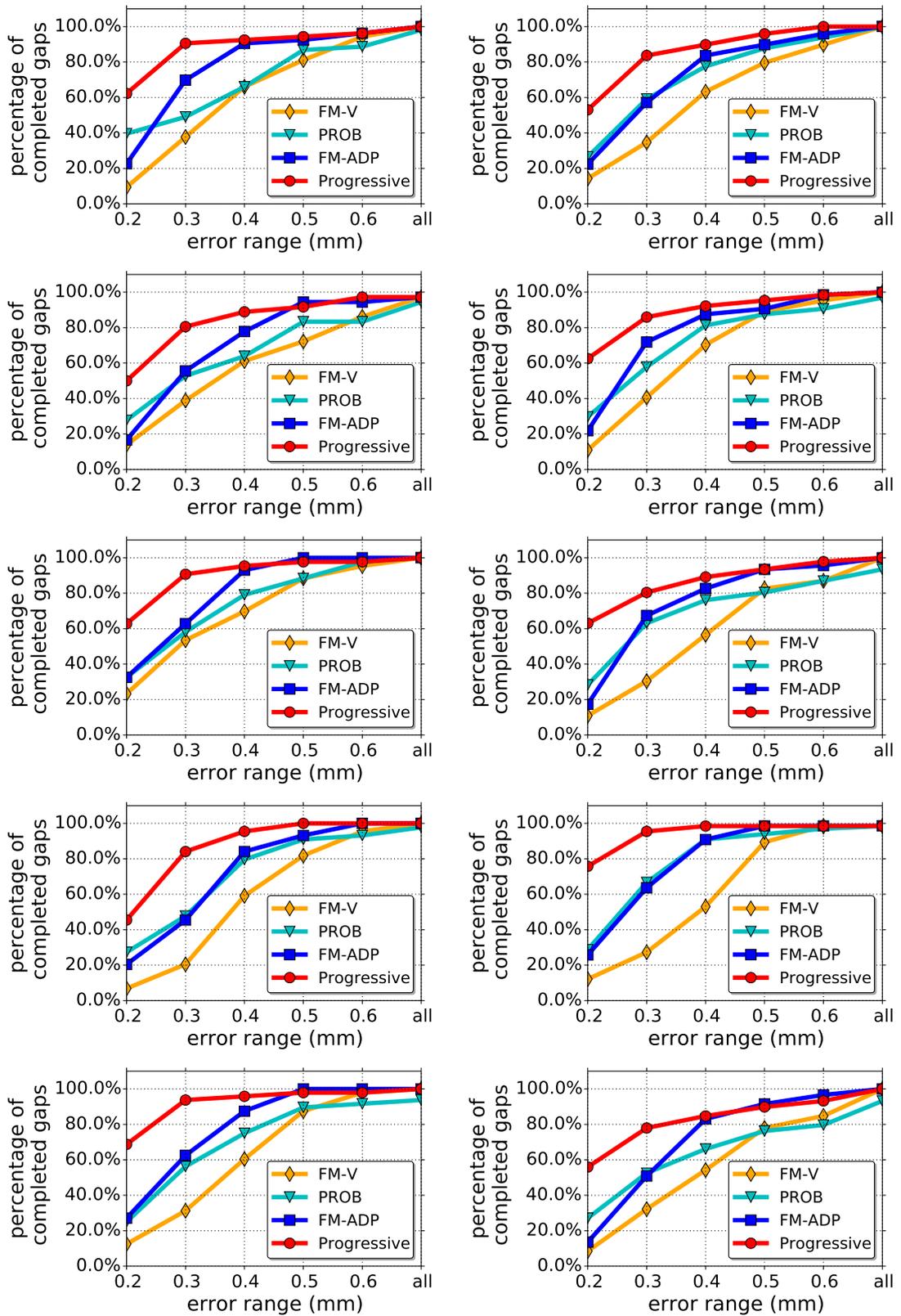


Figure 7.13: Percentage of the total number of completed gaps for ten 3D 7T MRA images of the LSA region of the human brain using different minimal path methods.

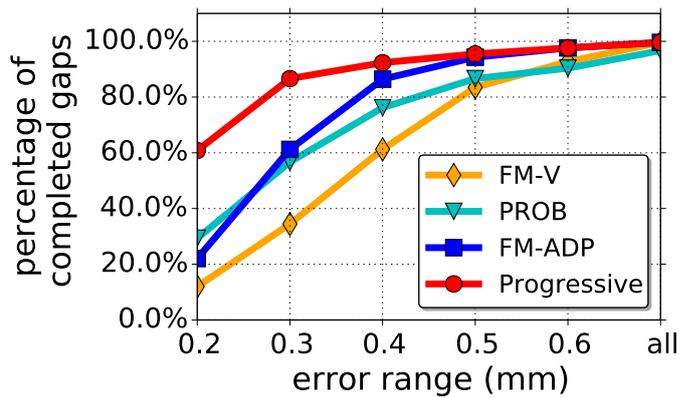


Figure 7.14: Percentage of the total number of completed gaps for 3D 7T MRA images: Average over all 10 3D 7T MRA images.

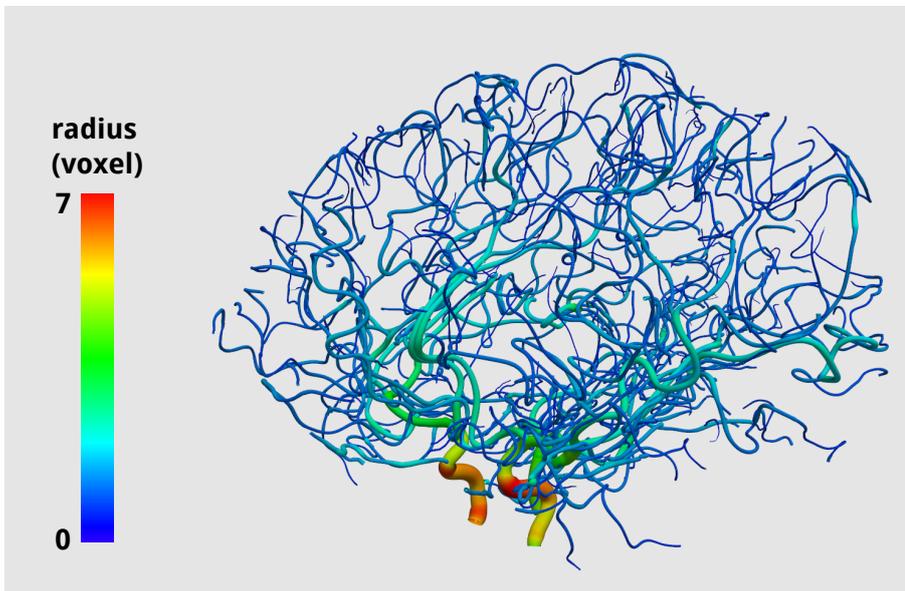


Figure 7.15: Segmentation of vessels of the whole brain from a 3D 7T MRA image using our progressive method in conjunction with the model-based approach. The color-coded radius ranges from 0 to 7 voxels.

7.3 Joint Appearance and Geometric Feature

We also applied our progressive minimal path method using the joint appearance and geometric feature (Sect. 6.2.3 above) to segment rivers and detect bridges in satellite images from Google Earth. We used 10 images that contain different types of bridges, rivers, and environment. The size of the images varies from 363×129 to 1617×1029 pixels. For example, Fig. 7.16a shows a narrow river with four bridges, while Fig. 7.16b shows a much wider river, along with two long bridges. In Fig. 7.16c, the river in the urban area is surrounded by buildings,

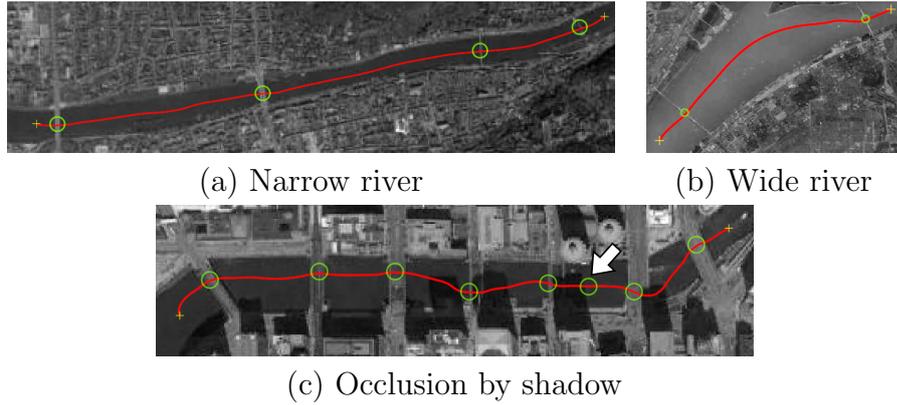


Figure 7.16: Bridge detection in satellite images with different types of bridges, rivers, and environments. Extracted rivers are indicated by the red lines and detected bridges by green circles. The white arrow indicates a false positive.

which cast shadows onto the river. Our approach was able to detect all bridges in all images. In the case that the river is partially occluded by shadows (Fig. 7.16c), the detection is more difficult, and a false positive is caused by shadows of the buildings (see the white arrow).

Fig. 7.17 shows the result of applying our approach to a satellite image with 37 bridges. This image is challenging since this river is very long and its width varies strongly. Furthermore, there are many objects on the river and streets near the river, which have similar widths as the bridges. We compared our method (red line) with the method in Sect. 6.2.1, which uses a local path γ_{local} but no tangent t_{local} (yellow line), and with the classical fast marching method, which uses neither t_{local} nor γ_{local} (blue line). It turns out that by exploiting both t_{local} and γ_{local} in our method, the result is improved compared to the other two methods. First, bridges are detected, which is not possible with the other two methods. Second, rivers are detected more accurately. The results for four regions (cyan boxes) have been enlarged. In region 3, a branch of the river (top left) was not found due to the general limitation of minimal path methods, and consequently the three bridges in that branch cannot be detected. All remaining 34 bridges have been correctly detected. There is one false positive in the regions 1 and 3 (indicated by the white arrow), caused by objects on the river.

Note that for bridge detection, it is crucial to accurately compute the orientation of the bridges to determine $\bar{\mathcal{V}}_{\text{B}}(t_{\text{local}}(\mathbf{x}))$ defined in Sect. 6.2.3 above. For example, the river in Fig. 7.17 resembles roughly a semi-circle, and the 37 bridges generally have different orientations. Our approach can not only cope with 37 orientations, but also with an arbitrary number of different orientations at *constant*

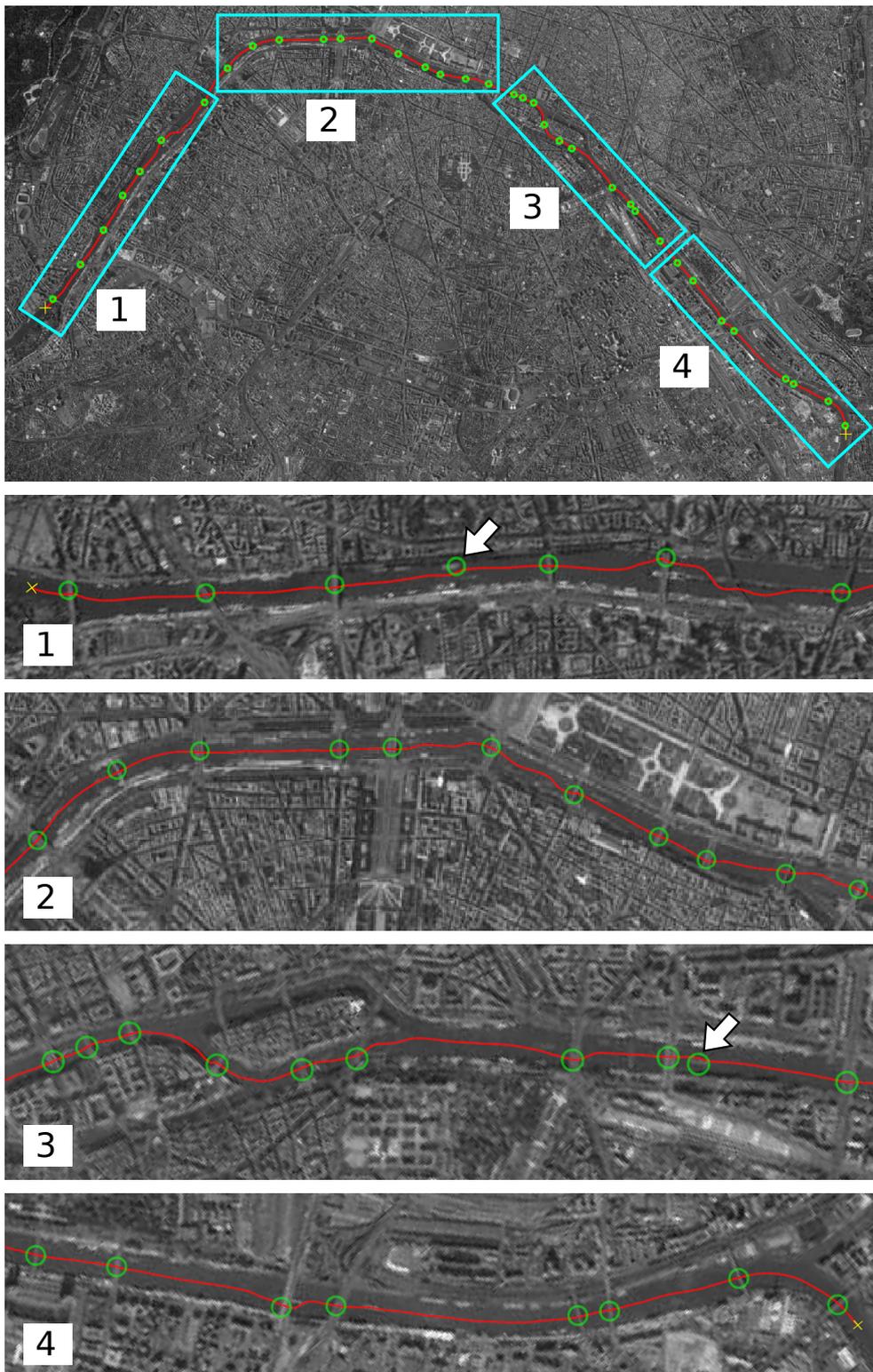


Figure 7.17: Bridge detection in a satellite image. Whole image and enlarged regions marked by the cyan boxes 1, 2, 3, and 4. Extracted rivers are indicated by red (our approach), yellow (appearance feature only), and blue (classical fast marching) lines. Detected bridges are indicated by green circles. White arrows indicate false positives.

7.3. Joint Appearance and Geometric Feature

cost due to the continuous nature of our approach. This is a difference to discrete minimal path approaches, where the number of orientations is limited (e.g., 16 orientations can be handled using 32 edges for each node in [56]).

Chapter 8

Summary and Conclusion

In this thesis, we introduced two different methods for the segmentation of 2D and 3D line structures. The first method includes two *direction-preserving minimal path methods* for segmentation of cerebral vessels from high-resolution 3D 7T MRA images of the human brain as well as for segmentation of vessels in 2D retinal images. The second method is a novel *progressive minimal path method* for efficient segmentation of line structures from 2D and 3D images.

From a theoretical point of view, these methods present algorithmic extensions to minimal path methods, which are commonly used for the segmentation of line structures. From a practical point of view, these methods cope with the limitations of previous minimal path methods, and the segmentation of 2D and 3D line structures are enhanced in different application areas.

8.1 Direction-Preserving Minimal Path Methods

We introduced two minimal path methods which incorporate prior directional information for vessel segmentation (Chapter 4). These methods were used within a novel automatic hybrid 3D segmentation and quantification approach to extract lenticulostriate arteries (LSAs) from high-resolution 3D 7T MRA images of human brains. We also introduced an approach to segment retinal vessels in 2D images using an iterative direction-preserving minimal path method.

Our hybrid 3D approach for vessel segmentation combines our direction-preserving 3D minimal path methods with a robust 3D model-based approach. After an automatic initialization, thick vessels and most parts of thin vessels are segmented using the model-based approach. Furthermore, at the ends of segmented vessels, the directions of the vessels in gaps are determined. Then gaps between vessels

are completed using our minimal path approaches, which take into account directional information. We have proposed two different minimal path approaches: An explicit approach based on probabilistic sampling, and an implicit approach based on fast marching with anisotropic directional prior.

Our 2D approach for vessel segmentation extends the fast marching approach with anisotropic directional prior to extract 2D retinal vessels iteratively. In each iteration, a directional prior derived from the last iteration is combined with vesselness and radius information to segment a vessel centerline and to quantify the vessel radius within the current ROI, as well as to find the center of the next ROI. The results from each iteration are concatenated to form the final segmentation.

We conducted an extensive evaluation for our 3D and 2D vessel segmentation approaches, including qualitative and quantitative experiments using synthetic and real images (Chapter 5). For our 3D approach, we used over 2300 3D synthetic images which resemble typical vessels in real 3D 7T MRA images, and 40 3D 7T MRA images of the LSA region. The 3D synthetic images in the experiments were generated using typical image parameters (angle between vessels, gap length, noise level) of real 3D 7T MRA images. It turned out that our two direction-preserving minimal path approaches yield more accurate results than a previous fast marching approach, which uses the same speed function as one of our methods, but does not use direction information. Furthermore, our fast marching approach with anisotropic directional prior achieves relatively low mean errors and standard deviations for a much larger range of image parameters. For the experiments with real 3D 7T MRA images, a quantitative evaluation with ground truth showed that our approaches achieve lower errors than the previous fast marching approach. Qualitatively, the vessels segmented by our approaches have a more realistic shape, i.e., the transitions between the vessel branches and the main vessels are smooth, and the curvature is well represented. These two properties cannot be satisfied by the previous fast marching approach. Furthermore, our approaches were successfully applied in two clinical studies concerning stroke and vascular dementia. In both studies, accurate segmentation and quantification of cerebral vessels turned out to be crucial for distinguishing between normal and pathological cases. While the fast marching approach with anisotropic directional prior is more efficient and achieves a lower error than the approach based on probabilistic sampling in most cases, there are few cases where the latter approach performs better. Also, the approach based on probabilistic sampling is easier to implement. For our 2D approach, the results show that our approach can deal well with two common difficulties of fast marching approaches, i.e., short cuts and vessel cross-

ings. Quantitative evaluation using 91 vessels from the DRIVE dataset shows that compared with a previous approach (Li-Yezzi), our approach achieves consistently a higher true positive rate and a lower false positive rate, each with a lower standard deviation. Furthermore, our approach is significantly more efficient than the previous approach.

Future work includes the segmentation and quantification of more 3D data of cerebral vessels, as well as application to further MRA imaging sequences, such as susceptibility weighted imaging. Also, the 2D iterative approach based on fast marching with anisotropic directional prior could be extended to segment vessels in 3D images.

8.2 Progressive Minimal Path Method

We also introduced a novel *progressive* minimal path method based on dynamic speed and path features (Chapter 6). We presented three instances of our *progressive* minimal path framework, including an appearance feature, a geometric feature, and a joint appearance and geometric feature. Previous minimal path approaches used *point features*. To avoid short cuts, different improvements based on anisotropic speed, domain lifting, line graph, directional information, and additional key points have been proposed. But these approaches are computationally demanding, introduce discretization and metrication errors, have limitations in the numerical scheme, or use limited features. In comparison, our method enables coping with these limitations by introducing *path features* for local paths. These features are created by aggregating point features, which can be computed efficiently for arbitrary dimensions. The path features can only be determined during the wave propagation, so they cannot be incorporated in previous minimal path approaches using static speed. With our progressive minimal path method, we can significantly improve the segmentation result of line structures in 2D and 3D images while avoiding short cuts. In addition, we combined segmentation based on minimal paths with object detection to detect bridges in satellite images. Our approach does not suffer from discretization or metrication errors.

We conducted an extensive evaluation of our method using 7560 2D synthetic images, 50 2D retinal images from the STARE and DRIVE datasets, 30 satellite images from Google Maps and Google Earth, 300 3D synthetic images, 40 real high-resolution 3D 7T MRA images of the human brain around the lenticulostriate region, and three high-resolution 3D 7T MRA images of the whole human brain (Chapter 7). Our quantitative comparisons with previous approaches showed that

our method achieves more accurate results.

Currently, our method only extracts the centerlines of line structures. An extension would be to determine also the radius of the line structures. In addition, more application-specific speed functions could be exploited instead of the general vesselness measure.

Bibliography

- [1] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active Contour Models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [2] C. Xu and J. Prince, “Snakes, Shapes, and Gradient Vector Flow,” *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.
- [3] J. Boykov and G. Funka-Lea, “Graph Cuts and Efficient N-D Image Segmentation,” *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.
- [4] C. Rother, V. Kolmogorov, and A. Blake, ““GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts,” *ACM Transactions on Graphics*, vol. 23, pp. 309–314, 2004.
- [5] P. Kohli, P. Kumar, and P. H. S. Torr, “P3 & Beyond: Move Making Algorithms for Solving Higher Order Functions,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1645–1656, 2009.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105.
- [7] P. Kohli, A. Osokin, and S. Jegelka, “A Principled Deep Random Field Model for Image Segmentation,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’13)*. Portland, OR/USA: IEEE Computer Society, June 2013, pp. 1971–1978.
- [8] E. Shelhamer, J. Long, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 640–651, no. 39, p. 4, 2017.
- [9] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, “Learning to Refine Object Segments,” in *Proc. European Conference on Computer Vision*

- (*ECCV'16*), ser. Lecture Notes in Computer Science, vol. 9905. Amsterdam, The Netherlands: Springer Berlin Heidelberg, Oct. 2016, pp. 75–91.
- [10] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” in *Proc. International Conference on Computer Vision (ICCV'17)*. Venice, Italy: IEEE Computer Society Press, Oct. 2017, pp. 2980–2988.
- [11] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A Database of Human Segmented Natural Images and Its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics,” in *IEEE International Conference on Computer Vision (ICCV'01)*. Vancouver, British Columbia, Canada: IEEE Computer Society Press, July 2001, pp. 416–423.
- [12] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes Challenge: A Retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.
- [13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-Based Learning Applied to Document Recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [14] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Tech. Rep., 2009.
- [15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *Proc. International Conference on Computer Vision (ICCV'09)*. Kyoto, Japan: IEEE Computer Society Press, Sept.–Oct. 2009, pp. 248–255.
- [16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” in *Proc. European Conference on Computer Vision (ECCV'14)*, ser. Lecture Notes in Computer Science, vol. 8693. Zurich, Switzerland: Springer Berlin Heidelberg, Sept. 2014, pp. 740–755.
- [17] H. Kolb, “How the Retina Works,” *American Scientist*, vol. 91, no. 1, pp. 28–35, 2003.
- [18] D. Nathan, “Long-Term Complications of Diabetes Mellitus,” *The New England Journal of Medicine*, vol. 328, pp. 1676–1685, 1993.

- [19] K. Malpass, “Retinal Changes Predict Subsequent Vascular Events in Ischemic Stroke,” *Nature Review Neurology*, vol. 7, p. 538, 2011.
- [20] J. Flammer, S. Orgül, V. P. Costa, N. Orzalesi, G. K. Krieglstein, L. M. Serra, J.-P. Renard, and E. Stefánsson, “The Impact of Ocular Blood Flow in Glaucoma,” *Progress in Retinal and Eye Research*, vol. 28, no. 11, pp. 1768–1783, 2002.
- [21] M. B. Sasongko, T. Y. Wong, T. T. Nguyen, C. Cheung, J. E. Shaw, and J. J. Wang, “Retinal Vascular Tortuosity in Persons with Diabetes and Diabetic Retinopathy,” *Diabetologia*, vol. 54, no. 9, pp. 2409–2416, 2011.
- [22] P. Mitchell, H. Leung, J. J. Wang, E. Rochtchina, A. J. Lee, T. Y. Wong, and R. Klein, “Retinal Vessel Diameter and Open-Angle Glaucoma: the Blue Mountains Eye Study,” *Ophthalmology*, vol. 112, no. 2, pp. 245–250, 2005.
- [23] J. Gao, Y. Liang, F. Wang, R. Shen, T. Wong, Y. Peng, D. S. Friedman, and N. Wang, “Retinal Vessels Change in Primary Angle-Closure Glaucoma: the Handan Eye Study,” *Scientific Reports*, vol. 9585, 2015.
- [24] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, “Ridge based Vessel Segmentation in Color Images of the Retina,” *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.
- [25] Z.-H. Cho, C.-K. Kang, J.-Y. Han, S.-H. Kim, K.-N. Kim, S.-M. Hong, C.-W. Park, and Y.-B. Kim, “Observation of the Lenticulostriate Arteries in the Human Brain In Vivo using 7.0T MR Angiography,” *Stroke*, vol. 39, no. 5, pp. 1604–1606, 2008.
- [26] D. Purves, G. J. Augustine, D. Fitzpatrick, W. C. Hall, A.-S. LaMantia, J. O. McNamara, and L. E. White, *Neuroscience*, 3rd ed. Sinauer Associates Inc., 2004.
- [27] G. Roman, T. Erkinjuntti, A. Wallin, L. Pantoni, and H. Chui, “Subcortical Ischaemic Vascular Dementia,” *Lancet Neurology*, vol. 1, pp. 426–436, 2002.
- [28] J. Zwanenburg and M. van Osch, “Targeting Cerebral Small Vessel Disease with MRI,” *Stroke*, vol. 48, no. 11, pp. 3175–3182, 2017.

- [29] A. Qureshi, S. Tuhim, J. Broderick, H. Batjer, H. Hondo, and D. Hanley, “Spontaneous Intracerebral Hemorrhage,” *New England Journal of Medicine*, vol. 344, pp. 1450–1460, 2001.
- [30] J. Heverhagen, E. Bourekas, S. Sammet, M. Knopp, and P. Schmalbrock, “Time-of-Flight Magnetic Resonance Angiography at 7 Tesla,” *Investigative Radiology*, vol. 43, no. 8, pp. 568–573, 2008.
- [31] W. Nowinski, B. Chua, Y. Marchenko, F. Puspitsari, I. Volkau, and M. Knopp, “Three-Dimensional Reference and Stereotactic Atlas of Human Cerebrovasculature from 7 Tesla,” *NeuroImage*, vol. 55, pp. 986–998, 2011.
- [32] P. van Ooij, J. Zwanenburg, F. Visser, C. Majoie, E. vanBavel, J. Hendrikse, and A. Nederveen, “Quantification and Visualization of Flow in the Circle of Willis: Time-Resolved Three-Dimensional Phase Contrast MRI at 7 T Compared with 3 T,” *Magnetic Resonance in Medicine*, vol. 69, pp. 868–876, 2013.
- [33] P. Koopmans, R. Manniesing, W. Niessen, M. Viergever, and M. Barth, “MR Venography of the Human Brain using Susceptibility Weighted Imaging at Very High Field Strength,” *Magnetic Resonance Materials in Physics, Biology and Medicine*, vol. 21, pp. 149–158, 2008.
- [34] G. Grabner, A. Dal-Bianco, S. Hametner, H. Lassmann, and S. Trattnig, “Group Specific Vein-Atlasing: An Application for Analyzing the Venous System under Normal and Multiple Sclerosis Conditions,” *Journal of Magnetic Resonance Imaging*, vol. 40, pp. 655–661, 2014.
- [35] K. Diedrich, J. Roberts, R. Schmidt, C.-K. Kang, Z.-H. Cho, and D. Parker, “Validation of an Arterial Tortuosity Measure with Application to Hypertension Collection of Clinical Hypertensive Patients,” *BMC Bioinformatics*, vol. 12, Suppl 10, p. S15, 2011.
- [36] A. Matese, P. Toscano, S. F. D. Gennaro, L. Genesio, F. P. Vaccari, J. Primicerio, C. Belli, A. Zaldei, R. Bianconi, and B. Gioli, “Intercomparison of UAV, Aircraft and Satellite Remote Sensing Platforms for Precision Viticulture,” *Remote Sensing*, vol. 7, no. 3, pp. 2971–2990, 2015.
- [37] W. Liao, K. Rohr, C.-K. Kang, Z.-H. Cho, and S. Wörz, “A Generative MRF Approach for Automatic 3D Segmentation of Cerebral Vasculature from 7 Tesla MRA Images,” in *Proc. IEEE International Symposium on*

- Biomedical Imaging: From Nano to Macro (ISBI'11)*. Chicago, IL/USA: IEEE Computer Society, March–April 2011, pp. 2041–2043.
- [38] W. Liao, S. Wörz, and K. Rohr, “Vessel Segmentation using an Iterative Fast Marching Approach with Directional Prior,” in *Proc. SPIE Medical Imaging 2012: Image Processing (MI'12)*, vol. 8314. San Diego, CA/USA: SPIE Bellingham, WA/USA, Feb. 2012.
- [39] W. Liao, K. Rohr, C.-K. Kang, Z.-H. Cho, and S. Wörz, “Automatic Human Brain Vessel Segmentation from 3D 7 Tesla MRA Images using Fast Marching with Anisotropic Directional Prior,” in *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'12)*. Barcelona, Spain: IEEE Computer Society, May 2012, pp. 1140–1143.
- [40] W. Liao, K. Rohr, C.-K. Kang, Z.-H. Cho, and S. Wörz, “Automatic 3D Segmentation and Quantification of Lenticulostriate Arteries from High-Resolution 7 Tesla MRA Images,” *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 400–413, 2016.
- [41] L. Cohen and R. Kimmel, “Global Minimum for Active Contour Models: A Minimal Path Approach,” *International Journal of Computer Vision*, vol. 24, no. 1, pp. 57–78, 1997.
- [42] Q. Lin, “Enhancement, Extraction, and Visualization of 3D Volume Data,” Ph.D. dissertation, Department of Electrical Engineering, Linköpings University, Linköping, Sweden, 2003.
- [43] S. Jbabdi, P. Bellec, R. Toro, J. Daunizeau, M. Pelegriani-Issac, and H. Bernali, “Accurate Anisotropic Fast Marching for Diffusion-Based Geodesic Tractography,” *International Journal of Biomedical Imaging*, vol. 2008, pp. 1–12, 2008.
- [44] F. Benmansour and L. D. Cohen, “Tubular Structure Segmentation based on Minimal Path Method and Anisotropic Enhancement,” *International Journal of Computer Vision*, vol. 92, no. 2, pp. 192–210, 2011.
- [45] P. Lo, B. van Ginneken, and M. de Bruijne, “Vessel Tree Extraction using Locally Optimal Paths,” in *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI'10)*. Rotterdam, The Netherlands: IEEE Computer Society, April 2010, pp. 680–683.

-
- [46] S. Wörz and K. Rohr, “Segmentation and Quantification of Human Vessels using a 3-D Cylindrical Intensity Model,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 1994–2004, 2007.
- [47] W. Liao, S. Wörz, and K. Rohr, “Globally Minimal Path Method Using Dynamic Speed Functions Based on Progressive Wave Propagation,” in *Proc. Asian. Conference on Computer Vision (ACCV’12)*, ser. Lecture Notes in Computer Science, vol. 7725. Dajeon, Korea: Springer Berlin Heidelberg, Nov. 2012, pp. 25–37.
- [48] W. Liao, K. Rohr, and S. Wörz, “Globally Optimal Curvature-Regularized Fast Marching for Vessel Segmentation,” in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’13)*, ser. Lecture Notes in Computer Science, vol. 8149. Nagoya, Japan: Springer Berlin Heidelberg, Sept. 2013, pp. 550–557.
- [49] W. Liao, S. Wörz, C.-K. Kang, Z.-H. Cho, and K. Rohr, “Progressive Minimal Path Method for Segmentation of 2D and 3D Line Structures,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 696–709, 2018.
- [50] A. Frangi, W. Niessen, K. Vincken, and M. Viergever, “Multiscale Vessel Enhancement Filtering,” in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’98)*, ser. Lecture Notes in Computer Science, vol. 1496. Cambridge, MA/USA: Springer Berlin Heidelberg, Oct. 1998, pp. 130–137.
- [51] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, T. Gerig, and R. Kikinis, “Three-Dimensional Multi-Scale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images,” *Medical Image Analysis*, vol. 2, no. 2, pp. 143–168, 1998.
- [52] M. Law and A. Chung, “Three Dimensional Curvilinear Structure Detection using Optimally Oriented Flux,” in *Proc. European Conference on Computer Vision (ECCV’08)*, ser. Lecture Notes in Computer Science, vol. 5305. Marseille, France: Springer Berlin Heidelberg, Oct. 2008, pp. 368–382.
- [53] V. Mnih and G. Hinton, “Learning to Detect Roads in High-Resolution Aerial Images,” in *Proc. European Conference on Computer Vision (ECCV’10)*, ser. Lecture Notes in Computer Science, vol. 6316. Heraklion, Crete, Greece: Springer Berlin Heidelberg, Sept. 2010, pp. 210–223.

- [54] A. Sironi, E. Turetken, V. Lepetit, and P. Fua, “Multiscale Centerline Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1327–1341, 2016.
- [55] T. Schoenemann, S. Masnou, and D. Cremers, “The Elastic Ratio: Introducing Curvature Into Ratio-Based Image Segmentation,” *IEEE Transactions on Image Processing*, vol. 20, no. 9, pp. 2565–2581, 2011.
- [56] H. Ulén, P. Strandmark, and F. Kahl, “Shortest Paths with Higher-Order Regularization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2588–2600, 2015.
- [57] G. Parker, C. Wheeler-Kingshott, and G. Barker, “Estimating Distributed Anatomical Connectivity using Fast Marching Methods and Diffusion Tensor Imaging,” *IEEE Transactions on Medical Imaging*, vol. 21, no. 5, pp. 505–512, 2002.
- [58] V. Joshi, M. Garvin, J. Reinhardt, and M. Abramoff, “Identification and Reconnection of Interrupted Vessels in Retinal Vessel Segmentation,” in *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI’11)*. Chicago, IL/USA: IEEE Computer Society, March–April 2011, pp. 1416–1420.
- [59] M. Krueger, P. Delmas, and G. Gimel’farb, “Robust and Efficient Object Segmentation Using Pseudo-Elastica,” *Pattern Recognition Letters*, vol. 34, pp. 833–845, 2013.
- [60] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Pearson, 2012.
- [61] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Pearson, 2001.
- [62] E. R. Davies, *Computer and Machine Vision*. Academic Press, 2012.
- [63] R. Szeliski, *Computer Vision*. Springer-Verlag, 2011.
- [64] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Cengage Learning, 2014.
- [65] S. J. D. Prince, *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

-
- [66] C. Kirbas and F. Quek, “A Review of Vessel Extraction Techniques and Algorithms,” *ACM Computing Surveys*, vol. 36, no. 2, pp. 81–121, 2004.
- [67] D. Lesage, E. Angelini, I. Bloch, and G. Funka-Lea, “A Review of 3D Vessel Lumen Segmentation Techniques: Models, Features and Extraction Schemes,” *Medical Image Analysis*, vol. 13, no. 6, pp. 819–845, 2009.
- [68] D. J. Withey and Z. J. Koles, “A Review of Medical Image Segmentation: Methods and Available Software,” *International Journal of Bioelectromagnetism*, vol. 10, no. 3, pp. 125–148, 2008.
- [69] A. Elnakib, G. Gimel’farb, J. S. Suri, and A. El-Baz, *Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies*. Springer New York, 2011, ch. Medical Image Segmentation: A Brief Survey, pp. 1–39.
- [70] T. Koller, G. Gerig, G. Székely, and D. Dettwiler, “Multiscale Detection of Curvilinear Structures in 2-D and 3-D Image Data,” in *Proc. Fifth International Conference on Computer Vision (ICCV’95)*. Cambridge, MA/USA: IEEE Computer Society Press, June 1995, pp. 864–869.
- [71] T. Lindeberg, “Edge Detection and Ridge Detection with Automatic Scale Selection,” *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117–156, 1998.
- [72] R. Manniesing, M. Viergever, and W. Niessen, “Vessel Enhancing Diffusion: A Scale Space Representation of Vessel Structure,” *Medical Image Analysis*, vol. 10, no. 6, pp. 815–825, 2006.
- [73] A. Vasilevskiy and K. Siddiqi, “Flux Maximizing Geometric Flows,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1565–1578, 2002.
- [74] M. W. K. Law and A. C. S. Chung, “An Oriented Flux Symmetry Based Active Contour Model for Three Dimensional Vessel Segmentation,” in *Proc. European Conference on Computer Vision (ECCV’10)*, ser. Lecture Notes in Computer Science, vol. 6313. Heraklion, Crete, Greece: Springer Berlin Heidelberg, Sept. 2010, pp. 720–734.
- [75] E. Tueretken, C. Becker, P. Glowacki, F. Benmansour, and P. Fua, “Detecting Irregular Curvilinear Structures in Gray Scale and Color Imagery

- using Multi-Directional Oriented Flux,” in *Proc. International Conference on Computer Vision (ICCV'13)*. Sydney, Australia: IEEE Computer Society Press, Dec. 2013, pp. 1553–1560.
- [76] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” *arXiv:1603.04467*, 2016.
- [77] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional Architecture for Fast Feature Embedding,” *arXiv:1408.5093*, 2014.
- [78] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhan, “MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems,” *arXiv:1512.01274*, 2015.
- [79] F. Seide and A. Agarwal, “CNTK: Microsoft’s Open-Source Deep-Learning Toolkit,” in *Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, 2016, pp. 2135–2135.
- [80] R. Collobert, K. Kavukcuoglu, and C. Farabet, “Torch7: A Matlab-like Environment for Machine Learning,” in *BigLearn, Workshop of Advances in Neural Information Processing Systems*, 2011.
- [81] F. Chollet *et al.*, “Keras,” 2015. [Online]. Available: <https://github.com/keras-team/keras>
- [82] A. Vedaldi and K. Lenc, “Matconvnet – convolutional neural networks for matlab,” in *CoRR*, *abs/1412.4564*, 2014.
- [83] S. Dieleman, J. Schlüter, C. Raffel, E. Olson, S. K. Sønderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. D. Fauw, M. Heilman, D. M. de Almeida, B. McFee, H. Weideman, G. Takács, P. de Rivaz, J. Crall, G. Sanders, K. Rasul, C. Liu, G. French, and J. Degraeve, “Lasagne: First release.” Aug. 2015. [Online]. Available: <http://dx.doi.org/10.5281/zenodo.27878>

-
- [84] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’16)*. Las Vegas, NV/USA: IEEE Computer Society, June 2016, pp. 779–788.
- [85] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 39, pp. 1137 – 1149, 2016.
- [86] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully Convolutional Instance-aware Semantic Segmentation,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’17)*. Honolulu, HI/USA: IEEE Computer Society, July 2017, pp. 4438–4446.
- [87] E. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, “Exploiting Linear Structure Within Convolutional Networks for Efficient Evaluation,” in *Proc. Advances in Neural Information Processing Systems (NIPS’14)*. Montreal, Canada: Curran Associates, Inc., Dec. 2014, pp. 1269–1277.
- [88] P. Liskowski and K. Krawiec, “Segmenting Retinal Blood Vessels with Deep Neural Networks,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 11, pp. 2369 – 2380, 2016.
- [89] E. Nasr-Esfahani, S. Samavi, N. Karimi, S. Soroushmehr, K. Ward, and M. Jafari, “Vessel Extraction in X-Ray Angiograms Using Deep Learning,” in *Proc. International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC’16)*. Orlando, FL/USA: IEEE Engineering in Medicine and Biology Society, Aug. 2016, pp. 643–646.
- [90] Y. Zheng, M. Loziczonek, B. Georgescu, S. K. Zhou, F. Vega-Higuera, and D. Comaniciu, “Machine Learning Based Vesselness Measurement for Coronary Artery Segmentation in Cardiac CT Volumes,” in *SPIE Medical Imaging*, vol. 7962, no. 1, 2011.
- [91] A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua, “Learning Separable Filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 94–106, 2015.
- [92] A. Gooya, H. Liao, K. Matsumiya, K. Masamune, Y. Masutani, and T. Dohi, “A Variational Method for Geometric Regularization of Vascular Segmenta-

- tion in Medical Images,” *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1295–1312, 2008.
- [93] S. Esneault, C. Lafon, and J. Dillenseger, “Liver Vessels Segmentation using a Hybrid Geometrical Moments/Graph Cuts Method,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 2, pp. 276–283, 2010.
- [94] O. Friman, M. Hindennach, C. Kühnel, and H.-O. Peitgen, “Multiple Hypothesis Template Tracking of Small 3D Vessel Structures,” *Medical Image Analysis*, vol. 14, no. 2, pp. 160–171, 2010.
- [95] A. El-Baz, A. Elnakib, F. Khalifa, M. A. El-Ghar, P. McClure, A. Soliman, and G. Gimel’farb, “Precise Segmentation of 3-D Magnetic Resonance Angiography,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 7, pp. 2019–2029, 2012.
- [96] S. Osher and J. Sethian, “Fronts Propagating with Curvature-Dependent Speed: Algorithms based on Hamilton-Jacobi Formulations,” *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.
- [97] Wikipedia. An illustration of the level set method. Accessed: 2016 Feb. 8. [Online]. Available: https://en.wikipedia.org/wiki/Level_set_method#/media/File:Level_set_method.jpg
- [98] V. Caselles, R. Kimmel, and G. Sapiro, “Geodesic Active Contours,” *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.
- [99] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, 2005.
- [100] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer-Verlag, 2006.
- [101] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.
- [102] J. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [103] R. Kimmel, *Numerical Geometry of Images*. Springer-Verlag, 2004.

-
- [104] T. F. Chan and L. A. Vese, “Active Contours without Edges,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [105] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag, 2002.
- [106] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, “A PDE-based fast local level set method,” *Journal of Computational Physics*, vol. 155, no. 2, pp. 410–438, 1999.
- [107] J. Gomes and O. Faugeras, “Reconciling distance functions and level sets,” *Journal of Visual Communication and Image Representation*, vol. 11, no. 2, pp. 209–223, 2000.
- [108] C. Li, C. Xu, C. Gui, and M. D. Fox, “Distance Regularized Level Set Evolution and Its Application to Image Segmentation,” *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3243–3254, 2010.
- [109] F. Catte, F. Dibos, and G. Koepfler, “A Morphological Scheme for Mean Curvature Motion and Applications to Anisotropic Diffusion and Motion of Level Sets,” *SIAM Journal of Numerical Analysis*, vol. 32, no. 6, pp. 1895–1909, 1995.
- [110] F. Guichard and J.-M. Morel, “Geometric Partial Differential Equations and Iterative Filtering,” in *Mathematical Morphology and Its Applications to Image and Signal Processing (GSMM’98)*, Amsterdam, The Netherlands, 1998, pp. 127–138.
- [111] P. Marquez-Neila, L. Baumela, and L. Alvarez, “A Morphological Approach to Curvature-Based Evolution of Curves and Surfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 2–17, 2014.
- [112] M. Niethammer and C. Zach, “Segmentation with Area Constraints,” *Medical Image Analysis*, vol. 17, pp. 101–112, 2013.
- [113] L. Cohen and I. Cohen, “Finite Element Methods for Active Contour Models and Balloons for 2D and 3D Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1131–1147, 1993.
- [114] Y. Shang, R. Deklerck, E. Nyssen, A. Markova, J. de Mey, X. Yang, and K. Sun, “Vascular Active Contour for Vessel Tree Segmentation,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 4, pp. 1023–1032, 2011.

- [115] S. Geman and D. Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.
- [116] G. Sapiro, *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.
- [117] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, 2009.
- [118] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [119] J. Besag, “On the statistical analysis of dirty pictures,” *Journal of the Royal Statistical Society, Series B*, vol. 48, no. 3, pp. 48–259, 1986.
- [120] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA, USA: MIT Press, 1987.
- [121] J. Sun, N.-N. Zheng, and H.-Y. Shum, “Stereo Matching Using Belief Propagation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, 2003.
- [122] V. Kolmogorov, “Convergent Tree-Reweighted and Message and Passing for Energy and Minimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1568–1583, 2006.
- [123] P. Hammer, P. Hansen, and B. Simeone, “Roof duality, complementation and persistency in quadratic 0-1 optimization,” *Mathematical Programming*, vol. 28, pp. 121–155, 1984.
- [124] V. Kolmogorov and C. Rother, “Minimizing Non-Submodular Functions with Graph Cuts – a Review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 7, pp. 1274–1279, 2007.
- [125] Y. Boykov, O. Veksler, and R. Zabih, “Fast Approximate Energy Minimization via Graph Cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.
- [126] Y. Boykov and V. Kolmogorov, “Computing Geodesics and Minimal Surfaces via Graph Cuts,” in *Proc. IEEE International Conference on Computer Vision (ICCV’03)*. Nice, France: IEEE Computer Society Press, Oct. 2003, pp. 26–33.

-
- [127] L. Ladicky, C. Russel, P. Kohli, and P. H. S. Torr, “Graph Cut Based Inference with Co-Occurrence Statistics,” in *Proc. European Conference on Computer Vision (ECCV’10)*, ser. Lecture Notes in Computer Science, vol. 6315. Heraklion, Crete, Greece: Springer Berlin Heidelberg, Sept. 2010, pp. 239–253.
- [128] P. Kohli, L. Ladicky, and P. H. Torr, “Robust Higher Order Potentials for Enforcing Label Consistency,” *International Journal of Computer Vision*, vol. 82, no. 3, pp. 302–324, 2009.
- [129] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- [130] A. Blake, P. Kohli, and C. Rother, *Markov Random Fields For Vision And Image Processing*. Cambridge, MA/USA: MIT Press, 2011.
- [131] C. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.
- [132] Y. Boykov and V. Kolmogorov, “An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.
- [133] V. Kolmogorov and R. Zabih, “What Energy Functions Can Be Minimized via Graph Cuts?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.
- [134] L. Grady, “Random Walks and for Image and Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.
- [135] C. Couprie, L. Grady, L. Najman, and H. Talbot, “Power Watershed: A Unifying Graph-Based Optimization Framework,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1384–1399, 2011.
- [136] A. Vedaldi and S. Soatto, “Quick Shift and Kernel Methods for Mode Seeking,” in *Proc. European Conference on Computer Vision (ECCV’08)*, ser. Lecture Notes in Computer Science, vol. 5305. Marseille, France: Springer Berlin Heidelberg, Oct. 2008, pp. 705–718.

- [137] Y. Zhao, Y. Liu, X. Wu, S. P. Harding, and Y. Zheng, “Retinal Vessel Segmentation: An Efficient Graph Cut Approach with Retinex and Local Phase,” *PLOS ONE*, vol. 10, no. 4, pp. 1–22, 2015.
- [138] J. Zheng, P.-R. Lu, D. Xiang, Y.-K. Dai, Z.-B. Liu, D.-J. Kuai, H. Xue, and Y.-T. Yang, “Retinal Image Graph-Cut Segmentation Algorithm Using Multiscale Hessian-Enhancement-Based Nonlocal Mean Filter,” *Computational and Mathematical Methods in Medicine*, vol. 2013, pp. 1–7, 2013.
- [139] B. Chen, Y. Sun, and S. H. Ong, “Liver Vessel Segmentation Using Graph Cuts with Quick Shift Initialization,” in *International Conference on Biomedical Engineering (ICBE’13)*, vol. 43, 2013, pp. 188–191.
- [140] P. R. Wankhede and K. B. Khanchandani, “Retinal Blood Vessel Segmentation using Graph Cut Analysis,” in *International Conference on Industrial Instrumentation and Control (ICIC’15)*, Pune, India, May 2015, pp. 1429–1432.
- [141] A. Sinop and L. Grady, “Accurate Banded Graph Cut Segmentation of Thin Structures using Laplacian Pyramids,” in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’06)*, ser. Lecture Notes in Computer Science, vol. 4191. Copenhagen, Denmark: Springer Berlin Heidelberg, Oct. 2006, pp. 896–903.
- [142] T. Nuzhnaya, E. Cheng, H. Ling, D. Kontos, P. R. Bakic, and V. Megalooikonomou, “Segmentation of Anatomical Branching Structures Based on Texture Features and Graph Cut,” in *Proc. IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI’11)*. Chicago, IL/USA: IEEE Computer Society, March–April 2011, pp. 673–676.
- [143] S. Vicente, V. Kolmogorov, and C. Rother, “Graph Cut Based Image Segmentation with Connectivity Priors,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’08)*. Anchorage, AK/USA: IEEE Computer Society, June 2008, pp. 1–8.
- [144] S. Jegelka and J. Bilmes, “Submodularity Beyond Submodular Energies: Coupling Edges in Graph Cuts,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’11)*. Colorado Springs, CO/USA: IEEE Computer Society, June 2011, pp. 1897–1904.

-
- [145] N. Zhu and A. C. Chung, “Graph-Based Optimization with Tubularity Markov Tree for 3D Vessel Segmentation,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’13)*. Portland, OR/USA: IEEE Computer Society, June 2013, pp. 2219–2226.
- [146] E. Dijkstra, “A Note on Two Problems in Connection with Graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [147] J. Tsitsiklis, “Efficient Algorithms for Globally Optimal Trajectories,” *IEEE Transactions on Automatic Control*, vol. 40, pp. 1528–1538, 1995.
- [148] V. Kolmogorov and Y. Boykov, “What Metrics Can Be Approximated by Geo-Cuts, or Global Optimization of Length/Area and Flux,” in *Proc. IEEE International Conference on Computer Vision (ICCV’05)*. Beijing, China: IEEE Computer Society Press, Oct. 2005, pp. 564–571.
- [149] J. A. Sethian and A. Vladimirov, “Fast Methods for the Eikonal and Related Hamilton-Jacobi Equations on Unstructured Meshes,” *Proc. the National Academy of Sciences*, vol. 97, no. 11, pp. 5699–5703, 2000.
- [150] J.-M. Mirebeau, “Anisotropic Fast-Marching on Cartesian Grids Using Lattice Basis Reduction,” *SIAM Journal on Numerical Analysis*, vol. 52, no. 4, pp. 1573–1599, 2014.
- [151] J.-M. Mirebeau, “Efficient Fast Marching with Finsler Metrics,” *Numerische Mathematik*, vol. 126, no. 3, pp. 515–557, 2014.
- [152] D. Chen, J.-M. Mirebeau, and L. D. Cohen, “Global Minimum for a Finsler Elastica Minimal Path Approach,” *International Journal of Computer Vision*, vol. 122, no. 3, pp. 458–483, 2017.
- [153] L. Yatziv, A. Bartesaghi, and G. Sapiro, “O(N) Implementation of the Fast Marching Algorithm,” *J Comp Phys*, vol. 212, pp. 393–399, 2006.
- [154] H. Li and A. Yezzi, “Vessels as 4-D Curves: Global Minimal 4-D Paths to Extract 3-D Tubular Surfaces and Centerlines,” *IEEE Transactions on Medical Imaging*, vol. 26, no. 9, pp. 1213–1223, 2007.
- [155] M. Pechaud, R. Keriven, and G. Peyre, “Extraction of Tubular Structures Over an Orientation Domain,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’09)*, vol. 1. Miami, FL/USA: IEEE Computer Society, June 2009, pp. 336–342.

- [156] D. Chen, J.-M. Mirebeau, and L. D. Cohen, “Global Minimum for Curvature Penalized Minimal Path Method,” in *Proceedings of the British Machine Vision Conference (BMVC’15)*. BMVA Press, September 2015, pp. 86.1–86.12.
- [157] P. Strandmark. C++ shortest path with curvature and torsion taken into account. Accessed: 2015 Dec. 2. [Online]. Available: https://github.com/PetterS/curve_extraction
- [158] T. Deschamps, “Curve and Shape Extraction with Minimal Path and Level-Sets techniques. Applications to 3D Medical Imaging,” Ph.D. dissertation, University of Paris-Dauphine, Paris, France, 2001.
- [159] T. Deschamps and L. Cohen, “Fast Extraction of Tubular and Tree 3D Surfaces with Front Propagation Methods,” in *Proc. the International Conference on Pattern Recognition (ICPR’02)*, Quebec, Canada, Aug. 2002, pp. 731–734.
- [160] P. Lo, J. Sparring, J. J. H. Pedersen, and M. de Bruijne, “Airway Tree Extraction with Locally Optimal Paths,” in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’09)*, ser. Lecture Notes in Computer Science, vol. 5762. London, UK: Springer Berlin Heidelberg, Sept. 2009, pp. 51–58.
- [161] M. Poon, G. Hamarneh, and R. Abugharbieh, “Live-Vessel: Extending Livewire for Simultaneous Extraction of Optimal Medial and Boundary Paths in Vascular Images,” in *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI’07)*, ser. Lecture Notes in Computer Science, vol. 4792. Brisbane, Australia: Springer Berlin Heidelberg, Nov. 2007, pp. 444–451.
- [162] F. Benmansour and L. Cohen, “Fast Object Segmentation by Growing Minimal Paths from a Single Point on 2D or 3D Images,” *Journal of Mathematical Imaging and Vision*, vol. 33, no. 2, pp. 209–221, 2009.
- [163] V. Kaul, A. Yezzi, and Y. Tsai, “Detecting Curves with Unknown Endpoints and Arbitrary Topology using Minimal Paths,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 10, pp. 1952–1965, 2012.
- [164] D. Chen, L. Cohen, and J.-M. Mirebeau, “Vessel Extraction Using Anisotropic Minimal Paths and Path Score,” in *Proc. IEEE International*

-
- Conference on Image Processing (ICIP'14)*. IEEE Computer Society, Oct. 2014, pp. 1570–1574.
- [165] J. Stühmer, P. Schröder, and D. Cremers, “Tree Shape Priors with Connectivity Constraints using Convex Relaxation on General Graphs,” in *Proc. IEEE International Conference on Computer Vision (ICCV'13)*, Dec. 2013, pp. 336–2343.
- [166] M. Rempfler, M. Schneider, I. G. D., X. Xiao, R. S. Stock, J. Klohs, G. Szekely, B. Andres, and B. H. Menze, “Extracting Vascular Networks under Physiological Constraints via Integer Programming,” in *Proc. MIC-CAI'14*, ser. LNCS, no. 8674, Sept. 2014, pp. 505–512.
- [167] E. Türetken, F. Benmansour, B. Andres, P. Glowacki, H. Pfister, and P. Fua, “Reconstructing Curvilinear Networks Using Path Classifiers and Integer Programming,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2515 – 2530, 2016.
- [168] P. Glowacki, M. A. Pinheiro, A. Mosinska, E. Tueretken, D. Lebrecht, A. Holtmaat, R. Sznitman, J. Kybic, and P. Fua, “Reconstructing Evolving Tree Structures in Time Lapse Sequences by Enforcing Time-Consistency,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 755–761, 2017.
- [169] International Business Machines Corporation, “IBM ILOG CPLEX Optimization Studio.” [Online]. Available: <https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>
- [170] Gurobi Optimization, Inc., “Gurobi Optimization.” [Online]. Available: <http://www.gurobi.com>
- [171] Zuse Institute Berlin, “SCIP Optimization Suite.” [Online]. Available: <http://scip.zib.de/>
- [172] M. Freiman, N. Broide, M. Natanzon, L. Weizman, E. Nammer, O. Shilon, J. Frank, L. Joskowicz, and J. Sosna, “Vessels-Cut: A Graph based Approach to Patient-Specific Carotid Arteries Modeling,” in *Proc. Workshop on 3D Physiological Human (VH'09)*, ser. LNCS, vol. 5903, Zermatt, Switzerland, Nov. 2009, pp. 1–12.

- [173] N. Forkert, A. Schmidt-Richberg, D. Säring, T. Illies, J. Fiehler, and H. Handels, “Closing of Interrupted Vascular Segmentations: An Automatic Approach based on Shortest Paths and Level Sets,” in *Proc. SPIE Medical Imaging 2010: Image Processing*, vol. 7623. San Diego, CA/USA: SPIE Bellingham, WA/USA, Feb. 2010.
- [174] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (Third Edition)*. MIT Press, 2009.
- [175] J. Kleinberg and E. Tardos, *Algorithm Design*. MA/USA: Addison Wesley, 2005.
- [176] S. Skiena, *The Algorithm Design Manual*, 2nd ed. Springer-Verlag, 2008.
- [177] J. A. Sethian and A. Vladimirsky, “Fast Methods for the Eikonal and Related Hamilton-Jacobi Equations on Unstructured Meshes,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 11, pp. 5699–5703, 2000.
- [178] J. A. Sethian, “A Fast Marching Level Set Method for Monotonically Advancing Fronts,” *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.
- [179] S. Wörz and K. Rohr, “3D Adaptive Model-Based Segmentation of Human Vessels,” in *Proc. SPIE Medical Imaging 2007: Physiology, Function, and Structure from Medical Images*, vol. 6511. San Diego, CA/USA: SPIE Bellingham, WA/USA, Feb. 2007.
- [180] C.-K. Kang, S. Wörz, W. Liao, C.-A. Park, Y.-B. Kim, C.-W. Park, Y.-B. Lee, K. Rohr, and Z.-H. Cho, “Three Dimensional Model-Based Analysis of the Lenticulostriate Arteries and Identification of the Vessels Correlated to the Infarct Area: Preliminary Results,” *International Journal of Stroke*, vol. 7, pp. 558–563, 2012.
- [181] S. Seo, C.-K. Kang, S. Kim, D. Yoon, W. Liao, S. Wörz, K. Rohr, Y.-B. Kim, D. Na, and Z.-H. Cho, “Measurements of Lenticulostriate Arteries Using 7T MRI: New Imaging Markers for Subcortical Vascular Dementia,” *Journal of the Neurological Sciences*, vol. 322, pp. 200–205, 2012.
- [182] A. Hoover, V. Kouznetsova, and M. Goldbaum, “Locating Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response,” *IEEE Transactions on Medical Imaging*, vol. 19, no. 3, pp. 203–211, 2000.