

GEOMETRIC TRACKING AND GROUPING OF
COMPLEX VASCULATURE IN MEDICAL IMAGES

NICKY J. VAN DEN BERG

November 2025

COLOPHON

This document was typeset using the typographical `classicthesis` template for `LATEX` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available at:

<https://bitbucket.org/amiede/classicthesis/>

The work described in this thesis has received funding from the Dutch Foundation of Science NWO under the Talent Programme VICI 2020 Exact Sciences (Duits, Geometric Learning for Image Analysis, VI.C. 202-031).

A catalogue record is available from the Eindhoven University of Technology Library.

ISBN: 978-90-386-6543-6.

Nicky J. van den Berg: *Geometric Tracking and Grouping of Complex Vasculature in Medical Images*, © November 2025. All rights are reserved. No part of this publication may be reproduced by print, photocopy or any other means without permission of the copyright owner.

GEOMETRIC TRACKING AND GROUPING OF COMPLEX VASCULATURE IN MEDICAL IMAGES

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr. S.K. Lenaerts, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op vrijdag 28 november 2025 om 11:00 uur

door

NICKY JANNEKE VAN DEN BERG

geboren te Maastricht

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter: prof.dr.ir. B.F. van Dongen
promotor: dr.ir. R. Duits
copromotoren: dr. T.T.J.M. Berendschot (Universiteit Maastricht)
 prof.dr. O. Mula Hernández (University of Vienna)
leden: prof.dr.ir. C.I. Sánchez Gutiérrez (Universiteit van Amsterdam)
 prof.dr.rer.nat. J. Modersitzki (University of Lübeck)
 prof.dr.ir. B. Koren
adviseurs: prof.dr. G. Citti (Università di Bologna)
 dr. M. Veta

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

PUBLICATIONS

- [1] N. J. VAN DEN BERG, S. ZHANG, B. M. N. SMETS, T. T. J. M. BERENDSCHOT, and R. Duits. "Geodesic Tracking of Retinal Vascular Trees with Optical and TV-Flow Enhancement in $SE(2)$." In: *Scale Space and Variational Methods in Computer Vision*. Ed. by L. CALATRONI, M. DONATELLI, S. MORIGI, M. PRATO, and M. SANTACESARIA. Cham: Springer International Publishing, 2023, pp. 525–537.
- [2] N. J. VAN DEN BERG, O. MULA, L. Vis, and R. Duits. *Connected Components on Lie Groups and Applications to Multi-Orientation Image Analysis*. arXiv preprint arXiv:2409.18002 (in revision). 2024.
- [3] N. J. VAN DEN BERG, B. M. N. SMETS, G. PAI, J.-M. MIREBEAU, and R. Duits. "Geodesic Tracking via New Data-Driven Connections of Cartan Type for Vascular Tree Tracking." In: *JMIV* 66 (2024), pp. 198–230.
- [4] A. KOMPANETS, R. Duits, D. LEONETTI, N. J. VAN DEN BERG, and H. H. SNIJDER. "Segmentation Tool for Images of Cracks." In: *Advances in Information Technology in Civil and Building Engineering*. Ed. by S. SKATULLA and H. BEUSHAUSEN. Cham: Springer International Publishing, 2024, pp. 93–110.
- [5] N. J. VAN DEN BERG, F. M. SHERRY, T. T. J. M. BERENDSCHOT, and R. Duits. "Crossing-Preserving Geodesic Tracking on Spherical Images." In: *Scale Space and Variational Methods in Computer Vision*. Ed. by T. A. BUBBA, R. GABURRO, S. GAZZOLA, K. PAPAFITSOROS, M. PEREYRA, and C.-B. SCHÖNLIEB. Cham: Springer Nature Switzerland, 2025, pp. 192–204.
- [6] A. GARCÍA-CASTELLANOS, D. R. WESSELS, N. J. VAN DEN BERG, R. Duits, D. M. PELET, and E. J. BEKKERS. *Equivariant Eikonal Neural Networks: Grid-Free, Scalable Travel-Time Prediction on Homogeneous Spaces*. arXiv preprint arXiv:2505.16035 (accepted to NeurIPS). 2025.

CONTENTS

Introduction	1
I Theoretical Preliminaries	
1 Space of Planar Positions and Orientations \mathbb{M}_2	19
1.1 Definition of \mathbb{M}_2 and Corresponding Lie Group	19
1.2 Metric Tensor Fields and Finsler Functions	20
1.3 Lifting from \mathbb{R}^2 to \mathbb{M}_2	25
1.4 Geodesic Tracking	27
1.4.1 Symmetric Reeds-Shepp Car Model	29
1.4.2 Asymmetric Reeds-Shepp Car Model	29
1.4.3 Cost Function	30
2 Space of Spherical Positions and Orientations \mathbb{W}_2	35
2.1 Definition of \mathbb{W}_2 and Corresponding Lie Group	35
2.2 Metric Tensor Fields and Finsler Functions	37
2.3 Mapping from \mathbb{R}^2 to S^2	38
II Geodesic Tracking	
3 Introduction to Geodesic Tracking	45
3.1 Limitations and Their Solutions	46
3.1.1 Misalignment due to orientation sampling	46
3.1.2 Camera artifacts: uneven illumination and haze	49
3.1.3 Spherical Images	51
4 Geodesic Tracking via New Data-Driven Cartan Connections	53
4.1 Preliminaries on Cartan Connections and fast-marching	54
4.1.1 Cartan Connections	54
4.1.2 Anisotropic fast-marching	55
4.1.3 Flowchart and Overview of the Methodology	57
4.2 Data-Driven Metric & Data-Driven Cartan Connection	58
4.3 The New Geometric Tracking Model	62
4.3.1 Combine Optimally Straight and Short: A new Data-Driven Version ∇^U of the Cartan Connection	66
4.3.2 Main Theorems	67
4.3.3 Asymmetric Finsler Extension to Automatically Deal with Bifurcations	70
4.4 Numerical Solutions to the Eikonal PDE System	71
4.4.1 Asymmetric quadratic eikonal PDE	72

4.4.2	Discretization and consistency	74
4.4.3	Anisotropic Fast-Marching for Computing Distance Maps of Data-driven Left-invariant Finsler Models	76
4.4.4	Steepest Descent for the Finslerian Geodesics	77
4.5	Experiments	78
4.5.1	Curvature Adaptation	78
4.5.2	Complete Vascular Tree Tracking	79
4.5.3	Accuracy of the Model	84
4.6	Conclusion and Future Work	87
5	Vascular Tracking with Optical and TV-Flow Enhancement	89
5.1	Illumination Enhancement	90
5.2	TV-flow Enhancement	91
5.3	A Finsler Metric on \mathbb{M}_2 that Includes the Enhancements	91
5.4	Experimental Results	94
5.5	Conclusion	97
6	Crossing-Preserving Geodesic Tracking on Spherical Images	99
6.1	Crossing-Preserving Mapping from \mathbb{M}_2 to \mathbb{W}_2	99
6.2	Geodesic Tracking Model on \mathbb{W}_2	101
6.3	Eikonal PDE Solver	102
6.4	Experimental Results	104
6.5	Conclusion	105
III	Perceptual Grouping in Lie Groups	
7	Introduction - Perceptual Association Fields	109
7.1	Context and Goals	109
7.2	Fundamentals of Connected Component Algorithms	109
7.2.1	Topological Data Analysis	113
7.3	Connectivity on the Lie Group SE(2)	113
7.4	Our approach	114
7.5	Main Contributions	116
7.6	Structure of the Chapter	118
8	Connected Components on Lie Groups and Applications	119
8.1	Background on Lie Groups and Left-Invariant Distances	119
8.2	Algorithm to Find the δ -Connected Components	122
8.2.1	The Notion of δ -Connectedness	122
8.2.2	A General Algorithm for (Disjoint) Connected Components	124
8.2.3	How the Algorithm Works in an Example	126
8.3	Computing a Thickened Set A_ϵ with Hamilton-Jacobi Equations	126
8.4	Computing δ -Connected Components	131

8.4.1	Fixing $\alpha = 1$ in the δ -Connected Component Algorithm	131
8.4.2	δ -Connected Component Algorithm	132
8.4.3	Convergence Analysis	133
8.5	Choosing δ with Persistence Diagrams	136
8.6	Affinity Matrices between δ -Connected Components	139
8.7	Experiments	144
8.7.1	Experimental Set-Up	144
8.7.2	δ -Connected Components	147
8.7.3	Affinity Matrices	153
8.8	Conclusion and Future Work	155
iv	Geometric Learning for Artery-Vein Classification	
9	Artery-Vein Classification	159
9.1	Introduction	159
9.2	Related Work	161
9.3	Method	163
9.3.1	Dataset and Data augmentation	163
9.3.2	Network Architecture	165
9.3.3	Post-Processing	166
9.3.4	δ -Connected Components for Consistent Classification	166
9.3.5	Evaluation of the Method	167
9.4	Experiments	168
9.4.1	Synthetic Examples	168
9.4.2	Network with Limited Training Data	169
9.4.3	Usage of a Neural Network with Sufficient Training Data	172
9.4.4	Accuracy of the Vesselness Filter	173
9.4.5	Adapting the Model: Lifting Arteries and Veins Separately	176
9.4.6	Choosing the Metric Parameters of δ -Connected Components	179
9.4.7	Transferability of the Performance to other Datasets	182
9.5	Conclusion and Future Work	183
Discussion and Outlook		185
Bibliography		191
v	Appendix	
A	Proofs of Various Theorem's and Lemma's	213
A.1	Proof of Lemma 4.3.1	213

A.2	Proof of Theorem 4.3.1	214
A.3	Proof of Lemma 4.4.2	219
A.4	Multiple consecutive morphological dilations	221
A.5	Proof of Proposition 8.4.2	222
B	Data-Driven Left-Invariant Metric Tensor Field	225
C	Multi-scale Crossing-Preserving Cost Function	229
D	Adaptation to Asymmetric Data-Driven Finsler Functions	231
E	Logarithmic Norm Approximation	233
E.1	Logarithmic Norm Approximation in SE(2)	234
E.2	Logarithmic Norm Approximation in SO(3)	235
F	Reflectional Symmetries	237
G	For $\alpha > 1$ the equivalence relation breaks	239
H	Comparison to Existing Methods	241
H.1	Segment Anything Model 2	241
H.2	Topological Mode Analysis Tool (ToMATo)	242
I	Color Temperature	247
	Summary	249
	About the Author	251
	Acknowledgments	253

INTRODUCTION

Your eyes offer a window to your health.

Many aspects of a person's health are contained in the characteristics of their blood vessels. Based on these characteristics, doctors can diagnose several diseases, such as diabetes, hypertension, and Alzheimer's disease [60, 173, 197] (Fig. 1). Since the characteristics of the blood vessels are similar throughout the entire body, the blood vessels on the retina (in the eye) offer an easy, non-invasive way to get a snapshot of a patient's vascular system. These snapshots can be easily taken using a fundus camera (Fig. 1a). To objectively evaluate a person's health via retinal fundus images, we use image analysis techniques on those retinal images. We aim to develop methods that automatically quantify various features in a retinal image, such as the sinuosity of the arteries and veins [174]. These measures will allow us to monitor an individual's health closely over time, and to take appropriate measures when necessary.

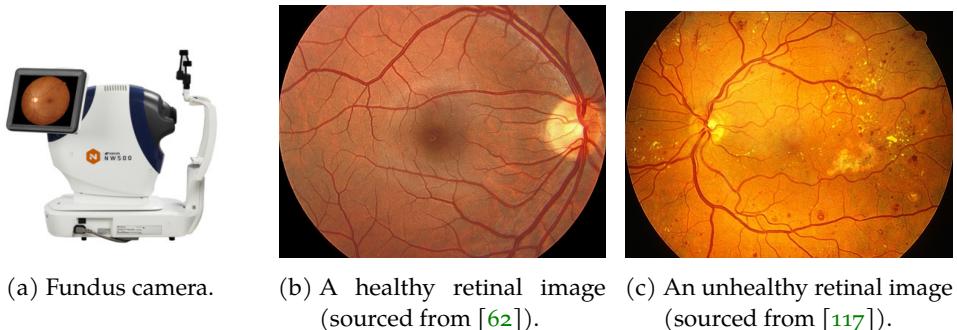
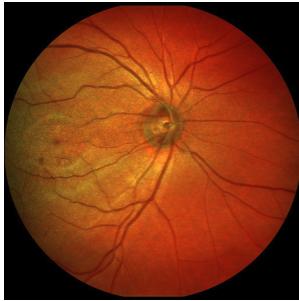


Figure 1: Retinal pictures are taken with a fundus camera (Fig. 1a). There are clear differences between the pictures taken from a healthy individual (Fig. 1b) and a diabetic patient (Fig. 1c). The vessels of diabetic patients typically have a high sinuosity and are also more likely to leak, resulting in hemorrhaging and microaneurysms [193]. Eventually, this can lead to blindness.

IMAGE ANALYSIS METHODS FOR RETINAL IMAGE EVALUATION

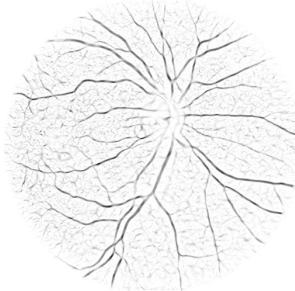
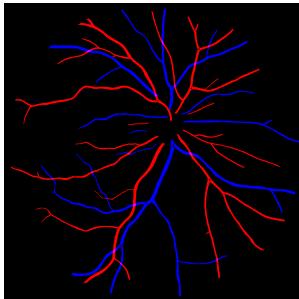


Over the past decades, different methods for image analysis have been developed. We will give a concise overview of some of the most relevant approaches for monitoring changes in retinal images, categorized into the following three tasks: segmentation of the vasculature, identification of connected components, and tracking complex vascular trees.

Segmentation

The first task is to locate blood vessels on an image of the retina. This is an example of a technique called *segmentation*, in which each pixel is assigned to a class. The assignment can be continuous (fuzzy) or discrete (non-fuzzy) [151]. In the case of vessel segmentation, there are two classes: vessel and non-vessel.

The Frangi multi-scale vesselness enhancement filter [98] is an example of fuzzy segmentation. The method outputs a grayscale image with low values in the vessel structures and high values in other locations. It can be interpreted as the probability that the pixel belongs to the vasculature. A binary non-fuzzy segmentation, in contrast, would assign either a 0 or 1 depending on whether the pixel is considered vessel or background, respectively.



Besides segmenting images to locate blood vessels, one can also use segmentation to *distinguish between arteries and veins* (and background). In ophthalmology¹, it is well known that these vessels have different characteristics and should therefore be treated differently. Hence, it is important to develop models that correctly classify vascular structures as either an artery or a vein. Currently, the approaches that have

been suggested rely mainly on neural networks. A selection is discussed in the review [147]. However, the proposed solutions often require post-processing to compensate for a fragmented classification, where within a vascular tree, the classification switches numerous times between artery and vein [52].

¹ Dutch: "oogheelkunde".

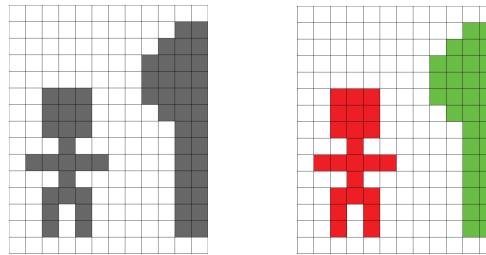
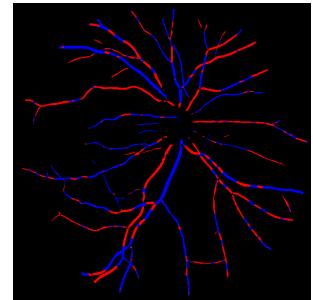


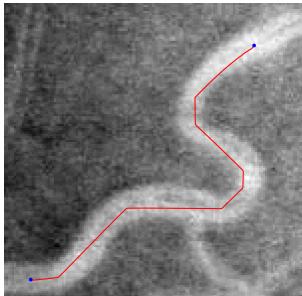
Figure 2: Connected components can be used to differentiate between different structures. Left: reference image, right: identified connected components (person and tree).

Identification of Connected Components

To avoid fragmented artery-vein classification, one can include knowledge of vascular trees. Once we have a (binary) segmentation of a retinal image, we are interested in identifying full vascular trees. The vascular trees are generally connected, so we can consider working with *connected components*. Connected component algorithms are grouping algorithms that identify parts of a set or image that belong to the same group, such as the person and the tree in Fig. 2. Algorithms often rely on mathematical morphology, which we discuss in Chapter 8.



Geodesic Tracking



Once we have an artery-vein classification of the retinal image, we aim to assess the characteristics of the vascular structures automatically. An example of such a characteristic is the sinuosity of the blood vessel, as this is an important indicator of diseases such as diabetes [174]. Sinuosity can be extracted from the centerline of the blood vessels. This centerline can be approximated by *geodesic tracking models*. These models calculate geodesics, i.e., shortest (optimal) paths between two points. The many different geometric models (Riemannian, sub-Riemannian, Finslerian, sub-Finslerian) dictate the behavior of the geodesics and are designed in such a way that the corresponding optimal paths follow the vasculature in the image. An overview of literature will follow in Chapter 3.

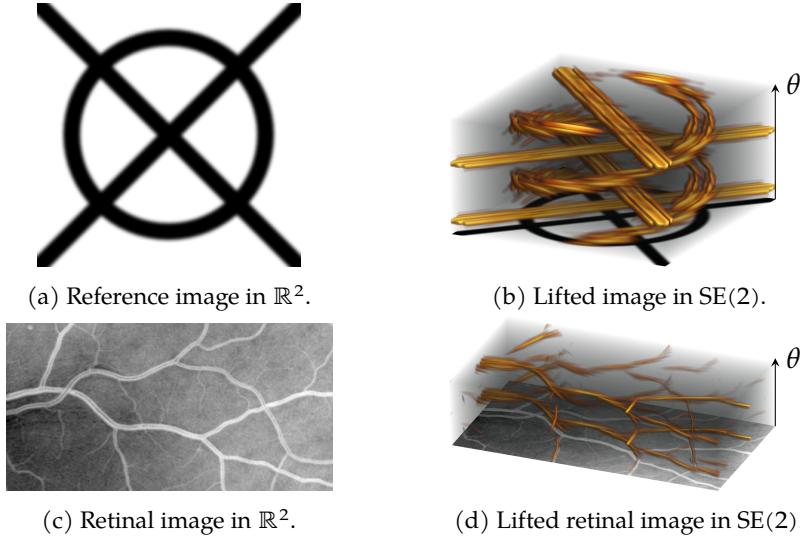


Figure 3: Image processing on the \mathbb{R}^2 image can lead to issues at crossing structures. These are disentangled when the image is lifted to $\text{SE}(2)$.

Generic Approach to Deal with Complex Structures: Lifting the Image

However, image processing tasks such as tracking in a two-dimensional image can lead to issues in distinguishing between different crossing structures. A possible solution is to create a higher-dimensional image in which these structures are automatically disentangled, cf. Fig. 3. In this lifted image, the crossing structures are disentangled based on their local orientation [77, 78]. Here, one can again perform (continuous) image segmentation [108] and tracking [17, 19]. The mathematical details on how the lifting is done will follow in Section 1.3.

THIS THESIS

In this thesis, we focus on the development of methods contributing to a fully automatic evaluation of retinal images. Our approach consists of three different objectives:

1. Tracking of complex vascular trees in retinal images;
2. Identifying connected components on non-fuzzy segmentations of the vasculature;
3. Classifying vessels as arteries or veins.

Next, we detail our contributions to each of these objectives.

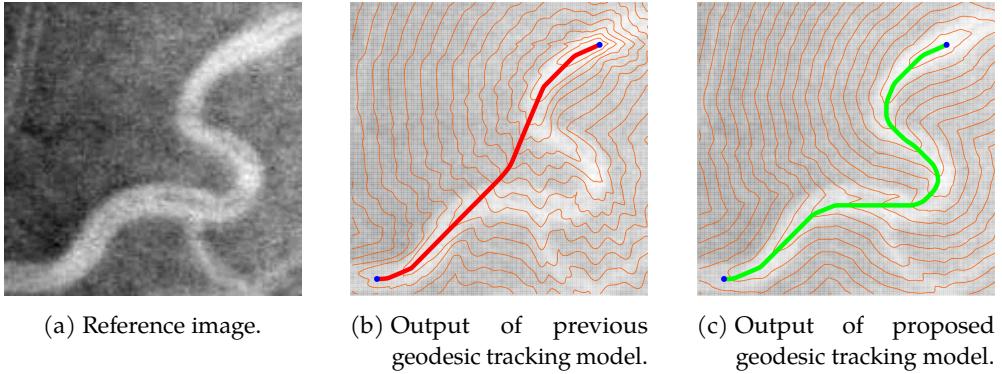


Figure 4: Tracking results of the previous and proposed model on a sinuous vessel.
More information about the underlying techniques and mathematics of the new tracking model follow in Chapter 4.

1. Tracking of complex vasculature in retinal images

First, we focused on developing tracking algorithms that automatically follow the vasculature (i.e., vessel structure) in a retinal image. Here, we took three approaches to improve various aspects of existing tracking algorithms.

The first issue we tackle deals with very sinuous structures as in Fig. 4a. Existing tracking methods tend to find them very challenging and often struggle to follow them correctly while staying in the vasculature, as visualized in Fig. 4b. Therefore, we propose a new model with more knowledge of the (direction of) local vascular structures by including more of the underlying image data. By design, movement along vascular structures is encouraged and therefore performs better to track highly curved, sinuous structures Fig. 4c. Furthermore, we show that the method can be used to track a complete vascular tree in one single run of the tracking algorithm.

The second issue relates to the influence of the underlying image. Retinal images are typically of varying quality, like the fact that the lighting is often uneven, as in Fig. 5a, making it challenging to differentiate between different structures. We apply image enhancement methods to compensate for a) uneven illumination [206] in Fig. 5b and b) noise while preserving crossing structures [185] in Fig. 5c. Then, we compare the tracking results on the unprocessed image with the illumination enhanced images without and with crossing-preserving denoising (Figs. 5d to 5f respectively). We conclude that especially the illumination enhancement significantly improves tracking results, and the crossing-preserving denoising step slightly improves the results. Additionally, we notice that prior knowledge of the connectivity of start and

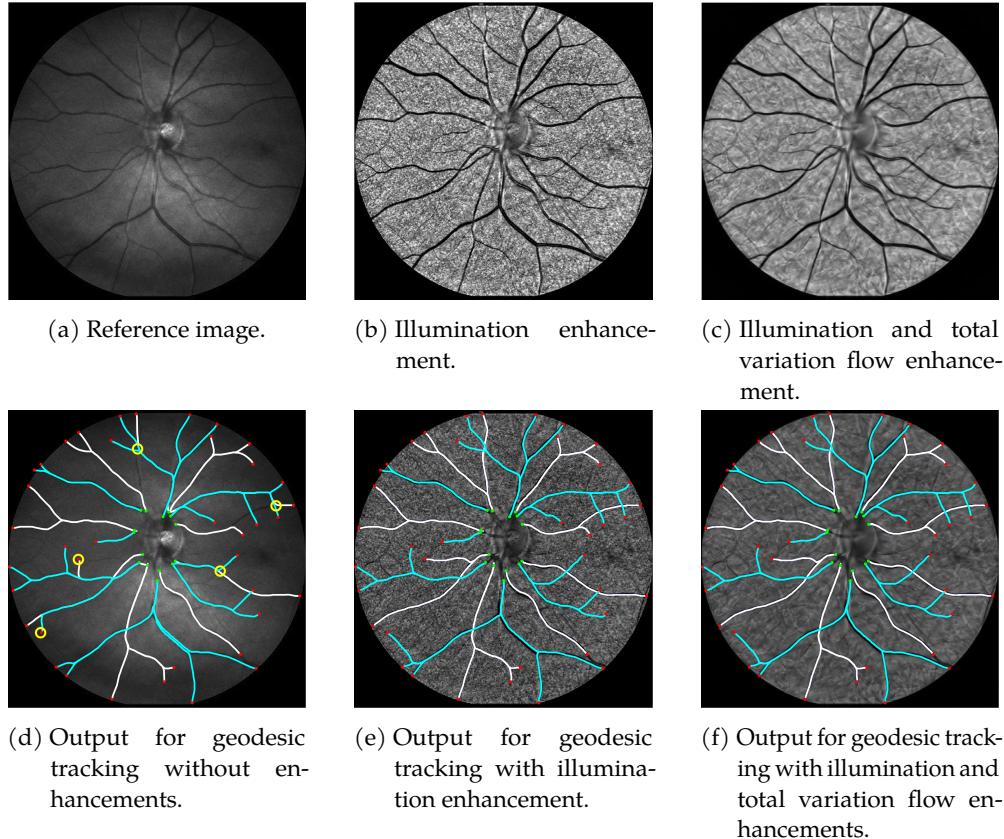
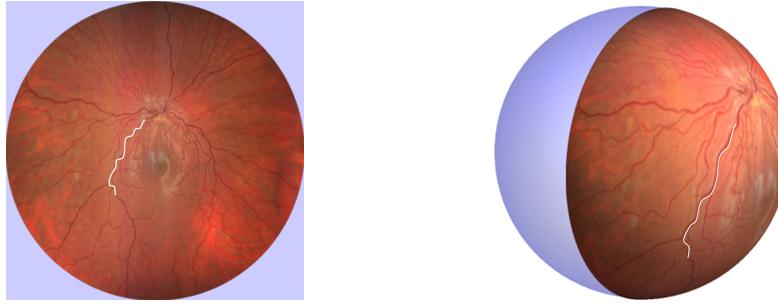


Figure 5: Tracking results on the same picture enhanced different preprocessing techniques (illumination and TV-flow enhancement). More information about the underlying techniques and mathematics of the new tracking model follow in Chapter 5.



(a) Tracking on the flat image.

(b) Tracking on the spherical image.

Figure 6: Tracking on different manifolds (flat image vs. spherical image). More information about the underlying techniques and mathematics of the new tracking model follow in Chapter 6.

end points, or knowledge of the type of vessel they are located on (artery or vein), significantly improves the tracking results as well.

Lastly, we usually treat the retinal image as a flat surface when we track the vasculature (Fig. 6a). However, this does not reflect the true geometry of the retina in the eye, which is spherical (Fig. 6b). Therefore, we extend the existing methods to a lifted space of the sphere, as done by [135]. Being able to perform tracking on the sphere allows for considering the deformations arising from the projection to flat images. Here, we extend the existing theory by developing a *crossing-preserving* tracking model on the spherical image. Additionally, we develop an asymmetric, crossing-preserving tracking model on the sphere, which avoids the occurrence of cusps.

2. Identifying connected components on non-fuzzy segmentations of the vasculature

Besides tracking, we focus on the identification of connected components. Usually, connected component algorithms cannot differentiate between different crossing structures, as in the example in Fig. 7b. Here, all crossing lines are identified to belong to the same connected component. However, this is not always desirable: crossing vessels in an image of the vasculature should not be classified as the same vessel structure.

Therefore, we extend the existing theory on connected components to Lie groups. We develop a framework that can a) differentiate between different crossing components, and b) group well-aligned components leveraging the Lie group structure. We call the connected components δ -connected components, as they allow for gaps up to distance δ to occur within one component.

We analyze our method by 1) proving that the provided algorithm always converges to the correct δ -connected component in a finite number of steps and

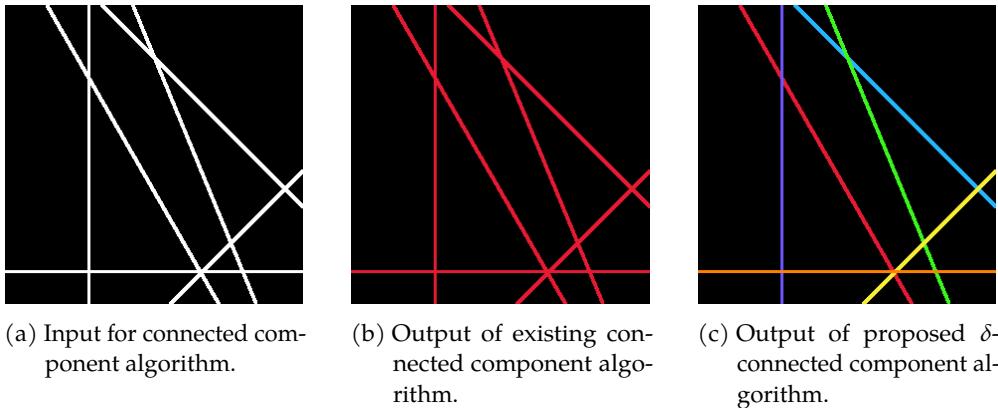


Figure 7: Visualization of the output of existing and the proposed connected component algorithms. More information about the underlying techniques and mathematics of δ -connected components follow in Part III.

2) show the reflectional symmetries of our algorithm and how they generalize to other Lie groups. We propose to use specific affinity matrices, allowing us to measure alignment (both spatial and in orientation) between different δ -connected components. Lastly, we briefly mention the idea of *persistence homology* of δ -connected components, allowing us to analyze the stability of all components, depending on the choice of δ .

The introduced δ -connected component algorithm allows us to identify vascular trees in the segmentation of an image (Fig. 8c).

3. Classifying vessels as arteries or veins

Finally, we explore combining neural networks for the classification of arteries and veins with geometrical tools. Experience shows that it is challenging to get a consistent classification along vascular structures using only neural networks (Fig. 9b). Therefore, we develop a framework that combines existing neural network techniques with a geometrical technique in this thesis: δ -connected components, resulting in a classification that is consistent along a vascular tree (Fig. 9c).

Our framework inputs a retinal image on which we apply a neural network segmenting the pixels of the input image into 4 categories: 1) background, 2) artery, but not a vein, 3) vein, but not an artery, and 4) artery and vein. Then, we use the δ -connected components to improve the classification results and check their correctness.

The presented preliminary research shows that combining the neural network output with the δ -connected components leads to promising and more

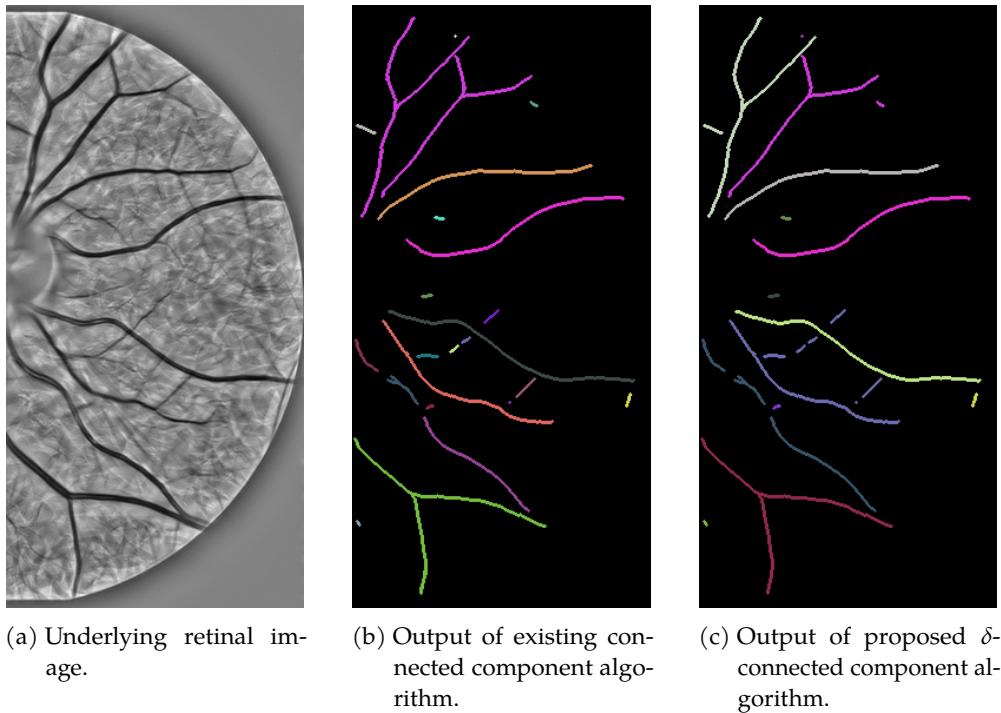


Figure 8: Visualization of our contribution to connected component algorithms applied to a retinal image. More information about the underlying techniques and mathematics of δ -connected components follow in Part III.

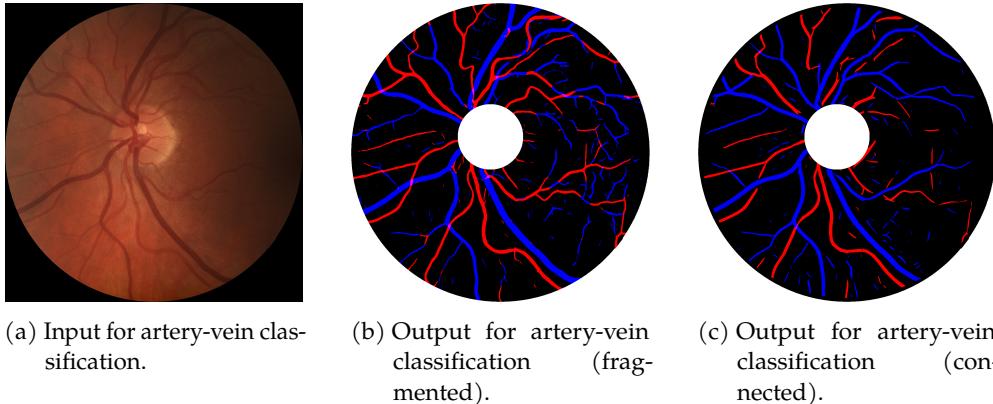


Figure 9: Visualization of the contribution to each artery-vein classification, discussed in this thesis. More information about the underlying techniques and mathematics follow in Part IV.

consistent classification results, still leaving room for future improvements. Additionally, we dive into the accuracy of the vesselness and lifting binary data. Lastly, we consider the effect of data augmentation on test data belonging to the training dataset, and a completely independent dataset.

Outline

This thesis consists of four parts:

- In Part I, we cover the necessary theoretical preliminaries.
 - In Chapters 1 and 2, we discuss the preliminaries on the space of *planar* positions and orientations \mathbb{M}_2 and on the space of *spherical* positions and orientations \mathbb{W}_2 respectively.
- In Part II, we cover the research on geodesic tracking.
 - Chapter 3 gives a global introduction to the different chapters in Part II.
 - Chapter 4 covers the data-driven left-invariant tracking model via Cartan connections and is based on [30].
 - Chapter 5 covers work on the influence of image enhancements and is based on [31].
 - Chapter 6 covers the extension of crossing-preserving tracking from the Lie group $SE(2)$ to another Lie group $SO(3)$ to deal with the fact that the retina is a sphere (and not a plane). This chapter is based on [29].

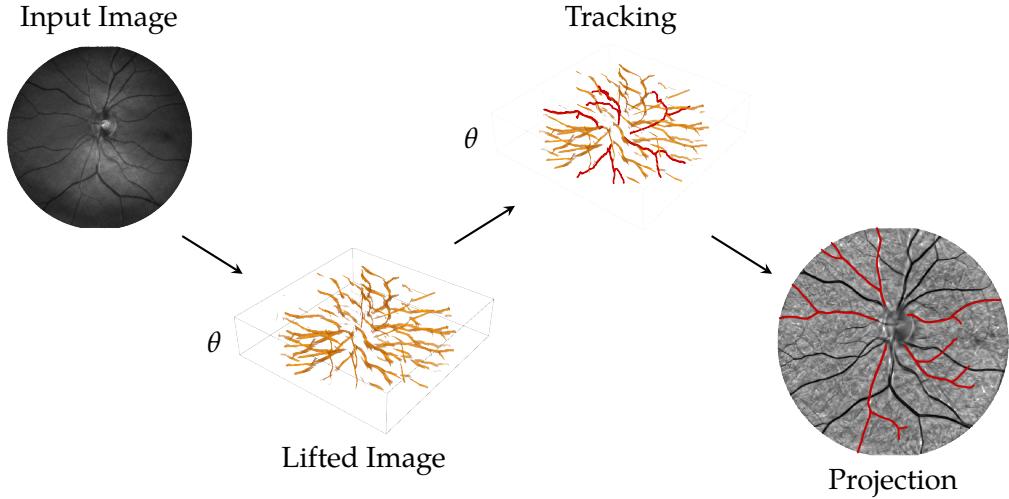


Figure 10: Visualization of the main steps in performing the different geodesic tracking models. First, the image is lifted to a higher-dimensional space ($SE(2)$), in which the geodesic tracking model is applied. Afterward, the result is projected back onto the input image.

- In Part III, we cover the research on crossing-preserving ($\delta-$)connected component algorithms on Lie groups with applications to the cases $SE(2)$ and $SO(3)$. Chapter 8 is based on [28].
- In Part IV, we cover our research towards artery-vein classification using the techniques introduced in Part III, i.e., the δ -connected components.

List of Technical Contributions

1) Tracking of complex vascular trees in retinal images: We have made three new geodesic tracking methods to track *complex full vascular trees* in retinal images. Each method first lifts the input image to a higher-dimensional space, on which we apply our tracking algorithm. All obtained vessel tracks are projected back onto the original image. The process is visualized in Fig. 10.

The first geodesic tracking model deals with very sinuous blood vessels and includes the *underlying image data in the structure of the model* (in [30]):

- i. We introduce a new geodesic tracking model that uses a crossing-preserving approach for tracking complex vasculature in the lifted space of planar positions and orientations. Our model uses a new anisotropic fast-marching algorithm to compute cusp-free data-driven geodesics. The induced geometric vessel tracking better adapts for vessel curvature

and orientation sampling biases, compared to the previous model in [83], as shown in Figs. 4a to 4c.

- ii. We mathematically analyze these solutions (the family of all geodesics) via our data-driven version of the plus Cartan connection that underlies the Hamiltonian flow.
- iii. We demonstrate our method on highly challenging examples of retinal images with complex vasculature. We obtain adequate tracking results with only two runs of the proposed anisotropic fast-marching algorithm.

Secondly, we develop new *crossing-preserving enhancement methods* to deal with uneven illumination, low contrast, and noise as a consequence of the fundus camera (in [31]):

- i. We develop a new asymmetric, data-driven left-invariant Finsler geometric model that includes contextual contrast enhancement via TV-flows on SE(2),
- ii. We conduct experiments that show that application of this new Finsler geometric model reduces many tracking errors compared to previous left-invariant models [18, 83] and the recent data-driven model [30] (cf. Fig. 5),
- iii. We conclude that the new model performs very well on realistic, unevenly illuminated retinal images and allows full vasculature computations from a *single* distance map. The *inclusion* of the optical and TV-flow enhancements in the Finsler function no longer requires a 2-step approach as in [30]. A single run of building the distance map suffices with the techniques presented in this work.

Thirdly, we develop new geodesic tracking methods for *spherical images*, such as (wide-field) retinal images (in [29], cf. Figs. 6a and 6b):

- i. We create a *crossing-preserving* tracking algorithm on the sphere S^2 using a cost function defined on the space of spherical positions and orientations \mathbb{W}_2 . We show examples where our new geodesic tracking method no longer misinterprets crossings for bifurcations, taking the wrong exit at crossings, due to our crossing-preserving cost function, which is new compared to existing work [135].
- ii. We include a forward gear constraint to get *cusp-free* spherically projected \mathbb{W}_2 -geodesics. We provide a new spherical version of the forward gear Reeds-Shepp car [83].

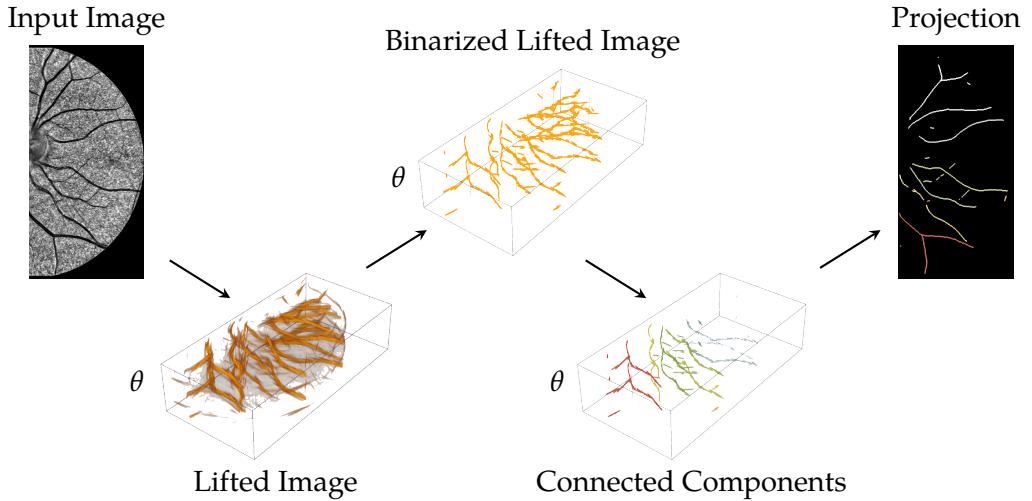


Figure 11: Visualization of the main steps in performing the δ -connected component algorithm. First, the image is lifted to a higher-dimensional space ($SE(2)$), in which the connected component algorithm is applied. Afterward, the result is projected back onto the input space.

- iii. We apply the method to wide-field images instead of standard images. They cover 120° of the eyeball with deformations up to 52.2 % when projecting from the spherical image to the planar image. Note that regular optical images cover 72° of the eye and consequently deal with smaller deformations (up to 22.2 %), cf. Fig. 6b.
- iv. We compute the distance maps with fast, simple, accurate GPU-code using the PDE-approach in [19] as an alternative to anisotropic fast-marching [137].

2) *Identifying connected components on non-fuzzy segmentations of vasculature:* We develop a new connected component algorithm on any Lie group G able to differentiate between crossing structures (in [28]). The methodology is developed for general Lie groups with the primary use cases $G = SE(2)$, and $G = SO(3)$ to identify connected components in orientation-based lifts of flat and spherical images respectively.

- i. We develop a new method to find the connected components of a binary function defined on a Lie group G equipped with a left-invariant (sub)-Riemannian metric. The method is roto-translation equivariant and relies on standard tools from topological data analysis (e.g., Čech complexes of radius $\delta/2$ [12, 33, 46, 67, 85, 149, 187] are closely related to our δ -connected components). We compute them via morphological PDEs on

Lie groups. Here, we employ efficient left-invariant solvers, relying on iterative morphological group convolutions with analytic PDE kernels. Such morphological group convolutions are parallelizable and employ the group structure on the Lie group G .

- ii. We mathematically analyze our method:
 - a) We prove that the provided algorithm always converges to the correct δ -connected component in a finite number of steps in Theorem 8.4.1.
 - b) We show reflectional symmetries of our connected component algorithm (Corollary F.1 in Appendix F) and how this generalizes to other Lie groups G of dimension $n < \infty$ (Lemma F.1). This fundamental property is due to the invariance of both the Riemannian distance and its logarithmic norm approximation under the 2^n reflectional symmetries in the Lie algebra. For $G = \text{SE}(2)$ and $G = \text{SO}(3)$ inclusion of these symmetries (cf. Fig. F.1) is desirable in the connected component algorithm.
- iii. Along with this method we publish code (in Mathematica and Python [27]), for the morphological convolutions and the connected component algorithm.
- iv. The iterative morphological convolutions are parallelizable, fast and flexible, thanks to intuitively parameterized analytic kernels on the Lie group G . They do not require more expensive state-of-the-art anisotropic fast-marching schemes [139, 140] for computing the Riemannian distance maps.
- v. We show how our method combined with (variants of) standard methodology from topological data analysis (affinity matrices [1, 3], persistence homology based clustering [46, 183]), is very beneficial in multi-orientation image analysis of complex vascular trees in retinal imaging. We present experiments of (improved) grouping and segmentation of blood vessels.

- 3) *Classifying vessels as arteries or veins:* We develop a new artery-vein classification method that automatically identifies all arteries and veins in a retinal image. This problem is addressed in Chapter 9. We follow the steps in Fig. 12. In this last part, we focus on:

- i. We designed and tested a model to make the artery-vein classification consistent on a network trained with a limited amount of data, where the results improved. Unfortunately, these results did not transfer to properly trained neural networks.

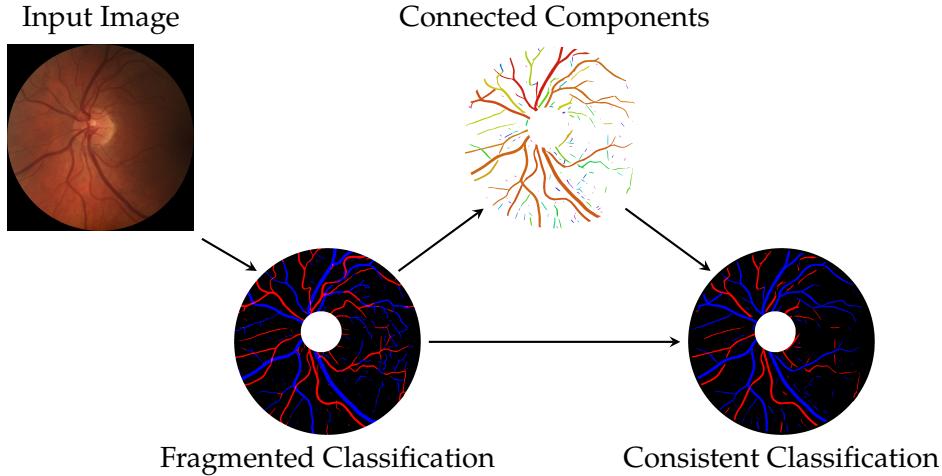


Figure 12: Visualization of the main steps in performing the artery-vein classification. First, a neural network makes a prediction of the location of arteries and veins. This output is fragmented, therefore we use our δ -connected component algorithm to make the classification consistent along vessels.

- ii. We investigated the precision of the segmentation by the vesselness filter [30, App. C] and compared it to the precision of the segmentation of a U-net.
- iii. We investigated the precision of various ways to lift a binary image before applying the δ -connected component algorithm (to separate crossing line structures). We determined the best way to preprocess the image to get the most accurate results.
- iv. We made variations of our model to improve the precision of the artery-vein classification. Their success varied: removing more than only the optic disk was not useful, but lifting the artery and vein output of the network separately before applying our method showed promising results.
- v. We investigated the transferability of the results of our neural network to other datasets of retinal images. We compared the dice scores of three differently trained U-nets (training dataset was augmented using different techniques) on two test datasets: one belonging to the training dataset and one independent dataset.

Part I

THEORETICAL PRELIMINARIES

1

SPACE OF PLANAR POSITIONS AND ORIENTATIONS \mathbb{M}_2

In the introduction, we briefly mentioned applying image analysis techniques to the two-dimensional image does not always result in the desired outcomes: algorithms often have trouble distinguishing between different crossing structures. To avoid these difficulties, it is beneficial to do image processing in a higher-dimensional representation of the input image. Instead of only using positional information, one can include information about local directions in the image, as we do in this thesis.

In this chapter, we first explain what the space of positions and orientations \mathbb{M}_2 is. Next, we introduce a frame which we will use in this space. After that, we introduce the metric tensor fields and Finsler functions on \mathbb{M}_2 . Finally, we detail how we lift the data.

1.1 DEFINITION OF \mathbb{M}_2 AND CORRESPONDING LIE GROUP

From a two-dimensional image, we aim to create a three-dimensional image representation. Before explaining how we do that, we have to clarify in what space this three-dimensional representation of the image will live. Intuitively, we assign to every spatial position $x \in \mathbb{R}^2$ in the image an orientation vector $\mathbf{n}_\theta = (\cos \theta, \sin \theta) \in S^1$ for all $\theta \in \mathbb{R}^2 / (2\pi\mathbb{Z})$. We call this three-dimensional space the space of planar positions and orientations. The formal definition follows next:

Definition 1.1.1 (Space of planar positions and orientations). *The space of planar positions and orientations \mathbb{M}_2 is defined as the smooth manifold*

$$\mathbb{M}_2 := \mathbb{R}^2 \times S^1, \tag{1.1}$$

where $S^1 \cong \mathbb{R}/(2\pi\mathbb{Z})$ using the identification

$$(\cos \theta, \sin \theta) \longleftrightarrow \theta.$$

To move from one position in the space of planar positions and orientations to another, we have to use a group action that changes the position and orientation appropriately. For this, we use a roto-translation from the special Euclidean

group $\text{SE}(2) := \mathbb{R}^2 \rtimes \text{SO}(2)$. Applying the roto-translation $g = (\mathbf{b}, R_\phi) \in \text{SE}(2)$ to the point $\mathbf{p} = (\mathbf{x}, \mathbf{n}_\theta) \in \mathbb{M}_2$, results in the (left) group action

$$L_g(\mathbf{p}) = L_{(\mathbf{b}, R_\phi)}(\mathbf{x}, \mathbf{n}_\theta) := (\mathbf{b} + R_\phi \mathbf{x}, R_\phi \mathbf{n}_\theta) = (\mathbf{b} + R_\phi \mathbf{x}, \mathbf{n}_{\theta+\phi}), \quad (1.2)$$

with $R_\phi \in \text{SO}(2)$ a counter-clockwise planar rotation over angle ϕ . We also denote the left group action by $g \cdot \mathbf{p}$ and gp .

Note that we can identify any orientation matrix $R_\theta \in \text{SO}(2)$ with just its orientation $\theta \in \mathbb{R}/(2\pi\mathbb{Z})$, i.e.,

$$(\cos \theta, \sin \theta) \longleftrightarrow \theta \longleftrightarrow R_\theta \in \text{SO}(2). \quad (1.3)$$

Hence, points in \mathbb{M}_2 or group elements of $\text{SE}(2)$ can be parametrized as

$$(\mathbf{x}, (\cos \theta, \sin \theta)) = (\mathbf{x}, \mathbf{n}_\theta), (\mathbf{x}, \theta), \text{ or } (\mathbf{x}, R_\theta) \text{ for } \mathbf{x} \in \mathbb{R}^2, \theta \in \mathbb{R}/(2\pi\mathbb{Z}).$$

This allows us to describe any point in \mathbb{M}_2 and any group element of $\text{SE}(2)$ by a position $\mathbf{x} \in \mathbb{R}^2$ and an orientation $\theta \in \mathbb{R}/(2\pi\mathbb{Z})$ that can be written down in different ways: as a unit vector \mathbf{n}_θ , a rotation (matrix) R_θ , or an angle θ depending on what is convenient.

The Lie group $\text{SE}(2)$ and the space \mathbb{M}_2 can be identified with each other in the specific case where we chose a reference point $\mathbf{p}_0 := ((0, 0), (1, 0)) \in \mathbb{M}_2$. Then,

$$\mathbb{M}_2 \cong \text{SE}(2)/\text{Stab}_{\text{SE}(2)}(\mathbf{p}_0) \cong \text{SE}(2)/\{(\mathbf{0}, I)\} \cong \text{SE}(2). \quad (1.4)$$

We say that \mathbb{M}_2 is the principal homogeneous space of $\text{SE}(2)$.

1.2 METRIC TENSOR FIELDS AND FINSLER FUNCTIONS

We have just introduced the space of planar positions and orientations in which we will do image processing. The image processing applications of our focus in this thesis rely on the notion of distances. To calculate distances, we need to introduce the metric tensor fields, which describe how distances are calculated, and the frames on which these metric tensor fields build.

Intuition Behind Metric Tensor Fields and Example

Metric tensor fields offer a way to describe distances and lengths. They dictate in which directions it is cheap and expensive to move. We will illustrate this with an example in \mathbb{R}^2 .

Let us consider a point $\mathbf{x} \in \mathbb{R}^2$ where we want to know the length of the tangent vector $\dot{\mathbf{x}} \in T\mathbb{R}^2$, for example such as visualized in Fig. 1.1a. We can decompose this vector relative to a frame ∂_1, ∂_2 . Often, we use

the standard frame $\partial_x = (1, 0)$, $\partial_y = (0, 1)$, as in the example in Fig. 1.1b, but this can be any pair of tangent vectors that are linearly independent (cf. Fig. 1.1d). Then, the vector $\dot{\mathbf{x}}$ can be decomposed as $\dot{\mathbf{x}} = \dot{x}^1 \partial_1 + \dot{x}^2 \partial_2$ with $\dot{x}^1, \dot{x}^2 \in \mathbb{R}$.

The decomposition of the vector relative to the frame allows us to calculate the distance. The definition of the distance relies on the dual frame d^1, d^2 . The dual frame maps vectors to their coordinates relative to a specific frame. More specifically, the dual frame satisfies $d^i(\partial_j) = \delta_j^i$. By definition, the distance in \mathbb{R}^2 is given by

$$\|\dot{\mathbf{x}}\| = \sqrt{\sum_{i,j=1}^2 g_{ij} d^i(\dot{\mathbf{x}}) d^j(\dot{\mathbf{x}})}, \quad (1.5)$$

where $g_{ij} : \mathbb{R}^2 \rightarrow \mathbb{R}$ so that $[g_{ij}]_{ij}$ is positive definite. For the standard, unweighted (Euclidean) distance, we use $g_{ij} = \delta_{ij}$, resulting in

$$\|\dot{\mathbf{x}}\| = \sqrt{\sum_{i=1}^2 d^i(\dot{\mathbf{x}}) d^i(\dot{\mathbf{x}})} = \sqrt{(x^1)^2 + (x^2)^2}. \quad (1.6)$$

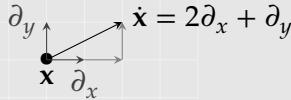
In Figs. 1.1a to 1.1d, we show that, depending on the chosen frame and weights g_{ij} , the length of the same vector differs. All figures show the point \mathbf{x} with tangent vector $\dot{\mathbf{x}}$, but use different frames to decompose the vector $\dot{\mathbf{x}}$. The isocontours for distances 1, 2, and 3 (and distance $\|\dot{\mathbf{x}}\|$) are plotted in gray dashed (black solid) lines.

First, we consider the standard frame ∂_x, ∂_y in Fig. 1.1b. Here, the vector $\dot{\mathbf{x}}$ can be expressed as $\dot{\mathbf{x}} = 2\partial_x + \partial_y$, which tells us, using Eq. (1.6), that it has unweighted distance $\sqrt{5}$.

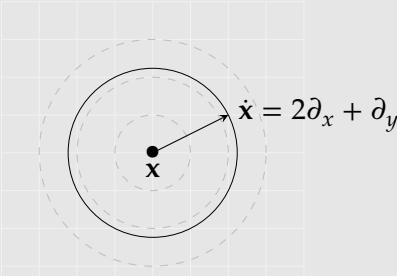
Alternatively, one could make movement in the ∂_y direction twice as expensive, i.e., $2\partial_2 = \partial_y$ (cf. Fig. 1.1c). In that case, the unweighted distance relative to the new frame ∂_1, ∂_2 of $\|\dot{\mathbf{x}}\|$ equals $\sqrt{8}$. Note that we would have obtained the same result in Fig. 1.1b using distance weights $g_{11} = 1, g_{22} = 4, g_{12} = g_{21} = 0$.

Lastly, we observe that the frame does not have to be orthogonal, as shown in Fig. 1.1d. We would have found the same distance metric when using weights $g_{11} = 1, g_{22} = 2, g_{12} = g_{21} = -1$.

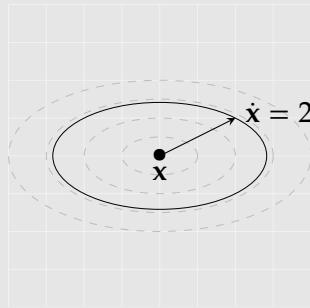
Hence, distances are defined relative to a frame, dictating the costs of moving in specific directions.



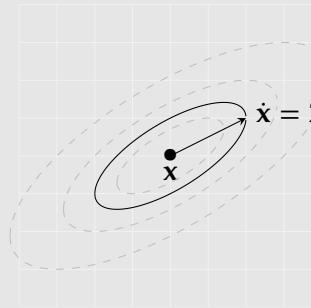
(a) Point $x \in \mathbb{R}^2$ and a tangent vector $\dot{x} \in T\mathbb{R}^2$.



(b) Standard distance of the tangent vector \dot{x} with respect to the standard frame, i.e., $\|\dot{x}\| = \sqrt{dx(\dot{x})^2 + dy(\dot{x})^2} = \sqrt{(2)^2 + (1)^2} = \sqrt{5}$.



(c) Weighted distance of the tangent vector \dot{x} with respect to the standard frame with weights $g_{11} = 1, g_{22} = 4$ and $g_{12} = g_{21} = 0$, i.e., $\|\dot{x}\| = \sqrt{dx(\dot{x})^2 + 4dy(\dot{x})^2} = \sqrt{8}$.



(d) Weighted distance of the tangent vector \dot{x} with respect to the standard frame with weights $g_{11} = 1, g_{22} = 2$ and $g_{12} = g_{21} = -1$, i.e., $\|\dot{x}\| = \sqrt{dx(\dot{x})^2 - 2dx(\dot{x})dy(\dot{y}) + 2dy(\dot{x})^2} = \sqrt{2}$.

Figure 1.1: Examples of different weighted and unweighted distances calculated with respect to different metric tensor fields.

First, we introduce the static frame denoted by $\{\partial_x, \partial_y, \partial_\theta\} \subset \Gamma(T\mathbb{M}_2)$, where $\Gamma(\cdot)$ denotes the smooth section of a given vector bundle, induced by the coordinates x, y, θ for all points in \mathbb{M}_2 . Its dual frame is denoted by $\{dx, dy, d\theta\} \subset \Gamma(T^*\mathbb{M}_2)$, and can be used to express the metric tensor field \mathcal{G} .

For our applications, it is advantageous to use left-invariant vector fields. Left-invariant vector fields on \mathbb{M}_2 guarantee that distances remain equivariant under the group of roto-translations. More specifically, that means that tracking and grouping of structures (processing) is independent of the roto-translation of the

image: processing on a roto-translation of the image is identical to processing on the original image and roto-translating the result.

Definition 1.2.1 (Frame of left-invariant vector fields). *The frame of left-invariant vector fields (left-invariant frame) is obtained by a pushforward of the static frame at the origin p_0 . We define the pushforward $(L_g)_* : T_p(\mathbb{M}_2) \rightarrow T_{gp}(\mathbb{M}_2)$ by*

$$(L_g)_* \partial_x|_p U = \partial_x|_p (U \circ L_g),$$

for all smooth functions $U : \mathbb{M}_2 \rightarrow \mathbb{C}$ and the left group action L_g defined in Eq. (1.2). Then, the left-invariant frame $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ is defined by

$$\mathcal{A}_i|_{(x,y,\theta)} = (L_{(x,y,\theta)})_* \mathcal{A}_i|_{p_0}.$$

After computations, we obtain the left-invariant vector fields

$$\mathcal{A}_1|_{(x,y,\theta)} = \cos \theta \partial_x|_{(x,y,\theta)} + \sin \theta \partial_y|_{(x,y,\theta)}, \quad (1.7a)$$

$$\mathcal{A}_2|_{(x,y,\theta)} = -\sin \theta \partial_x|_{(x,y,\theta)} + \cos \theta \partial_y|_{(x,y,\theta)}, \quad (1.7b)$$

$$\mathcal{A}_3|_{(x,y,\theta)} = \partial_\theta|_{(x,y,\theta)}. \quad (1.7c)$$

The corresponding dual frame is given by $\{\omega^i\}_{i=1}^3$ where $\omega^i(\mathcal{A}_j) = \delta_j^i$. A brief computation gives

$$\omega^1 = \cos \theta dx + \sin \theta dy, \quad \omega^2 = -\sin \theta dx + \cos \theta dy \text{ and } \omega^3 = d\theta. \quad (1.8)$$

In addition to that, we define the left-invariant metric tensor field:

Definition 1.2.2. *A metric tensor field $\mathcal{G} \in \Gamma(T^{(0,2)}(\mathbb{M}_2))$ on \mathbb{M}_2 is left-invariant iff*

$$\mathcal{G}_{g,p}((L_g)_*\dot{p}, (L_g)_*\dot{p}) = \mathcal{G}_p(\dot{p}, \dot{p})$$

for all $p \in \mathbb{M}_2$, all $\dot{p} \in T_p(\mathbb{M}_2)$ and all $g \in \text{SE}(2)$.

Remark 1.2.1 (Left-invariant metric tensor field). Let \mathcal{G} denote a left-invariant metric tensor field on G . Then there exists a unique constant matrix $[g_{ij}]_{ij} \in \mathbb{R}^{3 \times 3}$ such that

$$\mathcal{G} = \sum_{i,j=1}^3 g_{ij} \omega^i \otimes \omega^j, \quad (1.9)$$

where \otimes denotes the usual tensor product.

In the standard left-invariant model, we restrict ourselves to the case where $g_{ij} = g_{ii}\delta_{ij}$. Then the metric tensor field \mathcal{G} is diagonal with respect to the co-frame $\{\omega^i\}_i$, i.e., $\mathcal{G} = \sum_{i=1}^3 g_{ii} \omega^i \otimes \omega^i$.

The metric tensor fields define the distance on a manifold or group as follows:

Definition 1.2.3 (Distance w.r.t. metric tensor field \mathcal{G}). *The distance from $p \in \mathbb{M}_2$ to a point $q \in \mathbb{M}_2$ with respect to a metric tensor field \mathcal{G} is defined by*

$$d_{\mathcal{G}}(p, q) = \inf_{\substack{\gamma \in \Gamma_1 \\ \gamma(0)=p, \gamma(1)=q}} \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt, \quad (1.10)$$

where $\Gamma_1 := PC^1([0, 1], \mathbb{M}_2)$ are the piecewise continuously differentiable curves in \mathbb{M}_2 .

Often, we want to describe movement that is not symmetric, such as only forward movement (to avoid cusps, see Section 1.4.2). To describe this, we cannot use a (symmetric) Riemannian metric tensor field but can resort to the general Finsler geometry as done in [83] and [137]. Essentially, this means that we replace the symmetric norm $\sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}$ in the Riemannian distance/metric d_G by an asymmetric Finsler norm $\mathcal{F}(\gamma(t), \dot{\gamma}(t))$, given by

$$|\mathcal{F}(p, \dot{p})|^2 = \mathcal{G}_p(\dot{p}, \dot{p}) + g_{11}(\varepsilon^{-2} - 1) \min \{0, \omega_p^1(\dot{p})\}^2, \quad (1.11)$$

where the relaxation parameter $0 < \varepsilon \ll 1$ penalizes backward spatial motion in the A_1 -direction. For convergence results of geodesics and distances if $\varepsilon \downarrow 0$, see [83, 97]. The Finslerian distance is defined as

$$d_{\mathcal{F}}(p, q) = \inf_{\substack{\gamma \in \Gamma_1 \\ \gamma(0)=p, \gamma(1)=q}} \int_0^1 \mathcal{F}(\gamma(t), \dot{\gamma}(t)) dt. \quad (1.12)$$

Lastly, we introduce the concept of *control sets*. These sets contain all vectors at most distance 1 from a reference point relative to a specific metric tensor field.

Remark 1.2.2 (Control sets). The control set in the tangent bundle $T(\mathbb{M}_2)$ is defined as

$$\mathcal{B}_{\mathcal{G}}(p) := \left\{ \dot{p} \in T_p(\mathbb{M}_2) \mid \sqrt{\mathcal{G}_p(\dot{p}, \dot{p})} \leq 1 \right\},$$

with $p \in \mathbb{M}_2$ and \mathcal{G} the underlying Riemannian metric tensor field.

The corresponding asymmetric control set in the tangent bundle $T(\mathbb{M}_2)$ is defined as

$$\mathcal{B}_{\mathcal{F}}(p) := \left\{ \dot{p} \in T_p(\mathbb{M}_2) \mid \mathcal{F}(p, \dot{p}) \leq 1 \right\}, \quad (1.13)$$

with $\mathbf{p} = (\mathbf{x}, \mathbf{n}) \in \mathbb{M}_2$ and \mathcal{F} the underlying Finsler function. In the limiting case where backward motions become prohibited as $\varepsilon \downarrow 0$ (i.e., the sub-Finslerian setting), we only get half of the Riemannian control sets

$$\lim_{\varepsilon \downarrow 0} \mathcal{B}_{\mathcal{F}}(\mathbf{p}) = \{\dot{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}}) \in \mathcal{B}_{\mathcal{G}}(\mathbf{p}) \mid \dot{\mathbf{x}} \cdot \mathbf{n} \geq 0\}, \quad (1.14)$$

with the Finsler function \mathcal{F} given in Eq. (1.11)

1.3 LIFTING FROM \mathbb{R}^2 TO \mathbb{M}_2

We aim to create a three-dimensional representation of the two-dimensional image wherein crossing structures are disentangled. To that end, we have introduced the space of planar positions and orientations \mathbb{M}_2 . The only thing left is to explain how we create the three-dimensional image representation which we call *orientation scores*, which we will do in this section.

Definition 1.3.1 (Orientation scores). *The orientation score transform $W_\psi : \mathbb{L}_2(\mathbb{R}^2) \rightarrow \mathbb{L}_2(\mathbb{M}_2)$ maps an image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ to an orientation score $U = W_\psi f : \mathbb{M}_2 \rightarrow \mathbb{R}$ using anisotropic wavelet ψ . The orientation score is given by*

$$(W_\psi f)(\mathbf{x}, \theta) := \int_{\mathbb{R}^2} \psi(R_\theta^{-1}(\mathbf{y} - \mathbf{x})) f(\mathbf{y}) d\mathbf{y}. \quad (1.15)$$

For the anisotropic wavelet ψ , we use the real part of the cake wavelets [18, 73] depicted in Fig. 1.2c. This choice of wavelet results in an invertible orientation score transform $W_\psi f$; one can reconstruct the image f by integrating over the orientation θ . More information on the orientation score transform W_ψ , its range, its invertibility, and the choice of wavelets ψ can be found in previous works [16, 73, 75, 78].

In other words, we calculate the orientation score $U = W_\psi f$ of an input image f . The orientation score is a three-dimensional representation of the two-dimensional image f . Each point in the image $f(\mathbf{x})$ has multiple representatives in the orientation score $W_\psi f(\mathbf{x}, \cdot)$: one for each orientation $\theta \in \mathbb{R}/(2\pi\mathbb{Z})$ being $W_\psi f(\mathbf{x}, \theta)$. The value $W_\psi f(\mathbf{x}, \theta)$ measures correspondence between the local structure in the image at point \mathbf{x} and the orientation θ (represented by the counter-clockwise rotation of the wavelet ψ). By calculating this value $W_\psi f(\mathbf{x}, \theta)$ for a range of different orientations θ , we lift the image from \mathbb{R}^2 to \mathbb{M}_2 .

The orientation score $W_\psi f$ is built in such a way that applying a roto-translation to the image and then calculating the orientation score transform is the same as first calculating the orientation score and then applying a left action on the score, i.e.,

$$W_\psi \circ \mathcal{U}_g = \mathcal{L}_g \circ W_\psi, \quad (1.16)$$

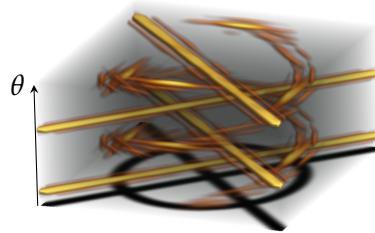
(a) Reference image in \mathbb{R}^2 .(b) Lifted image in $\text{SE}(2)$.(c) Visualization of the real part of the cake wavelet, with orientations $\theta = 0$ (left) and $\theta = 3\pi/16$ (right), used to lift image f to the orientation score.

Figure 1.2: Visualization of a grayscale image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ and its corresponding orientation score $U : \mathbb{M}_2 \rightarrow \mathbb{R}$ in the space of positions and orientations \mathbb{M}_2 given by (1.15), using a standard cake wavelet as depicted in 1.2c. We use a volume rendering where the orange spirals indicate data-points $\mathbf{p} = (x, y, \theta) \in \mathbb{M}_2$ with high amplitudes $|U(\mathbf{p})|$.

where the roto-translation of the image is given by

$$(\mathcal{U}_g f)(\mathbf{y}) = f(R_\theta^{-1}(\mathbf{y} - \mathbf{x})), \quad (1.17)$$

and the left action on the score by

$$\mathcal{L}_g U(\mathbf{p}) := U(L_{g^{-1}}(\mathbf{p})), \quad (1.18)$$

where we use that the identity element $e \in \text{SE}(2)$ is given by $e = (\mathbf{0}, I)$, and hence the inverse g^{-1} of an element $g = (\mathbf{x}, R_\theta)$ is given by $g^{-1} = (-R_{-\theta}\mathbf{x}, R_{-\theta})$.

In [73, Thm. 21], it was shown that the orientation score processing is left-invariant (i.e., equivariant with respect to left actions \mathcal{L}_g and roto-translations \mathcal{U}_g). The key reason is the relation Eq. (1.16). This is also crucial to the image-processing applications that are discussed in this thesis:

- Rotating and translating an image should yield an equally rotated and translated (lifted) tracking output curve.
- Rotating and translating an image should yield equally rotated and translated (lifted) connected components.

The orientation scores are constructed such that the vector \mathcal{A}_1 points along the local orientation in the score. To make this more concrete, we use the concept of horizontality.

Definition 1.3.2 (Horizontality on \mathbb{M}_2). *On \mathbb{M}_2 , we choose the distribution $\Delta^{\mathbb{M}_2} := \langle \mathcal{A}_1, \mathcal{A}_3 \rangle \subset T\mathbb{M}_2$. Then a smooth curve $\gamma : \mathbb{R}^2 \rightarrow \mathbb{M}_2$ is said to be horizontal if $\dot{\gamma}(t) \in \Delta_{\gamma(t)}^{\mathbb{M}_2}$ for all t .*

By construction, the orientation score transform in Eq. (1.15) lifts curves in \mathbb{R}^2 to horizontal curves in \mathbb{M}_2 [99, Sec. 2.8.5]. Hence, we see that the tensor fields $\omega^1 \otimes \omega^1$, $\omega^2 \otimes \omega^2$, and $\omega^3 \otimes \omega^3$ measure forward, lateral, and angular movement respectively. Consequently, the vectors \mathcal{A}_1 , and \mathcal{A}_2 point spatially along, and spatially perpendicular to the structure, whereas \mathcal{A}_3 points into the angular direction.

1.4 GEODESIC TRACKING

Geodesic tracking methods are designed to identify the shortest paths that trace the structure of the biological blood vessels in the retinal image. Such shortest paths are obtained by finding minimizing geodesics, defined as curves with the shortest length functionals. Typically, such length functionals are driven by a cost function that is small at locations of the blood vessels and high at all other places. Many different approaches to determine the minimizing geodesics have

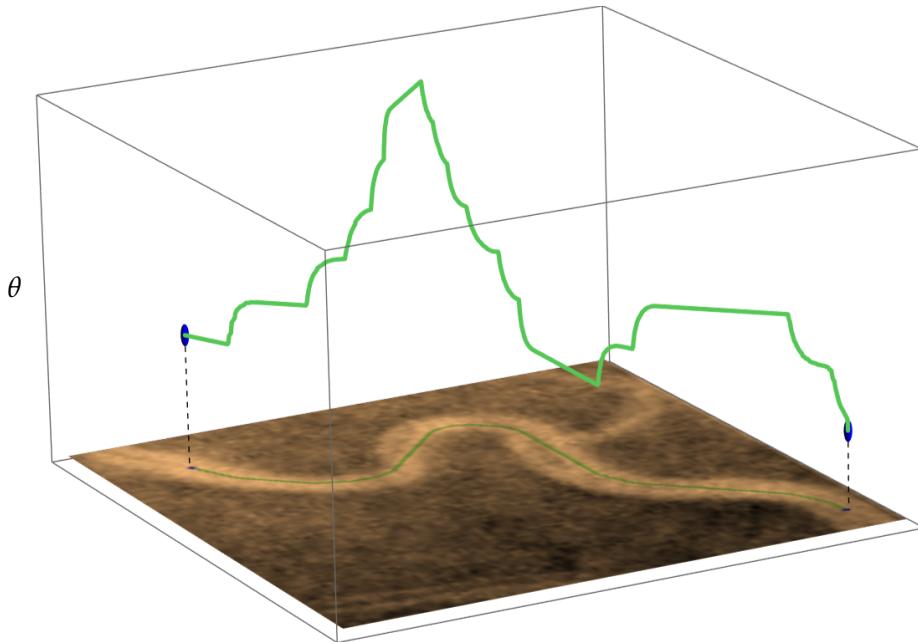


Figure 1.3: The tracking performed in the lifted space of positions and orientations \mathbb{M}_2 is projected back onto the input 2D image.

been proposed over the years, ranging from classical geodesic tracking in the image domain [44] to tracking in higher-dimensional homogeneous spaces [16, 19, 24, 83].

As already explained in Section 1.3, we lift the input image f from \mathbb{R}^2 to \mathbb{M}_2 using orientation scores in homogeneous spaces ([75, 78], see Fig. 1.2). The tracking process involves computing a geodesic distance map in \mathbb{M}_2 and then using steepest descent to find the *shortest curve* in the lifted space. Finally, we project the curve back onto the input image in \mathbb{R}^2 to get the final tracking result, cf. Fig. 1.3 for an example.

Over time, different models have been introduced that describe how the geodesics should behave. One of them, the *Reeds-Shepp “car” model*, describes the problem of the shortest paths for cars [163]. Here, one can imagine a car moving along a geodesic between an initial and final point. In the Reeds-Shepp forward model [83], the car’s “reverse gear” is turned off. In both the symmetric and asymmetric models, the spatially projected geodesics (optimal paths) tend to follow blood vessels in medical images well.

1.4.1 SYMMETRIC REEDS-SHEPP CAR MODEL

The left-invariant metric tensor field of the symmetric Reeds-Shepp car model, \mathcal{G} , is given by the symmetric tensor field

$$\begin{aligned}\mathcal{G} &= C^2 \left(\xi^2 \omega^1 \otimes \omega^1 + \frac{\xi^2}{\zeta^2} \omega^2 \otimes \omega^2 + \omega^3 \otimes \omega^3 \right) \quad \text{which gives} \quad (1.19) \\ \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}}) &= C(\mathbf{p})^2 \left(\xi^2 |\dot{\mathbf{x}} \cdot \mathbf{n}|^2 + \frac{\xi^2}{\zeta^2} \|\dot{\mathbf{x}} \wedge \mathbf{n}\|^2 + \|\dot{\mathbf{n}}\|^2 \right).\end{aligned}$$

for all $\mathbf{p} = (\mathbf{x}, \mathbf{n}) \in \mathbb{M}_2$, $\dot{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}}) \in T_{\mathbf{p}}(\mathbb{M}_2)$ with $\|\dot{\mathbf{x}} \wedge \mathbf{n}\|^2 := \|\dot{\mathbf{x}}\|^2 - |\dot{\mathbf{x}} \cdot \mathbf{n}|^2$. The anisotropy parameter ζ penalizes vectors with large sideways components. Note that the classical sub-Riemannian model corresponds to the limit $\zeta \downarrow 0$. For formal convergence results of the Riemannian model to the sub-Riemannian model see [83, Thm.2]. In practice, choosing $\zeta = 0.1$ usually provides a good enough approximation of the sub-Riemannian model for our purposes. The last parameter ξ , a weighting parameter, influences the flexibility of the tracking. It either stimulates or discourages angular movement over spatial movement [83].

The cost function $C : \mathbb{M}_2 \rightarrow [\delta, 1]$, with $\delta > 0$, discourages movement at specific locations, e.g., outside vessel structures, on which we elaborate in Section 1.4.3.

1.4.2 ASYMMETRIC REEDS-SHEPP CAR MODEL

Besides the symmetric version of the left-invariant metric tensor field of the Reeds-Shepp car model, an asymmetric version has been introduced in [83]. The forward gear version of this model is given by the asymmetric Finsler norm/function

$$|\mathcal{F}(\mathbf{p}, \dot{\mathbf{p}})|^2 = \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}}) + C(\mathbf{p})^2 (\varepsilon^{-2} - 1) \xi^2 |(\dot{\mathbf{x}} \cdot \mathbf{n})_-|^2 \quad (1.20)$$

for all $\mathbf{p} = (\mathbf{x}, \mathbf{n})$, $\dot{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}})$, with $a_- := \min\{0, a\}$. Eq. (1.20) coincides with Eq. (1.11) with $g_{11} = \xi^2$, $g_{22} = \xi^2/\zeta^2$, $g_{33} = 1$.

The parameters ζ and ξ and the cost function C have the same meaning as in the symmetric model. However, we consider an extra variable $\varepsilon \in (0, 1]$ in the asymmetric Reeds-Shepp car model. This parameter determines how strongly the model needs to adhere to the forward gear. Note that when $\varepsilon = 1$, we find the symmetric Reeds-Shepp car model, and when $\varepsilon \rightarrow 0$, backward movement becomes prohibited. In that case, cusps become in-place rotations to change orientation (cf. Fig. 1.4). These asymmetric Finslerian models are also highly beneficial in image segmentation as shown by [49].

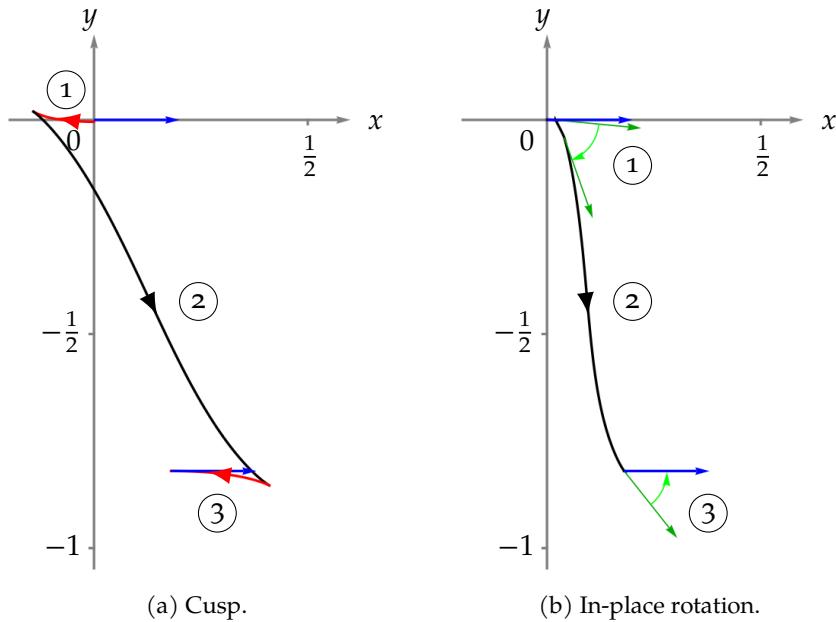


Figure 1.4: Visualization of two cusps in a spatially projected geodesic in the Riemannian manifold $(\mathbb{M}_2, \mathcal{G})$ (left, red) and in-place rotations in spatially projected Finslerian geodesic $(\mathbb{M}_2, \mathcal{F})$ (right, green).

1.4.3 COST FUNCTION

In both the symmetric and asymmetric Reeds-Shepp car model, the cost function $C : \mathbb{M}_2 \rightarrow [\delta, 1]$, with $\delta > 0$, discourages movement at specific locations, e.g., outside vessel structures. In this thesis, we slightly adapted the multi-scale crossing-preserving vesselness map [108], which is based on the multi-scale vessel enhancement filter [98]. We will discuss the existing vesselness filters next.

1.4.3.1 Multi-scale Vessel Enhancement Filter

We aim to identify light (tubular) structures on a dark background. These structures can range from straight lines to vascular trees and cracks in bridges. We introduce the multi-scale vessel enhancement filter [98].

We consider an image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ that contains light vascular structures or line structures on a dark background. We assume that this image contains some noise.

To reduce the impact of the noise, and to focus on structures of a specific width, we blur the image a little bit with a Gaussian kernel with scale $s \in S$, given by

$$G_s(\mathbf{x}) := \frac{1}{2\pi s^2} e^{-\frac{\|\mathbf{x}\|^2}{2s^2}},$$

where the set S contains a finite number of different scales.

We aim to find the tubular structures in the resulting image $G_s * f : \mathbb{R}^2 \rightarrow \mathbb{R}$. We do so by calculating the Hessian corresponding to this image:

$$H_S f(\mathbf{x}) := \begin{pmatrix} \frac{\partial^2}{\partial x^2} (G_s * f)(\mathbf{x}) & \frac{\partial^2}{\partial x \partial y} (G_s * f)(\mathbf{x}) \\ \frac{\partial^2}{\partial y \partial x} (G_s * f)(\mathbf{x}) & \frac{\partial^2}{\partial y^2} (G_s * f)(\mathbf{x}) \end{pmatrix}.$$

To simplify calculations, we use that

$$\frac{\partial}{\partial x} (G_s * f)(\mathbf{x}) = \left(f * \frac{\partial}{\partial s} G_s \right) (\mathbf{x}).$$

Once we have calculated the Hessian, we can determine its eigenvectors $v_1(\mathbf{x}), v_2(\mathbf{x})$ and the corresponding eigenvalues $\lambda_1(\mathbf{x}), \lambda_2(\mathbf{x})$, with $|\lambda_1| \geq |\lambda_2|$. Here, the eigenvector corresponding to the largest eigenvalue v_1 describes the principal direction of the tubular structure (if present), whereas the other eigenvector $v_2(\mathbf{x})$ is perpendicular to $v_1(\mathbf{x})$, as the Hessian is a symmetric matrix.

Then, we say that there is a tubular structure if $|\lambda_1(\mathbf{x})| \gg |\lambda_2(\mathbf{x})|$. We use this information to define the “anisotropy measure”

$$\mathcal{R}(\mathbf{x}) = \left(\frac{\lambda_2(\mathbf{x})}{\lambda_1(\mathbf{x})} \right)^2,$$

which describes the ratio between $\lambda_1(\mathbf{x})$ and $\lambda_2(\mathbf{x})$.

Besides the local anisotropy, we also need to describe likelihood of being on a local structure. For this, we use the structure measure

$$\mathcal{S}(\mathbf{x}) = \frac{\lambda_1(\mathbf{x})^2 + \lambda_2(\mathbf{x})^2}{\|\lambda_1^2 + \lambda_2^2\|_\infty}.$$

Lastly, we use the convexity criterion

$$\mathcal{Q}(\mathbf{x}) = \lambda_1(\mathbf{x})$$

to disregard dark tubular structures.

This allows us to define the vesselness belonging to scale s

$$V_s^{\mathbb{R}^2}(\mathbf{x}) := \begin{cases} \exp\left(-\frac{\mathcal{R}^2}{2\alpha^2}\right)\left(1 - \exp\left(-\frac{\mathcal{S}^2}{2\epsilon^2}\right)\right) & \text{if } \mathcal{Q}(\mathbf{x}) > 0 \\ 0 & \text{else.} \end{cases} \quad (1.21)$$

The parameters α, ϵ are standard set to $\alpha = 0.5, \epsilon = 0.5\|\mathcal{S}\|_\infty$. Then, the multi-scale vesselness is defined as

$$V^{\mathbb{R}^2}(\mathbf{x}) := \max_{s \in S} V_s^{\mathbb{R}^2}(\mathbf{x}) \quad (1.22)$$

This allows us to define the cost function as

$$C^{\mathbb{R}^2}(\mathbf{x}) = \frac{1}{1 + \lambda|V^{\mathbb{R}^2}(\mathbf{x})|^p},$$

with parameters $\lambda, p > 0$.

We show an example in Fig. 1.5b.

1.4.3.2 Multi-scale Crossing-Preserving Vessel Enhancement Filter

The multi-scale vessel enhancement filter introduced in Section 1.4.3.1 tends to find lower responses at crossing structures. This is due to the lack of a main direction in the crossing structures. To handle this issue, [108] introduced a crossing-preserving version of the multi-scale vessel enhancement filter. Here, a similar approach is followed on the orientation score $W_\psi f$ instead of on the image f .

First, we blur the orientation score with scales σ_s, σ_a in the spatial and angular direction respectively. Then, we work with the blurred orientation scores

$$(W_\psi f)^{\sigma_s, \sigma_a}(\mathbf{x}, \theta) := (G_{\sigma_s, \sigma_s, \sigma_a} * W_\psi f)(\mathbf{x}, \theta),$$

where

$$G_{\sigma_s, \sigma_s, \sigma_a}(\mathbf{x}, \theta) := G_{\sigma_s}(x) G_{\sigma_s}(y) G_{\sigma_a}(\theta).$$

Now, we redefine the anisotropy measure, structure measure, and convexity criterion using the blurred orientation score:

$$\mathcal{R}(\mathbf{x}, \theta) = \frac{\mathcal{A}_1 \mathcal{A}_1(W_\psi f)^{\sigma_s, \sigma_a}(\mathbf{x}, \theta)}{\mathcal{A}_2 \mathcal{A}_2(W_\psi f)^{\sigma_s, \sigma_a}(\mathbf{x}, \theta)}, \quad (1.23)$$

$$\mathcal{S}(\mathbf{x}, \theta) = \sum_{i=1}^2 \left(\mathcal{A}_i \mathcal{A}_i(W_\psi f)^{\sigma_s, \sigma_a}(\mathbf{x}, \theta) \right)^2, \quad (1.24)$$

$$\mathcal{Q}(\mathbf{x}) = \mathcal{A}_2 \mathcal{A}_2(W_\psi f)^{\sigma_s, \sigma_a}(\mathbf{x}, \theta). \quad (1.25)$$

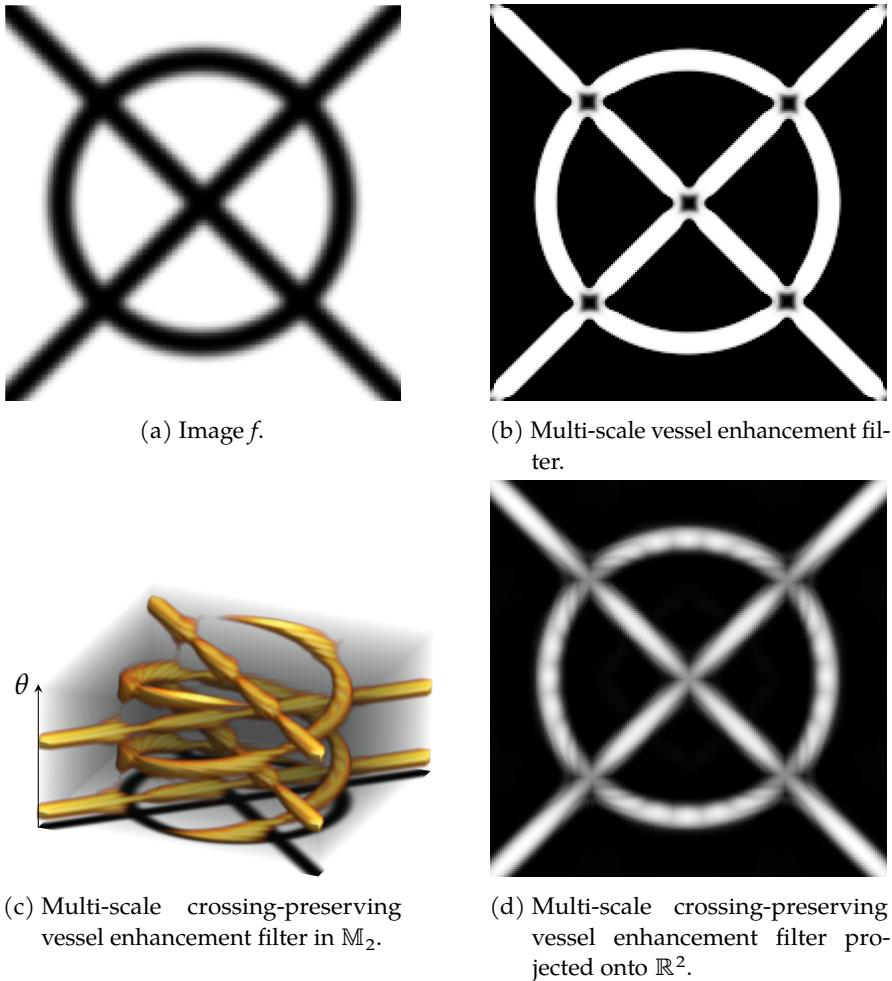


Figure 1.5: Visualization of the multi-scale vessel enhancement filter (Fig. 1.5b) and the multi-scale crossing-preserving vessel enhancement filter in \mathbb{M}_2 and its projection onto \mathbb{R}^2 in Fig. 1.5c and Fig. 1.5d respectively.

Then, the vesselness $V_s^{\mathbb{M}_2}$ in \mathbb{M}_2 is defined as Eq. (1.21), using the new measures in Eqs. (1.23) to (1.25). The multi-scale crossing-preserving vessel enhancement filter is defined slightly different compared to $V^{\mathbb{R}^2}$ in Eq. (1.22):

$$V^{\mathbb{M}_2}(\mathbf{x}, \theta) := \frac{\sum_{i=1}^{N_s} V_{s_i}^{\mathbb{M}_2}(\mathbf{x}, \theta)}{\left\| \sum_{i=1}^{N_s} V_{s_i}^{\mathbb{M}_2} \right\|_\infty}.$$

This allows us to define the cost function as

$$C^{\mathbb{M}_2}(\mathbf{x}, \theta) = \frac{1}{1 + \lambda |V^{\mathbb{M}_2}(\mathbf{x}, \theta)|^p}, \quad (1.26)$$

We have defined two different cost functions: one on \mathbb{R}^2 and one on \mathbb{M}_2 . The first cost function $C^{\mathbb{R}^2}$ gave low responses at crossing structures (cf. Fig. 1.5b). On the other hand, the cost function $C^{\mathbb{M}_2}$ does not suffer from the low responses at the crossings (cf. Fig. 1.5c). As we consider rather complex vasculature it is often more intuitive to display the minimum projection of the cost function $C^{\mathbb{M}_2}$ over θ , cf. Figure 1.5d.

The concepts in this chapter will help us to perform crossing-preserving, left-invariant image processing on images.

2

SPACE OF SPHERICAL POSITIONS AND ORIENTATIONS \mathbb{W}_2

In this thesis, we perform image processing on a higher-dimensional image representation to deal with crossing structures. In the previous chapter, we detailed how we lift the image from the flat two-dimensional space \mathbb{R}^2 to the space of planar positions and orientations \mathbb{M}_2 . In the case of retinal images, it makes more sense to lift the image to the space of spherical positions and orientations, as the eye is a sphere. Therefore, this chapter will focus on the space of *spherical* positions and orientations.

In this chapter, we first introduce the space of positions and orientations \mathbb{W}_2 . Next, we propose a frame that we will use in this space. After that, we introduce the metric tensor fields and Finsler functions on \mathbb{W}_2 . Finally, we detail how we map an image from \mathbb{R}^2 to the sphere S^2 . Further details on lifting from \mathbb{R}^2 to \mathbb{W}_2 are provided in Chapter 6.

2.1 DEFINITION OF \mathbb{W}_2 AND CORRESPONDING LIE GROUP

From a two-dimensional (spherical) image, we aim to create a three-dimensional image representation. First, we note that every position in the image corresponds to a position on the sphere $\mathbf{n} \in S^2$. Then, we assign to every position on the sphere an orientation vector $\dot{\mathbf{n}}_\phi$ where $\phi \in \mathbb{R}/(2\pi\mathbb{Z})$. We call this three-dimensional space the space of spherical positions and orientations. The formal definition follows next:

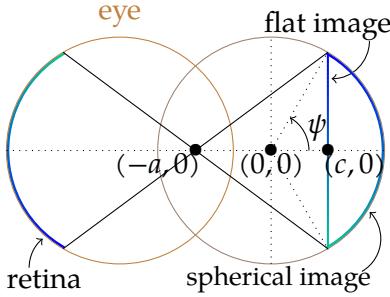
Definition 2.1.1 (Space of spherical positions and orientations). *The space of spherical positions and orientations \mathbb{W}_2 is defined as the smooth manifold*

$$\mathbb{W}_2 := S^2 \times S^1, \tag{2.1}$$

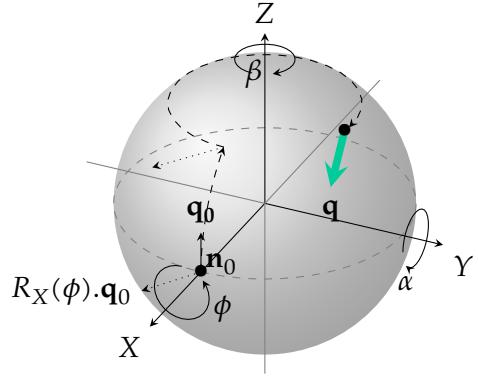
where $S^2 \cong (\mathbb{R}/(2\pi\mathbb{Z}))^2$ and $S^1 \cong \mathbb{R}/(2\pi\mathbb{Z})$ using the identifications

$$\mathbf{n} = (\cos \alpha \cos \beta, \cos \alpha \sin \beta, \sin \alpha) \leftrightarrow (\alpha, \beta), \quad \dot{\mathbf{n}}_\phi = (\cos \phi, \sin \phi) \leftrightarrow \phi.$$

To move from one position in the space of spherical positions and orientations to another, we have to use a group action that changes the position and orientation appropriately. For this, we use a rotation from the three-dimensional



(a) Schematic of medical setup: the curved retina is captured in a flat image, which we map to a curved image with the same geometry as the retina.



(b) Coordinates in $\text{SO}(3) \equiv \mathbb{W}_2$.

Figure 2.1: Visualization of the eye and the captured spherical image, and the used coordinates $\{\alpha, \beta, \phi\}$ of the point $\mathbf{q} = R(\alpha, \beta, \phi) \cdot \mathbf{q}_0 \in \mathbb{W}_2$ with $\mathbf{q}_0 = (\mathbf{n}_0, \dot{\mathbf{n}}_0) \in T(S^2)$ on the sphere. Parameter ψ denotes the max. angle of the wide-field image, $c > 0$ is the distance from the eyeball center to the flat image, $(-a, 0)$ is the focal point.

special orthogonal group $\text{SO}(3)$. Applying a rotation $R \in \text{SO}(3)$ to a point $(\mathbf{n}, \dot{\mathbf{n}}) \in \mathbb{W}_2$, this results in the group action

$$R \cdot (\mathbf{n}, \dot{\mathbf{n}}) := (R\mathbf{n}, R_*\dot{\mathbf{n}}), \text{ with pushforward } R_*, \quad (2.2)$$

where we assume the action $R \in \text{SO}(3)$ on the point $\mathbf{n} \in S^2$ is known. The easiest representation of this action is the embedding of both spaces in \mathbb{R}^3 , i.e., $(\mathbf{n}, \dot{\mathbf{n}}) \in \mathbb{R}^3 \times \mathbb{R}^3$, with $\|\mathbf{n}\| = \|\dot{\mathbf{n}}\| = 1$ and \mathbf{n} orthogonal to $\dot{\mathbf{n}}$. Then, $R \in \text{SO}(3)$ acts in Eq. (2.2).

The Lie group $\text{SO}(3)$ and the space \mathbb{W}_2 can be identified with each other in the specific case where we choose a reference point $\mathbf{q}_0 = ((1, 0, 0), (0, 0, 1)) \in \mathbb{W}_2$. Then,

$$\mathbb{W}_2 \cong \text{SO}(3) / \text{Stab}_{\text{SO}(3)}(\mathbf{q}_0) \cong \text{SO}(3) / \{I\} \cong \text{SO}(3). \quad (2.3)$$

We conclude that \mathbb{W}_2 is the principal homogeneous space of $\text{SO}(3)$.

We use the following coordinates in $\text{SO}(3)$ (see Fig. 2.1b):

$$R \longleftrightarrow (\alpha, \beta, \phi) \Leftrightarrow R = R_Z(-\beta) \circ R_Y(-\alpha) \circ R_X(\phi), \quad (2.4)$$

with $R_X(t), R_Y(t), R_Z(t)$ counter-clockwise rotations around the X -, Y -, and Z -axis by t radians. We use the same coordinates on \mathbb{W}_2 because $\mathbb{W}_2 \cong \text{SO}(3)$.

2.2 METRIC TENSOR FIELDS AND FINSLER FUNCTIONS

We have just introduced the space of spherical positions and orientations in which we can do image processing on a sphere. The image processing applications that we focus on, rely on the notion of distances. To calculate these distances, we need to introduce the metric tensor fields describing how distances are calculated, and the frames on which these metric tensor fields are built.

The frame we use in \mathbb{W}_2 depends on the static frame. This static frame is denoted by $\{\partial_\alpha, \partial_\beta, \partial_\phi\} \subset \Gamma(T\mathbb{W}_2)$, where $\Gamma(\cdot)$ denotes the smooth section of a given vector bundle, and induced by coordinates α, β, ϕ for all points in \mathbb{W}_2 . Its dual frame is denoted by $\{d\alpha, d\beta, d\phi\} \subset \Gamma(T^*\mathbb{W}_2)$, and can be used to express the metric tensor field \mathcal{G} .

Definition 2.2.1 (Frame of left-invariant vector fields on \mathbb{W}_2). *The frame of left-invariant vector fields (left-invariant frame) is obtained by a pushforward of the static frame at the origin q_0 . We define the pushforward $(R_g)_* : T_h(G) \rightarrow T_{gh}(G)$ by*

$$(R_g)_* \partial_\alpha|_{q_0} U = \partial_\alpha|_{q_0} (U \circ R_g),$$

for all smooth functions $U : \mathbb{W}_2 \rightarrow \mathbb{C}$ and the left group action R_g defined in Eq. (2.4). Then, the left-invariant frame $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ is defined by

$$\mathcal{B}_i|_{(\alpha, \beta, \phi)} = (R_{(\alpha, \beta, \phi)})_* \mathcal{B}_i|_{q_0}$$

$$\text{with } \mathcal{B}_1|_{q_0} = \partial_\alpha|_{q_0}, \mathcal{B}_2|_{q_0} = \partial_\beta|_{q_0}, \mathcal{B}_3|_{q_0} = \partial_\phi|_{q_0}.$$

The corresponding dual frame is given by $\{\nu^i\}_{i=1}^3$ where $\nu^i(\mathcal{B}_j) = \delta_j^i$.

We calculate the left-invariant vector fields on \mathbb{W}_2 explicitly. The left-invariant frame can be expressed as $\mathcal{B}_i = \sum_{j=1}^3 c_i^j \partial_{x_j}$. The coefficient functions c_i^j can be determined by solving

$$R(\alpha, \beta, \phi) \partial_{x_i} R(x_1, x_2, x_3)|_{(0,0,0)} = \sum_{j=1}^3 c_i^j \partial_{x_j} R(x_1, x_2, x_3)|_{(\alpha, \beta, \phi)},$$

with $x_1 = \alpha, x_2 = \beta, x_3 = \phi$. We obtain vector fields

$$\begin{aligned} \mathcal{B}_1|_{(\alpha, \beta, \phi)} &= \cos(\phi) \partial_\alpha|_{(\alpha, \beta, \phi)} + \frac{\sin(\phi)}{\cos \alpha} \partial_\beta|_{(\alpha, \beta, \phi)} \\ &\quad + \sin(\phi) \tan(\alpha) \partial_\phi|_{(\alpha, \beta, \phi)} \end{aligned} \tag{2.5a}$$

$$\begin{aligned} \mathcal{B}_2|_{(\alpha, \beta, \phi)} &= -\sin(\phi) \partial_\alpha|_{(\alpha, \beta, \phi)} + \frac{\cos(\phi)}{\cos \alpha} \partial_\beta|_{(\alpha, \beta, \phi)} \\ &\quad + \cos(\phi) \tan(\alpha) \partial_\phi|_{(\alpha, \beta, \phi)} \end{aligned} \tag{2.5b}$$

$$\mathcal{B}_3|_{(\alpha, \beta, \phi)} = \partial_\phi|_{(\alpha, \beta, \phi)}. \tag{2.5c}$$

A brief computation gives the corresponding dual frames ν^j

$$\nu^1 = \cos \phi \, d\alpha + \cos \alpha \sin \phi \, d\beta, \quad (2.6a)$$

$$\nu^2 = -\sin \phi \, d\alpha + \cos \alpha \cos \phi \, d\beta, \quad (2.6b)$$

$$\nu^3 = -\sin \alpha \, d\beta + d\phi. \quad (2.6c)$$

This frame of left-invariant metric tensor fields allows us to construct a left-invariant metric tensor field \mathcal{G} easily.

Definition 2.2.2. A metric tensor field \mathcal{G} on \mathbb{W}_2 is left-invariant iff

$$\mathcal{G}_{g \cdot q}((L_g)_* \dot{q}, (L_g)_* \dot{q}) = \mathcal{G}_q(\dot{q}, \dot{q})$$

for all $q \in \mathbb{W}_2$, all $\dot{q} \in T_q(\mathbb{W}_2)$ and all $g \in \text{SO}(3)$.

If the metric tensor field \mathcal{G} is left-invariant, there exists a unique constant matrix $[g_{ij}]_{ij} \in \mathbb{R}^{3 \times 3}$ such that

$$\mathcal{G} = \sum_{i,j=1}^3 g_{ij} \nu^i \otimes \nu^j,$$

where \otimes denotes the standard tensor product.

We introduced the metric tensor fields to describe distances, which we define next:

Definition 2.2.3 (Distance w.r.t. metric tensor field \mathcal{G}). The distance from $p \in \mathbb{W}_2$ to a point $q \in \mathbb{W}_2$ with respect to a metric tensor field \mathcal{G} is defined by

$$d_{\mathcal{G}}(p, q) = \inf_{\substack{\gamma \in \Gamma_T \\ \gamma(0)=p, \gamma(1)=q}} \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt, \quad (2.7)$$

where $\Gamma_T := PC^1([0, T], \mathbb{W}_2)$ piecewise continuously differentiable curves in \mathbb{W}_2 .

We will use the distance measure when tracking blood vessels, to find the shortest path while staying on the vascular structure.

2.3 MAPPING FROM \mathbb{R}^2 TO S^2

In this thesis, we mainly focus on the analysis of retinal images. In these images, we capture a 2D projection of the eye's vasculature. As the eye is spherical, we can determine the relation between the positions on the planar retinal image (projection) and the spherical positions in the eye. This section establishes this relation (similar to [135]).

We consider the setup illustrated in Fig. 2.1a. We will denote the planar (flat image) coordinates by $(x, y) \in \mathbb{R}^2$, and the spherical (curved image) coordinates by $(\alpha, \beta) \in [0, 2\pi) \times [-\pi/2, \pi/2)$. We define the projection $\Pi : [0, 2\pi) \times [-\pi/2, \pi/2) \rightarrow \mathbb{R}^2$ describing the coordinate transformation from spherical coordinates to planar coordinates:

$$\begin{aligned}(x, y) &= \Pi(\alpha, \beta) \\ &= \left(\frac{(a+c)\sin\alpha}{a+\cos\alpha\cos\beta}\eta, \frac{(a+c)\cos\alpha\sin\beta}{a+\cos\alpha\cos\beta}\eta \right),\end{aligned}\tag{2.8}$$

where $a, c \in \mathbb{R}$ denote the nodal point and reference distance of the flat image ($c = \cos(\psi)$) respectively, and with scaling parameter $\eta \in \mathbb{R}^+$ denoting the distance to the flat image.

From this coordinate mapping, one can also construct the inverse projection map $\Pi^{-1} : \mathbb{R}^2 \rightarrow [0, 2\pi) \times [-\pi/2, \pi/2)$, from the planar coordinates in the projection (x, y) to spherical coordinates (α, β) . For $\eta = 1$, this map is given by

$$\begin{aligned}(\alpha, \beta) &= \Pi^{-1}(x, y) \\ &= (\arcsin(x \bar{p}(x, y)), \arg(p_1(x, y) + i y \bar{p}(x, y))),\end{aligned}\tag{2.9}$$

where

$$\begin{aligned}\bar{p}(x, y) &= \frac{a(a+c) + \Xi_{a,c}(x, y)}{(x^2 + y^2) + (a+c)^2}, \\ p_1(x, y) &= \frac{(a+c)\Xi_{a,c}(x, y) - a(x^2 + y^2)}{(x^2 + y^2) + (a+c)^2},\end{aligned}$$

with $\Xi_{a,c}(x, y) = \sqrt{(x^2 + y^2)(1 - a^2) + (a+c)^2}$.

The coordinate transition from spherical to planar coordinates leads to image distortion. Normally, we sample the planar image with points of equal distance in the horizontal and vertical direction ($\Delta x, \Delta y \in \mathbb{R}$) as visualized in Fig. 2.2a. However, this sampling does not correspond to an equidistant sampling on the spherical image as visualized in Fig. 2.2b, i.e., $\Delta\alpha = \Delta\alpha(x, y)$, $\Delta\beta = \Delta\beta(x, y)$. Additionally, we see that distortion in the x - or y -coordinate leads to distortion in *both* the α - and β -coordinates. We can analyze each distortion individually, by looking at

$$\begin{aligned}(\Delta\alpha^x, \Delta\beta^x)(x_0, y_0) &= \frac{\Pi^{-1}(x_0 + \Delta x, y_0) - \Pi^{-1}(x_0, y_0)}{\Delta x} \xrightarrow{\Delta x \downarrow 0} \frac{d}{dx} \Pi^{-1}(x, y)|_{(x_0, y_0)}, \\ (\Delta\alpha^y, \Delta\beta^y)(x_0, y_0) &= \frac{\Pi^{-1}(x_0, y_0 + \Delta y) - \Pi^{-1}(x_0, y_0)}{\Delta y} \xrightarrow{\Delta y \downarrow 0} \frac{d}{dy} \Pi^{-1}(x, y)|_{(x_0, y_0)}.\end{aligned}$$

Then, we describe the (maximal) distortion by the amount $(\Delta\alpha^x, \Delta\beta^x)$ and $(\Delta\alpha^y, \Delta\beta^y)$ deviate from moving into the direction of the α - and β -coordinate,

respectively. We both describe the deviation in length and angle. In the case of no distortion, the angles for moving into the α - and β -direction would be 0 and $\pi/2$ respectively.

For the standard parameter settings $a = 13/21, c = \cos \psi, \eta = 1$, the distortion that comes from change in the x -direction corresponds to lengths $\|\cdot\|$ and angles \angle in the (α, β) -coordinates in the intervals

$$\begin{aligned} \left\| \frac{d}{dx} \Pi^{-1}(x, y) \right\| &\in (0.86, 1.14), \\ \angle \left(\frac{d}{dx} \Pi^{-1}(x, y), (1, 0) \right) &\in (-0.18, 0.18) \approx (-3\pi/10, 3\pi/10); \\ \left\| \frac{d}{dx} \Pi^{-1}(x, y) \right\| &\in (0.74, 1.45), \\ \angle \left(\frac{d}{dx} \Pi^{-1}(x, y), (1, 0) \right) &\in (-0.55, 0.55) \approx (-9\pi/10, 9\pi/10) \end{aligned}$$

for $\psi = \pi/5$ and $\psi = \pi/3$ respectively. Similarly, the distortion resulting from change in the y -direction corresponds to lengths and angles in the (α, β) -coordinates in the intervals

$$\begin{aligned} \left\| \frac{d}{dy} \Pi^{-1}(x, y) \right\| &\in (0.95, 1.24), \\ \angle \left(\frac{d}{dy} \Pi^{-1}(x, y), (1, 0) \right) &\in (1.36, 1.78) \approx (3\pi/14, 2\pi/7); \\ \left\| \frac{d}{dy} \Pi^{-1}(x, y) \right\| &\in (0.86, 2), \\ \angle \left(\frac{d}{dy} \Pi^{-1}(x, y), (1, 0) \right) &\in (1.05, 2.1) \approx (\pi/3, 2\pi/3) \end{aligned}$$

for $\psi = \pi/5$ and $\psi = \pi/3$ respectively. Here, parameter setting $\psi = \pi/5$ corresponds to [135].

Alternatively, we can look at the change in the area of a given size. This change in the projected area of a unit square on the sphere is expressed by

$$\det J_{\Pi}(\alpha, \beta) = \frac{(a + c)^2 \cos \alpha (1 + a \cos \alpha \cos \beta)}{(a + \cos \alpha \cos \beta)^3},$$

where J_{Π} denotes the Jacobian of the coordinate transformation Π . Then, the change in the area is, for standard parameter settings $a = 13/21, c = \cos \psi, \eta = 1$, and for $\psi = \pi/5$ and $\psi = \pi/3$ respectively, given by

$$0.778 = \det J_{\Pi}(0, 0) \leq \det J_{\Pi}(\alpha, \beta) \leq \det J_{\Pi}(\psi, \psi) = 1.12,$$

$$0.478 = \det J_{\Pi}(0, 0) \leq \det J_{\Pi}(\alpha, \beta) \leq \det J_{\Pi}(0.59, \psi) = 1.18.$$

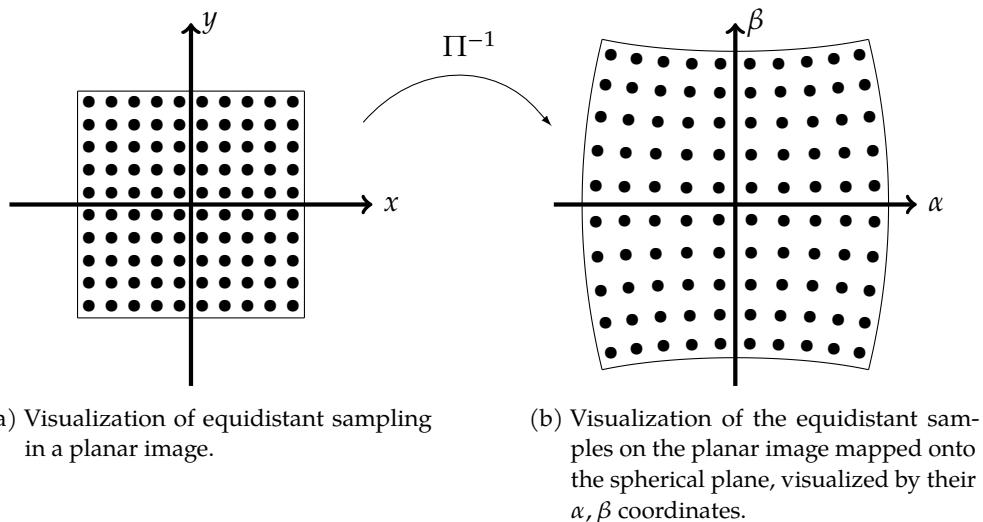


Figure 2.2: Visualization of equidistant sampling on the planar image (and mapped to the spherical coordinates using $a = 13/21, c = 0.5$).

We conclude that changing parameters leads to a relatively large distortion from both ways of describing the distortion (comparison of local distortions in step size and projected area). Therefore, we will look into the effect of this change of parameters on our tracking model, in Chapter 6.

Part II

GEODESIC TRACKING

3

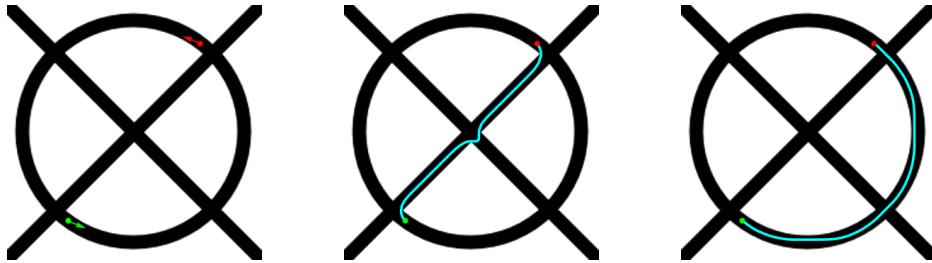
INTRODUCTION TO GEODESIC TRACKING

Retinal images are commonly used to examine the vascular system using optical scanning devices that capture the vasculature in the retina without invasive procedures. The vasculature in the eye is known to be typically representative of the vasculature throughout the body. This allows doctors to monitor the circulatory system and aids in the diagnosis of different kinds of diseases like diabetes, hypertension [16, 173, 197] and Alzheimer's disease [60]. Typically, high levels of tortuosity in the vasculature are a biomarker for such diseases [21, 53, 116, 174]. Successful automatic vessel tracking detects complex vasculature and aids the effective diagnosis of such diseases. Here, geometric models come into play via geodesic tracking methods, which compute the shortest paths that follows the biological blood vessels. They help in tracking and subsequent analysis of the vascular tree in the retina originating from the optic nerve [19, 48, 128, 129].

Geodesic tracking has been extensively studied where many prevalent approaches perform the tracking in the standard 2D image domain [45, 51, 58, 61, 155, 172]. For many methods in this category, calculating the geodesics in \mathbb{R}^2 leads to certain difficulties in accurately following the blood vessel. For example, one common difficulty is the inaccurate tracking of crossing structures and bifurcations, cf. Fig. 3.1b. This has motivated methods that aim to lift the image function to higher dimensional spaces. Examples include the space of positions and orientations [153, 158] or radius-lifted spaces [47] where the lifting yields the benefit of disentangling seemingly complex crossing structures in the retinal images.

In this thesis, we focus on the methods [18, 19, 24, 47] that perform the geodesic tracking in the 3D-space of positions and orientations \mathbb{M}_2 . This is based on lifted images or so-called *orientation scores*, as introduced in Chapter 1. The well-known benefit of this lifted approach is that lines involved in crossings are manifestly disentangled in \mathbb{M}_2 . As visualized in Fig. 1.2, the crossing circles in the image become disjoint spirals (cf. Fig. 1.2b) in the homogeneous space of positions and orientations.

In previous works by various authors, it has been shown that PDE-based geometric tracking algorithms [30, 83, 97, 130] perform well in \mathbb{M}_2 . Here,



(a) Reference image in \mathbb{R}^2 . (b) Tracking performed in \mathbb{R}^2 . (c) Tracking performed in \mathbb{M}_2 .

Figure 3.1: Tracking results (cyan) from the green point/arrow to the red point/arrow in $\mathbb{R}^2/\mathbb{M}_2$. Tracking algorithms in \mathbb{R}^2 mistake crossings for bifurcations, whereas tracking algorithms in \mathbb{M}_2 can differentiate between both, and hence stay on the correct structure.

one first calculates a distance map which is based on the image data. After computing the distance map, the steepest descent algorithm is applied to find each shortest path from a tip (an endpoint) to the corresponding seed (a starting point).

3.1 LIMITATIONS AND THEIR SOLUTIONS

Performing tracking in the lifted space of positions and orientations \mathbb{M}_2 fixes the issue of mistaking a crossing for a bifurcation. However, new problems arise in this higher-dimensional space. In this thesis, we focus on three of them:

1. Adjusting for the misalignment of the dominant direction \mathcal{A}_1 caused by orientation sampling in the lifting step, referred to as *deviation from horizontality*, by modifying the left-invariant metric and the associated Cartan connection (Chapter 4);
2. Addressing artifacts produced by fundus cameras, dehazing and illumination in particular, through optical and TV-flow enhancements on the retinal image (Chapter 5);
3. Adjusting for the curvature of the eye, instead of treating it as flat, by implementing crossing-preserving geodesic tracking on *spherical* images (Chapter 6).

3.1.1 MISALIGNMENT DUE TO ORIENTATION SAMPLING

Practical considerations of working in (the domain \mathbb{M}_2 of) orientation scores, like memory reduction and enabling low computation times, result in some

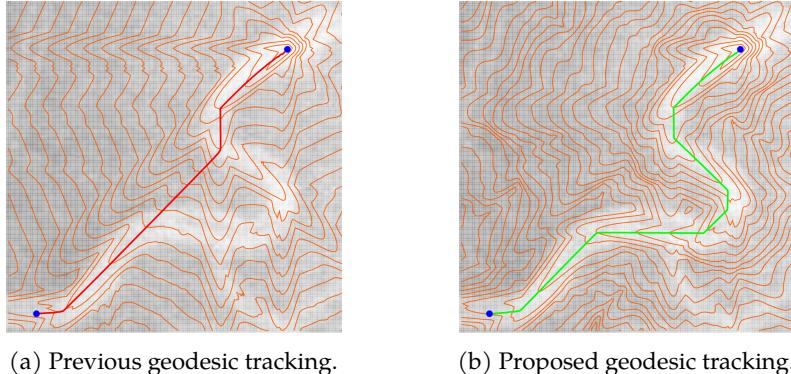


Figure 3.2: Orientations sampling bias in geodesic tracking. Sampling bias can lead to wrong tracking results, and our new model, presented in Chapter 4, will overcome this as we will show later in more detail (Fig. 4.5).

undesirable effects. For example, using a limited number of orientations leads to imperfections in the computation of the orientation scores. As a result, some vessels may be assigned a near angular coordinate that do not accurately reflect their true orientation. Consequently, it does not align with the vessel data correctly. We denote this problem as *misalignment* (also referred to as *deviation from horizontality* [100]). Moreover, considering a limited number of orientations results in a sampling bias on orientations, and thereby the possibility of missing high curvature regions yielding poor adaptation to vessel curvature (cf. Fig. 3.2).

In Chapter 4, we provide a novel, data-driven tracking model that improves upon existing geodesic tracking methods. Our model demonstrates an improved curvature adaptation, reduces misalignment, and exhibits a high degree of geometric interpretability.

We will aim for a *single* geometric Finslerian model to deal with complex vasculature without requiring heavy preprocessing (e.g. placement of anchor points, pre-skeletonization) and associated extra parameters, and without suffering from the “cusp problems” reported in [19, 38, 74].

The cusp problem is tackled by creating an asymmetric Finslerian model¹ (M_2, \mathcal{F}^U) extension of the data-driven Riemannian manifold similar to the much less data-driven techniques in [83]. For a quick impression of such a “cusp” in a spatially projected sub-Riemannian geodesic, see Fig. 1.4a in Section 1.4. Clearly, cusps are undesirable for vascular tracking, and an asymmetric Finslerian version of the Riemannian manifold tackles this problem. Intuitively, cusps in spatial projections of sub-Riemannian geodesics arise sometimes as

¹ In Finsler geometry [14, 83], the norm of tangent vectors may not be induced by an inner product. Recall that in the Riemannian setting, one does have $\mathcal{F}(p, \dot{p}) = \sqrt{g_p(\dot{p}, \dot{p})}$.

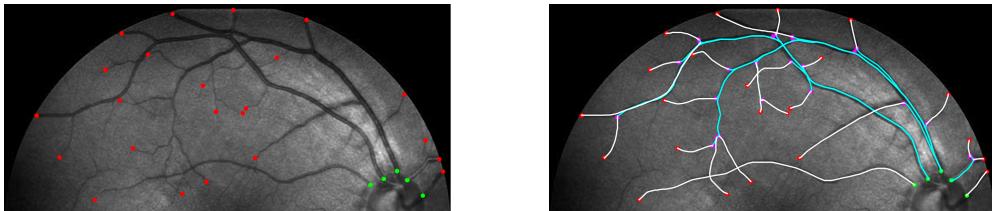


Figure 3.3: This tracking result (right) of a vascular tree in an optical image of the eye (left) is calculated with only two runs of the anisotropic fast-marching algorithm. In the images, seeds, bifurcations, and tips are indicated by green, purple, and red points respectively. The white and cyan lines denote the tracking results obtained in the first and second run respectively. Details follow in Section 4.5.

optimal paths of a *Reed-Shepp car* (imagine a car driving along the geodesic track) [83, 163] where the car was required to use its reverse gear to follow the optimal path, cf. Section 1.4. In the asymmetric Finslerian model we turn off the reverse gear of the car, while allowing for “in-place rotations” see Fig. 1.4b.

Preprocessing techniques for geodesic tracking such as pre-skeletonization and iterative placement of anchor/key points are typically used in conjunction with Bézier curves [5] or splines on Lie groups [5, 17], but often require additional parameters and fine-tuning. Specifically, extensive use of anchor points implies that anchors get relatively close to each other, and then the choice of geometric model in between becomes increasingly less relevant (even non-data-driven sub-Riemannian distance approximations suffice as shown in the work by Bekkers et al. [17]). As a result, this reduces the geometric interpretability of the overall model. In this work, we therefore aim for a single geometrical model. Hence, we will not use preprocessing, pre-skeletonization [47], multiple anchor points [50], and connectivity by perceptual grouping [2, 17, 58], even though these techniques are theoretically interesting and applicable.

In tracking an entire vascular tree, we limit the number of anchor points to at most one (which is computed without explicit manual supervision) and only use the boundary conditions for each vessel (see Fig. 3.3). Thanks to our new modified version of the anisotropic fast-marching algorithm [137], we can now better adapt to curvature and spatial misalignment efficiently (see Fig. 3.4). We also address common pitfalls at complex overlapping structures, where one must impose additional constraints to avoid taking wrong exits in the tracking. The implementation of such constraints is easily accounted for in our model.

Similar to the plus-control variant of previous geometric curve optimization models [83] we replace symmetric, anisotropic, (sub-)Riemannian geodesic tracking by *asymmetric*, anisotropic Finslerian models that avoid cusps and allow for automatic placement of in-place rotations (recall Fig. 1.4 and see [83,

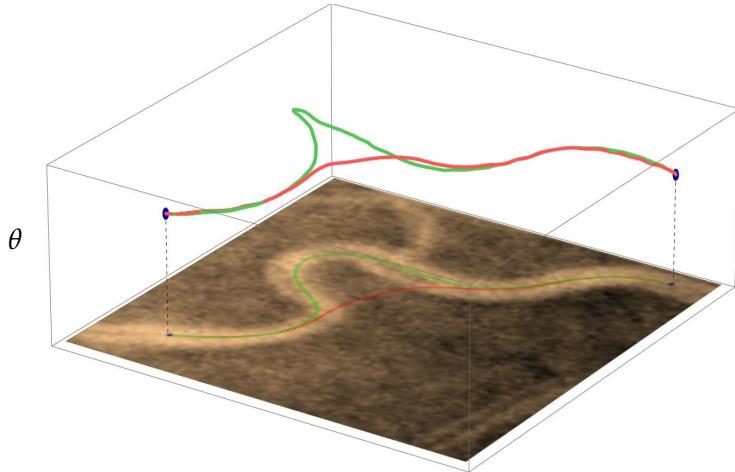


Figure 3.4: Tracking result of the previous left-invariant model [83] (red), and the new data-driven left-invariant model (green). The tracking performed in the lifted space of positions and orientations is projected back onto the input 2D image. Our proposed model (in green) demonstrates a significantly improved accuracy in adapting to curvature of blood vessels in the optical image.

134]). As a result, our model automatically accounts for bifurcations, thereby reducing the number of anchor points.

We discuss the new *data-driven geodesic tracking model* in detail in Chapter 4.

3.1.2 CAMERA ARTIFACTS: UNEVEN ILLUMINATION AND HAZE

Diseases such as cataracts can result in cloudy retinal images [201], while camera movements lead to motion artifacts [143] and uneven illumination [208]. These artifacts reduce the clarity and visibility of the vasculature, complicating tracking procedures. To cope with the limitations in the quality of ophthalmology images in practice, we must integrate both contrast enhancement from optical image processing [206] and crossing-preserving contextual TV-flows, in our geodesic tracking model for vasculature.

In our new tracking model, we integrate PDE-enhancements, like crossing-preserving total variation flow (TV-flow) enhancement in \mathbb{M}_2 [185]. We will show this improves the results. Furthermore, optical enhancement [206] of limited-quality retinal images is required to keep equal contrast and intensity across the whole vasculature. This inevitably creates small noisy structures that are non-aligned with other structures in the data. Applying the TV-flow enhancement in \mathbb{M}_2 leads to crossing-preserving contextual denoising that preserves crossings, and line structures, and removes noisy non-aligned struc-

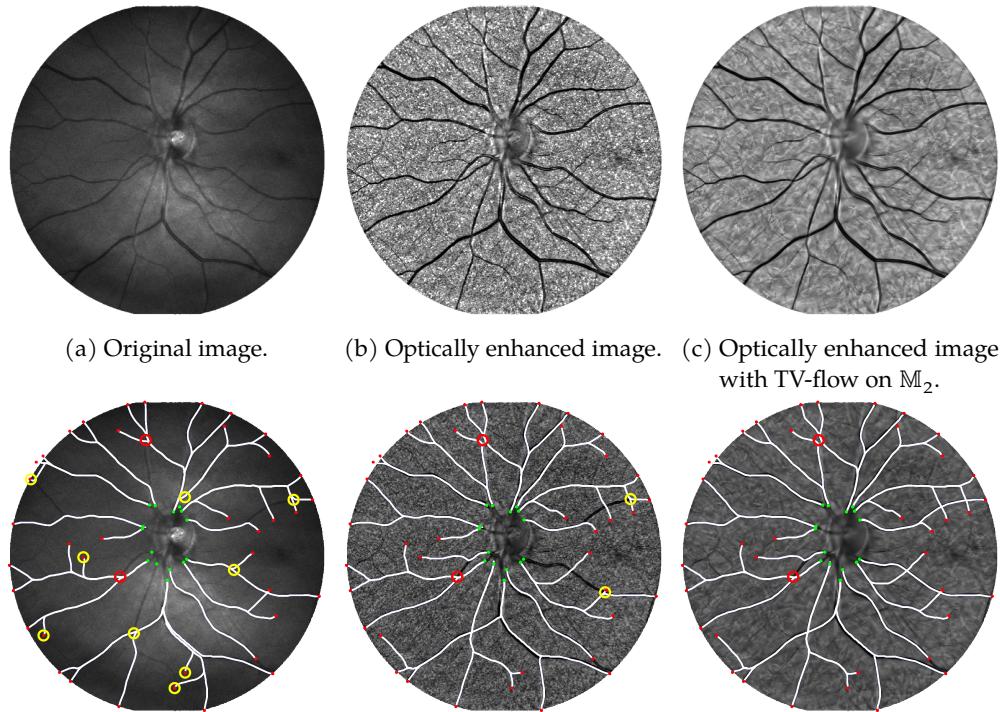


Figure 3.5: Geodesic tracking on the original image, (contrast-)enhanced image, and enhanced image after which TV-flow enhancement is done (left to right). The seeds and tips are indicated in respectively green and red. Yellow (/ red) circles indicate tracking mistakes that are (/ are *not*) fixed in the tracking of another column.

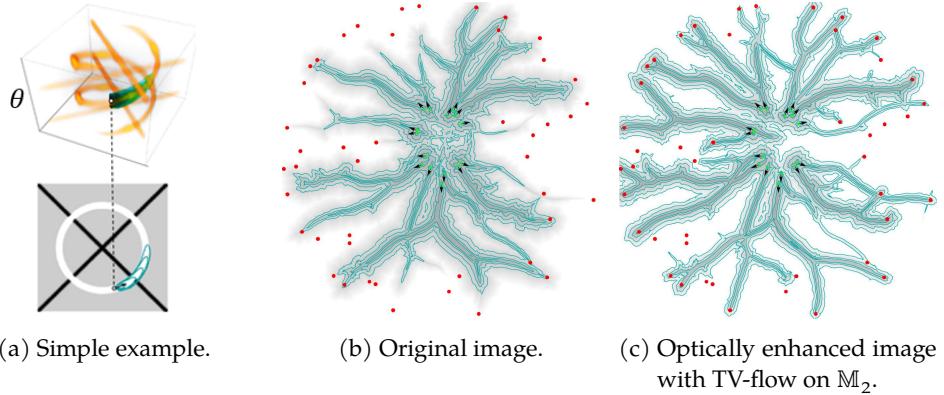


Figure 3.6: Distance map built in \mathbb{M}_2 (Fig. 3.6a; top) and isocontours of the minimum projection over the orientations back onto \mathbb{R}^2 (Fig. 3.6a; bottom). The isocontours of the minimum projection of the distance map are constructed based on the original image and optically enhanced image after which TV-flow enhancement is done (in Figs. 3.6b and 3.6c respectively).

tures from the optical enhancement. Altogether the scheme results in a vascular tracking algorithm that provides better results as wavefronts follow the complex branching vasculature better than the approach in [30], see Fig. 3.5. Even a single geodesic front propagation (building the distance map initializing all seeds at the same time), where fronts follow the entire vasculature in one run produces good results, see Fig. 3.6.

In Chapter 5, we discuss the details of the new *geodesic tracking model with optical and TV-flow enhancements*.

3.1.3 SPHERICAL IMAGES

Existing geodesic tracking models on \mathbb{M}_2 implicitly assume that the retinal image accurately captures the geometry of the retina. However, since the image is flat while the retina is spherical, the distances in the projection deviate from the actual distances. This deviation influences the tracking results. This concern was first put forward by Mashtakov et al. [135]: processing spherical images requires data-driven versions of sub-Riemannian [25, 32, 37] geodesics on the space of *spherical* positions and orientations $\mathbb{W}_2 \equiv \text{SO}(3)$ instead of on \mathbb{M}_2 . The sub-Riemannian geometry can intuitively be visualized (as pointed out by Boscain & Rossi [37]) with a Reeds-Shepp car traveling over the blood vessels on the retinal sphere, cf. Fig. 3.7, where the car can only move forward and backward and turn the wheel with finite costs, while sideward motions have infinite cost.

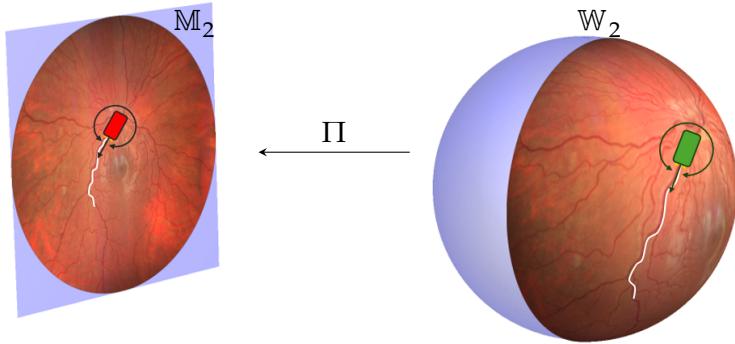


Figure 3.7: Visualization of a tracking result of a blood vessel (white) on the plane (left) and on the sphere (right). We also visualize cars (red on the plane, green on the sphere) that can only move forward and change orientation. This gives intuition behind existing models on $M_2 \cong \text{SE}(2)$ [83, 163] (red car), and our proposed, and previous [135] models, on $W_2 \cong \text{SO}(3)$ (green car). We show that the projection map Π is uniquely determined, cf. Theorem 6.1.1.

A limitation of sub-Riemannian geometry for vascular tracking is the presence of cusps, which do not occur in blood vessels [37], [135, Fig. 8]. This was solved in [83] by introducing the sub-Finslerian forward gear Reeds-Shepp car model. Since the car in Fig. 3.7 now also cannot move backward, it cannot create cusps.

In our new geodesic tracking model, we provide a data-driven extension of [25, 32, 37] of the sub-Riemannian geodesics model on W_2 . We practically improve the model and algorithm of Mashtakov et al. [135]. We create a *crossing-preserving* tracking algorithm on the sphere S^2 , using a cost function defined on the space of spherical positions and orientations W_2 . The new cost function ensures that the tracking method stays on the correct blood vessel, and does not mistake a crossing for a bifurcation. The model is designed such that no cusps occur (cf. Fig. 1.4a), since they are cumbersome in our specific application of tracking vasculature. We apply our new method to wide-field images [177]. These images cover up to 120° of the eyeball, resulting in deformations up to 52.2% between the flat image and the curved eye. Regular optical images only cover up to 72° of the eyeball and consequently deal with smaller deformations (up to 22.2%).

In Chapter 6, we discuss the details of this new tracking method.

4

GEODESIC TRACKING VIA NEW DATA-DRIVEN CARTAN CONNECTIONS

The first geodesic tracking model that we developed deals with very sinuous blood vessels and includes the underlying image data in the structure of the model. We do this by adapting the Cartan connection on which the model relies. In this chapter, we will elaborate on the details. Its content is based on [30].

In Section 4.1, we provide background on the geometrical tools underlying our method (Cartan connections and anisotropic fast-marching). In Section 4.2 and Section 4.3, we describe our model. Section 4.2 introduces a new data-driven Cartan connection ∇^U , which is associated with a data-driven left-invariant metric tensor field \mathcal{G}^U . These geometric tools allow for curvature adaptation and correction of misalignments in existing geodesic tracking algorithms in \mathbb{M}_2 . In Section 4.3 we use the data-driven Cartan connections and data-driven left-invariant metric tensor fields. Our main theoretical result, presented in Theorem 4.3.1,

- characterizes “straight curves” and “shortest curves” in data-driven left-invariant Riemannian manifolds on a finite-dimensional Lie group G ,
- analyzes the Hamiltonian flow of all geodesics together,
- provides the geodesic backtracking formula of the new geodesic tracking model,
- addresses the symmetries of the geodesics and connections of the new model.

Then in Section 4.4 we employ the geometrical models and tools and present a numerical algorithm to compute the distance map for the special case where the Lie group equals the roto-translation group $G = \text{SE}(2)$. Additionally, we explain how to compute the backtracking of geodesics from end to source point. We present a new version of the anisotropic fast-marching algorithm [140] that applies to our new data-driven model.

In Section 4.5, we report an extensive experimental evaluation of geodesic tracking in retinal images from the STAR dataset [1, 205], and show that our new model allows for adequate geometric tracking of highly complex vasculature. In Section 4.6, we end with a brief discussion of future work and conclude.

4.1 PRELIMINARIES ON CARTAN CONNECTIONS AND ANISOTROPIC FAST-MARCHING

4.1.1 CARTAN CONNECTIONS

The theory of Cartan connections was developed by Élie Cartan. His viewpoint on differential geometry relies on moving frames of reference (*repère mobile*). The idea is to connect tangent spaces by group actions on homogeneous spaces. This geometric tool allows us to understand the geodesic flow associated with the Riemannian distance Eq. (1.10) and its data-driven extensions.

For crossing-preserving image processing in 2D, we use the homogeneous space of positions and orientations \mathbb{M}_2 . Here, the pushforward $(L_g)_*$ of the left-multiplication connects $T_e(\mathbb{M}_2)$ to $T_g(\mathbb{M}_2)$ as it maps $T_e(\mathbb{M}_2)$ (isometrically¹) onto $T_g(\mathbb{M}_2)$ and $(L_g^{-1})_*$, known as the Cartan-Ehresmann form, maps $T_g(G)$ back to $T_e(G)$.

First, we introduce the general definition of Cartan connections, after which we also introduce the Cartan plus connection [84]. In this chapter, we will introduce a data-driven version of the Cartan plus connection, leading to a generalization of the existing theory on shortest and straight curves in \mathbb{M}_2 .

Definition 4.1.1 (Cartan connection [84]). *A Cartan connection on a Lie group G is a tangent bundle connection with the following additional properties*

1. *left invariance: if X, Y are left-invariant vector field then $\nabla_X Y$ is a left-invariant vector field,*
2. *for any $A \in T_e(G)$ the exponential curve is auto-parallel, i.e., $\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0$ where $\gamma(t) = \gamma(0) \exp(tA)$.*

We use the following special case of a Cartan connection to define shortest and straight curves. Note that this Cartan connection is easily expressed in the left-invariant frame [47, 57, 84, 121, 145, 156].

Definition 4.1.2 (Cartan plus connection [84]). *Consider a Lie group G of finite dimension n , with Lie brackets $[\cdot, \cdot]$ and structure constants $c_{ij}^k \in \mathbb{R}$ s.t.*

$$[\mathcal{A}_i, \mathcal{A}_j] = \sum_{k=1}^n c_{ij}^k \mathcal{A}_k.$$

Then the Cartan plus connection is given by

$$\nabla^{[+]} := \sum_{k=1}^n \left(\sum_{i=1}^n \omega^i \otimes (\mathcal{A}_i \circ \omega^k) + \sum_{i,j=1}^n \omega^i \otimes \omega^j c_{ij}^k \right) \mathcal{A}_k. \quad (4.1)$$

¹ W.r.t. norm induced by the metric tensor field \mathcal{G} in Eq. (1.9).

Remark 4.1.1. Note that the \circ symbol denotes the composition of functions such that for example

$$\mathcal{A}_i \circ \omega^k \left(\sum_{l=1}^n \alpha^l \mathcal{A}_l \right) = \mathcal{A}_i \left(\omega^k \left(\sum_{l=1}^n \alpha^l \mathcal{A}_l \right) \right) = \mathcal{A}_i(\alpha^k).$$

For explicit coordinate expressions, see [84]. Next, we explain how to read and compute Eq. (4.1). The covariant derivative $\nabla_X Y$ of a vector field Y with respect to a vector field X is again a vector field. Indeed the above formula gives

$$\nabla_X^{[+]} Y = \sum_{k=1}^n \left(\sum_{i=1}^n \omega^i(X) (\mathcal{A}_i \circ \omega^k(Y)) + \sum_{i,j=1}^n \omega^i(X) \omega^j(Y) c_{ij}^k \right) \mathcal{A}_k, \quad (4.2)$$

so that it becomes clear where X and Y typically enter in the open slots of the expression (4.1). Note that vector field \mathcal{A}_i in Eq. (4.2) is a differential operator applied to the smooth function $G \ni g \mapsto \omega_g^k(Y_g) \in \mathbb{R}$.

Remark 4.1.2. The connection $\nabla^{[+]}$ is called *Cartan plus connection* as we add the two sums between the two large round brackets. In differential geometry one also has Cartan connections with a real-valued scalar factor in front of the second term, but this does not serve our applications [84].

Having introduced Cartan connections, we now return to the central goal: designing a geometric model in which projected geodesics accurately follow the blood vessels.

4.1.2 ANISOTROPIC FAST-MARCHING

We provide here a brief overview of the partial differential equation (PDE) framework associated with geodesic distance maps, and of their numerical computation, see Section 4.4 for further details. We already mentioned in Section 1.4 that it is common to calculate minimizing geodesics in two steps: first, calculating the geodesic distance map and then, calculating the shortest curve using steepest descent. To get a first impression of how this looks in practice, see Fig. 4.1. The geodesic distance map is characterized in the PDE framework as the viscosity solution of a static first-order Hamilton-Jacobi-Bellman equation, known as the Eikonal Equation. For numerically solving the Eikonal equation, it is discretized using, for instance, finite differences [19], leading to a coupled non-linear system of equations, which is typically solved using a front propagation method such as the fast-marching algorithm (FMM). Classical references on FMM include [155, 179], anisotropic variants are presented in [137, 138, 140], and the details of our new model will follow in Section 4.4.1.

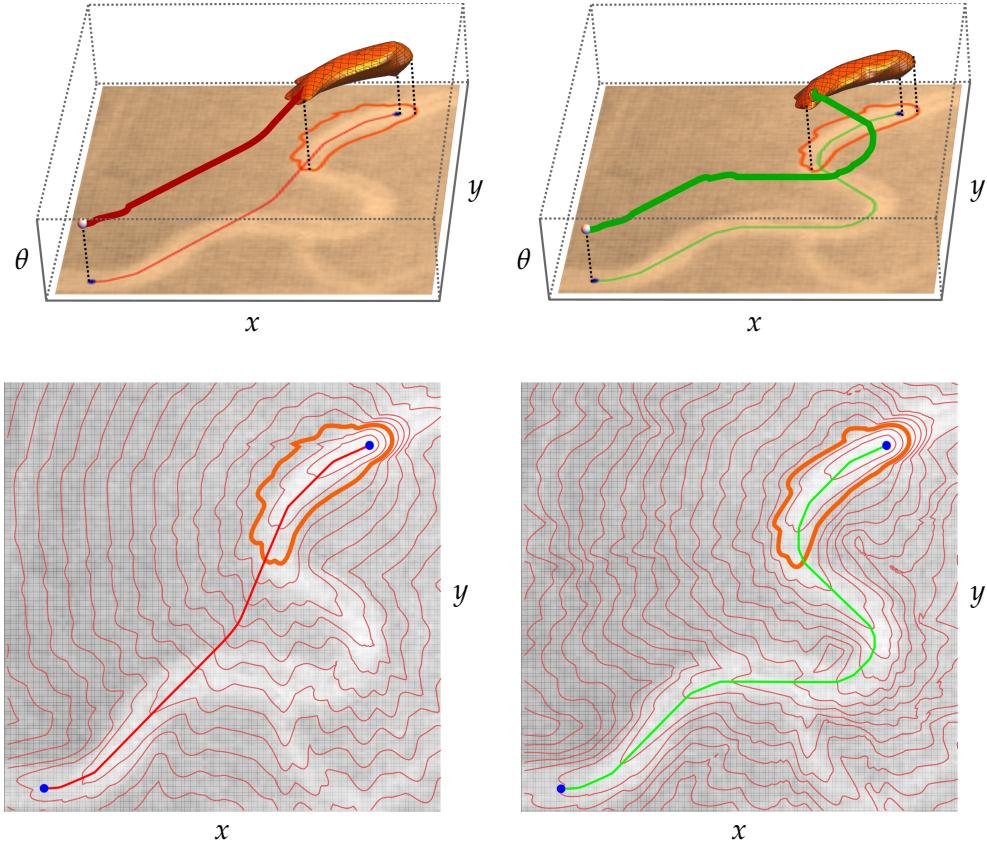


Figure 4.1: Top row: All points in the orange surface have the same distance to the seed. The isocontours are projected back onto the image, as depicted in the ground plane. Bottom row: Several isocontours are projected onto the image and a projection of the curve is visualized. Left: The Riemannian geodesic with parallel velocity to the Cartan plus connection $\nabla^{[+]}$ (red) takes a wrong shortcut. Right: The Riemannian geodesic (green) with parallel velocity to the Cartan plus connection ∇^U in the Riemannian manifold $(\mathbb{M}_2, \mathcal{G}^U)$ –or more precisely the Finslerian manifold $(\mathbb{M}_2, \mathcal{F}^U)$ given by Eq. (4.4)– does not, and moreover adapts for curvature in \mathbb{M}_2 , cf. Fig. 4.3). Explicit formulas for ∇^U and $\nabla^{[+]}$ will follow later in Table 4.2.

The fast-marching algorithm numerically solves the coupled system of equations that discretize the eikonal PDE. The algorithm proceeds in only one pass over the domain hence providing significant efficiency gains, but also requiring that the numerical scheme satisfies two conditions (monotonicity and causality), see [137, 140, Def. 2.1]. The proposed variant of this method uses Selling's algorithm [178] to calculate in a preliminary step a decomposition of the quadratic forms defining the dual metric. This dual metric suitably only involves positive weights and vectors with integer coordinates, see Proposition 4.4.1. These ingredients are used to devise an adaptive finite differences scheme, discretizing the anisotropic Eikonal PDE and obeying the required conditions, see Section 4.4.2. The eikonal PDE is solved via an anisotropic fast-marching algorithm, and its solution provides the desired distance map. Finally, the minimizing geodesic is calculated by solving an ordinary differential equation defined in terms of the distance map [137, 140].

In previous studies of the Reeds-Shepp model and variants [83, 137], the geodesic metric tensor matrix featured a block diagonal structure, which was exploited in the discretization. However, while working with data-driven metric tensor fields, this block format does not apply! Therefore we adapt the anisotropic fast-marching algorithm to cope with the general setting. In this chapter, we will briefly discuss the changes that were necessary to solve data-driven metric tensor fields. Such data-driven geometric models, described in the next section, yield improved tracking results compared to previous approaches, as one can see in Fig. 4.1. In addition to that, using the anisotropic fast-marching algorithm to calculate the geodesics, only a limited number of runs (one for a single vessel, Fig. 4.1, and only two for a full vasculature, Fig. 3.3) are needed to correctly track the vascular structures.

4.1.3 FLOWCHART AND OVERVIEW OF THE METHODOLOGY

Before presenting the details of our method, we outline a sequential flowchart and indicate where each component is discussed in more detail.

1. Create an orientation score of input image f (Eq. (1.15));
2. Calculate the Hessian (Appendix B);
3. Extract the Data-Driven frame from the Hessian (Eq. (4.4));
4. Determine the local cost function for tracking: Vesselness Map (Appendix C);
5. Identify the Finsler Function (with $+$ -control in Appendix D) with the special Riemannian case (Eq. (1.20) with $\varepsilon = 1$) in Theorem 4.3.1;

6. Identify the Dual Finsler Function (with +-control in Lemma 4.4.1 and Appendix D);
7. Analyze the Hamiltonian Flow of all geodesics (Theorem 4.3.1);
8. Determine the Eikonal PDE distance map (symmetric case: Eq. (4.25), with +-control: Eq. (4.40));
9. Numerically solve the Eikonal PDE using Anisotropic fast-marching:
 - a) Calculate the stencils using Selling matrix decomposition assuring causality for single parts (Proposition 4.4.1);
 - b) Follow the procedure of Far-Trial-Accepted points (Section 4.4.3);
10. Apply steepest descent on the distance map (Section 4.4.4);
11. Project the geodesic spatially by

$$P(x(t), y(t), \theta(t)) = (x(t), y(t)).$$

4.2 DATA-DRIVEN METRIC & DATA-DRIVEN CARTAN CONNECTION

In multi-orientation image processing, it is beneficial (for vessel segmentation [205]) to rely on locally adaptive frames [82, 175]. However, locally adaptive frames in \mathbb{M}_2 typically require a stable selection of the principal eigenvector (the eigenvector corresponding to the largest eigenvalue) of a symmetrized Hessian of the function $U : \text{SE}(2) \rightarrow \mathbb{R}$. Recall that a Hessian is defined via a dual connection. Even if one uses the left Cartan connection, selecting the principal eigenvector can be locally unstable [82], and the largest eigenvalue may not be unique. For instance, if line structures are not locally present at all. To address this, we propose a novel approach to construct an unconditionally stable, data-driven, left-invariant metric tensor field

Definition 4.2.1 (Data-driven left-invariant metric). *Let G be a Lie group. Then the metric tensor field \mathcal{G}^U is data-driven left invariant when it satisfies for all $(g, \dot{g}) \in T(G)$ and all $q \in G$:*

$$\mathcal{G}_g^U(\dot{g}, \dot{g}) = \mathcal{G}_{qg}^{\mathcal{L}_q U}((L_q)_*\dot{g}, (L_q)_*\dot{g}). \quad (4.3)$$

Recall that in our case of interest where $G = \text{SE}(2)$ and where $U = \mathcal{W}_\psi f$ is an orientation score of the image f , the equivariance relation in Eq. (1.16) holds, so roto-translation of an image $f \mapsto \mathcal{U}_g f$ is equivalent to roto-translation $U \mapsto \mathcal{L}_g U$ of the score.

Consequently (as will follow in Theorem 4.3.1) if a metric tensor field is data-driven left-invariant then a roto-translation $\mathcal{U}_g f$ of the input image f produces a new geodesic γ_{new} that is rotated and translated accordingly: $\gamma_{new}(\cdot) = g\gamma(\cdot)$.

Thus, Definition 4.2.1 is a valid constraint in our application as we want the vessel tracking along geodesics to be equivariant with respect to roto-translations.

By creating such a data-driven metric tensor field \mathcal{G}^U on our Lie group of interest $G = \text{SE}(2) \cong \mathbb{M}_2$, data-driven corrections are made for spatial and angular misalignment in existing models relying on the standard left-invariant frame [77, 82, 100]. We demonstrate that a better-fitted metric tensor field \mathcal{G}^U significantly improves tracking performance for highly tortuous vessels, as shown in Fig. 4.1. For our case of interest \mathbb{M}_2 , a reasonable choice that satisfies the constraint, and that we use in our experiments, is given by:

$$\begin{aligned}\mathcal{G}_{\mathbf{p}}^U(\dot{\mathbf{p}}, \dot{\mathbf{p}}) &= \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}}) + \lambda C^2(\mathbf{p}) \frac{\|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2}{\max_{\|\dot{\mathbf{q}}\|=1} \|HU|_{\mathbf{p}}(\dot{\mathbf{q}}, \cdot)\|_*^2}, \\ |\mathcal{F}^U(\mathbf{p}, \dot{\mathbf{p}})|^2 &= |\mathcal{F}(\mathbf{p}, \dot{\mathbf{p}})|^2 + \lambda C^2(\mathbf{p}) \frac{\|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2}{\max_{\|\dot{\mathbf{q}}\|=1} \|HU|_{\mathbf{p}}(\dot{\mathbf{q}}, \cdot)\|_*^2}\end{aligned}\quad (4.4)$$

where \mathcal{G} and \mathcal{F} are given in Eqs. (1.19) and (1.20),

with $\mathbf{p} = (\mathbf{x}, \mathbf{n}) \in \mathbb{M}_2$ and $\dot{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}}) \in T_{\mathbf{p}}(\mathbb{M}_2)$.

Here the Hessian field HU is defined in Lemma B.1 in Appendix B, and $\|\cdot\|_*$ the dual norm corresponding to the primal norm given by $\sqrt{\mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}})}$ with $\zeta = 1$ in Eq. (1.19).

Parameter $\lambda > 0$ regulates inclusion of data-driven 2nd order line-adaptation to the orientation score data U , cf. Fig. 1.2.

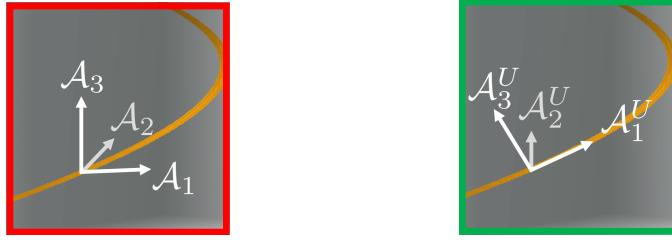
Finally, the data-driven left-invariant metric tensor field relies on the usual Reeds-Shepp car models \mathcal{G} respectively \mathcal{F} with external smooth cost $C(\mathbf{p})$ satisfying:

$$0 < \delta \leq C \leq 1, \quad (4.5)$$

computed from the orientation score U , as explained in Appendix C. There we combine ideas on crossing-preserving vesselness maps from [83, 108, 205].

Remark 4.2.1. Within \mathcal{G} and \mathcal{F} in Eq. (4.4) we set $\zeta^2 = 0.01 = g_{11}/g_{22}$ as relative costs for sideward motion, recall Eq. (1.19). Ideally we want this to be high, but as we will prove in Section 4.4.2, a spatial anisotropy of $\zeta^2 = 0.01$ still guarantees numerical accuracy. We follow [19, 83] and we set the bending stiffness parameters as $\xi^2 = g_{11} = 0.01$ and $g_{33} = 1$.

Proposition 4.2.1. *Metric tensor field \mathcal{G}^U given by Eq. (4.4) is indeed data-driven left-invariant (i.e., satisfying Eq. (4.3)).*



(a) Left-invariant frame.

(b) Data-driven left-invariant frame.

Figure 4.2: Visualizations of the left-invariant frame and the data-driven left-invariant frame in \mathbb{M}_2 . Locally along one of the spirals in the orientation score depicted in Fig. 1.2. In Fig. 4.2a, the main direction of \mathcal{A}_1 is not properly aligned with the underlying 3D structure, whereas in Fig. 4.2b \mathcal{A}_1^U is.

Proof. See Lemma B.3 in Appendix B. \square

Next, we list a few remarks that underpin and explain our specific choice of metric tensor field.

Remark 4.2.2. In geometric image analysis [106], eigenvectors of the Hessian typically provide a local coordinate frame along lines. In orientation scores, this is not different [100]. In \mathbb{M}_2 , Hessians $HU = \nabla^{[+],*} dU$ are not symmetric and we rely on a singular value decomposition via the dual norm in Eq. (4.4) which only relies on the symmetric product, see Remark 4.2.4.

Remark 4.2.3. Formally speaking, the (old) metric tensor fields \mathcal{G} and asymmetric version \mathcal{F} are also data-driven if it comes to scalar-adaptation via cost function C , but as they do not adapt for any kind of directional data-adaptation (as illustrated in Fig. 4.2) we do *not* refer to them as “the data-driven model”.

Remark 4.2.4. Via the identification in Eq. (1.3) we write $\mathbf{p} = (x, y, \theta)$ for short. In fixed coordinates on $\mathbb{R}^2 \times \mathbb{R}/(2\pi\mathbb{Z})$, we write $\dot{\mathbf{p}} = (\dot{x}, \dot{y}, \dot{\theta})^\top$. The dual norm expression $\|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2$ in Eq. (4.4) then boils down to a straightforward Euclidean norm:

$$\|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2 = \left\| M_\xi \begin{pmatrix} U_{xx}(\mathbf{p}) & U_{xy}(\mathbf{p}) & U_{x\theta}(\mathbf{p}) + U_y(\mathbf{p}) \\ U_{yx}(\mathbf{p}) & U_{yy}(\mathbf{p}) & U_{y\theta}(\mathbf{p}) - U_x(\mathbf{p}) \\ U_{\theta x}(\mathbf{p}) & U_{\theta y}(\mathbf{p}) & U_{\theta\theta}(\mathbf{p}) \end{pmatrix}^\top \dot{\mathbf{p}} \right\|^2 \quad (4.6)$$

where $M_\xi = \text{diag}(\xi^{-1}, \xi^{-1}, 1) \in \mathbb{R}^{3 \times 3}$.

For details of Hessians of functions on manifolds with a connection, see Appendix B. For now let us focus on the notion of data-driven left-invariant frames, where we improve upon the “Locally Adaptive Derivatives (LADs)” in [82, 205].

Definition 4.2.2 (Data-driven left-invariant frame). *Any data-driven metric tensor field \mathcal{G}^U can be diagonalized:*

$$\mathcal{G}^U = \sum_{i=1}^3 \alpha_i^U(\cdot) \omega_U^i \otimes \omega_U^i \quad (4.7)$$

and this defines the positively-oriented data-driven left-invariant co-frame $\{\omega_U^i\}_{i=1}^3$, dual to the primal frame $\{\mathcal{A}_j^U\}_{j=1}^3$ related by $\langle \omega_U^i, \mathcal{A}_j^U \rangle = \delta_j^i$.

Remark 4.2.5 (Advantages of our data-driven metric and frame). The local frame of reference $\{\mathcal{A}_i^U\}$ depends on the image data, cf. Fig. 4.2. In fact, Eq. (4.7) is used to define the dual of the data-driven left-invariant frame via diagonalization. This approach differs from LADs in previous work [84, 205], and offers the advantage of coercivity

$$\mathcal{G}^U \geq \mathcal{G} \geq \delta > 0, \quad (4.8)$$

recall Eq. (4.5), independent of the orientation score data U , which ensures unconditional stability of the tracking algorithms. Another benefit over LADs is that when U is constant, the data-driven left-invariant frame \mathcal{A}_i^U coincides with the standard left-invariant frame \mathcal{A}_i .

In order to calculate distances using the new data-driven metric tensor field, we need to introduce the data-driven Riemannian distance.

Definition 4.2.3 (Data-driven Riemannian distance). *The data-driven Riemannian distance $d_{\mathcal{G}^U}$ from a point $p \in \mathbb{M}_2$ to a point $q \in \mathbb{M}_2$ is given by*

$$d_{\mathcal{G}^U}(p, q) = \inf_{\substack{\gamma \in \Gamma_1, \\ \gamma(0) = p, \gamma(1) = q}} \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}^U(\dot{\gamma}(t), \dot{\gamma}(t))} dt \quad (4.9)$$

where $\Gamma_1 := PC^1([0, 1], \mathbb{M}_2)$ piecewise continuously differentiable curves in \mathbb{M}_2 , and $\dot{\gamma}(t) := \frac{d}{dt}\gamma(t)$.

Remark 4.2.6. If image U is constant, then $\mathcal{G}^U = \mathcal{G}$, $d_{\mathcal{G}^U} = d_{\mathcal{G}}$.

Remark 4.2.7. Note that this distance can be transformed into a quasi-distance when using the forward gear version of the model:

$$d_{\mathcal{F}^U}(p, q) = \inf_{\substack{\gamma \in \Gamma_1, \\ \gamma(0) = p, \gamma(1) = q}} \int_0^1 \mathcal{F}_U(\gamma(t), \dot{\gamma}(t)) dt.$$

	Metric Tensor Field	Diagonalization	Cartan Connection
Earlier work	\mathcal{G}	$\{\omega^i\}_{i=1}^n$	$\nabla^{[+]}$
Current work	\mathcal{G}^U	$\{\omega_U^i\}_{i=1}^n$	∇^U

Table 4.1: Comparison of (notation of) current and previous work. Diagonalization is w.r.t. dual frame associated to the frames depicted in Fig. 4.2.

Using the new data-driven metric frame, recall Definition 4.2.2, we introduce the data-driven Cartan plus connection, which will be used to express “short” and “straight” curves in Section 4.3.

Definition 4.2.4 (Data-driven Cartan plus connection). *The data-driven Cartan plus connection is given by*

$$\nabla^U := \sum_{k=1}^n \left(\sum_{i=1}^n \omega_U^i \otimes (\mathcal{A}_i^U \circ \omega_U^k) + \sum_{i,j=1}^n \omega_U^i \otimes \omega_U^j c_{ij}^k \right) \mathcal{A}_k^U.$$

Explicit coordinate expressions will follow in Lemma 4.3.1.

In Table 4.1, an overview of the notation used for the new concepts introduced in this work and concepts introduced in earlier work is given.

In Fig. 4.3, the exponential curves and the control sets for both discussed Cartan connections, $\nabla^{[+]}$ and ∇^U are visualized. In addition to that, the tracking results relying on different models are plotted. One sees that the data-driven Cartan connection better adapts for curvature leading to more accurate tracking results.

4.3 THE NEW GEOMETRIC TRACKING MODEL: ASYMMETRIC FINSLER FUNCTIONS STEERED BY LOCALLY ADAPTIVE FRAMES

We introduce a new data-driven version of the Cartan connection, applicable to all Lie groups G of finite dimension $\dim(G) = n$. Note that $\text{SE}(2) \cong \mathbb{M}_2$, but the result does not apply to all homogeneous spaces (like \mathbb{M}_d for $d > 2$). The notation used in this section is summarized in Table 4.2.

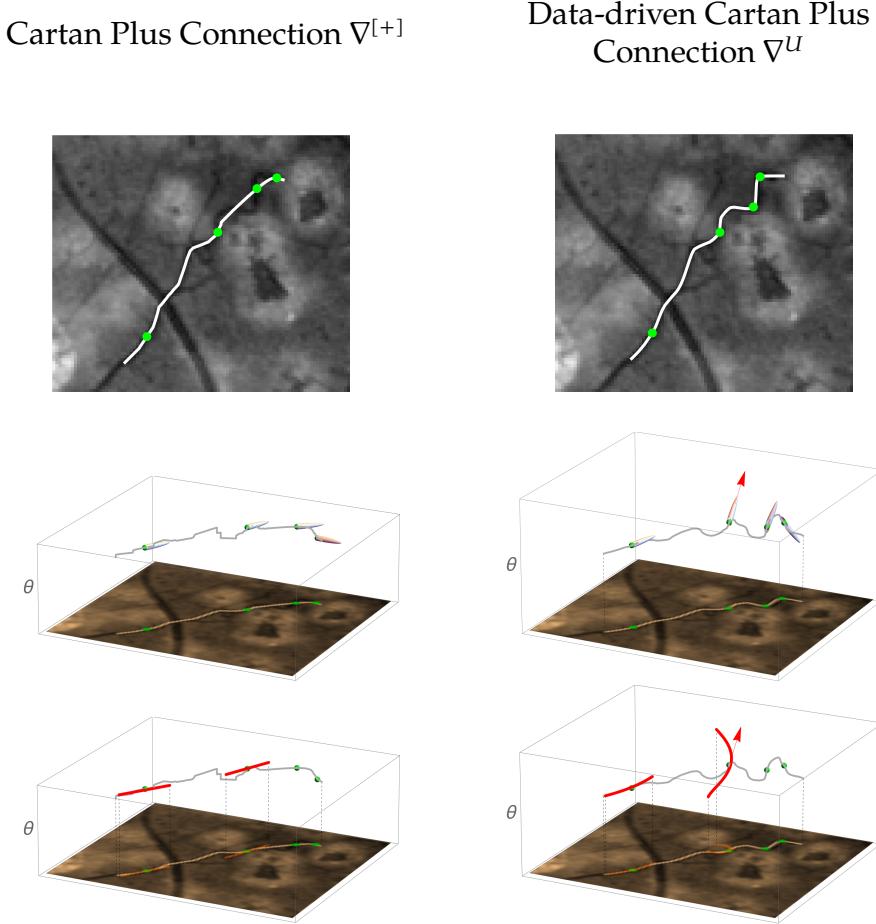


Figure 4.3: The advantage of using data-driven Cartan connections ∇^U instead of the non-data-driven Cartan plus connection $\nabla^{[+]}$. In grey, a shortest curve (geodesic) between two points in \mathbb{M}_2 is visualized, along with its spatial projection. The left geodesic has parallel momentum w.r.t. ∇^U (cf. Theorem 4.3.1) and the right w.r.t. $\nabla^{[+]}$ [84, Thm.1]. The new geodesic better adapts for curvature (and spatial alignment). This is also visible in the corresponding control sets (1.14) depicted by the white closed surfaces above at several green points on the geodesics. The red arrow indicates the principal direction of the local metric tensor (left: \mathcal{G} , right: data-driven \mathcal{G}^U). The control sets for $\nabla^{[+]}$ are only aligned to the underlying structure in the spatial domain, whereas those for ∇^U align with the appropriate curvature in the tangent space as well. In the bottom row, we depict exponential curves through the green points with a tangent in the principal direction (left of \mathcal{G} , right of \mathcal{G}^U). They are straight-curves of $\nabla^{[+]}$ (left) and ∇^U (right), and "steer" the geodesic tracking as we will show in Theorem 4.3.1.

$T(G) := \{(g, \dot{g}) \dot{g} \in T_g(G)\}$	$\{\partial_{x^i}\}_{i=1}^n;$	$\{\mathcal{A}_i\}_{i=1}^n;$	$\{\mathcal{A}_i^U\}_{i=1}^n.$
bases			
$T^*(G)$ bases	$\{dx^i\}_{i=1}^n;$	$\{\omega^i\}_{i=1}^n;$	$\{\omega_U^i\}_{i=1}^n.$
Notation duals	$\langle \hat{a}, b \rangle = \hat{a}(b).$		
Relation between bases $T(G)$ and $T^*(G)$	$\langle dx^i, \partial_{x^j} \rangle = dx^i(\partial_{x^j}) = \delta_j^i;$	$\langle \omega^i, \mathcal{A}_j \rangle = \delta_j^i;$	$\langle \omega_U^i, \mathcal{A}_j^U \rangle = \delta_j^i.$
LIV metric tensor field	$\mathcal{G} = \sum_{i=1}^n g_{ii} \omega^i \otimes \omega^i$	For specific choices of coefficients g_{ii} , see Eq. (1.19).	
Data-driven LIV metric tensor field	$\mathcal{G}^U = \sum_{i=1}^n \alpha_i^U(\cdot) \omega_U^i \otimes \omega_U^i$	that must satisfy Eq. (4.3). We choose to use Eq. (4.4) and write $\alpha_i := \alpha_i^U$. For its asymmetric Finslerian extension \mathcal{F}^U see also Eq. (4.4).	
Velocity components	$\dot{\gamma} = \sum_{i=1}^n \dot{\gamma}^i \partial_{x^i} = \sum_{i=1}^n \dot{\gamma}^i \mathcal{A}_i = \sum_{i=1}^n \dot{\tilde{\gamma}}^i \mathcal{A}_i^U.$		
Momentum components	$\lambda = \sum_{i=1}^n \bar{\lambda}_i dx^i = \sum_{i=1}^n \lambda_i \omega^i = \sum_{i=1}^n \tilde{\lambda}_i \omega_U^i.$		
Lagrangian	$\mathcal{L}(\gamma, \dot{\gamma}) = \frac{1}{2} \mathcal{G}_\gamma^U(\dot{\gamma}, \dot{\gamma}) ^2 = \sum_{i=1}^n \alpha_i^U(\gamma) \dot{\tilde{\gamma}}^i ^2.$		
Hamiltonian	$\mathfrak{h}(\lambda) = \frac{1}{2} \sum_{i=1}^n \tilde{\lambda}_i \tilde{\lambda}^i = \frac{1}{2} \sum_{i=1}^n \tilde{\lambda}_i \alpha^i(\cdot) \tilde{\lambda}_i$	with $\alpha^i = \alpha_i^{-1}$.	(4.11)

Fundamental symplectic form $\sigma \equiv \sum_{i=1}^n dx^i \wedge d\bar{\lambda}_i = \sum_{i=1}^n \omega^i \wedge d\lambda_i = \sum_{i=1}^n \omega_U^i \wedge d\tilde{\lambda}_i$,

where $\langle d\lambda_i, \partial_{\lambda_j} \rangle = \delta_j^i$, but $d\lambda_i \neq d\lambda_i$.

Hamiltonian flow $\begin{cases} \dot{\nu} = \vec{h}(\nu) \\ \nu(0) = \nu_0, \end{cases}$

where $\nu = (\gamma, \lambda)$, with γ denoting the geodesic and λ the momentum along it.

Lie bracket $[X, Y] = XY - YX$.

(4.12)

Poisson bracket $\{f, g\} = \sum_{i=1}^n \partial_{x^i} g \frac{\partial f}{\partial \bar{\lambda}_i} - \frac{\partial g}{\partial \bar{\lambda}_i} \partial_{x^i} f = \sum_{i=1}^n (\mathcal{A}_i^U g) \frac{\partial f}{\partial \tilde{\lambda}_i} - (\mathcal{A}_i^U f) \frac{\partial g}{\partial \tilde{\lambda}_i}$.

Structure functions $[\mathcal{A}_i^U, \mathcal{A}_j^U] = \sum_{k=1}^n \tilde{c}_{ij}^k(\cdot) \mathcal{A}_k^U$.

Structure constants $[\mathcal{A}_i, \mathcal{A}_j] = \sum_{k=1}^n c_{ij}^k \mathcal{A}_k$, where $c_{ij}^k = -c_{ji}^k$.

Data-driven Cartan connection $\nabla^U = \sum_{k=1}^n \left(\sum_{i=1}^n \omega_U^i \otimes \mathcal{A}_i^U \circ \omega_U^k(\cdot) + \sum_{i,j=1}^n (\omega_U^i \otimes \omega_U^j) \tilde{c}_{ij}^k(\cdot) \right) \mathcal{A}_k^U$. see (4.18)

Dual data-driven Cartan connection $(\nabla^U)^* = \sum_{i=1}^n \left(\sum_{j=1}^n \omega_U^j \otimes (\mathcal{A}_j^U \circ \mathcal{A}_i^U) + \sum_{j,k=1}^n (\omega_U^j \otimes \mathcal{A}_k^U) \tilde{c}_{ij}^k(\cdot) \right) \omega_U^i$. see (4.19)

Table 4.2: Table of geometric tools and notations.

We consider a locally adaptive frame $\{\mathcal{A}_i^U\}_{i=1}^n$ with dual frame $\{\omega_U^i\}_{i=1}^n$. This can be any well-defined frame that depends on the underlying data. The (data-driven) metric tensor field that is considered, is given by Eq. (4.7). The data-driven terms can adapt for curvature and deviation from horizontality where the direction of the left-invariant frame deviates from the underlying line structure.

4.3.1 COMBINE OPTIMALLY STRAIGHT AND SHORT: A NEW DATA-DRIVEN VERSION ∇^U OF THE CARTAN CONNECTION

In previous works, the Cartan plus connection, which relies on the left-invariant frame, has been used to describe straight and shortest curves in Lie groups [84]. However, this frame is not always adequate in multi-orientation image processing as it does not always align perfectly with the underlying line structures in the orientation scores (see Fig. 4.2). To improve the tracking results, we, therefore, switch to using a data-driven Cartan connection associated with the data-driven metric tensor field \mathcal{G}^U given by Eq. (4.4). Let us first define what we mean by a *data-driven Cartan connection*.

Definition 4.3.1. *The data-driven Cartan connection and its corresponding dual are given by*

$$\nabla^U = \sum_{k=1}^n \left(\sum_{i=1}^n \omega_U^i \otimes (\mathcal{A}_i^U \circ \omega_U^k) + \sum_{j=1}^n (\omega_U^i \otimes \omega_U^j) \tilde{c}_{ij}^k \right) \mathcal{A}_k^U, \quad (4.15)$$

$$(\nabla^U)^* = \sum_{i=1}^n \left(\sum_{j=1}^n \omega_U^j \otimes (\mathcal{A}_j^U \circ \mathcal{A}_i^U) + \sum_{k,j=1}^n (\omega_U^j \otimes \mathcal{A}_k^U) \tilde{c}_{ij}^k \right) \omega_U^i, \quad (4.16)$$

where $(\nabla^U)_X^* \lambda := (\nabla^U)^*(X, \lambda)$ and $\nabla_X^U Y := \nabla^U(X, Y)$.

Remark 4.3.1. The relation between ∇ and its dual ∇^* is

$$\langle \nabla_X^* \lambda, Y \rangle := X \langle \lambda, Y \rangle - \langle \lambda, \nabla_X Y \rangle \quad (4.17)$$

for all vector fields X, Y and all covector fields λ on G , which may be interpreted as a product rule for the pairing between the vectors and co-vectors. In particular for $X = \mathcal{A}_i^U$, $Y = \mathcal{A}_j^U$ and $\lambda = \omega_U^k$ we obtain

$$\begin{aligned} \tilde{c}_{ji}^k &\stackrel{(4.16)}{=} \langle (\nabla^U)^*_{\mathcal{A}_i^U} \omega_U^k, \mathcal{A}_j^U \rangle \\ &\stackrel{(4.17)}{=} \mathcal{A}_i^U \langle \omega_U^k, \mathcal{A}_j^U \rangle - \langle \omega_U^k, \nabla_{\mathcal{A}_i^U}^U \mathcal{A}_j^U \rangle \\ &= \mathcal{A}_i^U (\delta_j^k) - \langle \omega_U^k, \nabla_{\mathcal{A}_i^U}^U \mathcal{A}_j^U \rangle \\ &= -\langle \omega_U^k, \nabla_{\mathcal{A}_i^U}^U \mathcal{A}_j^U \rangle \stackrel{(4.15)}{=} -\tilde{c}_{ij}^k. \end{aligned}$$

In the next lemma, we will express the data-driven Cartan connection and its corresponding dual explicitly in coordinates, which will provide us an expression on which we will build in the proof of our main theorem, Theorem 4.3.1.

Lemma 4.3.1. *When expressing Eqs. (4.15) and (4.16) more explicitly in data-driven left-invariant frame components (gauge frame components for short), one finds*

$$(\nabla^U)_X Y = \sum_{k=1}^n \left(\dot{\tilde{y}}^k + \sum_{i,j=1}^n \tilde{c}_{ij}^k(\cdot) \tilde{x}^i \tilde{y}^j \right) \mathcal{A}_k^U, \quad (4.18)$$

and for the dual connection

$$(\nabla^U)_X^* \lambda = \sum_{i=1}^n \left(\dot{\tilde{\lambda}}_i + \sum_{k,j=1}^n (\tilde{x}^j \tilde{\lambda}_k) \tilde{c}_{ij}^k(\cdot) \right) \omega_U^i, \quad (4.19)$$

where $X = \sum_{i=1}^n \tilde{x}^i \mathcal{A}_i^U|_\gamma$, $Y = \sum_{i=1}^n \tilde{y}^i \mathcal{A}_i^U|_\gamma$ and $\lambda = \sum_{i=1}^n \tilde{\lambda}_i \omega_U^i$, and where derivations of the components of Y and λ equal

$$\begin{aligned} \dot{\tilde{y}}^k(t) &:= \frac{d}{dt} \tilde{y}^k(\gamma(t)) = (X(\tilde{y}^k))(\gamma(t)), \\ \dot{\tilde{\lambda}}_i(t) &:= \frac{d}{dt} \tilde{\lambda}_i(\gamma(t)) = (X(\tilde{\lambda}_i))(\gamma(t)), \end{aligned}$$

along a flow-line² $\gamma : [0, 1] \rightarrow \mathbb{M}_2$ of smooth vector field X .

Proof. See Appendix A.1. □

4.3.2 MAIN THEOREMS

Our goal is to analyze and structure the Hamiltonian flow belonging to the new data-driven geometric model determined by a data-driven metric tensor field \mathcal{G}^U . For convenience, we restrict ourselves in our main theorem to the case where the homogeneous space equals a full finite-dimensional Lie group G as the base manifold. We are mainly interested in the case $G = \text{SE}(2) \equiv \mathbb{M}_2$ with $n = 3$ and with data-driven metric tensor field \mathcal{G}^U given by Eq. (4.4).

In geometric control curve optimization problems, the Hamiltonian flow is a powerful mechanism [6, 7, 132, 144, 146]. It typically allows us to analyze the behavior of *all* geodesics (and their momentum) together, see e.g., [144]. In the left-invariant (non-data-driven) setting, the Hamiltonian flows were characterized [84, Thm.1&2] via the plus Cartan connection, where shortest curves (geodesics) have parallel momentum. It has led to exact formulas [74, 81] for specific settings (uniform cost case in the left-invariant model \mathcal{G} given

² A curve γ satisfying $\dot{\gamma}(t) = X_{\gamma(t)}$.

in Eq. (1.19), $C = 1$ which was introduced on $\mathbb{M}_2 \equiv \text{SE}(2)$ by Citti and Sarti [56] and deeply analyzed by Sachkov [168]). Remarkably, the Cartan plus connection $\nabla^{[+]}$ has torsion resulting in different “straight curves” (i.e., those with parallel velocity) and “shortest curves” (i.e., those with parallel momentum), and it underlies many multi-orientation image analysis applications [84].

Before stating the main theoretical result, that generalizes [84, Thm.1&2] to *data-driven* metric tensor fields \mathcal{G}^U , we introduce the concepts of parallel momentum and parallel velocity. They are now determined by the data-driven Cartan connection ∇^U and its dual $(\nabla^U)^*$.

Definition 4.3.2 (Parallel momentum). *Let $\gamma : [0, 1] \rightarrow G$ be a smooth curve. Then, the curve $\gamma(\cdot)$ has ∇^U parallel momentum $\lambda(\cdot)$*

$$\Leftrightarrow \begin{cases} (\nabla^U)^* \lambda = 0 \\ \mathcal{G}^U \dot{\gamma} = \lambda. \end{cases} \quad (4.20)$$

Definition 4.3.3 (Parallel velocity). *Let $\gamma : [0, 1] \rightarrow G$ be a smooth curve. Then, the curve $\gamma(\cdot)$ has parallel velocity $\dot{\gamma}(\cdot)$ w.r.t. connection ∇^U*

$$\Leftrightarrow (\nabla^U)_{\dot{\gamma}} \dot{\gamma} = 0. \quad (4.21)$$

Remark 4.3.2. Using the antisymmetry of the structure functions (4.13) and (4.18) in Lemma 4.3.1 we can rewrite Eq. (4.21) to

$$(\nabla^U)_{\dot{\gamma}} \dot{\gamma} = 0 \Leftrightarrow \exists_{(c^1, \dots, c^n) \in \mathbb{R}^n} \text{constant s.t. } \dot{\gamma} = \sum_{i=1}^n c^i A_i^U|_{\gamma}.$$

Next, we formulate the main theoretical results where we generalize the main results [84, Thm.1&2] from the left-invariant setting (G, \mathcal{G}) with connection $\nabla^{[+]}$, to the new data-driven geometric models (G, \mathcal{G}^U) with connection ∇^U . In more detail, the next theorem shows

1. how to compute globally optimal distance minimizers in a geometry that relies on data-driven left-invariant frames: These geodesics have parallel momentum w.r.t. connection ∇^U (Definition 4.3.2).
2. that the locally optimal straight curves are the straight curves w.r.t. connection ∇^U : These curves have parallel velocity (i.e., are auto-parallel) w.r.t. ∇^U (Definition 4.3.3).

Note that the equation for geodesics of the new data-driven model $(\mathbb{M}_2, \mathcal{G}^U)$ gives a wild expression in the left-invariant frame. In the fixed frame it is even worse. However, our new tool of the connection ∇^U expressed in the locally

adaptive frame ω_i^U allows us to describe these geodesic equations (and the Hamiltonian flow) concisely and intuitively by the next theorem, using the tools listed in Table 4.2.

Theorem 4.3.1 (Straight and shortest curves: parallel velocity and momentum w.r.t. connection ∇^U). *In a Riemannian manifold (G, \mathcal{G}^U) with data-driven left-invariant metric tensor field \mathcal{G}^U satisfying Eq. (4.3), and with induced Riemannian metric $d_{\mathcal{G}^U}$ Eq. (4.9), we have:*

- γ is a straight curve with respect to $\nabla^U \stackrel{\text{def.}}{\Leftrightarrow} \nabla_{\dot{\gamma}}^U \dot{\gamma} = 0$

$$\Leftrightarrow \exists (c^1, \dots, c^n) \in \mathbb{R}^n \text{ constant s.t. } \dot{\gamma} = \sum_{i=1}^n c^i \mathcal{A}_i^U|_{\gamma}.$$

- γ is a shortest curve with respect to $\nabla^U \Rightarrow$ the curve-momentum pair $v = (\gamma, \lambda) : [0, 1] \rightarrow T^*(G)$ satisfies the Hamiltonian flow

$$\dot{v} = \vec{\mathfrak{h}}(v) \Leftrightarrow \begin{cases} (\nabla^U)^*_\dot{\gamma} \lambda = 0 \\ \mathcal{G}^U \dot{\gamma} = \lambda, \end{cases} \quad (4.22)$$

where one has the following pull-back symmetry³ of the data-driven Cartan connection

$$(L_q)^*(\nabla^{\mathcal{L}_q U})^* = (\nabla^U)^* \text{ for all } q \in G, \quad (4.23)$$

with left actions L and \mathcal{L} given by Eqs. (1.2) and (1.18) respectively.

The shortest curve $\gamma : [0, 1] \rightarrow G$ with $\gamma(0) = g$ and $\gamma(1) = g_0$ may be computed by steepest descent backtracking on the distance map $W(g) = d_{\mathcal{G}^U}(g, g_0)$

$$\gamma(t) := \gamma_{g, g_0}^U(t) = \text{Exp}_g(t v(W)) \quad t \in [0, 1], \quad (4.24)$$

where Exp integrates the following vector field on G :

$v(W) := -W(g)(\mathcal{G}^U)^{-1} dW = -W(g) \sum_{k=1}^n |\alpha_k|^{-1} \mathcal{A}_k^U(W) \mathcal{A}_k^U$ and where W is the viscosity solution of the eikonal PDE system

$$\begin{cases} \|\text{grad}_{\mathcal{G}^U} W\| = 1, \\ W(g_0) = 0, \end{cases} \quad (4.25)$$

where we assume g is neither a 1st Maxwell point nor a conjugate point. As $v(W)$ is data-driven left-invariant, the geodesics carry the symmetry

$$\underline{\gamma_{qg, qg_0}^U(t) = q \gamma_{g, g_0}^U(t) \text{ for all } q, g, g_0 \in G, t \in [0, 1].} \quad (4.26)$$

³ For the definition of pullback of a dual connection, see Remark A.2.4 in Appendix A.2.

Proof. See Appendix A.2. □

Remark 4.3.3 (Assumptions on point g in backtracking (4.24)). For the geodesic backtracking formulated above, we need a differentiable distance map along the path and a well-posed Hamiltonian flow (i.e., the mapping from $\nu(0)$ to $\nu(t)$ arising from Eq. (4.22) must have a non-vanishing Jacobian) along the path. If g would be a first Maxwell point (i.e., two distinct geodesics meet for the first time at g) the distance map is not differentiable at g . If g would be a conjugate point (often limits of first Maxwell points [19]) then the Hamiltonian flow breaks down. In the latter case, local optimality is lost. In the first case, global optimality is lost. Fortunately the viscosity property of the viscosity solution [90] of Eq. (4.25) kills non-optimal fronts [19] and one may resort to multi-valued backtracking via sub-gradient backtracking.

The following three remarks illustrate how the Hamiltonian flow, the eikonal PDE, and the backtracking of geodesics simplify when expressed in the data-driven left-invariant frame.

Remark 4.3.4. Eq. (4.20) is in gauge frame components simply:

$$\begin{cases} \dot{\tilde{\gamma}}^i = \tilde{\lambda}^i & i = 1, \dots, n \\ \dot{\tilde{\lambda}}^i = \sum_{j=1}^n \sum_{k=1}^n \tilde{c}_{ji}^k(\gamma(\cdot)) \tilde{\lambda}_k \tilde{\lambda}^j & i = 1, \dots, n. \end{cases}$$

Remark 4.3.5. Eq. (4.25) is in gauge frame components simply:

$$\begin{cases} \sum_{j=1}^n \alpha_j^U(\cdot)^{-1} (\mathcal{A}_j^U W)^2 = 1 \\ W(\gamma(0)) = 0. \end{cases}$$

Remark 4.3.6. Eq. (4.24) is in gauge frame components simply:

$$\dot{\tilde{\gamma}}^k = \frac{1}{W(g)} |\alpha_k^U|^{-1} (\mathcal{A}_k^U W)(\tilde{\gamma}), \quad k = 1, \dots, n. \quad (4.27)$$

This explains the definition of $\nu(W)$ below Eq. (4.24). A more explicit integration formula for Eq. (4.24) can be obtained in a similar way as in [74, 81] (where exact solutions are derived for $C = U = 1$) via momentum preservation laws.

4.3.3 ASYMMETRIC FINSLER EXTENSION TO AUTOMATICALLY DEAL WITH BIFURCATIONS

One can always decide to include a positive control variant, to avoid cusps in the spatial projection of geodesics in $G = \text{SE}(2)$. This is done by considering

the geodesics in the asymmetric Finslerian manifold $(\mathbb{M}_2, \mathcal{F}^U)$, recall Eq. (4.4), rather than the geodesics in the Riemannian manifold $(\mathbb{M}_2, \mathcal{G}^U)$.

In practice, a slight modification of the eikonal PDE (taking the positive part of one momentum component) will guarantee that all optimal geodesic wave-fronts propagate in the preferred forward direction around the source point, as can be observed in Fig. 4.1 where the fronts initially move “down-left” (and not “up-right”).

The asymmetric Finslerian model $(\mathbb{M}_2, \mathcal{F}^U)$ is still well-posed (controllable and piecewise regular geodesics) even if $\zeta \downarrow 0$. In fact, such asymmetric Finslerian geodesics are by construction piecewise concatenations of Riemannian geodesics and in-place rotations. These observations follow by a straightforward generalization of [83, Thm.1, 2, 4] to the data-driven setting, where the control set formulation of the geodesic distances, still applies:

$$d_{\mathcal{F}^U}(\mathbf{p}, \mathbf{q}) = \inf\{T \geq 0 \mid \exists \gamma \in \Gamma_T, \gamma(0) = \mathbf{p}, \gamma(T) = \mathbf{q}, \dot{\gamma} \in \mathcal{B}_{\mathcal{F}^U}(\gamma)\} \quad (4.28)$$

where $\Gamma_T := PC([0, T], \mathbb{M}_2)$ piecewise continuous curves in \mathbb{M}_2 . Moreover, the field of control sets given by $\mathbf{p} \mapsto \mathcal{B}_{\mathcal{F}^U}(\mathbf{p})$ recall Eq. (1.13) remains continuous when using \mathcal{F}^U or \mathcal{G}^U (instead of \mathcal{F} or \mathcal{G}), which directly follows by [83, Lemma 6] in conjunction with the important coercivity property (4.8).

The nice thing in practice is that in-place rotations are automatically placed at optimal locations by the eikonal PDE system (solved by the anisotropic fast-marching algorithm that we discuss in the next section). It is not surprising that, when using a reasonable cost function C (see Figs. 4.8 and 4.9), these in-place rotations are automatically placed at bifurcations in complex vascular trees as can be observed in the upcoming Fig. 4.12.

4.4 NUMERICAL SOLUTIONS TO THE EIKONAL PDE SYSTEM: EXTENSION OF THE ANISOTROPIC FAST-MARCHING ALGORITHM THAT ALLOWS FOR ALL LEFT-INVARIANT DATA-DRIVEN METRICS

In this section we describe the computation of globally minimizing geodesics for the new models \mathcal{F}^U considered in this chapter, whose fundamental ingredient is the numerical solution to an anisotropic eikonal PDE. The *Reeds-Shepp forward* optimal control model \mathcal{F} , of which a variant \mathcal{F}^U is considered in this chapter, has already been addressed numerically using a Semi-Lagrangian [83] and Eulerian [137] discretization of the corresponding eikonal PDE. Both works however take advantage of the fact that the original geodesic model \mathcal{F} regards the tangent spaces to the physical \mathbb{R}^2 and the angular S^1 domains as orthogonal to each other. However, in our case of interest (with model \mathcal{F}^U given by Eq. (4.4)), we cannot expect such a block-matrix structure in the fixed coordinate system (x, y, θ) .

To overcome this problem, we describe below the extension of [137] to the adaptive frame setting considered here, where this orthogonality relation is lost; in contrast, [83] could not be generalized in an efficient manner.

Remark 4.4.1 (Convenience notations for the numerical section). Throughout this section, we label the dependence w.r.t. the relaxation parameter $\varepsilon \in (0, 1]$, so as to analyze it more easily, and to investigate the limit $\varepsilon \rightarrow 0$. In contrast, we often drop the dependence w.r.t. the data U , which is regarded as fixed.

We also take advantage of the fact that the manifold $\mathbb{M}_2 \equiv \mathbb{R}^2 \times \mathbb{R}/(2\pi\mathbb{Z})$ has a trivial tangent bundle: $T_p(\mathbb{M}_2) \equiv \mathbb{R}^2 \times \mathbb{R} \equiv \mathbb{R}^3$ canonically for any $p \in \mathbb{M}_2$, and likewise $T_p^*(\mathbb{M}_2) \equiv \mathbb{R}^3$. As a result, by identifying co-vectors and vectors by their components in \mathbb{R}^3 , the functional brackets $\langle \cdot, \cdot \rangle$ below boil down to the ordinary dot product on \mathbb{R}^3 . Similarly, the tensor product $\omega \otimes \omega$ boils down to $\omega\omega^\top$.

4.4.1 ASYMMETRIC QUADRATIC EIKONAL PDE

The Reeds-Shepp *forward* model, is defined via a sub-Finslerian quasi-metric⁴, relaxed by a small parameter $\varepsilon > 0$, recall \mathcal{F} was given by Eq. (1.20) and its data-driven version \mathcal{F}^U was given by Eq. (4.4). Throughout this section, and in our vessel tracking experiments, we are concerned with the case where lateral motions and backward motions become very expensive: we set spatial anisotropy parameter $\zeta = \varepsilon$ with $0 < \varepsilon \ll 1$ in the Finsler norm \mathcal{F}^U given by Eq. (4.4).

The generic form of the data-driven Finsler metric function considered in Eq. (4.4) reads:

$$\mathcal{F}_\varepsilon(p, \dot{p})^2 = C(p)^2 \cdot (\langle \dot{p}, M^0(p)\dot{p} \rangle + \varepsilon^{-2} \langle \omega^2(p), \dot{p} \rangle^2 + \varepsilon^{-2} \langle \omega^1(p), \dot{p} \rangle_-^2), \quad (4.29)$$

for any point $p \in \overline{\Omega}$, within a given bounded connected domain $\Omega \subset \mathbb{M}_2$, and any tangent vector $\dot{p} \in T_p(\mathbb{M}_2) \equiv \mathbb{R}^3$, and where the two small parameters ε, ζ relate via

$$\varepsilon^{-2} := (\varepsilon^{-2} - 1)\zeta^2. \quad (4.30)$$

In the following analysis, we only use the property that the tensor field M^0 is pointwise positive definite, that the differential forms ω^1 and ω^2 are pointwise linearly independent, and that $M^0 : \overline{\Omega} \rightarrow S_3^{++}$, and $\omega^1, \omega^2 : \overline{\Omega} \rightarrow \mathbb{R}^3$ (following the conventions of Remark 4.4.1) have Lipschitz regularity. Here S_3^{++} denotes the space of 3×3 real symmetric positive definite matrices.

⁴ I.e., the cost associated with a velocity at a point is non-Riemannian, non-symmetric, and possibly infinite.

Remark 4.4.2. In the normal left-invariant setting $\mathcal{F}^{U=1} = \mathcal{F}$ the asymmetric metric expressed in the fixed frame $(\dot{x}, \dot{y}, \dot{\theta})$, of the tangent space at any coordinates (x, y, θ) , has a block-diagonal structure. In contrast the data-driven metrics $\mathcal{G}^U, \mathcal{F}^U$, in general, does *not* have a block-matrix structure, as the independent elements $\{\omega_U^i\}_{i=1}^3$ may point anywhere due to their data-driven nature, as can be seen in Fig. 4.2, keeping in mind the duality (4.10). Therefore, we must introduce a new modification of the anisotropic fast-marching algorithm.

The second term in Eq. (4.29) penalizes lateral motion, while the third term discourages reverse gear motion; both are excluded in the genuine Reeds-Shepp forward car model obtained in the limit (akin to [83, Thm.2]) as $\varepsilon \rightarrow 0$, which is non-holonomic.

The distance map $W : \overline{\Omega} \rightarrow \mathbb{R}$ from a given point $\mathbf{p}_0 \in \Omega$ and w.r.t. the Finsler function \mathcal{F}_ε , is the unique viscosity solution to the following anisotropic eikonal system

$$\begin{cases} \mathcal{F}_\varepsilon^*(\mathbf{p}, dW(\mathbf{p})) = 1, & \mathbf{p} \in \mathbb{M}_2 \\ W(\mathbf{p}_0) = 0 \end{cases} \quad (4.31)$$

where the dual Finsler function is defined as

$$\mathcal{F}_\varepsilon^*(\mathbf{p}, \hat{\mathbf{p}}) := \sup\{\langle \hat{\mathbf{p}}, \dot{\mathbf{p}} \rangle \mid \dot{\mathbf{p}} \in T_{\mathbf{p}}(\mathbb{M}_2) \text{ and } \mathcal{F}_\varepsilon(\mathbf{p}, \dot{\mathbf{p}}) \leq 1\},$$

with $\hat{\mathbf{p}} \in T_{\mathbf{p}}^*(\mathbb{M}_2)$.

The structure of the metric in Eq. (4.31), referred to as *asymmetric quadratic*, allows us to compute a closed form expression of the dual metric $\mathcal{F}_\varepsilon^*$, and thereby the eikonal PDE (4.31), as we will see below.

Lemma 4.4.1. *Let $M \in S_3^{++}$ and $\omega \in \mathbb{R}^3$, and define*

$$\mathcal{F}(\mathbf{p}, \dot{\mathbf{p}})^2 := \langle \dot{\mathbf{p}}, M \dot{\mathbf{p}} \rangle + \langle \omega, \dot{\mathbf{p}} \rangle_-^2.$$

Then \mathcal{F} is a quasi-norm (i.e., an asymmetric norm), whose dual quasi-norm reads for all $\hat{\mathbf{p}} \in \mathbb{R}^3$

$$\mathcal{F}^*(\mathbf{p}, \hat{\mathbf{p}})^2 = \langle \hat{\mathbf{p}}, D\hat{\mathbf{p}} \rangle + \langle \hat{\mathbf{p}}, \eta \rangle_+^2, \quad (4.32)$$

with $D := (M + \omega \omega^\top)^{-1}$ and $\eta := M^{-1}\omega / \sqrt{1 + \omega^\top M^{-1}\omega}$.

Proof. See [83, Lemma 4]. □

For concreteness, we apply Lemma 4.4.1 to our Finsler function \mathcal{F}_ε of interest (4.31), defined pointwise by the parameters

$$M_\varepsilon := M^0 + \varepsilon^{-2}\omega^2(\omega^2)^\top, \text{ and } \omega_\varepsilon := \varepsilon^{-1}\omega^1. \quad (4.33)$$

This then results in the dual Finsler functions described in Lemma 4.4.2.

Lemma 4.4.2 (Dual Finsler functions). *With our choice (4.29) of Finsler function \mathcal{F}_ϵ used in Eq. (4.31), the dual Finsler function \mathcal{F}_ϵ^* is given for all $\hat{\mathbf{p}} \in T_p^*(\mathbb{M}_2) \equiv \mathbb{R}^3$ by*

$$\mathcal{F}_\epsilon^*(\mathbf{p}, \hat{\mathbf{p}})^2 = \langle \hat{\mathbf{p}}, D_\epsilon \hat{\mathbf{p}} \rangle + \langle \hat{\mathbf{p}}, \eta_\epsilon \rangle_+^2, \text{ with} \quad (4.34)$$

$$D_\epsilon = \frac{\mathcal{A} \mathcal{A}^\top}{\mathcal{A}^\top M^0 \mathcal{A}} + O(\epsilon^2), \quad (4.35)$$

$$\eta_\epsilon = \frac{M^{-1}(\omega_U^1 - \alpha \omega_U^2)}{\sqrt{(\omega_U^1)^\top M^{-1}(\omega_U^1 - \alpha \omega_U^2)}} + O(\epsilon^2), \quad (4.36)$$

where we used shorthand notation $M^{-1} := (M^0)^{-1}$, the cross product $\mathcal{A} := \omega_U^1 \times \omega_U^2$, and the orthogonalization coefficient $\alpha := (\omega_U^2)^\top M^{-1} \omega_U^1 / (\omega_U^2)^\top M^{-1} \omega_U^2$.

Proof. Follows by Lemma 4.4.1 and Taylor expansion, for details, see Appendix A.3. \square

Note that by Eq. (4.30), $O(\epsilon^2) = O(\epsilon^2)$ for small values of ϵ .

Lemma 4.4.2 shows that one can define an ideal sub-Finsler function \mathcal{F}_0^* that arises in the limiting case $\epsilon \downarrow 0$, and that

$$\mathcal{F}_\epsilon^*(\mathbf{p}, \hat{\mathbf{p}}) = \mathcal{F}_0^*(\mathbf{p}, \hat{\mathbf{p}}) + O(\epsilon^2). \quad (4.37)$$

Our goal, achieved in Sections 4.4.1 and 4.4.2 below, is to design a numerical scheme that is consistent with the sub-Finslerian eikonal PDE $\mathcal{F}_0^*(\mathbf{p}, dW(\mathbf{p})) = 1$, and which satisfies the conditions that make the fast-marching algorithm applicable.

4.4.2 DISCRETIZATION AND CONSISTENCY

We discretize the eikonal PDE (4.31), which has an asymmetric quadratic structure (4.32), using adaptive finite differences on the Cartesian grid $\Omega_h := \Omega \cap h\mathbb{Z}^3$ of the domain Ω , where $h > 0$ denotes the grid scale. Note that $2\pi/h$ must be an integer, for consistency with the periodic boundary conditions in the angular coordinate. The numerical scheme construction relies on Selling's decomposition of positive definite matrices [178] and on a corollary related to the approximation of the squared positive part of a linear form. The versions of these results presented in [137, Corollary 4.12, Corollary 4.13] are gathered in the following proposition.

We denote $\mu(D) := \sqrt{\|D\| \|D^{-1}\|}$, for any $D \in S_3^{++}$, and $a_+ := \max\{0, a\}$, for any $a \in \mathbb{R}$.

Proposition 4.4.1 (Selling matrix decomposition, see [137]). *For any $D \in S_3^{++}$, there exists $\dot{\mathbf{e}}_1, \dots, \dot{\mathbf{e}}_I \in \mathbb{Z}^3$ and $\lambda_1, \dots, \lambda_I \geq 0$, such that for all $\hat{\mathbf{p}} \in \mathbb{R}^3$*

$$\langle \hat{\mathbf{p}}, D\hat{\mathbf{p}} \rangle = \sum_{1 \leq i \leq I} \lambda_i \langle \hat{\mathbf{p}}, \dot{\mathbf{e}}_i \rangle^2.$$

For any $\eta \in \mathbb{R}^3$, $\epsilon > 0$, there exists $\dot{\mathbf{f}}_1, \dots, \dot{\mathbf{f}}_I \in \mathbb{Z}^3$ and $\mu_1, \dots, \mu_I \geq 0$, such that for all $\hat{\mathbf{p}} \in \mathbb{R}^3$

$$\langle \hat{\mathbf{p}}, \eta \rangle_+^2 \leq \sum_{1 \leq i \leq I} \mu_i \langle \hat{\mathbf{p}}, \dot{\mathbf{f}}_i \rangle_+^2 \leq \langle \hat{\mathbf{p}}, \eta \rangle_+^2 + \epsilon^2 (\|\hat{\mathbf{p}}\|^2 \|\eta\|^2 - \langle \hat{\mathbf{p}}, \eta \rangle^2).$$

In addition $\|\dot{\mathbf{e}}_i\|, \dots, \|\dot{\mathbf{e}}_I\| \leq C\mu(D)$ and $\|\dot{\mathbf{f}}_i\|, \dots, \|\dot{\mathbf{f}}_I\| \leq C\epsilon^{-1}$. The above holds with the constants $I := 6$, $C := 4\sqrt{3}$.

Remark 4.4.3. A key aspect of Proposition 4.4.1 is that the vectors $(\dot{\mathbf{e}}_i)$ and $(\dot{\mathbf{f}}_j)$ have *integer* coordinates, hence we avoid (off-grid) interpolations in our discretization.

Proposition 4.4.1 suggests the following discretization of $\mathcal{F}^*(dW(\mathbf{p}))$, as in Eq. (4.32), for arbitrary $D \in S_3^{++}$, $\eta \in \mathbb{R}^3$, $\epsilon > 0$:

$$\begin{aligned} & \sum_{1 \leq i \leq I} \lambda_i \max \left\{ 0, \frac{W(\mathbf{p}) - W(\mathbf{p} - h\dot{\mathbf{e}}_i)}{h}, \frac{W(\mathbf{p}) - W(\mathbf{p} + h\dot{\mathbf{e}}_i)}{h} \right\}^2 \\ & + \sum_{1 \leq j \leq I} \mu_j \max \left\{ 0, \frac{W(\mathbf{p}) - W(\mathbf{p} - h\dot{\mathbf{f}}_j)}{h} \right\}^2 =: \mathfrak{F}W(\mathbf{p}), \end{aligned} \quad (4.38)$$

with suitable boundary conditions. This numerical scheme falls within the Hamiltonian fast-marching framework [140], and thus can be efficiently solved numerically, see Section 4.4.3. Before that, we discuss its consistency with the eikonal equation: inserting a first order Taylor expansion in Eq. (4.38), we obtain (using Proposition 4.4.1):

$$\mathfrak{F}W(\mathbf{p}) = \langle \nabla W(\mathbf{p}), D\nabla W(\mathbf{p}) \rangle + \langle \nabla W(\mathbf{p}), \dot{\eta} \rangle_+^2 + \mathcal{O}(Rh + \epsilon^2), \quad (4.39)$$

where $R := \max\{\mu(D), \epsilon^{-1}\} > 0$ denotes the stencil radius.

Next, we analyze the approximation error towards the ideal model as $\epsilon \rightarrow 0$ and $h \rightarrow 0$ suitably. To this end we denote by \mathfrak{F}_ϵ^h the finite differences scheme on Ω_h associated with the parameters D_ϵ and η_ϵ of our application (4.34). Note that the stencil radius is $R_\epsilon = \max\{\mu(D_\epsilon), \epsilon^{-1}\} = \mathcal{O}(\epsilon^{-1})$, since $\mu(D_\epsilon) = \mu(M_\epsilon) = \mathcal{O}(\epsilon^{-1})$ in view of Eq. (4.33). Now combining Eq. (4.37) with Eq. (4.39), we obtain the overall consistency error

$$\begin{cases} \mathfrak{F}_\epsilon^h W(\mathbf{p}) = \mathcal{F}_0^*(\mathbf{p}, dW(\mathbf{p})) + \mathcal{O}(\epsilon^{-1}h + \epsilon^2) = 1, \\ \mathfrak{F}_\epsilon^h W(\mathbf{p}_0) = 0. \end{cases} \quad (4.40)$$

The optimal order of the consistency error $\mathcal{O}(h^{\frac{2}{3}})$ is achieved with the scaling $\epsilon = h^{\frac{1}{3}}$. In our practical experiments however, we rather use the small fixed value $\zeta = \epsilon = 0.1$ which by Eq. (4.40) indeed yields a sufficiently accurate scheme [137].

4.4.3 ANISOTROPIC FAST-MARCHING FOR COMPUTING DISTANCE MAPS OF DATA-DRIVEN LEFT-INVARIANT FINSLER MODELS

In fast-marching methods (FMM), grid points are categorized into three sets: Far, Trial, and Accepted. In each step of the FMM, one selects the trial point \mathbf{p} whose function value $W(\mathbf{p})$ is minimal. The point \mathbf{p} is moved into the accepted set, and $W(\mathbf{p})$ is frozen. In contrast, all the trial or far points whose stencil contains \mathbf{p} see their function value *updated*, and they are moved into the trial set. This procedure generalizes the classical FMM [179]; for details see [47]. When all points have moved to the accepted category, the FMM terminates, and a geodesic can be easily calculated by steepest descent as described in Section 4.4.4.

To update of a single function value $W(\mathbf{p})$, we isolate it in the numerical scheme (4.38), and express it by the values of its neighbors so as to obey $\mathfrak{F}W(\mathbf{p}) = 1$. The latter equation admits by construction a unique solution, which is obtained as the largest root of a quadratic equation.

There are two crucial properties of the discretization \mathfrak{F} :

- Discrete Degenerate ellipticity:

$\mathfrak{F}W(\mathbf{p})$ is a non-decreasing function of the finite differences in Eq. (4.38).

- Causality:

$\mathfrak{F}W(\mathbf{p})$ only depends on the positive part of all finite differences in Eq. (4.38).

These are the two key⁵ assumptions of [138, Theorem 2.3], implying that *the discretized PDE (4.40) admits a unique solution W_h^ϵ , which is computable in a single pass over the discretization domain Ω_h , using anisotropic fast-marching*.

Following the steps of the proof [137] associated with the standard Reeds-Shepp forward model, and with straightforward adaptations, we obtain that W_h^ϵ converges uniformly as $\epsilon \rightarrow 0$ and $h/\epsilon \rightarrow 0$ to the solution W of the sub-Finslerian Eikonal PDE $\mathcal{F}_0^*(\mathbf{p}, dW(\mathbf{p})) = 1$.

⁵ There are two other minor assumptions, the existence of a sub- and super-solution to the scheme, and a perturbation property, which can be established analogously to the Riemannian case in [138, Proposition 2.5].

4.4.4 STEEPEST DESCENT FOR THE FINSLERIAN GEODESICS

In previous work [83, Thm.4], standard Riemannian steepest descent tracking on the distance map W , recall Eq. (4.24) in Theorem 4.3.1, is generalized from the symmetric Riemannian setting to the (possibly asymmetric) Finsler model setting. That idea also transfers to the new data-driven left-invariant model as the Finslerian back-tracking [83, App.B] still applies. Steepest descent tracking from \mathbf{p} to source point \mathbf{p}_0 , as described in Eq. (4.24), becomes (using the embedding of $\mathbb{M}_2 \subset \mathbb{R}^3$)

$$\gamma(t) = \mathbf{p} - \frac{1}{W(\mathbf{p})} \int_0^t d\mathcal{F}_\epsilon^*(\gamma(s), dW(\gamma(s))) ds, \quad t \in [0, 1], \quad (4.41)$$

with $0 < \epsilon \ll 1$ and where the derivative is taken with respect to the second entry of the dual Finsler function, and where W is the viscosity solution of the eikonal PDE system (4.31).

In the practical implementations we use a second order Runge-Kutta method for time integration approximations and at time $t = 1$ we arrive at the source-point \mathbf{p}_0 .

This geodesic backtracking in $(\mathbb{M}_2, \mathcal{F}^U)$ again boils down to piecewise concatenations of

1. symmetric Riemannian geodesics in \mathbb{M}_2 with metric tensor field $\mathcal{G}_{\mathbf{p}}^U(\dot{\mathbf{p}}, \dot{\mathbf{p}})$ recall Eq. (4.4), and
2. symmetric Riemannian geodesics in \mathbb{M}_2 with metric tensor field $\mathcal{G}_{\mathbf{p}}^U(\dot{\mathbf{p}}, \dot{\mathbf{p}}) + (\epsilon^{-2} - 1)|\omega_U^1(\dot{\mathbf{p}})|^2$ that are in-place rotations, at locations where $\mathcal{A}_1^U \approx \mathcal{A}_1$ if $0 < \epsilon \ll 1$.

Remark 4.4.4. Taking the negative part of the final term in Eq. (4.29) means that we switch between two Riemannian manifolds (one with the final term active and with the final term non-active). At locations where $\omega_U^3 \approx \omega^3$, for instance at locations where $\mathcal{A}_1^U \approx \mathcal{A}_1$ this means that one switches between anisotropic geodesics and spherical geodesics (in-place rotations). In the vessel tracking applications we want such in-place rotations to appear above bifurcations in the vasculature.

A closely related situation is discussed in [83, Thm.4], but now it is applied to the new data-driven model \mathcal{F}^U (4.4) with dual $(\mathcal{F}^U)^* = \lim_{\epsilon \downarrow 0} \mathcal{F}_\epsilon^*$.

By Theorem 4.3.1 the Riemannian geodesics have parallel momentum and their transition towards spherical geodesics is like C continuously differentiable. The surface \mathcal{S} where Finslerian geodesics of \mathcal{F}^U in \mathbb{M}_2 switch from one Riemannian manifold to the other is given by

$$\mathcal{S} = \{\mathbf{p} \in \mathbb{M}_2 \mid \omega_U^1(\nabla W(\mathbf{p})) = (\mathcal{A}_1^U W)(\mathbf{p}) = 0\}.$$



Figure 4.4: **Influence of *data-driven* metric tensor fields:** (left) Tracking with the vanilla left-invariant metric tensor field from Eq. (1.19). (right) Tracking with the proposed *data-driven* left-invariant metric tensor field from Eq. (4.4). To isolate its effect in the tracking process and record the effect of only directional adaptation of the underlying metric, we have set the cost function $C = 1$. We observe that the *data-driven* nature of our model allows for a better fidelity to the underlying vascular structure. The parameters are given by $g_{11} = 0.01, g_{22} = 1, g_{33} = 1, \lambda = 100$.

Interestingly, in the mixed-model \mathcal{F}^M that we will explain later in Section 4.5, the condition in Remark 4.4.4 above is satisfied at bifurcations. Then in-place rotations are indeed automatically placed at the bifurcations in the tracking of blood vessels, as can be seen in Fig. 4.12.

4.5 EXPERIMENTS

We choose the data-driven left-invariant metric tensor field with forward gear \mathcal{F}^U as given in Eq. (4.4). An elaboration on the calculation of the cost function can be found in Appendix C. We will discuss the new model’s ability to adapt for curvature. Additionally, we show and discuss some full vascular tree tracking results.

4.5.1 CURVATURE ADAPTATION

The data-driven left-invariant metric tensor field \mathcal{G}^U and its asymmetric variant \mathcal{F}^U both consist of a “standard” left-invariant metric tensor field to which a data-driven term is added, as introduced in Eq. (4.4). The idea of the second data-driven term in this equation is that it pushes the main direction of the model into the direction of the underlying vessel, as illustrated in Fig. 4.4, where no data-dependent cost function $C = 1$ was used to generate the tracking result. We see that the data-driven left-invariant metric tensor field takes the image data into account and steers the tracking in the direction of the underlying vascular structure, even when the cost function does not contain information about vessel locations and curvature.

The data-driven term significantly improves tracking results for highly tortuous vascular structures, as we see in Fig. 4.5. In Fig. 4.5a, the tracking results relying on (solely) the left-invariant metric tensor field \mathcal{F} fail to follow the un-

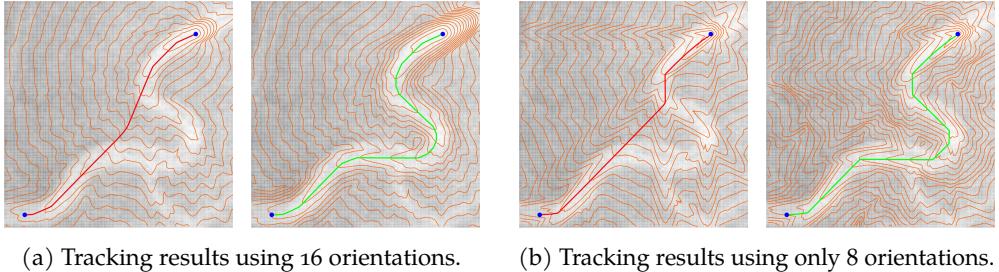


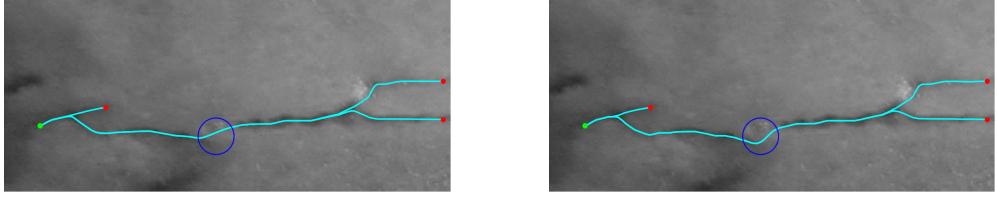
Figure 4.5: Comparison tracking results of left-invariant and data-driven left-invariant metric tensor field: Tracking results for left-invariant metric tensor field \mathcal{F} (left) and data-driven left-invariant metric tensor field \mathcal{F}^U (right). The parameters for the (data-driven) left-invariant metric tensor field are given by $g_{11} = 0.01, g_{22} = 0.16, g_{33} = 1, \lambda = 100$. The cost function is given by $C = 1/(1 + 200|U_f|^2)$. We see that the iso-contours of the data-driven metric tensor field follow the vessel structure better, and the tracking is correct (even with 8 orientations).

derlying vessel correctly, contrary to the new data-driven left-invariant model \mathcal{F}^U (4.4) which follows the vascular structure correctly. In addition, one sees that when using the left-invariant model, the wave fronts (indicated in orange) suffer from the discretization. In the data-driven left-invariant model, these discretization artifacts are gone, and the wavefronts follow the underlying vascular structure correctly. When only considering 8 orientations, as in Fig. 4.5b, the data-driven left-invariant frame is still able to follow the vascular structure correctly. Although the discretization is clearly visible in the tracking results, the data-driven left-invariant metric tensor field is still able to follow the vessel correctly. It is important to note that both models use the same cost function. The differences in the tracking results are due to the data-driven left-invariant metric tensor fields' ability to better adapt for:

1. gradual curvature change of blood vessels. The same applies to other applications such as the detection of cracks, see Fig. 4.6,
2. orientation biases by orientation score data- alignment as depicted in Fig. 4.2.

4.5.2 COMPLETE VASCULAR TREE TRACKING

In the previous section, we discussed the curvature adaptation feature of the new (asymmetric) data-driven left-invariant metric tensor field \mathcal{F}^U . This model also can automatically place “in-place” rotations in globally optimal geodesics which are typically placed at bifurcations.



(a) Tracking results with the left-invariant metric tensor field.

(b) Tracking results with the data-driven left-invariant metric tensor field with $\lambda = 100$.

Figure 4.6: Application in crack detection: Tracking results for the left-invariant and data-driven left-invariant metric tensor field on an image of cracks in a building. The presented results are calculated using 32 orientations, and parameter settings $g_{11} = 0.01$, $g_{22} = 1$, $g_{33} = 1$. In regions with high curvature, the data-driven model adapts more gradually for curvature to get more data evidence than the left-invariant model which tends to prefer in-place rotations.

However, these valuable abilities of the model may also lead to extremely complex structures to some undesirable behavior. In full vascular tree tracking, we see that the data-driven term may steer the tracking away from the actual vessel at crossings in extreme cases, as can be seen in Fig. 4.7.

To address this issue (see item c in Fig. 4.7) while retaining the benefits of improved data-adaptation (like in Figs. 4.1, 4.4 and 4.5), we introduce a (new) mixed model that prevents this undesirable behavior at (extreme) complex structures, where it locally relies on the old model. Then in between (extreme) complex structures we still benefit from the directional data adaptation in the orientation score.

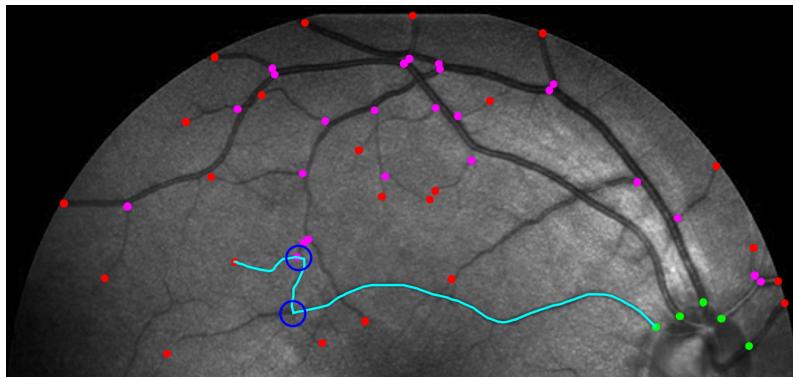
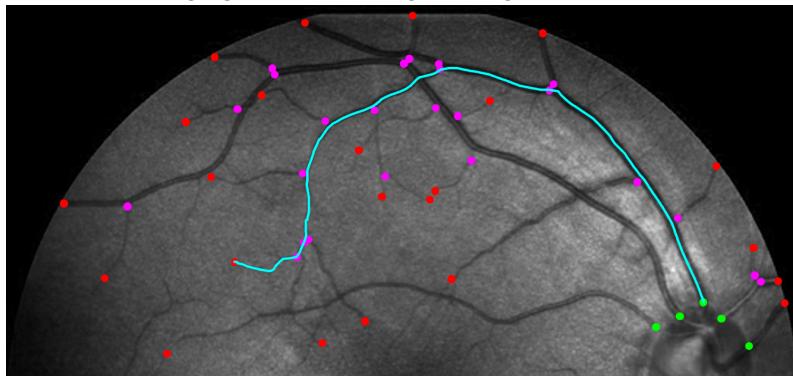
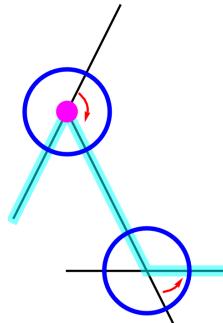
The mixed metric tensor field \mathcal{G}^M (and its asymmetric version \mathcal{F}^M) is given by the data-driven left-invariant metric tensor field away from the crossing structures, and by the left-invariant metric tensor field otherwise:

$$\begin{aligned}\mathcal{G}_p^M(\hat{\mathbf{p}}, \dot{\mathbf{p}}) &= \kappa(\mathbf{x}) \mathcal{G}_p(\hat{\mathbf{p}}, \dot{\mathbf{p}}) + (1 - \kappa(\mathbf{x})) \mathcal{G}_p^U(\hat{\mathbf{p}}, \dot{\mathbf{p}}), \\ |\mathcal{F}^M(\mathbf{p}, \dot{\mathbf{p}})|^2 &= \kappa(\mathbf{x}) |\mathcal{F}(\mathbf{p}, \dot{\mathbf{p}})|^2 + (1 - \kappa(\mathbf{x})) |\mathcal{F}^U(\mathbf{p}, \dot{\mathbf{p}})|^2,\end{aligned}\quad (4.42)$$

for all $\mathbf{p} = (\mathbf{x}, \mathbf{n}) \in \mathbb{M}_2$, and all $\hat{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}}) \in T_p(\mathbb{M}_2)$, and where $\kappa(\mathbf{x}) = \mathbb{1}_A * G_\sigma(\mathbf{x})$ with $A = \bigcup_{i=1}^N [\mathbf{x}_i - a, \mathbf{x}_i + a]$, where \mathbf{x}_i representing $N \in \mathbb{N}$ crossing locations in the image.

In our application, the results are generally insensitive to the choice of a and σ in our application as long as $a > 2$. Therefore we always set $a = 5$ and $\sigma = 1$ pixel-size in our experiments.

This construction of the metric tensor field ensures that the metric tensor field is not tempted to move in the wrong direction in extreme cases where

(a) Challenging case with wrong tracking result via model \mathcal{F}^U .(b) Correct tracking result via model \mathcal{F}^M .

(c) Schematic representation: Two vascular structures are visualized (black). One of them has a bifurcation point (purple). The tracking result (light blue) moves in the wrong direction at the bifurcation, after which it switches vascular structure at the crossing (rotation directions indicated in red).

Figure 4.7: **Motivation mixed model:** Too much curvature adaptation at crossings is dangerous in extreme cases. The mixed model, introduced in Eq. (4.42), is preferable as it only adapts for curvature (like in Fig. 4.5) *in between* those complex structures, and indeed provides correct tracts everywhere, as can be seen in Fig. 4.7b. The geodesics of both models are computed using $\lambda = 100$.

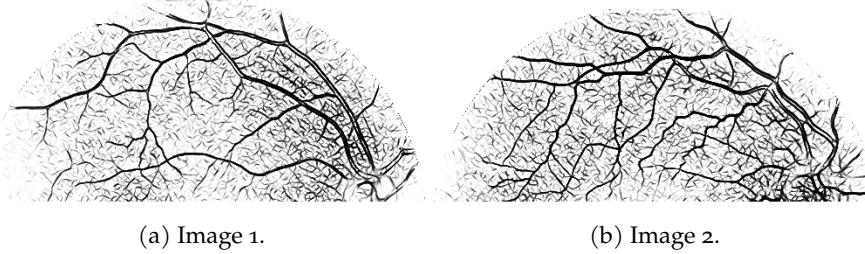


Figure 4.8: Projected cost functions for tracking in two steps - from bifurcation to tip:

Cost used to connect tips to the nearest bifurcation. Black and white mean low and high costs respectively. This cost function supports movement along the thin vessels very well. The multi-scale vesselness is computed as explained in Appendix C, and the considered spatial scale is $\sigma_s = 1$.

vessels cross each other. The tracking result computed with the mixed metric tensor field is visualized in Fig. 4.7b, and does not show the earlier mentioned undesirable behavior, as shown in Fig. 4.7c. Therefore, this new model will be used in all full vascular tree tracking results. All results presented in this section are calculated using parameters $g_{11} = 0.01$, $g_{22} = g_{33} = 1$. For the curve optimisation this is the same as setting $\varepsilon = \zeta = 0.1$ in Eq. (1.20) used in Eq. (4.4). Even for such extreme anisotropy settings, our numerical algorithm is appropriate as motivated in Section 4.4. We always observed that tracking is stable with respect to small variations in these parameters, so there was no point in fine-tuning them.

4.5.2.1 Asymmetric Double Step

The tracking results were computed in two steps; first tips are connected to the nearest bifurcation/seed (cost function visualized in Fig. 4.8), after which those points are connected to the nearest seed (cost function visualized in Fig. 4.9). The used cost functions (from tip to nearest bifurcation and from bifurcation to seed) support movement along the thin and thick vessels respectively. The tracking results that correspond to these cost functions are depicted in Fig. 4.10. The calculated geodesics are all correct, except for 2 difficulties when:

1. crossings and bifurcations are very close to each other,
2. vascular structures are kissing.

Next, we compare the results of the new mixed model \mathcal{F}^M and the left-invariant model \mathcal{F} in Fig. 4.11. There are some visible differences between both tracking methods, marked in pink and blue. First, we see that the tracking results relying on the mixed method ensure that the centerline is better followed,

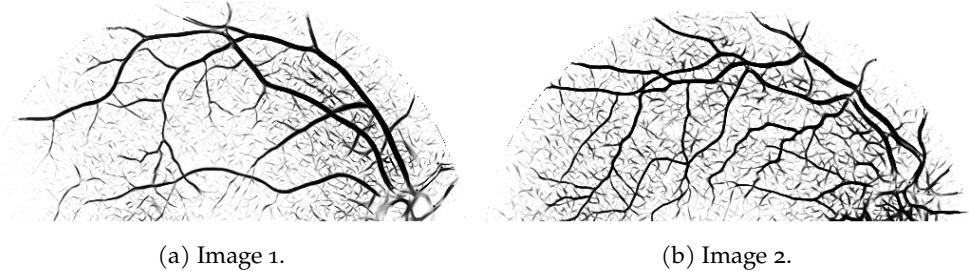


Figure 4.9: Projected cost functions for tracking in two steps - from seed to bifurcation: Cost used to connect bifurcations to optic nerve. Black and white mean low and high costs respectively. This cost function supports movement along the thick vessels very well. The multi-scale vesselness is computed as explained in Appendix C, and the considered spatial scales are $\sigma_s \in \{1, 2\}$.

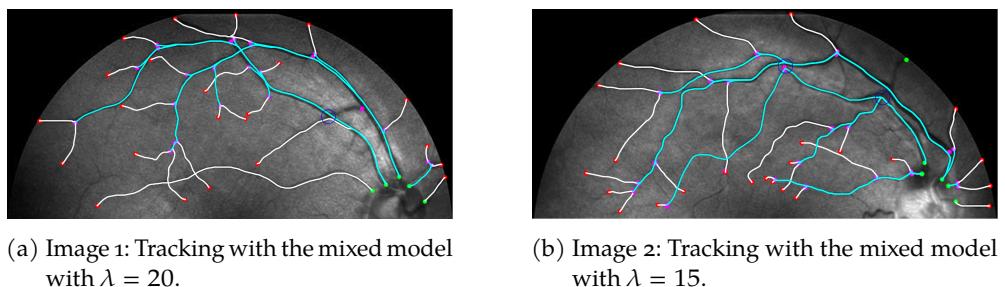


Figure 4.10: Two step tracking of vascular tree structures: Tracking with mixed model $(\mathbb{M}_2, \mathcal{F}^M)$ proposed in Eq. (4.42). The first step involves connecting the tips (marked in red) to the nearest bifurcation (marked in purple) using the cost function depicted in Fig. 4.8. Second, these bifurcations are now tracked to the seeds (marked in green) using the cost function depicted in Fig. 4.9.

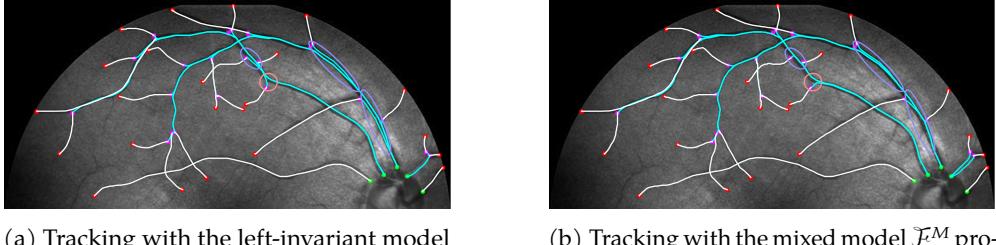


Figure 4.11: Two-step tracking of vascular tree structures: The first step involves connecting the tips (marked in red) to the nearest bifurcation (marked in purple). Second, these bifurcations are now tracked back to the seeds (marked in green). The cost functions used in the first (white) and second (blue) step are given in [83], with $\sigma = 800$ and $p = 4$, and Fig. 4.9 respectively. The main differences between both models indicate that the mixed model \mathcal{F}^M follows the vascular structure better and is better able to follow the centerline of a given vascular tree (marked by pink and blue circles respectively).

and multiple geodesics are at approximately the same place (in blue). Second, we see the ability of the mixed method to adapt to the direction of the vascular structure (in pink). Unlike the left-invariant model, which directs the tracking towards a bifurcation point away from the seed, the curvature adaptation ensures that the tracking results immediately move towards the seed it is connected to.

4.5.2.2 Asymmetric Single Step with Prior Classification of Seeds and Tips

Common practical setups in vascular tracking of retinal images include the prior knowledge of the locations of tips and seeds of vessel structures. We implemented our data-driven model using a prior classification of vessel tips and seeds. More specifically, in every run of the fast-marching algorithm, one of the seeds is considered together with its corresponding tips, and the connecting vessel structures are tracked. Fig. 4.12 shows our result in this setup and demonstrates that our approach can determine the geodesics that accurately follow the vascular structure in the retinal image.

4.5.3 ACCURACY OF THE MODEL

We now present quantitative evaluations to assess the accuracy of our data-driven metrics for geodesic tracking. We measure the mistake ratio E for the images in the STAR dataset. For these images, the ground truth of the vessels

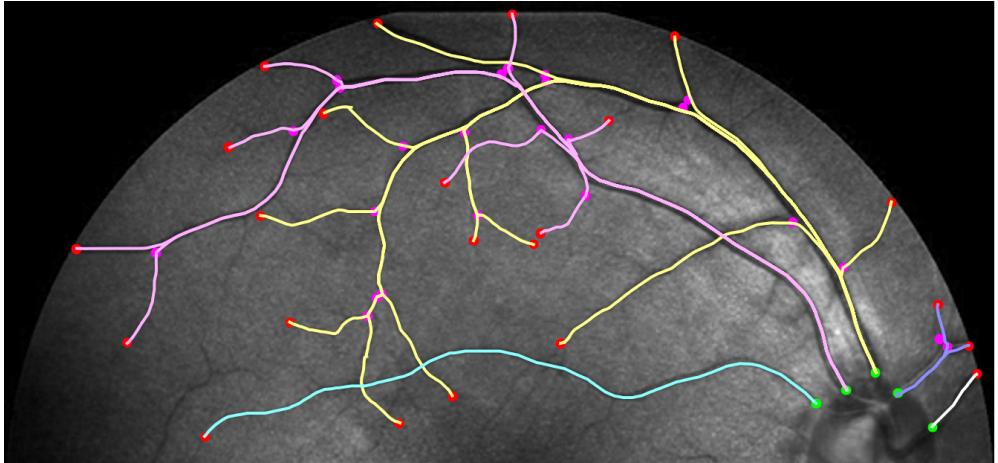
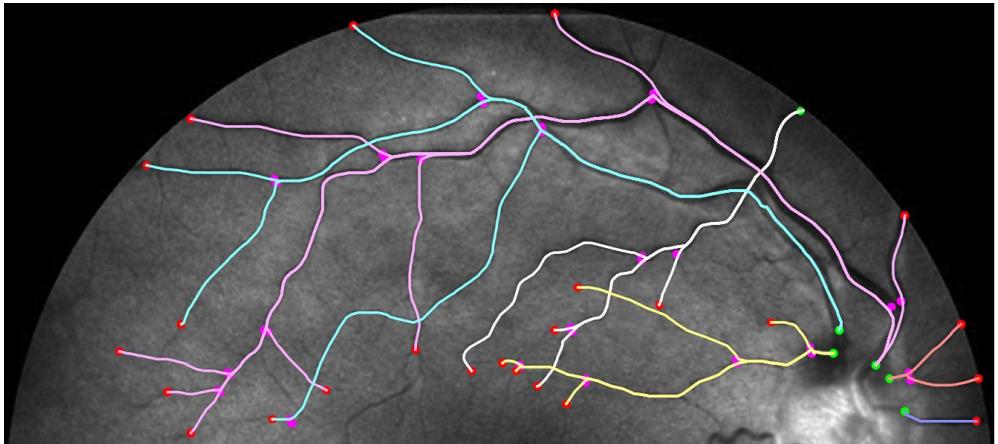
(a) Image 1: Tracking with the mixed model with $\lambda = 50$.(b) Image 2: Tracking with the mixed model with $\lambda = 15$.

Figure 4.12: Tracking of vascular tree per seed: Tracking with the mixed model $(\mathbb{M}_2, \mathcal{F}^M)$ proposed in Eq. (4.42). Prior grouping of tips (red) and seeds (green) results in perfect tracking of the vessel tree, using only one efficient anisotropic fast-marching run via the numerical method in Section 4.4. Both results are calculated with the cost function visualized in Fig. 4.9. At the purple points, we have bifurcations and our tracking is solely based on the mixed model produced (spatially projected) geodesics γ with (automatic) in-place rotations at such bifurcations.

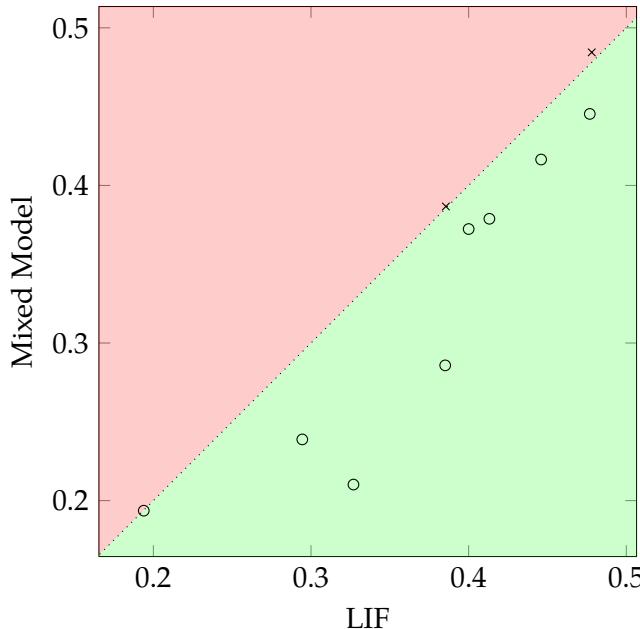


Figure 4.13: Visualization of the scatterplot of the accuracy of the mixed model vs. the left-invariant model applied on images in the STAR dataset (1,2,8,9,13,15,16,24,38,48). The accuracy is calculated on the calculated tracks between the tips and the nearest bifurcation for all vessels in one single run. The red area marks where the former left-invariant model performs better than the new mixed model, incidences indicated by an ‘x’. The green area marks where the new mixed model performs better than the former left-invariant model, incidences are indicated by an ‘o’. Most measurements show the improved performance of the mixed models compared to the left-invariant model (LIF).

is known, which allows us to calculate the percentage of the vessel that is not on the correct vascular structure, where

$$E = \frac{\# \text{ pixels not on correct vessel}}{\# \text{ pixels of all geodesics}}.$$

We have calculated this accuracy for images of the STAR dataset where we connect the tips to their nearest bifurcation, since one should use the new model away from crossing structures. The results are presented in Fig. 4.13. We see that for most tracks, the performance improves when switching to the new data-driven model, and in the cases where there is no improvement, the results do not get significantly worse. On average, we find an improvement of 10.7% of the calculated tracks for the considered images.

4.6 CONCLUSION AND FUTURE WORK

In this chapter, we introduced the concept of a data-driven left-invariant metric tensor field \mathcal{G}^U and its asymmetric variant \mathcal{F}^U . The metric tensor field is defined by the underlying image, where movement along line structures is encouraged by its design in Eq. (4.4). In addition, a data-driven version ∇^U of the plus Cartan connection, relying on \mathcal{G}^U , was introduced.

We used these geometrical tools to formulate a challenging data-driven version of [84, Thm.1], which was stated and proved in Theorem 4.3.1. In this theorem, “straight” and “short” curves are described with respect to the data-driven Cartan connection. In particular, it describes the entire Hamiltonian flow of the new Riemannian manifold model $(\mathbb{M}_2, \mathcal{G}^U)$ in terms of the new data-driven Cartan connection ∇^U , and explains the backtracking procedure for backtracking data-driven left-invariant geodesics in $(\mathbb{M}_2, \mathcal{G}^U)$. As subsequently explained this can be extended to the asymmetric Finsler model $(\mathbb{M}_2, \mathcal{F}^U)$ that often yields the same geodesics, but also automatically places in-place rotations. The latter is beneficial at bifurcations in complex vasculature when using crossing-preserving vesselness costs for the cost function C .

The diagonalization of the new data-driven left-invariant models \mathcal{G}^U and \mathcal{F}^U provides locally adaptive frames that are beneficial over previous approaches to locally adaptive frames in \mathbb{M}_2 [82, 185, 205] in the sense that:

1. they coincide with the usual left-invariant frame if the data is locally constant,
2. they are more stable as they are constructed by coercive metric tensor fields, recall Eq. (4.8).

To calculate the minimizing geodesics efficiently, an adaptation to the efficient anisotropic fast-marching algorithm was required and presented in Section 4.4. The metric tensor component matrix was no longer of block form in the fixed coordinate system, and the necessary changes to overcome this have been discussed and analyzed in Section 4.4. We also provide an asymptotic error analysis of our numerical scheme.

To show the performance of the data-driven metric tensor field and the mixed metric tensor field, we have tested them on 2D images of the retina. All experiments confirm that the new model is better able to adapt for curvature. In addition to that, for the tracking of a single vessel, a low number of orientations is sufficient to find the correct minimizing geodesic, as can be seen in Fig. 4.5. Full vascular tree tracking needs to be handled with care at difficult crossing structures, which is done in the mixed model \mathcal{F}^M introduced in Eq. (4.42).

In general, the tracking results perform very well in the discussed two-step approach (see e.g., Fig. 4.11), where tips are first connected to the nearest

bifurcation, after which the geodesics connecting these bifurcations to the corresponding seeds are calculated. After prior classification of seeds and tips belonging to the same vascular tree, the tracking results follow the vessels perfectly (see Fig. 4.12).

Despite some very appealing theoretical and practical advantages of our model, we still require considerable computation and runtime (tripling the overall processing time) to make the data-driven metric-tensor field and distance maps. Therefore, the exact usage of the proposed data-driven metric depends on the specific context of the tracking requirements.

For future work, it would be interesting to look into the possibilities to train the cost function C using PDE-G-CNNs [184], which is now geometrically computed as explained in Appendix C. In the past, this method had promising results for vessel segmentation. Besides using PDE-G-CNNs to construct the cost function, it would be worth looking into the possibilities to use neural networks to calculate the distance function as was done in [127].

5

VASCULAR TRACKING WITH OPTICAL AND TV-FLOW ENHANCEMENT

The second geodesic tracking model deals with the varying quality of retinal images. We apply illumination enhancement [206] to standardize image quality across the dataset. However, the enhanced images are quite noisy as a result of the design of the used method. We use PDE-enhancement techniques, such as crossing-preserving total variation flow (TV-flow) enhancement in \mathbb{M}_2 , to reduce this noise, while keeping the relevant structures sharp.

First, we elaborate on the enhancement methods that we used to improve the quality of the image. In Section 5.1, we focus on illumination enhancement which ensures that the illumination is approximately constant over the whole image. This leads to better visible vessel structures in parts of the image that were dark before. In Section 5.2, we dive into the crossing-preserving TV-flow enhancement. This is a crossing-preserving denoiser, removing most of the noise generated while improving the illumination of the image.

In Section 5.3, we discuss the developed model that includes both enhancements introduced in Sections 5.1 and 5.2. The model is an extension of the model introduced in Chapter 4. In Theorem 5.3.1, we show that the geodesics induced by the model can be computed by steepest descent on the calculated distance map. Additionally, it shows that the geodesics are data-driven left-invariant.

Finally, in Section 5.4, we show that our new model significantly improves the tracking results of prior models. Many of the previously incorrectly tracked vessels are now correct. The model performs well on realistic retinal images with uneven illumination, and allows computation of full vascular trees from a *single* distance map. Because of the *inclusion* of the optical and TV-flow enhancements in the model, we no longer require a two-step algorithm, as was still the case in Chapter 4 [30]. Additionally, we show that using more prior knowledge on the vasculature (such as artery-vein information or connectivity information) leads to significantly improved tracking results.

We end the chapter with a conclusion in Section 5.5.

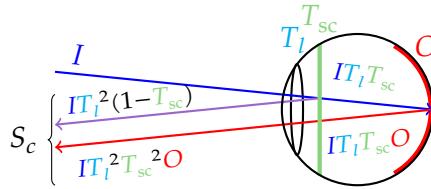


Figure 5.1: Visualization of the physical model when imaging the retina; O is the actual image we would like to recover, S_c is the perceived image (sum of purple and red reflected light). I stands for input illumination, T_l and T_{sc} respectively denote the transmission ratio of the lens and the intraocular scattering (including cataract).

5.1 ILLUMINATION ENHANCEMENT

Previous approaches in retinal vessel tracking typically consider the unprocessed picture S taken by the ophthalmologist, e.g., [18, 83, 108]. However, this may deviate from the actual retinal image O which we aim to recover, due to possible cataract and uneven illumination. The physical model of the construction of the output image is visualized in Fig. 5.1. This yields the following standard optics formula:

$$S_c(\mathbf{x}) = L(\mathbf{x}) (T_{sc}^2(\mathbf{x})O_c(\mathbf{x}) + 1 - T_{sc}(\mathbf{x})) \text{ with } L = I \cdot T_l^2, \quad (5.1)$$

where c denotes the color channel in RGB or the luminance channel Y in YPbPr color space, cf. Wikipedia, and $L, T_{sc} : \Omega \rightarrow [0, 1]$ denote the illumination from outside the eye and transmission of the intraocular scattering respectively on domain $\mathbf{x} \in \Omega \subset \mathbb{R}^2$. The illumination from outside the eye L is composed of the input illumination I and the transmission ratio of the lens T_l . We apply an illumination correction, as done in [207]. After determining the illumination L , we re-express the Y channel (in YPbPr color space) of Eq. (5.1) to

$$O_Y(\mathbf{x}) = \frac{L^{-1}(\mathbf{x})S_Y(\mathbf{x}) - 1 + T_{sc}(\mathbf{x})}{T_{sc}^2(\mathbf{x})} \in [0, 1], \text{ for all } \mathbf{x} \in \Omega, \\ \text{with } T_{sc} = 1 - A \left(L^{-1}(\cdot)S(\cdot) + \sum_{l=1}^n \frac{1}{n} \frac{1}{1+\exp(-\phi_l(\cdot))} G_{\sigma_l} * (L^{-1}(\cdot)S(\cdot)) \right), \quad (5.2)$$

with Gaussian kernel standard deviations $\sigma_l = \text{pixelsize} \cdot 2^{(l-1)}$ of the retinal image at scale level $l \in \{1, \dots, n\}$ where we took $n = 4$ and where the sigmoids on scale coefficients above are included to control the range in $[0, 1]$ and to allow for stable optimization below. The Y channel of the actual image O is obtained by solving the Euler-Lagrange equation of the Tikhonov regularization problem via the Karush-Kuhn-Tucker conditions including the constraints $O_Y \in [0, 1]$:

$$(\phi^{\min}, A^{\min}) = \operatorname{argmin}_{(\phi, A)} \{ \|O_Y - \bar{O}_Y\|_{\mathbb{L}_2(\Omega)}^2 + \lambda \|\nabla O_Y\|_{\mathbb{L}_2(\Omega)}^2 \} \quad (5.3)$$

where we optimize with respect to $A > 0$ and $\phi = (\phi_l)_{l=1}^n \in \mathbb{R}^n$, not O_Y . Here, $\overline{O_Y} \in \mathbb{R}$ is an estimation of the desired intensity level [207, Sec.3.5], and λ regulates the smoothness of O_Y . After optimal non-constant $O_Y = O_Y(\phi^{\min}, A^{\min})$ is retrieved by Eq. (5.2), image O follows by linear conversion of YPbPr to RGB colors, via updated Pb- and Pr-channels.

5.2 TV-FLOW ENHANCEMENT

TV-flow enhancement is a valuable technique for denoising surfaces, but at the same time preserve sharp edges. Recall that the metric intrinsic gradient is given by

$$\mathbb{M}_2 \ni \mathbf{p} \mapsto \nabla_{\mathcal{G}} \phi(\mathbf{p}) = \left((\mathcal{G})^{-1} d\phi \right)(\mathbf{p}) \in T_{\mathbf{p}}(\mathbb{M}_2),$$

using \mathcal{G} in Eq. (1.19) with $\zeta = C\xi = 1, C = 10$. Then TV-flow $U \mapsto W_0(\cdot, t)$ is given by

$$\begin{cases} \frac{\partial W_{\varepsilon}}{\partial t}(\mathbf{p}, t) = \operatorname{div} \left(\frac{\nabla_{\mathcal{G}} W_{\varepsilon}(\cdot, t)}{\varepsilon^2 + (\nabla_{\mathcal{G}} W(\cdot, t))^2} \right)(\mathbf{p}), & \mathbf{p} \in \mathbb{M}_2, t \geq 0, \\ W_{\varepsilon}(\mathbf{p}, 0) = U(\mathbf{p}) \end{cases}$$

and $W_0(\mathbf{p}, t) = \lim_{\varepsilon \downarrow 0} W_{\varepsilon}(\mathbf{p}, t)$. For proof of the \mathbb{L}_2 -convergence, see [185]. Training the end-time t of the TV-flow is unnecessary as for all lifted optically enhanced (cf. Section 5.1) images U of the STAR dataset [205], end-time $t = 0.5$ is nearly optimal for subsequent tracking, and $\Delta t = 0.1$ always remains in the stability region [185]. The same settings provided optimal PSNR-ratios in [185].

5.3 A FINSLER METRIC ON \mathbb{M}_2 THAT INCLUDES THE ENHANCEMENTS

Our goal was to track vasculature accurately. In order to do so, one needs a metric tensor field that describes distances on the manifold. In some cases, it is beneficial to construct a metric tensor field \mathcal{G}^U that depends explicitly on the underlying orientation score data U . This *data-driven* metric tensor field needs to be left invariant with respect to the roto-translation of the underlying data:

Definition 5.3.1 (Data-driven left-invariant metric (DDLM)). *The metric tensor fields \mathcal{G}^U and \mathcal{F}^U on \mathbb{M}_2 are data-driven left invariant when they satisfy for all $(\mathbf{p}, \dot{\mathbf{p}}) \in T(\mathbb{M}_2)$ and all $g \in \operatorname{SE}(2)$:*

$$\mathcal{G}_{\mathbf{p}}^U(\dot{\mathbf{p}}, \dot{\mathbf{p}}) = \mathcal{G}_{g\mathbf{p}}^{\mathcal{L}_g U}((L_g)_*\dot{\mathbf{p}}, (L_g)_*\dot{\mathbf{p}}), \text{ and } \mathcal{F}^U(\mathbf{p}, \dot{\mathbf{p}}) = \mathcal{F}^{\mathcal{L}_g U}(g \cdot \mathbf{p}, (L_g)_*\dot{\mathbf{p}}), \quad (5.4)$$

where $\mathcal{L}_g U(h) := U(L_{g^{-1}} h) := U(g^{-1} \cdot h)$ for all $h \in \mathbb{M}_2$.

The data-driven left-invariant metric tensor fields introduced in Chapter 4 are given by

$$\mathcal{G}_p^U(\dot{\mathbf{p}}, \dot{\mathbf{p}}) = \mathcal{G}_p(\dot{\mathbf{p}}, \dot{\mathbf{p}}) + \mu C^2(\mathbf{p}) \frac{\|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2}{\max_{\|\dot{\mathbf{q}}\|=1} \|HU|_{\mathbf{p}}(\dot{\mathbf{q}}, \cdot)\|_*^2}, \quad (5.5)$$

$$|\mathcal{F}^U(\mathbf{p}, \dot{\mathbf{p}})|^2 = |\mathcal{F}(\mathbf{p}, \dot{\mathbf{p}})|^2 + \mu C^2(\mathbf{p}) \frac{\|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2}{\max_{\|\dot{\mathbf{q}}\|=1} \|HU|_{\mathbf{p}}(\dot{\mathbf{q}}, \cdot)\|_*^2}, \quad (5.6)$$

for all $\mathbf{p} = (\mathbf{x}, \mathbf{n}), \dot{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}})$, representing the symmetric and asymmetric metric tensor fields respectively. In Eqs. (5.5) and (5.6), the metric tensor fields \mathcal{G} and \mathcal{F} were introduced in Eqs. (1.19) and (1.20) respectively. The Hessian field HU is defined as $HU := \nabla(dU)$, w.r.t. plus Cartan connection $\nabla^{[+]}$ for computational details see [84], [30, Rem.8], and $\|\cdot\|_*$ denotes the dual norm w.r.t. $\sqrt{\mathcal{G}_p(\dot{\mathbf{p}}, \dot{\mathbf{p}})}$, where $\zeta = \xi = C = 1$.

The parameter $\mu > 0$ regulates the inclusion of the new data-driven term, and $C(\mathbf{p})$ denotes the cost function described in [30, App. D].

Remark 5.3.1. The construction of this cost now relies on the orientation score U of the optically enhanced image with TV-flow enhancement, whereas previously it relied on the orientation score of the unprocessed image. Akin to [30, App.C], one can show that the new Finsler/Riemannian metric tensor fields (5.5) are DDLIM.

The geodesics are calculated via steepest descent on distance maps using a metric that describes distances on \mathbb{M}_2 .

Definition 5.3.2 (Data-driven Riemannian distance). *The data-driven Riemannian distance $d_{\mathcal{G}^U}$ from a point $\mathbf{p} \in \mathbb{M}_2$ to a point $\mathbf{q} \in \mathbb{M}_2$ is given by*

$$d_{\mathcal{G}^U}(\mathbf{p}, \mathbf{q}) = \inf_{\substack{\gamma \in \Gamma_1, \\ \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q}}} \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}^U(\dot{\gamma}(t), \dot{\gamma}(t))} dt \quad (5.7)$$

where $\Gamma_1 := \{\gamma : [0, 1] \rightarrow \mathbb{M}_2 | \gamma \in PC^1([0, 1], \mathbb{M}_2)\}$ with PC^1 the space of piecewise continuously differentiable curves in \mathbb{M}_2 , and $\dot{\gamma}(t) := \frac{d}{dt}\gamma(t)$. The quasi-distance that belongs to the asymmetric Finslerian model (5.6) is given by

$$d_{\mathcal{F}^U}(\mathbf{p}, \mathbf{q}) = \inf_{\substack{\gamma \in \Gamma_1, \\ \gamma(0) = \mathbf{p}, \\ \gamma(1) = \mathbf{q}}} \mathfrak{L}_{\mathcal{F}^U}(\gamma) := \inf_{\substack{\gamma \in \Gamma_1, \\ \gamma(0) = \mathbf{p}, \\ \gamma(1) = \mathbf{q}}} \int_0^1 \mathcal{F}^U(\gamma(t), \dot{\gamma}(t)) dt. \quad (5.8)$$

Lemma 5.3.1. *If \mathcal{F}^U is DDLIM (Definition 5.3.1) then distance $d_{\mathcal{F}^U}$ satisfies:*

$$\forall_{g \in SE(2)} \forall_{\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{M}_2} : d_{\mathcal{F}^U(g \cdot \mathbf{p}_1, g \cdot \mathbf{p}_2)} = d_{\mathcal{F}^U}(\mathbf{p}_1, \mathbf{p}_2). \quad (5.9)$$

Proof. One has

$$\begin{aligned} d_{\mathcal{F}^{\mathcal{L}_g U}}(g \cdot \mathbf{p}_1, g \cdot \mathbf{p}_2) &= \inf_{\substack{\gamma \in \Gamma_1, \\ \gamma(0) = g \cdot \mathbf{p}_1, \\ \gamma(1) = g \cdot \mathbf{p}_2}} \mathfrak{L}_{\mathcal{F}^{\mathcal{L}_g U}}(\gamma) \stackrel{(5.4)}{=} \inf_{\substack{g^{-1} \cdot \gamma \in \Gamma_1, \\ g^{-1} \cdot \gamma(0) = \mathbf{p}_1, \\ g^{-1} \cdot \gamma(1) = \mathbf{p}_2}} \mathfrak{L}_{\mathcal{F}^U}(g^{-1} \cdot \gamma) \\ &= d_{\mathcal{F}^U}(\mathbf{p}_1, \mathbf{p}_2), \end{aligned}$$

where $g^{-1} \cdot \gamma \in \Gamma_1 \Leftrightarrow \gamma \in \Gamma_1$, from which Eq. (5.9) follows. \square

The shortest curves are computed using steepest descent on the distance map, departing from tip $\mathbf{p} \in \mathbb{M}_2$ towards seed $\mathbf{p}_0 \in \mathbb{M}_2$ as described in Theorem 5.3.1.

Theorem 5.3.1. *The shortest curve $\gamma : [0, 1] \rightarrow \mathbb{M}_2$ with $\gamma(0) = \mathbf{p}$ and $\gamma(1) = \mathbf{p}_0$ can be computed by steepest descent tracking on distance map $W(\mathbf{p}) = d_{\mathcal{F}^U}(\mathbf{p}, \mathbf{p}_0)$*

$$\gamma(t) := \gamma_{\mathbf{p}, \mathbf{p}_0}^U(t) = \text{Exp}_{\mathbf{p}}(t v(W)), \quad t \in [0, 1], \quad (5.10)$$

where Exp integrates the following vector field on \mathbb{M}_2 : $v(W) := -W(\mathbf{p}) \nabla_{\mathcal{F}^U} W$ and where W is the viscosity solution of the eikonal PDE system

$$\begin{cases} \mathcal{F}_U^*(\mathbf{p}, dW(\mathbf{p})) = 1 & \mathbf{p} \in \mathbb{M}_2, \\ W(\mathbf{p}_0) = 0, \end{cases} \quad (5.11)$$

assuming \mathbf{p} is neither a 1st Maxwell-point nor a conjugate point, with dual Finsler function $\mathcal{F}_U^*(\mathbf{p}, \hat{\mathbf{p}}) := \max\{\langle \hat{\mathbf{p}}, \dot{\mathbf{p}} \rangle \mid \dot{\mathbf{p}} \in T_{\mathbf{p}}(\mathbb{M}_2) \text{ with } \mathcal{F}^U(\mathbf{p}, \dot{\mathbf{p}}) \leq 1\}$. As $v(W)$ is data-driven left invariant, the geodesics carry the symmetry

$$\gamma_{g\mathbf{p}, g\mathbf{p}_0}^{\mathcal{L}_g U}(t) = g \gamma_{\mathbf{p}, \mathbf{p}_0}^U(t) \text{ for all } g \in \text{SE}(2), \mathbf{p}, \mathbf{p}_0 \in \mathbb{M}_2, t \in [0, 1]. \quad (5.12)$$

Proof. This is a special case of [30, Thm.1] with Lie group $\text{SE}(2) \equiv \mathbb{M}_2$. Then this yields the symmetric case $\|\nabla_{\mathcal{G}^U} W(\mathbf{p})\| = 1$ in the usual eikonal PDE form. Inclusion of the asymmetric front propagation (relying on asymmetric Finsler metric \mathcal{F}^U) requires a replacement of $\|\nabla_{\mathcal{G}^U} W(\mathbf{p})\| = 1$ with a dual norm expression $\mathcal{F}_U^*(\mathbf{p}, dW(\mathbf{p})) = 1$, where one takes the positive part of the spatial momentum component in direction $\cos \theta dx + \sin \theta dy \in T^*(\mathbb{M}_2)$. This is similar to the technique in [83, Thm.4] but due to the data-driven behavior \mathcal{F}^U this is subtle [30, Eq.43, Lem. 3] and also directly applies to our model Eq. (5.13) using cost function C (including optical & TF-flow enhancement) as explained in Remark 5.3.1. Also, the backtracking requires a subtle adaptation: instead of ordinary intrinsic gradient descent in direction $\nabla_{\mathcal{G}^U} W = (\mathcal{G}^U)^{-1} dW$ it now becomes more general descent in direction $\nabla_{\mathcal{F}^U} W(\cdot) := d\mathcal{F}_U^*(\cdot, dW(\cdot))$ as explained in [83, prop.4]. \square

In the experimental section, we rely on the mixed metric tensor field, which is needed to avoid wrong exits at complex structures, see [30], and is given by:

$$\begin{aligned}\mathcal{G}_p^M(\dot{\mathbf{p}}, \dot{\mathbf{p}}) &= \kappa(\mathbf{x}) \mathcal{G}_p(\dot{\mathbf{p}}, \dot{\mathbf{p}}) + (1 - \kappa(\mathbf{x})) \mathcal{G}_p^U(\dot{\mathbf{p}}, \dot{\mathbf{p}}), \quad \mathbf{p} = (\mathbf{x}, \mathbf{n}) \in \mathbb{M}_2 \\ \mathcal{F}^M(\mathbf{p}, \dot{\mathbf{p}})^2 &= \kappa(\mathbf{x}) \mathcal{F}(\mathbf{p}, \dot{\mathbf{p}})^2 + (1 - \kappa(\mathbf{x})) \mathcal{F}^U(\mathbf{p}, \dot{\mathbf{p}})^2,\end{aligned}\quad (5.13)$$

with $\kappa = \mathbb{1}_A * G_\sigma$ and A the crossing structure locations and Gaussian $G_{\sigma=1\text{pix}}$.

5.4 EXPERIMENTAL RESULTS

We rely on the asymmetric metric tensor field (5.13) to calculate the geodesics of the 3 different models. These models are constructed based on a) the picture taken by the ophthalmologist (original image), b) the original image with illumination enhancement as explained in Section 5.1 (optically enhanced image), and c) the optically enhanced image with crossing-preserving TV-flow enhancement discussed in Section 5.2. In this section, we illustrate the results for a specific image and refer to Table 5.1 for an overview of the performance of the different models on the STAR dataset [1, 205]. These results are consistent with the discussed example, and for reproducible code and all processed images see [26]. In all experiments we set standard parameter settings [19, Eq.5.1], [30, Eq.65] for cost-function C ($p = 3, \lambda = 1000$), for the metrics ($\xi = \zeta = 0.1$) in Eq. (1.19), for TV-flow ($t = 0.5$).

In prior research, the “original” image directly entered the metric tensor field on \mathbb{M}_2 when calculating the geodesics. These images are of varying quality, depending on the patient’s condition and the used equipment. Applying a tracking algorithm, like Anisotropic Fast-Marching [30], on the metric tensor field based on the original image, often results in tracking mistakes due to uneven illumination, both along vascular structures and on the background, making it hard to distinguish different structures. In Fig. 5.2a, one sees the tracking results on an original, unevenly illuminated, non-enhanced image, where all vessels were tracked in one single run. At a lot of locations (13), the tracking connects the seeds \mathbf{p}_0 and tips \mathbf{p} incorrectly.

The optical enhancement explained in Section 5.1 corrects for uneven illumination. Calculating the geodesics using the metric tensor field relying upon the optically enhanced image, reduces the number of tracking mistakes significantly (to 5), cf. Fig. 5.2b. Due to pointwise optimization in the optical enhancement, noise is generated. The crossing-preserving total variation flow enhancement suppresses this noise and indeed results in even fewer tracking mistakes (3), cf. Fig. 5.2c.

The tracking results shown in 5.2 are computed without prior knowledge of the vasculature, aside from seed and tip locations. One might incorporate

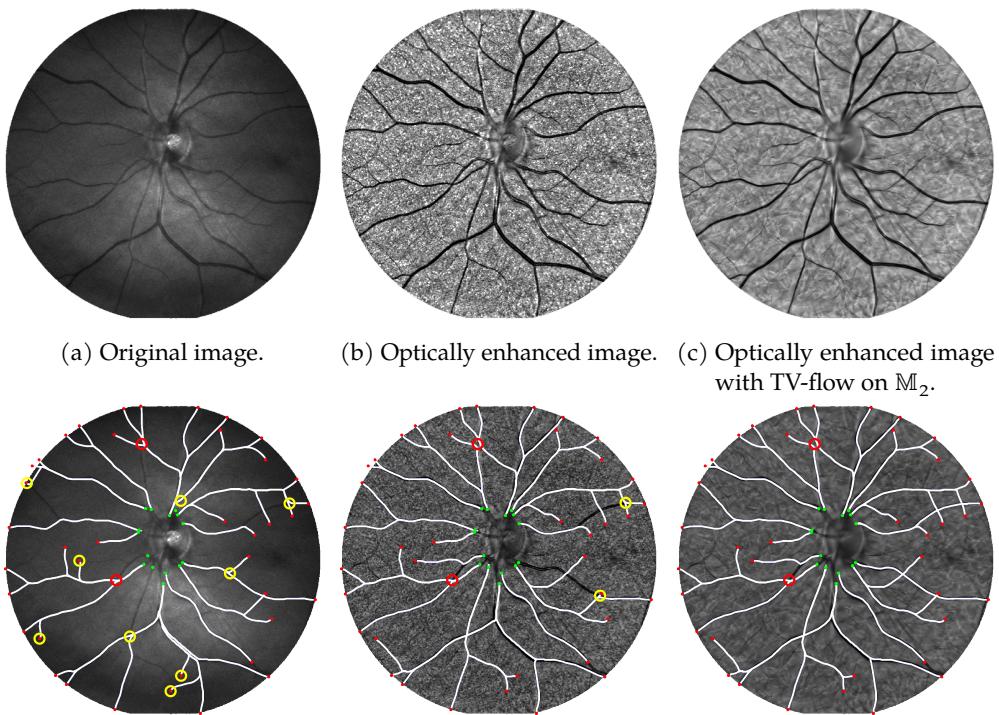


Figure 5.2: Geodesic tracking on the original image, (contrast-)enhanced image, and enhanced image after which TV-flow enhancement is done (left to right). The seeds and tips are indicated in respectively green and red. Yellow (/ red) circles indicate tracking mistakes that are (/ are *not*) fixed in the tracking of another column.

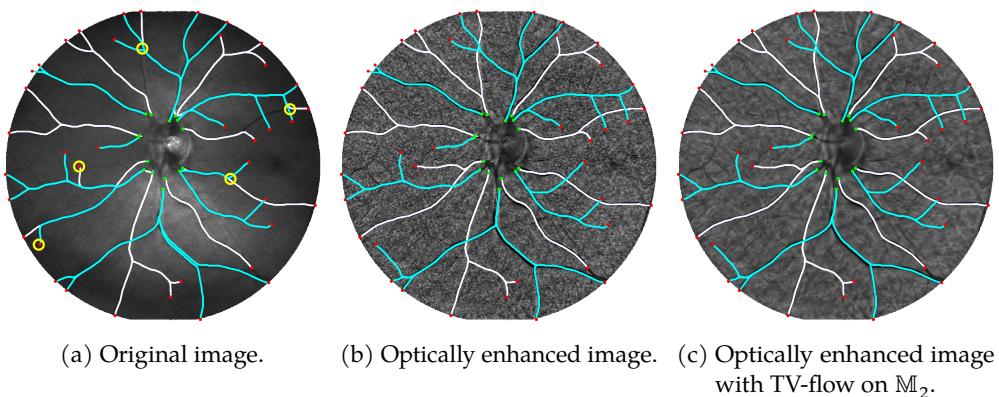


Figure 5.3: **Tracking of vascular tree per vessel type:** Tracking with the mixed model $(\mathbb{M}_2, \mathcal{F}^M)$ proposed in Eq. (5.13) with $\mu = 15$. Prior classification of vascular trees by type (artery/vein respectively white/cyan) only results in perfect tracking of the vessel tree on the enhanced images. Yellow circles indicate tracking mistakes.

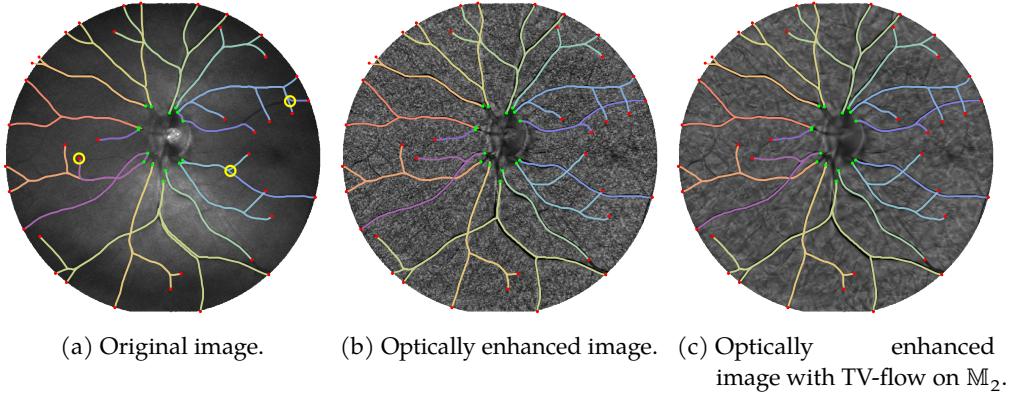


Figure 5.4: Tracking of vascular tree per seed on the optic disk: Tracking with the mixed model (M_2, \mathcal{F}^M) proposed in Eq. (5.13) with $\mu = 15$. Prior grouping of tips (in red) and seeds (in green) only results in perfect tracking of the vessel tree on the enhanced images. Tracking mistakes are indicated by yellow circles.

prior knowledge A) on vessel types (artery/vein), or B) on the connectivity of tips and seeds.

We start by investigating prior knowledge on vessel types (A), connecting all arterial tips to arterial seeds, and similarly for the tips and seeds on veins. Fig. 5.3 shows that the tracking results improve significantly for all cases; the number of tracking mistakes at crossings reduces from (13, 5, 3) to (5, 0, 0) for respectively the original, optically enhanced image excluding and including TV-flow enhancement.

Second, we investigate the impact of incorporating prior knowledge of seed-tip connectivity (B). In Fig. 5.4, the tracking results connecting the tips to their corresponding seed are presented. The number of correct tracks has improved once again, to only (3, 0, 0) mistakes for respectively the original, optically enhanced and optically enhanced with TV-flow image.

We report the tracking results for the three different approaches on images from STAR [1, 205], in particular the example in Figs. 5.2 to 5.4. We observe the same trend in performance for other images which we summarize in Table 5.1. We calculate the (weighted) percentage of incorrectly calculated geodesics by means of:

$$\epsilon := \frac{1}{|T|} \sum_{\mathbf{y} \in T} \left(1 - \frac{\|\mathbf{y} - \mathbf{x}_0(\mathbf{y})\|}{\sqrt{N_x^2 + N_y^2}} \right) C_0(\mathbf{x}(\mathbf{y}), \mathbf{y}) \geq 0, \quad (5.14)$$

Here S, T denote the sets of respectively seeds (near the optic disk) and tips. The image size is $N_x \times N_y$. The ground truth seed and calculated seed (first arriving front in the distance map, cf. Fig. 3.6) corresponding to the tip \mathbf{y} are

	Original image	Optically enhanced image	Optically enhanced image with crossing-preserving TV-flow
Single Run	0.34	0.23	0.20
Per Type (A/V)	0.25	0.12	0.10
Per Seed	0.23	0.09	0.09

Table 5.1: Error measure ϵ of each tracking applied to STAR images in [26], calculated by Eq. (5.14). Green: best results per tracking.

respectively denoted by $\mathbf{x}_0(\mathbf{y}) \in S$ and $\mathbf{x}(\mathbf{y}) \in S$. Function $C_0 : S \times T \rightarrow \{0, 1\}$ is given by $C_0(\mathbf{x}, \mathbf{y}) = 0$ if the tracking between \mathbf{x} and \mathbf{y} is correct and $C_0(\mathbf{x}, \mathbf{y}) = 1$ otherwise.

We evaluate with a strict error measure Eq. (5.14) where one crossing mistake (indicated by a circle) can propagate, leading to multiple errors, when the vessel bifurcates after the crossing.

The optical enhancement and TV-flow regularization applied on the original images, result in more accurate geodesics compared to those calculated directly on the original images, as can be seen in Table 5.1. The more prior information we use, the more accurate the geodesics follow the vasculature. Remarkably, tracking that incorporates artery-vein classification of seeds and tips performs similarly and is easier to automate than tracking with knowledge of seed-tip connectivity.

5.5 CONCLUSION

We developed a new asymmetric, data-driven left-invariant Finsler geometric model that includes contextual contrast enhancement via TV-flows on $SE(2)$. Experiments reveal that application of this new Finsler geometric model has benefits over previous left-invariant models [18, 83] and the recent data-driven model [30]. The new model reduces many errors and performs very well on both realistic and challenging low-quality retinal images where full vascular trees are computed from a *single* asymmetric Finslerian distance map. Although we have shown that both the contrast enhancement and the TV-flow on $SE(2)$ in the new Finslerian model are highly beneficial, there are still exceptional cases where vessel tracts take the wrong exit. This happens at places where both a crossing and a bifurcation occur (cf. the red circles in Fig. 5.2c). Therefore, in future work, we aim to tackle these cases by automatic artery vein classification via PDE-G-CNNs [184], as our experiments show this allows us to obtain the

same good practical results as with the “tracking per seed” (that requires too costly user-knowledge).

6

CROSSING-PRESERVING GEODESIC TRACKING ON SPHERICAL IMAGES

The third geodesic tracking model accounts for the geometry of the eye. Most existing tracking methods treat the retinal image as a flat surface. However, the image is a projection of the vasculature in the eye, a sphere. Therefore, we extend existing tracking methods to perform tracking on the sphere such that crossings are not interpreted as bifurcations.

First, we explain how we build the model having crossing-preserving properties on \mathbb{W}_2 in Section 6.1. We design a mapping Π that maps a horizontal curve on \mathbb{M}_2 (recall Definition 1.3.2) to a horizontal curve on \mathbb{W}_2 , which we prove in Theorem 6.1.1.

The map Π allows us to design a geodesic tracking model that exhibits similar behavior to existing Reeds-Shepp car models on \mathbb{M}_2 . In Section 6.2, we elaborate on how the construction models. To find the geodesics that belong to the new models, we have to adapt our Eikonal PDE solver on which we elaborate in Section 6.3.

In Section 6.4, we compare the performance of our new tracking model to existing ones: crossing-preserving tracking on the space of *planar* positions and orientations \mathbb{M}_2 and non-crossing-preserving tracking on the space of *spherical* positions and orientations \mathbb{W}_2 . We find that the crossing-preserving tracking on \mathbb{M}_2 and \mathbb{W}_2 give similar results. However, the new model does avoid the tracking to mistake a crossing of two bloodvessels for a bifurcation of one vessel, resulting in improved tracking results.

Finally, we conclude in Section 6.5.

6.1 CROSSING-PRESERVING MAPPING FROM \mathbb{M}_2 TO \mathbb{W}_2

We want to perform crossing-preserving tracking on \mathbb{W}_2 instead of \mathbb{M}_2 . For this, we need to define \mathbb{W}_2 analogues of the \mathbb{M}_2 curve optimizations in Eqs. (1.10) and (1.12), which in turn requires a cost function C on \mathbb{W}_2 . Since the tracking must be crossing-preserving, this cost must be computed from lifted data. We achieve this by mapping the crossing-preserving vesselness from \mathbb{M}_2 to \mathbb{W}_2 . Mashtakov et al. [135, Sec. 1.1] already derived a map between S^2 and \mathbb{R}^2 corresponding to the experimental setup (Fig. 6.1); we extend this map Π

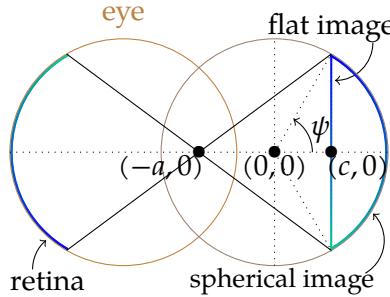


Figure 6.1: Schematic of medical setup: the curved retina is captured in a flat image, which we map to a curved image with the same geometry as the retina. Parameter ψ denotes the maximum angle of the wide-field image, $c > 0$ is the distance from the eyeball center to the flat image, $(-a, 0)$ is the focal point.

(cf. Fig. 3.7) from \mathbb{W}_2 to \mathbb{M}_2 . In Theorem 6.1.1, we prove that this extension is uniquely defined by requiring that horizontal curves in \mathbb{W}_2 get mapped to horizontal curves in \mathbb{M}_2 .

Definition 6.1.1 (Horizontality on \mathbb{W}_2). *On \mathbb{W}_2 , we choose the distribution $\Delta^{\mathbb{W}_2} := \text{Span}\{\mathcal{B}_1, \mathcal{B}_3\} \subset T\mathbb{W}_2$. Then a smooth curve $\delta : \mathbb{R} \rightarrow \mathbb{W}_2$ is said to be horizontal if $\dot{\delta}(t) \in \Delta_{\delta(t)}^{\mathbb{W}_2}$ for all t .*

The choice of distribution $\Delta^{\mathbb{W}_2} := \text{Span}\{\mathcal{B}_1, \mathcal{B}_3\} \subset T\mathbb{W}_2$ is such that the intuitive relationship between the left-invariant frame and the orientation score carries over from \mathbb{M}_2 to \mathbb{W}_2 (cf. $\Delta^{\mathbb{M}_2} := \text{Span}\{\mathcal{A}_1, \mathcal{A}_3\}$).

In our coordinates, the transition map $\pi : D(\pi) \rightarrow \mathbb{R}^2$ of [135] is given by

$$\pi(\alpha, \beta) = \left(\frac{(a+c) \sin(\alpha)}{a + \cos(\alpha) \cos(\beta)}, \frac{(a+c) \cos(\alpha) \sin(\beta)}{a + \cos(\alpha) \cos(\beta)} \right), \quad (6.1)$$

with domain $D(\pi) = \{R(\alpha, \beta, 0) \mathbf{n}_0 \in S^2 \mid \alpha, \beta \in (-\pi/2, \pi/2)\} \subset S^2$.

Here $a, c > 0$ are parameters of the experimental setup, see Fig. 6.1. We can then uniquely extend π to \mathbb{W}_2 by requiring the preservation of horizontality:

Theorem 6.1.1. *Let $\Pi : D(\Pi) \rightarrow \mathbb{M}_2$, with $D(\Pi) := D(\pi) \times S^1$, such that a) π is the spatial projection of Π , and b) for any horizontal curve δ on \mathbb{W}_2 the curve $\gamma := \Pi \circ \delta$ is horizontal on \mathbb{M}_2 . Then, Π is uniquely defined and is given by*

$$\Pi(\alpha, \beta, \phi) = (\pi(\alpha, \beta), \arg(\dot{\pi}^1(\alpha, \beta) + i \dot{\pi}^2(\alpha, \beta))) \quad (6.2)$$

where, for π^i the i -th component of π , we have

$$\dot{\pi}^i(\alpha, \beta) := \frac{\partial \pi^i}{\partial \alpha}(\alpha, \beta) \cos(\phi) + \frac{\partial \pi^i}{\partial \beta}(\alpha, \beta) \frac{\sin(\phi)}{\cos(\alpha)}.$$

Proof. We start by finding sufficient and necessary conditions for curves to be horizontal. Suppose $(x, y, \theta) \equiv \gamma : \mathbb{R} \rightarrow \mathbb{M}_2$ is horizontal. Then, we have $\dot{\gamma}(t) \in \text{Span}\{\mathcal{A}_1, \mathcal{A}_3\}$, which is true if and only if

$$\begin{aligned}\dot{\gamma}(t) &= c^1(t)\mathcal{A}_1 + c^3(t)\mathcal{A}_3 = \underbrace{c^1(t)\cos(\theta(t))\partial_x}_{\dot{x}(t)} + \underbrace{c^1(t)\sin(\theta(t))\partial_y}_{\dot{y}(t)} + \underbrace{c^3(t)\partial_\theta}_{\dot{\theta}(t)} \\ \iff \angle(\dot{x}(t), \dot{y}(t)) &= \theta(t).\end{aligned}\quad (6.3)$$

Similarly, suppose $(\alpha, \beta, \phi) \equiv \delta : \mathbb{R} \rightarrow \mathbb{W}_2$ is horizontal. Then, we have $\dot{\delta}(t) \in \text{Span}\{\mathcal{B}_1, \mathcal{B}_3\}$, which is true if and only if

$$\begin{aligned}\dot{\delta}(t) &= c^1(t)\mathcal{B}_1 + c^3(t)\mathcal{B}_3 = \underbrace{c^1(t)\cos(\phi(t))\partial_\alpha}_{\dot{\alpha}(t)} + \underbrace{c^1(t)\frac{\sin(\phi(t))}{\cos(\alpha(t))}\partial_\beta}_{\dot{\beta}(t)} + \underbrace{c^3(t)\partial_\phi}_{\dot{\phi}(t)} \\ \iff \angle(\dot{\alpha}(t), \dot{\beta}(t)\cos(\alpha(t))) &= \phi(t).\end{aligned}\quad (6.4)$$

Now, let $(\alpha_0, \beta_0, \phi_0) \in D(\Pi)$, and let $(\alpha, \beta, \phi) \equiv \delta : \mathbb{R} \rightarrow \mathbb{W}_2$ be a horizontal curve with $\delta(0) = (\alpha_0, \beta_0, \phi_0)$. Define $\gamma := \Pi \circ \delta \equiv (x, y, \theta)$. Then, by a) we have that $(x(t), y(t)) = \pi(\alpha(t), \beta(t))$, so in particular $(x_0, y_0) = \pi(\alpha_0, \beta_0)$. We next impose the horizontality constraint b):

$$\begin{aligned}\theta(t) &\stackrel{(6.3)}{=} \angle(\dot{x}(t), \dot{y}(t)) = \arg(\dot{x}(t) + i\dot{y}(t)) \\ &\stackrel{(6.4)}{=} \arg\left(\left(\frac{\partial\pi^1}{\partial\alpha}(\alpha(t), \beta(t))\cos(\phi(t)) + \frac{\partial\pi^1}{\partial\beta}(\alpha(t), \beta(t))\frac{\sin(\phi(t))}{\cos(\alpha(t))}\right)\right. \\ &\quad \left.+ i\left(\frac{\partial\pi^2}{\partial\alpha}(\alpha(t), \beta(t))\cos(\phi(t)) + \frac{\partial\pi^2}{\partial\beta}(\alpha(t), \beta(t))\frac{\sin(\phi(t))}{\cos(\alpha(t))}\right)\right).\end{aligned}$$

We now evaluate at $t = 0$ to see $\theta_0 := \theta(0) = \arg(\dot{\pi}^1(\alpha_0, \beta_0) + i\dot{\pi}^2(\alpha_0, \beta_0))$. We have found that $\Pi(\alpha_0, \beta_0, \phi_0) = (x_0, y_0, \theta_0)$, which agrees with Eq. (6.2). \square

6.2 GEODESIC TRACKING MODEL ON \mathbb{W}_2

By choosing to map horizontal curves to horizontal curves, the intuition that \mathcal{A}_1 points parallel to the local orientation while \mathcal{A}_2 points laterally carries over onto \mathbb{W}_2 : \mathcal{B}_1 points parallel to the local orientation while \mathcal{B}_2 points laterally. Hence, we can find analogues of the \mathbb{M}_2 Reeds-Shepp car models Eqs. (1.19) and (1.20) on \mathbb{W}_2 :

$$\mathcal{G}_{\mathbf{q}} = \begin{cases} C^2(\mathbf{q}) \left(\xi_{\mathbb{W}_2}^2 \nu^1 \otimes \nu^1 + \nu^3 \otimes \nu^3 \right) |_{\mathbf{q}}, & \text{on } \Delta_{\mathbf{q}}^{\mathbb{W}_2} \times \Delta_{\mathbf{q}}^{\mathbb{W}_2}, \\ +\infty, & \text{else, and} \end{cases} \quad (6.5)$$

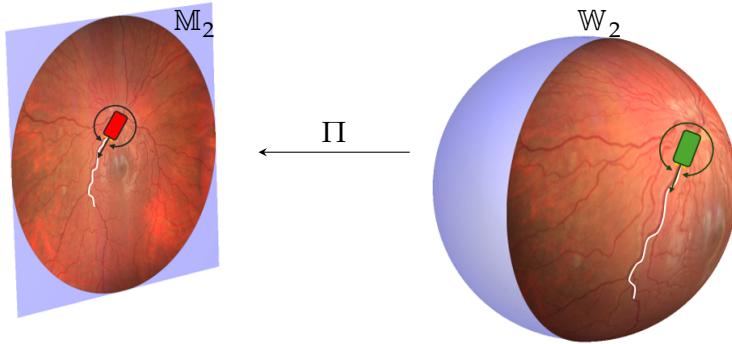


Figure 6.2: Visualization of a tracking result of a blood vessel (white) on the plane (left) and on the sphere (right). We also visualize cars (red on the plane, green on the sphere) that can only move forward and change orientation. This gives intuition behind existing models on $\mathbb{M}_2 \cong \text{SE}(2)$ [83, 163] (red car), and our proposed, and previous [135] models, on $\mathbb{W}_2 \cong \text{SO}(3)$ (green car). We show that the projection map Π is uniquely determined, cf. Theorem 6.1.1.

$$|\mathcal{F}(\mathbf{q}, \cdot)|^2 = \begin{cases} C^2(\mathbf{q}) (\xi_{\mathbb{W}_2}^2 \nu^1 \otimes \nu^1 + \nu^3 \otimes \nu^3) |_{\mathbf{q}'}, & \text{on } \Delta_{\mathbf{q}}^{\mathbb{W}_2,+}, \\ +\infty, & \text{else,} \end{cases} \quad (6.6)$$

where $\{\nu^i\}_{i=1}^3$ is the dual frame to $\{\mathcal{B}_i\}_{i=1}^3$, i.e., $\nu^i(\mathcal{B}_j) = \delta_j^i$, $C : \mathbb{W}_2 \rightarrow \mathbb{R}_{>0}$ is a positive cost function, $\xi_{\mathbb{W}_2}$ is the stiffness parameter regulating the relative cost of moving forwards compared to turning, and $\Delta_{\mathbf{q}}^{\mathbb{W}_2,+} := \{\dot{\mathbf{q}} \in \Delta_{\mathbf{q}}^{\mathbb{W}_2} \mid \nu^1(\dot{\mathbf{q}}) \geq 0\}$.

Remark 6.2.1. One can understand the car models by considering Fig. 6.2. The car, equipped only with a forward gear, can just move forward and steer, but cannot reverse. This corresponds to \mathcal{F} in Eq. (6.6), since curves with tangents in $\Delta_{\mathbf{q}}^{\mathbb{W}_2,+}$ only point forward and angularly. As such, we call \mathcal{F} the *forward gear* model. For the model \mathcal{G} in Eq. (6.5), the car can also move backward. We call \mathcal{G} the *sub-Riemannian* model. In both models, the cost function ensures that the car remains within the vasculature.

For the cost function, we pull back the \mathbb{M}_2 cost function (1.26) to \mathbb{W}_2 :

$$C^{\mathbb{W}_2} := C^{\mathbb{M}_2} \circ \Pi. \quad (6.7)$$

6.3 EIKONAL PDE SOLVER

We assume that vessels on the retina are well-modeled by geodesics (as e.g., [19, 129, 135]) of the distance maps induced by the car models in Eqs. (1.19),

(1.20), (6.5) and (6.6); this allows us to find vessels by backtracking on these distance maps. We will now discuss how we compute the distance map and find corresponding geodesics. Let $\mathcal{M} \in \{\mathbb{M}_2, \mathbb{W}_2\}$, and let \mathcal{F} be the (sub-)Finsler function corresponding to one of our controllers on \mathcal{M} .¹ Then, we are interested in computing the distance map with respect to some reference point $\mathbf{p}_0 \in \mathcal{M}$: $W(\mathbf{p}) := d_{\mathcal{F}}(\mathbf{p}_0, \mathbf{p})$. The reference point \mathbf{p}_0 is called the *seed* of the geodesic. It was shown [83] that W is the (viscosity) solution of the eikonal PDE

$$\begin{cases} \mathcal{F}^*(\mathbf{p}, dW(\mathbf{p})) = 1, & \text{on } \mathcal{M} \setminus \{\mathbf{p}_0\}, \\ W(\mathbf{p}_0) = 0, \end{cases} \quad (6.8)$$

where \mathcal{F}^* is the dual Finsler function. Since Eq. (6.8) is a boundary value problem, it is hard to solve. It can be solved efficiently with Fast-Marching, e.g., Mirebeau [138], though for technical reasons this requires sub-Riemannian and sub-Finslerian metrics to be approximated by highly anisotropic metrics. To enforce exact sub-Riemannian and sub-Finslerian constraints, we follow Bekkers et al. [19] in approximating W to accuracy $\epsilon > 0$ (in supremum norm) by iteratively solving a relaxed version of the eikonal PDE (6.8):

$$\begin{cases} \frac{\partial}{\partial r} W_{n+1}^\epsilon(\mathbf{p}, r) = 1 - \mathcal{F}^*(\mathbf{p}, dW_{n+1}^\epsilon(\mathbf{p}, r)), & \text{on } \mathcal{M} \times [0, \epsilon], \\ W_{n+1}^\epsilon(\mathbf{p}, 0) = W_n^\epsilon(\mathbf{p}, \epsilon), & \text{on } \mathcal{M} \setminus \{\mathbf{p}_0\}, \\ W_{n+1}^\epsilon(\mathbf{p}_0, r) = 0, & \text{on } [0, \epsilon], \end{cases} \quad (6.9)$$

with as initial condition

$$W_0^\epsilon(\mathbf{p}, 0) = \delta_{\mathbf{p}_0}^{\mathcal{M}}(\mathbf{p}) = \begin{cases} 0, & \text{if } \mathbf{p} = \mathbf{p}_0, \\ +\infty, & \text{else,} \end{cases} \quad (6.10)$$

the morphological delta. In the left-invariant ($C \propto 1$), sub-Riemannian case on $\text{SE}(2) \equiv \mathbb{M}_2$ it was shown for all $\mathbf{p} \in \text{SE}(2)$ [19, Thm. E.3] that

$$W(\mathbf{p}) = \lim_{\epsilon \downarrow 0} \lim_{n \rightarrow \infty} W_n^\epsilon(\mathbf{p}, 0). \quad (6.11)$$

The proof of convergence, which relies on $\text{SE}(2)$ morphological convolutions, can be readily extended to the left-invariant, sub-Riemannian case on $\mathbb{W}_2 \equiv \text{SO}(3)$, by instead using $\text{SO}(3)$ morphological convolutions. We *conjecture* that the distance map W solves the eikonal PDE (6.8) with any of the data-driven Finslerian car models considered in this work (and indeed

¹ The sub-Riemannian models also induce a Finsler function: $\mathcal{F}(\mathbf{p}, \dot{\mathbf{p}}) := \sqrt{g_p(\dot{\mathbf{p}}, \dot{\mathbf{p}})}$.

on more general Finsler manifolds), and that the convergence result (6.11) extends to these situations; we hope to prove this in future work. Given (an approximation of) the distance map W , we compute the geodesic connecting point \mathbf{p} to \mathbf{p}_0 by backtracking, i.e., by solving [83, Sec. 2.5]

$$\begin{cases} \dot{\gamma}(t) = -W(\mathbf{p}) d_{\hat{\mathbf{p}}} \mathcal{F}^*(\gamma(t), dW(\gamma(t))), & \text{on } [0, 1], \\ \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{p}_0, \end{cases} \quad (6.12)$$

where $d_{\hat{\mathbf{p}}} \mathcal{F}^*$ denotes the differential with respect to the second entry of \mathcal{F}^* .

6.4 EXPERIMENTAL RESULTS

We apply the new forward gear model on \mathbb{W}_2 to track vessels on a wide-field image [177] (maximum angle $2\psi = 120^\circ$ cf. Fig. 6.1) of a retina (patch in Fig. 6.3a), using the cost function $C^{\mathbb{W}_2}$ based on the crossing-preserving vesselness from [30]. We compare with two baselines: 1) tracking using the forward gear on \mathbb{M}_2 and the same crossing-preserving cost function, as in [83], to investigate the influence of the underlying manifold; 2) tracking using the forward gear on \mathbb{W}_2 and an \mathbb{R}^2 cost function based on the Frangi vesselness [98], as in [135], to investigate the influence of the crossing-preserving nature of the cost function. The implementations of the experiments (with parameters) can be found at [180].

We set one seed \mathbf{p}_0 near the optic disk. We use the eikonal PDE solver (6.9) discussed in Section 6.3 to compute an estimate of the distance maps induced by our forward gear models (Eqs. (1.20) and (6.6) on \mathbb{M}_2 and \mathbb{W}_2 , respectively). Afterward, we calculate the geodesics by backtracking on the distance map, see Eq. (6.12).

In the experiments, we use 3 different cost functions: $C^{\mathbb{M}_2}$, $C^{\mathbb{W}_2}$, and $C^{\mathbb{W}_2, \mathbb{R}^2}$. The first cost function $C^{\mathbb{M}_2}$ is computed from the crossing-preserving vesselness $V^{\mathbb{M}_2}$ (see Appendix C) using Eq. (1.26). The second cost function $C^{\mathbb{W}_2}$ is constructed from $C^{\mathbb{M}_2}$ using Eq. (6.7) (we set $a = \frac{13}{21}$, $c = \frac{1}{2}$). The final cost function $C^{\mathbb{W}_2, \mathbb{R}^2}$ is based on the Frangi multi-scale vessel enhancement filter $C^{\mathbb{R}^2}$ [98], as used in [135]: $C^{\mathbb{W}_2, \mathbb{R}^2}(\alpha, \beta, \phi) := C^{\mathbb{R}^2}(\pi(\alpha, \beta))$

Computing the distance maps and the geodesics on \mathbb{M}_2 and \mathbb{W}_2 is equally expensive, while pulling back the cost function from \mathbb{M}_2 to \mathbb{W}_2 requires negligible computational effort.

Baseline 1)

We first compare the tracking results using the forward gear on \mathbb{M}_2 Eq. (1.20) with cost function $C^{\mathbb{M}_2}$ (Fig. 6.3b) to the forward gear on \mathbb{W}_2 (6.6) with cost

function $C^{\mathbb{W}_2}$ (Fig. 6.3d). The methods perform similarly, with neither taking “shortcuts” that are not present in the underlying vasculature, i.e., they do not jump between different vascular trees at crossings [19]. This is in line with [135]: since the cost function parameter λ is large, it dominates the influence of the shape of the underlying manifold. In the white box, we see that the \mathbb{W}_2 model follows the vasculature better than the \mathbb{M}_2 model.

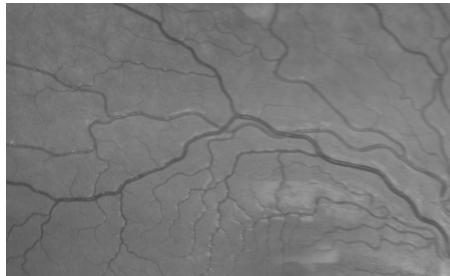
Baseline 2)

We next compare the tracking results using the forward gear on \mathbb{W}_2 (6.6) with the cost function $C^{\mathbb{W}_2, \mathbb{R}^2}$ (Fig. 6.3c) to the cost function $C^{\mathbb{W}_2}$ (Fig. 6.3d). We see that the model using $C^{\mathbb{W}_2, \mathbb{R}^2}$ – based on the Frangi filter which does not preserve crossings – has some issues following the right vasculature. This is caused by the fact that the cost function does not distinguish between bifurcations and crossings: at bifurcations, vessels can quickly turn, whereas at crossings, turning should be prohibited. Consequently, the model occasionally mistakes crossings for bifurcations, taking a “shortcut” (see bold dark green curve). On the other hand, the geodesics using $C^{\mathbb{W}_2}$ better follow the vasculature, as can be seen in Fig. 6.3d; the underlying crossing-preserving cost function has disentangled crossings, allowing the model to differentiate crossings from bifurcations and making it expensive to take a “shortcut”.

6.5 CONCLUSION

We have introduced a model on the lifted space of spherical positions and orientations \mathbb{W}_2 that accounts for local angular information. We have extended the model introduced in [135] to deal with wide-field retinal images and to induce cusp-free geodesics. Additionally, we identified the coordinate mapping between \mathbb{W}_2 and the space of planar positions and orientations \mathbb{M}_2 . This mapping is constructed in such a way that horizontal curves in \mathbb{W}_2 map to horizontal curves in \mathbb{M}_2 , using the same spatial coordinate mapping introduced in [135]. The coordinate mapping allows us to pull back the cost function from $C^{\mathbb{M}_2}$ to $C^{\mathbb{W}_2}$.

We validated the effectiveness of the extension of the coordinate mapping in the experimental section. We conclude that tracking models in the space of spherical positions and orientations \mathbb{W}_2 perform better when using a crossing-preserving cost function, which differentiates between structures at crossings of blood vessels. Additionally, we found that the tracking results (on wide-field images) in \mathbb{W}_2 can improve upon those calculated in the space of planar positions and orientations \mathbb{M}_2 .



(a) Underlying image.

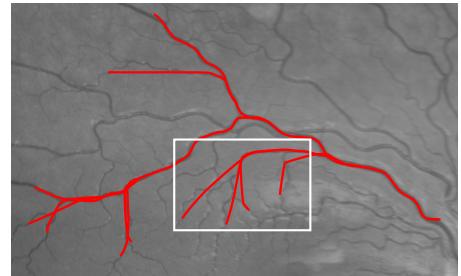
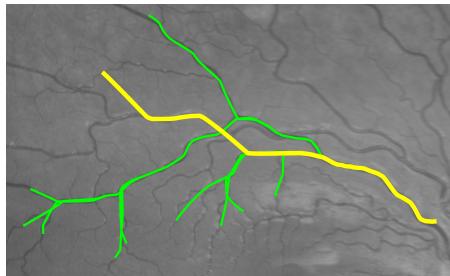
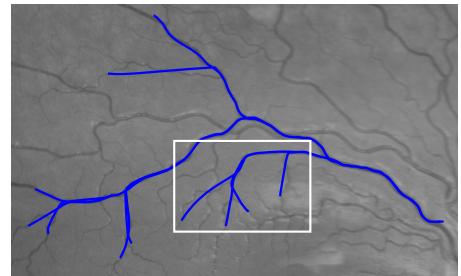
(b) Tracking in \mathbb{M}_2 with cost function $C^{\mathbb{M}_2}$ and $\xi_{\mathbb{M}_2} = 4$.(c) Tracking in \mathbb{W}_2 with cost function $C^{\mathbb{W}_2, \mathbb{R}^2}$ and $\xi_{\mathbb{W}_2} = 6$.(d) Tracking in \mathbb{W}_2 with cost function $C^{\mathbb{W}_2}$ and $\xi_{\mathbb{W}_2} = 6$.

Figure 6.3: Comparison tracking in \mathbb{M}_2 and \mathbb{W}_2 with cost function in \mathbb{R}^2 vs \mathbb{M}_2 . The tracking results c), d) on \mathbb{W}_2 perform better than on \mathbb{M}_2 (white box) and using a lifted cost function d) avoids wrong exits c) at crossings (bold, yellow tract).

Part III

PERCEPTUAL GROUPING IN LIE GROUPS

7

INTRODUCTION - PERCEPTUAL ASSOCIATION FIELDS

7.1 CONTEXT AND GOALS

In image analysis, the task of connecting components plays an important role in numerous applications such as image segmentation [54, 66, 136, 200], object recognition [40, 94, 105, 159, 182], motion tracking [93, 94, 105, 148, 152], and data compression [169, 199]. In this part, we develop a novel method for identifying connected components in images with line structures that possibly intersect with each other, as in Figs. 7.1 and 7.2. When applying a naive connected component algorithm in \mathbb{R}^2 on these images, the algorithm finds only one or two connected components, as it cannot differentiate between the different structures at crossings (cf. Figs. 7.1b and 7.2b).

We present a strategy that does not automatically merge different crossing structures. The image data is lifted from \mathbb{R}^2 to the space of positions and orientations, where crossing structures are disentangled based on their local orientation (cf. Figs. 7.1d and 7.2d).

After the lifting step, we apply our connected component algorithm on the Lie group SE(2). This algorithm is built for any Lie group G and uses theory involving (logarithmic approximations of) Riemannian distances. This results in the practical advantage of grouping different, possibly overlapping, anisotropic connected components based on their local alignment (measured by a Riemannian distance).

Once all connected components have been identified, the output is projected back onto the input image in \mathbb{R}^2 , as visualized in Figs. 7.1c and 7.2c. The new algorithm cannot only deal with overlapping structures, but also with small interruptions of lines, still assigning them to the same connected component, as shown in Fig. 7.3.

7.2 FUNDAMENTALS OF CONNECTED COMPONENT ALGORITHMS

Now that we have illustrated the main geometric idea on basic examples, let us zoom out, and consider the vast literature on connected components analysis.

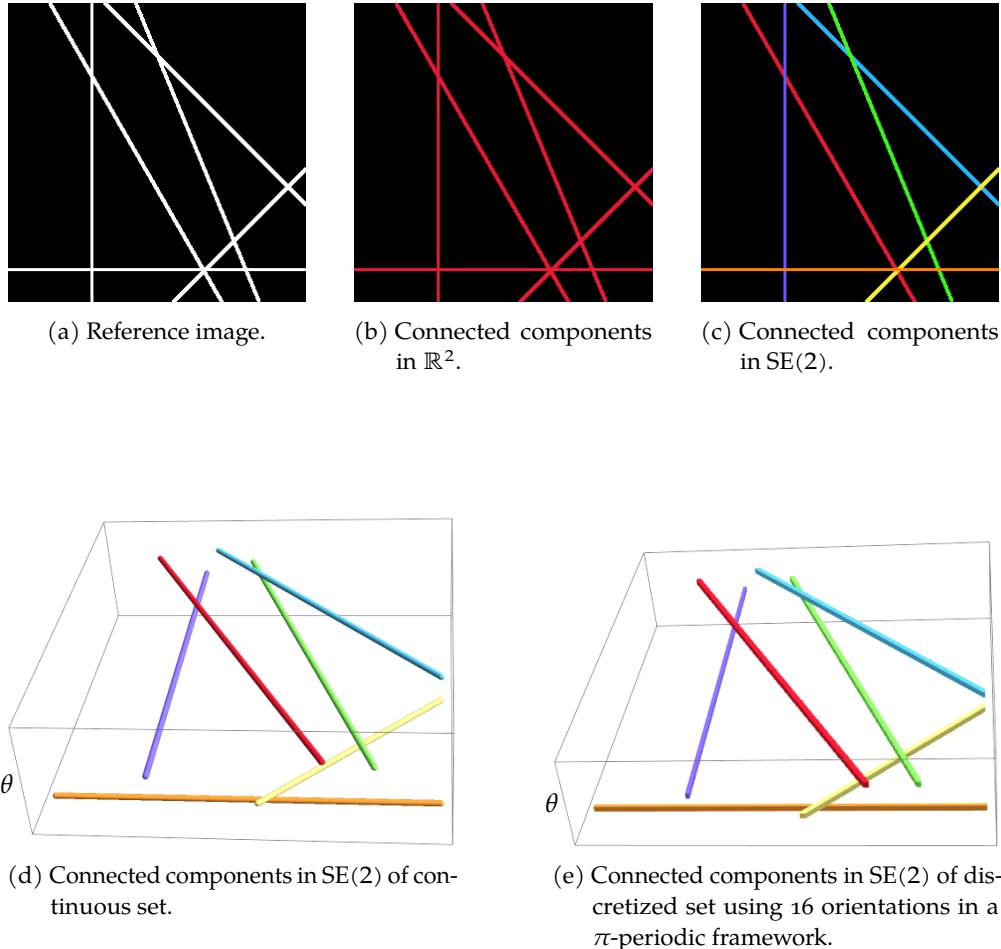


Figure 7.1: Visualization of the connected component algorithm in \mathbb{R}^2 and in the Lie group $SE(2)$ on an image of straight lines. The classical connected component algorithm in \mathbb{R}^2 cannot differentiate between the different line structures, unlike the extension to $SE(2)$. Here we applied the algorithm in Section 8.4.2 using parameters $(w_1, w_2, w_3) = (0.1, 1, 4)$ for the left-invariant metric in Eq. (8.3).

There are typically two main approaches to define connectivity: 1) the graph-theoretic approach and 2) the topological approach. In the first, one considers images as a set of pixels connected through a graph [68, 104, 111, 165]; in the second, images are rather seen as continuous functions on a smooth manifold [10, 42, 72, 191]. This work belongs to the second category. There are two main subclasses of topology-based algorithms: grey-level mathematical morphology [110, 196] and binary morphology [3, 17, 35, 65]. In this part, we use techniques

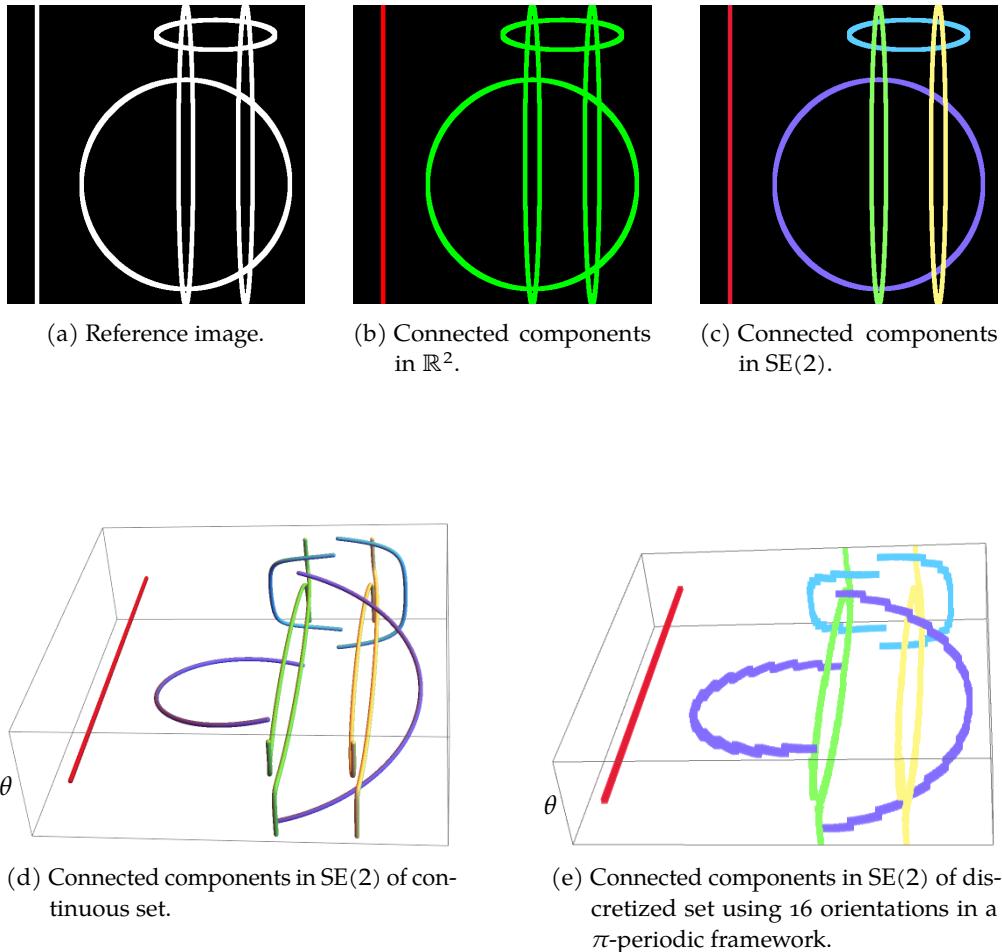


Figure 7.2: Visualization of the effect of performing the connected component algorithm in \mathbb{R}^2 and in the Lie group $SE(2)$ on an image of ovals and lines. The classical connected component algorithm in \mathbb{R}^2 is not able to differentiate between the different line structures, unlike the extension to $SE(2)$. Here we applied the algorithm in Section 8.4.2 using parameters $(w_1, w_2, w_3) = (0.2, 1, 4)$ for the left-invariant metric in Eq. (8.3).

from grey-level mathematical morphology, such as dilation, to solve binary morphology applications.

Connected components have received a lot of attention over the years, resulting in the development of various techniques and algorithms.

- Pixel connectivity: Initially, people were interested in identifying connected components in images and sets. They tackled the problem by looking at the direct neighbors of a pixel or voxel in the set, i.e., either

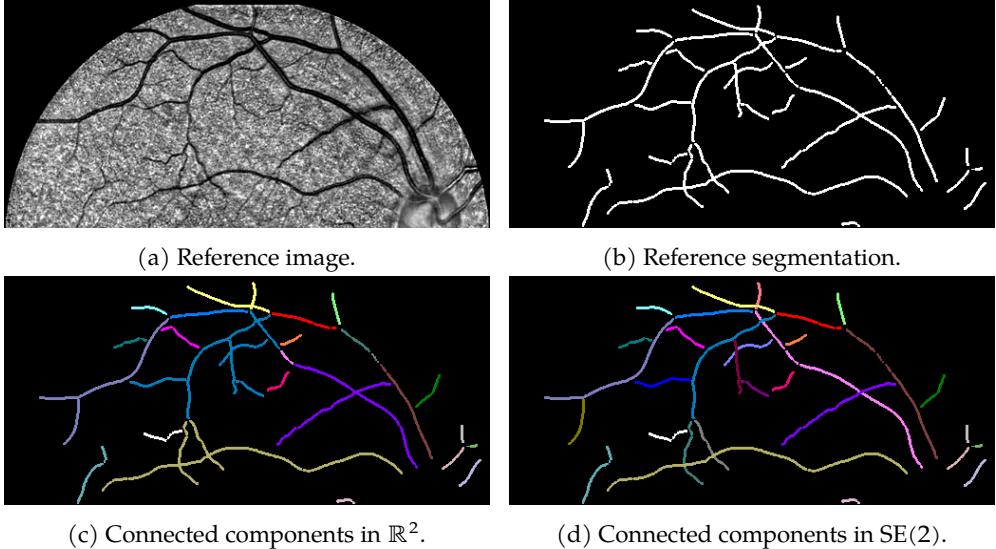


Figure 7.3: Visualization of the effect of the connected component algorithm in \mathbb{R}^2 and in the Lie group $SE(2)$. The classical connected component algorithm is not able to differentiate between crossing vessels, and additionally breaks up not perfectly connected vessels, into different components. The algorithm presented in Section 8.4.2 using parameters $(w_1, w_2, w_3) = (0.2, 1, 4)$ for the left-invariant metric in Eq. (8.3), can better differentiate between different crossing structures and can additionally group well-aligned structures resulting in a more intuitive result.

sharing an edge or corner in 2D, or sharing a face, edge, or corner in 3D [170]. This results in a very local analysis of a given input.

- Distance inclusion: This local connectivity approach is not always desirable depending on the application. Therefore, techniques were developed where the connectivity was defined in terms of the distances between neighboring points, using mathematical morphology [107]. This kind of connectivity allows to assign non-adjacent pixels to the same connected component. This is useful when one wants to identify multiple, previously separate connected components, as one, e.g., in the identification of paw prints or words [150].
- Symmetry inclusion: The grouping of line elements should be equivariant under roto-translations. This is guaranteed if the metric is left-invariant. Note that for isotropic distances, the equivariance under roto-translations is always satisfied.

7.2.1 TOPOLOGICAL DATA ANALYSIS

Although we focus on morphological (PDE-based) data analysis in this thesis, similar techniques were developed in the field of topological data analysis [33, 67, 85]. They developed a profound theory and a variety of methods to identify clusters in point clouds, often relying on distances between points [22, 43, 46]. In topological data analysis, clusters are created based on features of the point cloud, and the optimal threshold is chosen based on a persistence diagram [46, 86, 183, 209]. We apply a similar approach in Section 8.5. Topological data analysis typically focuses on the Euclidean and Riemannian settings on manifolds in general [12, 149, 187], but has not yet optimized its methods to Lie groups.

In the last decade, inclusion of efficient Lie group techniques and PDE-analysis has considerably improved geometric tracking [19, 83], denoising and image enhancement [55, 157, 181], inpainting [36], and geometric deep learning [20, 59, 184]. As we show in the Appendix H.2, topological data analysis techniques, such as ToMATo [46], do not include (sub-Riemannian) methods of neurogeometric perceptual organization on Lie groups [15, 23, 55, 74, 154], which we show to be crucial for tackling our specific applications.

7.3 CONNECTIVITY ON THE LIE GROUP SE(2)

In this chapter, we focus on dealing with overlapping structures by lifting an image from \mathbb{R}^2 to the Lie group SE(2). We highlight some of the works that have focused on this in the past.

Van de Gronde et al [104] suggested a graph-based approach. There, a local orientation tensor was identified for each vertex in the graph. This allowed for the identification of all possible paths containing every vertex while satisfying a set of constraints (local orientations of two adjacent pixels sufficiently aligned). The resulting set of possibly connected vertices represented one connected component. Since this method only considers adjacent pixels for connectivity, this is a local approach in the lifted space of roto-translations.

Besides the graph-based approach in the lifted space, different algorithms based on perceptual grouping (grouping of regions and parts of the visual scene to get higher order perceptual units such as objects or patterns [41]) have been introduced to identify vascular trees (in the lifted space) [3, 17]. In [3], the connectivity is learned from retinal images, after which the learned convection-diffusion kernel is used to determine the connectivity of the vessel fragments. On the other hand, in [17], a set of key points located on blood vessels are used as input for the algorithm. Then, the key points are connected

based on their mutual distance. It is important to note that these algorithms do not rely on mathematical morphology.

7.4 OUR APPROACH: CONNECTED COMPONENTS BY PDE-MORPHOLOGY IN THE LIFTED SPACE $\text{SE}(2)$

We aim for an approach that relies on mathematical morphology techniques to identify connected components. We consider Lie groups G equipped with a left-invariant metric tensor field \mathcal{G} . This metric tensor field induces a left-invariant distance $d_{\mathcal{G}}$ on the Lie group. The precise formulas for the metric $d_{\mathcal{G}}$ and metric tensor field \mathcal{G} will follow in Section 8.1 (cf. Eqs. (8.3) and (8.2)).

In the specific case $G = \text{SE}(2)$, the group of roto-translations, the metric tensor field \mathcal{G} is determined by three parameters w_1 , w_2 , and w_3 . Intuitively, these parameters represent tangential, lateral, and angular movement costs respectively. The influence of different choices for these parameters is illustrated in Fig. 7.4, where the reference image is lifted to the space of roto-translations. One single Riemannian ball is plotted in each figure (lifted to $\text{SE}(2)$ and projection on \mathbb{R}^2) to illustrate the influence of the metric tensor field.

The introduced algorithm will identify, for a given threshold $\delta > 0$, the largest cover of I such that all sets are separated by at least distance δ . We call every set in this cover a δ -connected component. The threshold distance $\delta > 0$ is a parameter to be tuned by the user.

Remark 7.4.1 (δ -connected components and Čech/Vietoris-Rips complexes of radius $\delta/2$). δ -connected components of discrete point clouds arise as connected components of the vertices of non-singleton elements in Čech (and Vietoris-Rips) complexes of radius $\delta/2 > 0$. Formally, non-singleton elements in Čech complexes are simplices (not points). The identification of δ -connected components and Čech complexes also generalizes from discrete point clouds to (para)compact sets [33, 67, 85, 119] (via nerves of δ -balls centered at points of the set). In our setting, where we only consider compact subsets of Lie groups, and in our analysis, the basic notion of δ -connected component suffices.

In calculating these δ -connected components, we rely on formal morphological convolutions via Hamilton-Jacobi-Bellman (HJB) equations and Riemannian distance approximations, allowing for fast, parallelizable morphological convolution algorithms.

In the experimental section of this part, we have restricted ourselves to the Lie group $G = \text{SE}(2)$. We consider two-dimensional images which we view

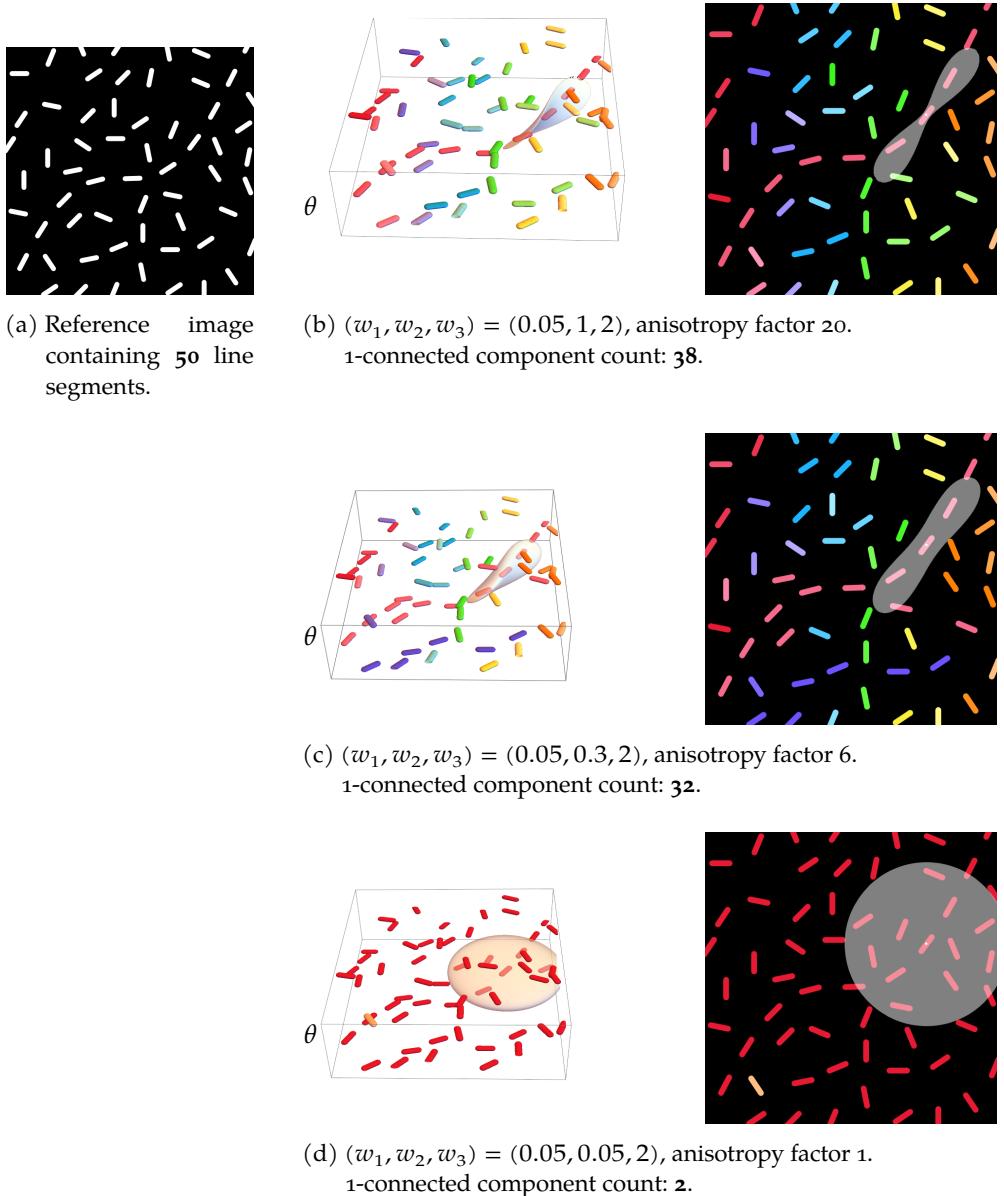


Figure 7.4: Visualization of the effect of the metric tensor field on the connected component algorithm in the Lie group $\text{SE}(2)$. Left to right: reference image, connected components visualized in $\text{SE}(2)$ and projected back onto \mathbb{R}^2 respectively. The output heavily relies on the chosen distance metric $d_{\mathcal{G}}$ introduced in Eq. (8.2), with metric tensor parameters as defined in Eq. (8.4).

as functions $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ belonging to the space $\mathbb{L}_2(\mathbb{R}^2, \mathbb{R})$. We next lift the images to the space of roto-translations $\text{SE}(2)$ with a mapping

$$W_\phi : \mathbb{L}_2(\mathbb{R}^2, \mathbb{R}) \rightarrow \mathbb{L}_2(\text{SE}(2), \mathbb{R}),$$

where, for every $f \in \mathbb{L}_2(\mathbb{R}^2, \mathbb{R})$ and for all $g = (\mathbf{x}, \theta) \in \text{SE}(2)$,

$$(W_\phi f)(g) := \int_{\mathbb{R}^2} \phi(R_\theta^{-1}(\mathbf{y} - \mathbf{x})) f(\mathbf{y}) d\mathbf{y}, \quad (7.1)$$

and where $\phi \in \mathbb{L}_1(\mathbb{R}^2) \cap \mathbb{L}_2(\mathbb{R}^2)$ is a rotating anisotropic wavelet. Here, we use the real-valued cake wavelets (for more information, see [75, 78]). The resulting function $W_\phi f$ is usually called the orientation score of the image f , and the operator W_ϕ is called the orientation score transform.

The δ -connected components are calculated on suitably binarized orientation scores $W_\phi f$, after which the resulting components are “projected back” onto $\mathbb{L}_2(\mathbb{R}^2, \mathbb{R})$ by inverting W_ϕ [73].

In the experimental section, we see that the δ -connected component algorithm allows us to identify aligned structures in images. In the experiments, we look at both artificial images (cf. Figs. 7.1, 7.2, 7.4 and 8.4 to 8.6) and images of the human retina (Figs. 7.3 and 8.9 to 8.13). On both datasets, we can identify connected components that do not suffer from the crossing structures. We have also performed a stability analysis on the δ -connected components as illustrated in Figs. 8.4 to 8.6. Lastly, we look at grouping different components in images of the human retina based on their mutual affinity (linked to perceptual grouping). By doing this, we can find even more complete vascular trees.

As the approach generalizes to other Lie groups and applications, cf. [15, 76, 80, 95, 135], we formulate the theory beyond the $\text{SE}(2)$ setting. Crucial ingredients are a good logarithmic norm approximation (valid for $G = H(d)$, $\text{SO}(d)$, $\text{SE}(d)$) and a lift via a unitary group representation so that a left action on the image is a left action on the score [73].

7.5 MAIN CONTRIBUTIONS

The main contributions in this chapter are:

1. We develop a new method to find the connected components of a binary function defined on a Lie group G equipped with a left-invariant (sub)-Riemannian metric. The method is roto-translation equivariant and relies on standard tools from topological data analysis (e.g., Čech complexes of radius $\delta/2$ [12, 33, 46, 67, 85, 149, 187] are closely related to our δ -connected components). We compute them via morphological PDEs on Lie groups. Here, we employ efficient left-invariant solvers, relying on iterative morphological group convolutions with analytic PDE kernels. Such morphological group convolutions are parallelizable and employ the group structure on the Lie group G .
2. We mathematically analyze our method:

- a) We prove that the provided algorithm always converges to the correct δ -connected component in a finite number of steps in Theorem 8.4.1.
 - b) We show reflectional symmetries of our connected component algorithm (Corollary F.1 in Appendix F) and how this generalizes to other Lie groups G of dimension $n < \infty$ (Lemma F.1). This fundamental property is due to the invariance of both the Riemannian distance and its logarithmic norm approximation under the 2^n reflectional symmetries in the Lie algebra. For $G = \text{SE}(2)$ and $G = \text{SO}(3)$ inclusion of these symmetries (cf. Fig. F.1) is desirable in the connected component algorithm.
3. Along with this method, we publish code (in Mathematica and Python [27]) for the morphological convolutions and the connected component algorithm.
 4. The iterative morphological convolutions are parallelizable, fast and flexible, thanks to intuitively parameterized analytic kernels on the Lie group G . They do not require more expensive state-of-the-art anisotropic fast-marching schemes [139, 140] for computing the Riemannian distance maps.
 5. We show how our method combined with (variants of) standard methodology from topological data analysis (affinity matrices [1, 3], persistence homology based clustering [46, 183]), is very beneficial in multi-orientation image analysis of complex vascular trees in retinal imaging. We present experiments of (improved) grouping and segmentation of blood vessels.

Generally, our algorithm performs well. It can be applied to multi-orientation analysis in flat and spherical images, relating to respectively Lie group $\text{SE}(2)$ [31] and Lie group $\text{SO}(3)$ [29]. In both cases, we rely on analytic distance approximations as explained in Appendix E. A basic comparison to existing approaches can be found in Appendix H.

Our method also has two main limitations: 1) we did not fully employ the possible parallelization of our Lie group processing, and 2) we did not perform automatic optimization of the parameters. Therefore, in future work, we aim for: 1) faster GPU implementations via TaiChi (as done for geodesic tracking in [29]) and 2) training of the $n = \dim(G)$ parameters of the connected component algorithm (controlling the ball shapes in G , recall Fig. 7.4 on $\text{SE}(2)$).

7.6 STRUCTURE OF THE CHAPTER

First, we briefly introduce Lie groups and left-invariant distances in Section 8.1. In Section 8.2, we introduce our notion of δ -connected components and elaborate on our approach towards identifying those components. Then, we introduce the morphological convolutions (Section 8.3) that are used in the presented δ -connected component algorithm (Section 8.4.2). We prove it converges to the desired components in a finite number of steps (Section 8.4.3). Additionally, we explain the choice of δ using persistence diagrams in Section 8.5. We define the affinity between different δ -connected components in Section 8.6, which allows us to quantify how well different components are aligned. Once all these theoretical concepts have been introduced, we move on to Section 8.7, where we present the experimental part of the chapter. Finally, we summarize and conclude in Section 8.8.

8

CONNECTED COMPONENTS ON LIE GROUPS AND APPLICATIONS

8.1 BACKGROUND ON LIE GROUPS AND LEFT-INVARIANT DISTANCES

The theory presented in this document applies to all finite-dimensional Lie groups G of dimension $n \in \mathbb{N}$, with a left-invariant Riemannian distance d . We recall that a Lie group is a smooth manifold G equipped with a group structure such that the group multiplication $\mu : G \times G \rightarrow G$ given by $\mu(x, y) = xy$ and the inverse map $\nu : G \rightarrow G$ given by $\nu(x) = x^{-1}$ are smooth maps.

The tangent space at $h \in G$ of Lie group G is the vector space of all tangent vectors of curves passing through $h \in G$, which we denote by $T_h(G)$.

Let $g \in G$. The left-action on G is denoted by $L_g : G \rightarrow G$ and defined as $L_g(h) = gh$, and its corresponding pushforward $(L_g)_* : T_h(G) \rightarrow T_{gh}(G)$ is defined as $(L_g)_* A_h u = A_h(u \circ L_g)$ for all smooth functions $u : G \rightarrow \mathbb{C}$, where $A_h \in T_h(G)$. In particular if we take $h = e$, the unit element, and a coordinate basis $\{\partial_{x^i}|_e\}_{i=1}^n$ at $T_e(G)$, then

$$(L_g)_* \partial_{x^i}|_e u = \partial_{x^i}|_e (u \circ L_g).$$

By definition, the Riemannian metric tensor field is a field of inner products $\mathcal{G}_g : T_g(G) \times T_g(G) \rightarrow \mathbb{R}$ for each $g \in G$. We always consider a left-invariant Riemannian metric tensor field \mathcal{G} , i.e., for all $g, h \in G$ and for all $\dot{h} \in T_h(G)$

$$\mathcal{G}_{gh}((L_g)_*\dot{h}, (L_g)_*\dot{h}) = \mathcal{G}_h(\dot{h}, \dot{h}). \quad (8.1)$$

Then, the Riemannian left-invariant distance $d_{\mathcal{G}}$ is defined by

$$d_{\mathcal{G}}(g, h) := \inf \left\{ \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt \mid \right. \\ \left. \gamma \in \Gamma_1, \gamma(0) = g, \gamma(1) = h \right\}, \quad (8.2)$$

where $\Gamma_1 = \text{PC}([0, 1], G)$ denotes the family of piece-wise continuously differentiable curves. Then from Eq. (8.1), it follows readily that $d_{\mathcal{G}}(gh_1, gh_2) = d_{\mathcal{G}}(h_1, h_2)$ for all $g, h_1, h_2 \in G$. Intuitively, this means that the distance is invariant to left-actions, and one has $d_{\mathcal{G}}(g, h) = d_{\mathcal{G}}(h^{-1}g, e)$, where $e \in G$ is the unit element.

Remark 8.1.1. The distance $d_{\mathcal{G}}$ always relies on a metric tensor \mathcal{G} . Henceforth, we omit the label \mathcal{G} , and write $d := d_{\mathcal{G}}$ for the distance whenever we do not need to stress the metric tensor used to calculate the distances.

Remark 8.1.2 (Explanation weights of metric tensor field). From Eq. (8.1), one can deduce that all left-invariant (recall Eq. (8.1)) metric tensor fields are given by

$$\mathcal{G}_g(\dot{g}, \dot{g}) = \sum_{i,j=1}^n c_{ij} \dot{g}^i \dot{g}^j \quad (8.3)$$

with *constant* coefficients $c_{ij} \in \mathbb{R}$, and where $\dot{g} = \sum_{i=1}^n \dot{g}^i \mathcal{A}_i|_g \in T_g(G)$ for a basis $\{\mathcal{A}_i\}_{i=1}^n$ of left-invariant vectors $G \ni g \mapsto \mathcal{A}_i|_g \in T_g(G)$, $i = 1, \dots, n$. Note that in general the matrix $[c_{ij}]$ is positive symmetric definite, but in this chapter we constrain ourselves to the diagonal case

$$c_{ij} = w_i \delta_{ij}, \quad (8.4)$$

with positive weights $w_i > 0$.

Remark 8.1.3. The theoretical results in this chapter also hold when the distance denotes a sub-Riemannian distance, though this requires different logarithmic approximations than in Eq. (8.7), for details see [23, 77, 78, 89].

We have a particular interest in the cases of the roto-translation group

$$G = \text{SE}(2) := \mathbb{R}^2 \rtimes \text{SO}(2)$$

and the rotation group

$$G = \text{SO}(3) := \{X \in \mathbb{R}^{3 \times 3} \mid X^\top X = I, \det(X) = 1\}.$$

The group product of $G = \text{SE}(2)$ is given by

$$g_1 g_2 = (x_1, R_1)(x_2, R_2) = (x_1 + R_1 x_2, R_1 R_2) \quad (8.5)$$

for all $(x_1, R_1), (x_2, R_2) \in \mathbb{R}^2 \rtimes \text{SO}(2)$, and the left-actions are given by roto-translations. Similarly, the group product $R_1, R_2 \in \text{SO}(3)$ is given by the ordinary matrix product $R_1 R_2 \in \text{SO}(3)$, where we note that $(R_1 R_2)(R_1 R_2)^\top = R_1 R_2 R_2^\top R_1^\top = R_1 I R_1^\top = I$ and additionally $\det(R_1 R_2) = \det(R_1) \det(R_2) = 1$.

Remark 8.1.4. In the case where $G = \text{SE}(2)$, we have $\dot{g} = \sum_{i=1}^3 \dot{g}^i \mathcal{A}_i|_g \in T_g(G)$ in Eq. (8.3), where

$$\mathcal{A}_1 = \cos \theta \partial_x + \sin \theta \partial_y, \mathcal{A}_2 = -\sin \theta \partial_x + \cos \theta \partial_y, \mathcal{A}_3 = \partial_\theta, \quad (8.6)$$

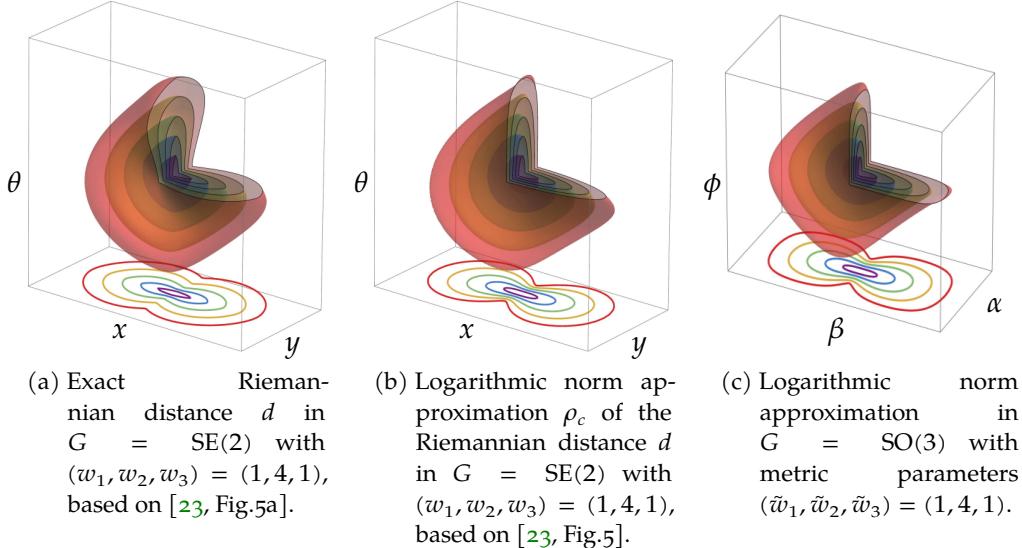


Figure 8.1: Visualization of distance balls and their logarithmic approximations in $G = \text{SE}(2)$ and $G = \text{SO}(3)$. We see isocontours of $d(p_0, \cdot)$ in G , and on the bottom, we see the min-projection over the orientation θ of these contours. The visible contours are $d = 0.5, 1, 1.5, 2, 2.5$ and the metric parameters are $(w_1, w_2, w_3) = (\tilde{w}_1, \tilde{w}_2, \tilde{w}_3) = (1, 4, 1)$. Parameter w_1 controls costs for tangential motion, w_2 controls costs for lateral motion on the base manifold (resp. \mathbb{R}^2 and S^2) and w_3 controls costs for changing the orientation along a geodesic. Thereby, together, they control the anisotropic shape of the Riemannian ball and the identification of connected components, recall white balls in column 2 of Fig. 7.4.

are left-invariant vector fields and where we identify

$$\theta \in \mathbb{R}/(2\pi\mathbb{Z}) \leftrightarrow R_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \in \text{SO}(2).$$

In that case, w_1 , w_2 , and w_3 in Eqs. (8.3) and (8.4) describe costs for forward, sideways, and angular movement in $\text{SE}(2)$ respectively.

The groups $G = \text{SE}(2)$ and $G = \text{SO}(3)$ have the special property that the exponential map $\text{Exp} : T_e(G) \rightarrow G$ is surjective and we have a complete logarithmic norm approximation

$$d_{\mathcal{G}}(g, h) = d_{\mathcal{G}}(h^{-1}g, e) \approx \|\log h^{-1}g\|_{\mathcal{G}} \quad (8.7)$$

for $g, h \in G$ close enough to each other, cf. Fig. 8.1. For details on the quality of this approximation for the $G = \text{SE}(2)$ case, see [23]. For a concise self-contained summary and explicit formulas for the logarithmic norm approximation in

Eq. (8.7) in coordinates, see Eq. (E.3) for $G = \text{SE}(2)$ and Eq. (E.6) for $G = \text{SO}(3)$ in Appendix E.

Remark 8.1.5 (Inclusion of symmetries). It is well-known in Lie group theory that left-invariant Riemannian metrics have a constant Gram matrix in any left-invariant frame. So in Eq. (8.3), $c_{ij} \in \mathbb{R}$ is independent of $g \in G$ if $(\mathcal{A}_i)_{i=1}^n$ is a left-invariant frame, i.e., a basis of left-invariant vector fields. The associated distance maps of such metrics and their logarithmic norm estimates carry $2^{\dim(G)}$ reflectional symmetries. In the specific Lie group case of roto-translations $G = \text{SE}(2)$ there are 2^3 reflectional symmetries (for intuitive illustration see Appendix F, Fig. F.1) which also become reflectional symmetries of our connected component algorithm.

Remark 8.1.6. The Lie group $\text{SE}(2)$ can be identified with the homogeneous manifold of positions and orientations in two dimensions \mathbb{M}_2 . However, this is not true for $n > 2$: $\text{SE}(n) \not\cong \mathbb{M}_n$.

8.2 AN ALGORITHM TO FIND THE δ -CONNECTED COMPONENTS OF THE COMPACT SET I

To explain our algorithm, we first introduce the notion of δ -connectedness of a compact set I . This concept is closely related to the Čech complexes of radius $\delta/2$ in computational geometry and topology [33, 67, 85], see Remark 7.4.1.

8.2.1 THE NOTION OF δ -CONNECTEDNESS

We start by recalling the notion of the covering number of I , for which we use the definition of a ball.

Definition 8.2.1 (Riemannian ball). *The ball around $g \in G$ with radius $\delta > 0$ is given by*

$$B(g, \delta) = \{h \in G \mid d(g, h) < \delta\}.$$

Definition 8.2.2 (δ -covering). *A set $C \subset I$ is a δ -covering of I if $I \subseteq \bigcup_{g \in C} B(g, \delta)$.*

Definition 8.2.3 (Covering number). *The covering number, denoted by $n_\delta(I)$, is the smallest cardinality of all possible δ -coverings of the set I , i.e.,*

$$n_\delta(I) := \min \{|C| : C \text{ is a } \delta\text{-covering of } I\}. \quad (8.8)$$

Since we assume that the set I is compact, its covering number $n_\delta(I)$ is finite. Next, we introduce the notion of δ -connectedness between points $g, h \in I$.

Definition 8.2.4 (δ -connectedness of points). Let $\delta > 0$. We say that two elements g and $h \in I$ are δ -connected, and we denote it as $g \mathcal{D} h$, if and only if there exists a finite number $m \in \mathbb{N}$ of elements $\{q_i\}_{i=0}^m \subset I$ such that $q_0 = g$, $q_m = h$ and

$$d(q_{i+1}, q_i) \leq \delta, \quad \forall i \in \{0, \dots, m-1\}.$$

Remark 8.2.1. Note that $m \in \mathbb{N}$ may depend on $g, h \in G$ and δ , so $m := m_\delta(g, h) \in \mathbb{N}$.

One can readily check that \mathcal{D} is an equivalence relation on I , and the equivalence class of a $g \in I$ under \mathcal{D} , denoted by $[g]$, is defined as

$$[g] := \{q \in I \mid q \mathcal{D} g\}.$$

These preliminary concepts allow us to give the following definition to the notion of δ -connected components.

Definition 8.2.5 (δ -connected components). The set of δ -connected components of a given set I is defined as its equivalence class $\tilde{I}^\delta := I / \mathcal{D}$, and each element $[g] \in \tilde{I}^\delta$ is called a δ -connected component.

The following result guarantees that there are at most $n_\delta(I)$ δ -connected components to search for when I is a compact set.

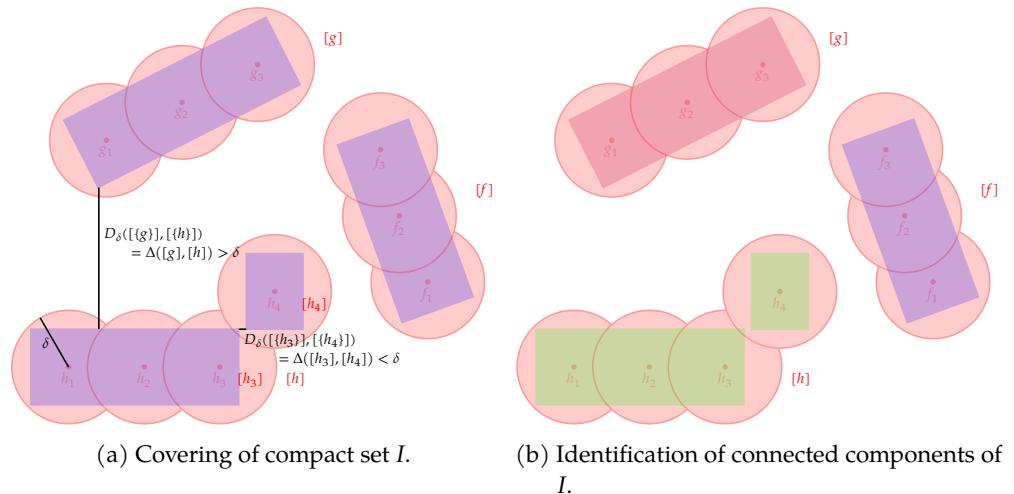


Figure 8.2: The set of connected components I (in blue in Fig. 8.2a) has a finite covering.

In this case, it is covered by 10 balls with radius δ (in red), and has covering number $n_\delta(I) = 10$. The distance between different connected components is larger than the radius of a ball δ . The example shows three different δ -connected components, indicated by color in Fig. 8.2b.

Lemma 8.2.1. *For every compact set I one has that the number of δ -connected components is bounded by the covering number, i.e., $|\tilde{I}^\delta| \leq n_\delta(I)$.*

Proof. We argue by contradiction. Suppose that the covering number is $n_\delta(I)$, and suppose there are $m > n_\delta(I)$ connected components in I . Then there are m points $\{g_i\}_{i=1}^m$ in I such that $d(g_i, g_j) > \delta$ for all $1 \leq i, j \leq m$. Therefore, we need at least m balls of radius δ to cover the set I , so $n_\delta(I) \geq m$. However, by assumption $m > n_\delta(I)$. \square

Remark 8.2.2. It follows from Lemma 8.2.1 that the distance between the closest pair of points between two δ -connected components is larger than δ , that is,

$$\inf_{(g,h) \in [g] \times [h]} d(g, h) > \delta \quad \text{if } [g] \neq [h].$$

We next define several notions of distances that will be needed in subsequent developments.

Definition 8.2.6 (Distance from point to set). *The distance from a point $g \in G$ to a non-empty set $A \subset G$ is given by*

$$d(g, A) := \inf_{a \in A} d(g, a).$$

Definition 8.2.7 (Distance between two sets). *The distance from a non-empty set $A \subset G$ to a non-empty set $B \subset G$ is given by*

$$d(A, B) := \inf_{a \in A, b \in B} d(a, b).$$

Additionally, we will need the concept of an ϵ -thickened set when we explain the δ -connected component algorithm:

Definition 8.2.8 (ϵ -thickened set). *Let $\epsilon > 0$. For every closed subset $A \subset G$, we define its ϵ -thickened version A_ϵ as the set*

$$A_\epsilon := \{g \in G \mid d(g, A) \leq \epsilon\} = \bigcup_{a \in A} \overline{B(a, \epsilon)}.$$

8.2.2 A GENERAL ALGORITHM FOR (DISJOINT) CONNECTED COMPONENTS

Given a threshold $\delta > 0$, the following strategy finds the δ -connected components of a compact set I . We defer the discussion on how to select an optimal value of δ to Section 8.5, and focus here on explaining the algorithm once this parameter is fixed. The strategy consists of a main algorithm `find_all_components` which identifies all δ -connected components relative to the compact set I . It relies on an algorithm that identifies the δ -connected component $[g] \in \tilde{I}^\delta$ of a given $g \in I$ (algorithm `find_full_component`).

Algorithm find_all_components**Find all δ -Connected Components** $[g] \in \tilde{I}^\delta$ for a Compact $I \subset G$ To find all $[g_i] \in \tilde{I}^\delta$ in I , we proceed iteratively:

- $k = 1$:
 1. Pick any $g_1 \in I$.
 2. Compute $[g_1]$ with `find_full_component` to compute δ -connected components, which is introduced in the next box.
 3. Set $\tilde{I}^\delta(1) := \{[g_1]\}$.
 4. Define $I^\delta(1) := \{g \in G \mid [g] \in \tilde{I}^\delta(1)\}$.
- $k > 1$: Given $\tilde{I}^\delta(k-1) = \{[g_1], \dots, [g_{k-1}]\}$ and its associated set $I^\delta(k-1)$.
 1. Search an element $g_k \in I \setminus I^\delta(k-1)$
 - a) if g_k exists: Update

$$\tilde{I}^\delta(k) := \tilde{I}^\delta(k-1) \cup \{[g_k]\}$$
 and $I^\delta(k)$ accordingly.
 - b) if g_k does not exist: all k δ -connected components are identified, i.e., $\tilde{I}^\delta(k-1) = \tilde{I}^\delta$. The algorithm ends.

Algorithm find_full_component**Find δ -Connected Component** $[g]$ for a $g \in G$ To find $[g] \in \tilde{I}^\delta$ for a given g , in practice, we again proceed iteratively:

- $n = 0$: We set $C(g, 0) = \{g\}$.
- $n \geq 1$: Given $C(g, n-1)$, find

$$C(g, n) = C_\delta(g, n-1) \cap I, \quad (8.9)$$

where $C_\delta(g, n-1) := (C(g, n-1))_\delta$ is the δ -thickened set of $C(g, n-1)$ (see Definition 8.2.8).

As we prove next in Lemma 8.2.2, if $C(g, n) = C(g, n-1)$, then $C(g, n) = [g]$ and the algorithm terminates. Otherwise, we go to step $n+1$.

In practice, $C(g, n)$ is computed by means of

$$C(g, n) = C_\delta(g, n - 1) \cap I = \left(\bigcup_{c \in C(g, n - 1)} B(c, \delta) \right) \cap I.$$

As a result, it suffices to compute $B(c, \delta) \cap I$ for every $c \in C(g, n - 1)$ to find the new set $C(g, n)$.

In practice, the thickened sets $C_\delta(g, n)$ in [find_full_component](#) are computed using morphological convolutions which we explain in Section 8.3.

Lemma 8.2.2. *Let $g \in G$ and $n \geq 1$. If $C(g, n) = C(g, n - 1)$, then $C(g, n) = [g]$.*

Proof. We prove the statement by contradiction: Suppose $C(g, n) = C(g, n - 1)$ and $C(g, n - 1) \subsetneq [g]$. Then, there exists a $h \in [g]$ such that $h \notin C(g, n - 1)$ and $d(h, C(g, n - 1)) \leq \delta$ (due to the definition of the equivariance classes $[g]$), so $h \in C_\delta(g, n - 1)$. Since we also have that $h \in I$, it follows that $h \in C_\delta(g, n - 1) \cap I = C(g, n)$. Hence $h \in C(g, n - 1)$, which is a contradiction. \square

8.2.3 HOW THE ALGORITHM WORKS IN AN EXAMPLE

An example of finding connected components in (a discrete subset of) \mathbb{R}^2 is given in Fig. 8.3. Here, we consider an image of a person and a tree. The set I consists of the center points of the pixels of width ρ . Then, we execute the algorithm with $\delta = \rho\sqrt{2}$.

The first δ -connected component is identified in row 2 to 4 of Fig. 8.3. We initiate the algorithm by selecting $g_1 = C(g_1, 0) \in I$, depicted by the green pixel in the first column. Then, the set $C_\delta(g_1, 0)$ is identified in transparent green and contains all pixels whose center point has at most distance δ to the point g_1 . Then, set $C(g_1, 1)$ is shown in non-transparent green in the 2nd row, 3rd column. These steps are repeated until stabilization is reached, so until $C(g_1, n) = C(g_1, n - 1)$ (for $n = 7$ in Fig. 8.3). The final two rows depict the identification of the second connected component in red.

8.3 COMPUTING A THICKENED SET A_ϵ WITH HAMILTON-JACOBI EQUATIONS

The above algorithm hinges on computing δ -thickened versions of the sets $C(g, n)$ in the sense of Definition 8.2.8. In this section, we give a practical algorithm based on morphological dilations on the Lie group G . We carry the discussion for a generic set A first. The transformations used to create a thickened set A_ϵ rely on morphological convolutions as defined next.

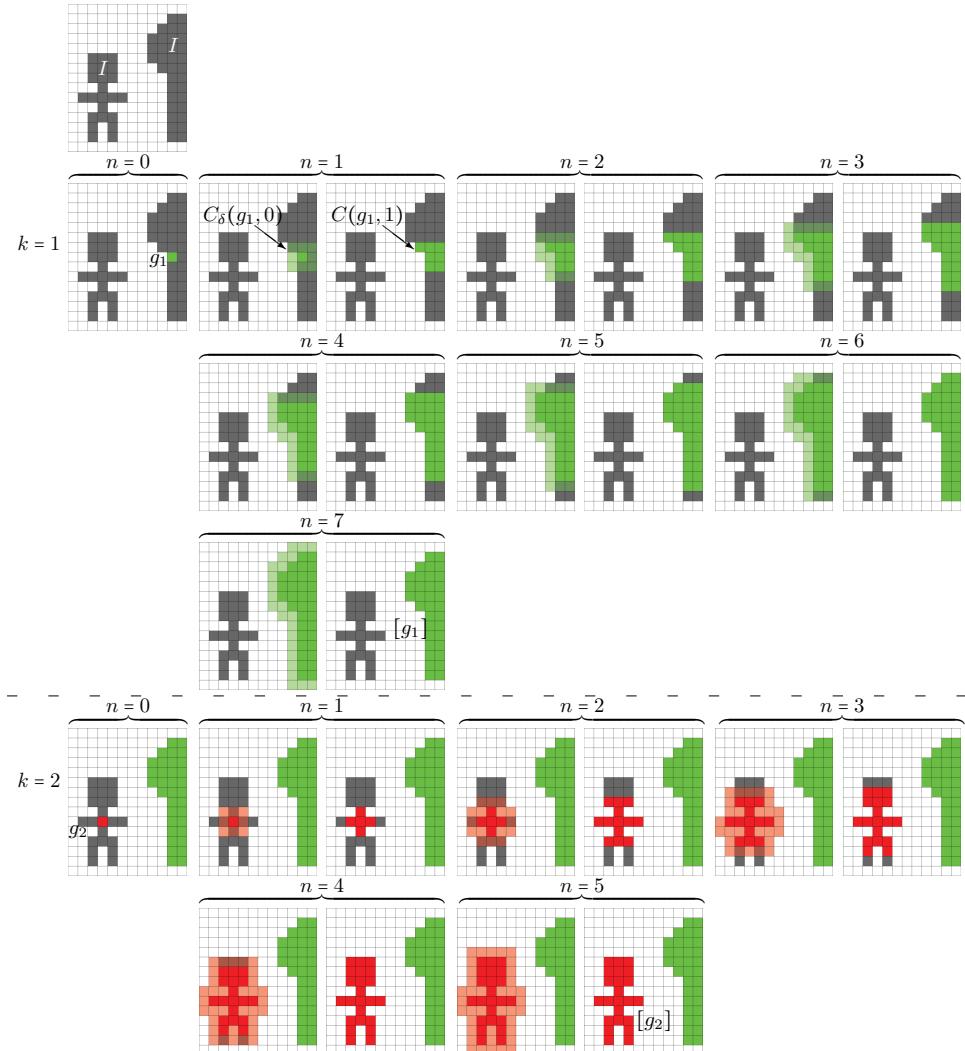


Figure 8.3: A visualization of a connected component algorithm on a set $\Omega = \{1, \dots, 12\} \times \{1, \dots, 15\}$. Here, one sees in the top figure $I = \{\text{grey pixel locations}\}$. There are two separate connected components which are identified in two steps: $k = 1$ shows the process of identifying $[g_1] := \{\text{green pixel locations}\}$ and $k = 2$ shows the process of finding the second connected component $[g_2] := \{\text{red pixel locations}\}$.

Definition 8.3.1 (Morphological convolution). Let G be a Lie group. Let $f_1, f_2 : G \rightarrow \mathbb{R}$ be lower semi-continuous functions, bounded from below. Then the morphological convolution $(f_1 \square f_2) : G \rightarrow \mathbb{R}$ is given by

$$(f_1 \square f_2)(g) = \inf_{h \in G} \{f_1(h^{-1}g) + f_2(h)\}.$$

To create an ϵ -thickened set A_ϵ , we rely on morphological dilations. The concept of morphological dilations is connected to the notion of viscosity solutions of Hamilton-Jacobi-Bellmann (HJB) equations. For a given $\alpha > 1$ and initial condition $f \in C(G)$, we search for a $W : G \times [0, T] \rightarrow \mathbb{R}$ that satisfies the Hamilton-Jacobi-Bellmann equation

$$\begin{cases} \frac{\partial W}{\partial t}(g, t) = \frac{1}{\alpha} \|\nabla_G W(g, t)\|^\alpha, & t \in (0, T] \\ W(g, 0) = f(g), & g \in G. \end{cases} \quad (8.10)$$

The above equation has a unique viscosity solution W which is given by

$$W(g, t) = -(k_t^\alpha \square -f)(g), \quad (8.11)$$

where

$$k_t^\alpha(g) := \frac{t}{\beta} \left(\frac{d(g, e)}{t} \right)^\beta =: \kappa_t^\alpha(d(g, e)) \quad (8.12)$$

is called a morphological kernel with $\frac{1}{\alpha} + \frac{1}{\beta} = 1$, $\alpha, \beta > 1$ and where $\kappa_t^\alpha(x) = \frac{t}{\beta}(x/t)^\beta$, $x \in \mathbb{R}$ denotes the 1D morphological kernel. Eq. (8.12) shows that the morphology on G relates to the 1D morphology on the distance map. For more details on this relation, cf. Appendix A.4.

Definition 8.3.2 (Morphological dilation). *We say that $W(\cdot, t)$ is a morphological dilation off with kernel k_t^α given by Eq. (8.11) if W is the viscosity solution of Eq. (8.10).*

For subsequent developments, it will be useful to view the viscosity solution $W(\cdot, t)$ given in Eq. (8.11) as the image of f under the flow map $\varphi_t^\alpha : C(G) \rightarrow \mathbb{R}$ given by

$$\varphi_t^\alpha(f) := W(\cdot, t) = -(k_t^\alpha \square -f). \quad (8.13)$$

The semigroup property of the flow implies that

$$\varphi_t^\alpha \circ \varphi_s^\alpha(f) = \varphi_{t+s}^\alpha(f), \quad \forall t, s \in \mathbb{R}, \forall f \in C(G). \quad (8.14)$$

Remark 8.3.1. Note that the semigroup property of the flow implies that $\varphi_t^\alpha \circ (n-1) \varphi_t^\alpha f = \varphi_{nt}^\alpha f$ for all $t > 0$. This is due to the well-posedness of Eq. (8.10) in terms of viscosity solutions, but can also be seen more explicitly by looking at the corresponding kernels, see Lemma A.4.1 in Appendix A.4.

Remark 8.3.2 (Relation to sub-Riemannian diffusions). Often, researchers rely on sub-Riemannian diffusion to determine the grouping of nearby objects and points [3, 17, 79, 171]. Linear left-invariant diffusions are solved by a convolution with a heat kernel h_t^α : $(h_t^\alpha * U)(g)$. It is expensive to compute this heat kernel h_t^α exactly, but if the Lie group G is of polynomial growth, one can find an upper and lower bound for the heat kernel with Maheux' heat kernel bounds [186, Lemma 6.6], i.e., then there exist constants $c_1, c_2 > 0$ and for every $\epsilon > 0$ there exists a $c_\epsilon > 0$ so that

$$c_1 \eta_t e^{-\frac{d_{\mathcal{G}}(g,e)^2}{4c_2 t}} \leq h_t^1(g) \leq c_\epsilon \eta_t e^{-\frac{d_{\mathcal{G}}(g,e)^2}{4(1+\epsilon)t}},$$

with normalization constant η_t given by $\eta_t := \mu_{\mathcal{G}}(B(e, \sqrt{t}))^{-1}$, where $\mu_{\mathcal{G}}$ is the volume measure induced by \mathcal{G} . Similarly, one can squeeze the approximated kernel by the real one, i.e., if G is of polynomial growth, there exist constants $C \geq 1, D_1 \in (0, 1), D_2 > D_1$ so that for all $t > 0$

$$\frac{1}{C} h_{D_1 t}^1(g) \leq h_t^{1,\text{approx}}(g) \leq C K_{D_2 t}^1(g),$$

see [184, Sec.5.2 Lemma 24]. However, dilations and erosions are *exactly* solved by $(k_t^\alpha \square U)(g)$, where k_t^α is introduced in Eq. (8.12), resulting in more accurate grouping results. Akin to optimal transport on SE(2) [34], one could rely for δ -connected components on Varadhan's Theorem, i.e., $d_{\mathcal{G}}^2(g, e) = -\lim_{t \rightarrow 0} (4t) \log k_t^\alpha(g, e)$ [189]. However, this would be less accurate [34, Fig.5.1&5.2] than applying HJB-solvers with morphological convolutions, also when using logarithmic norm approximations.

We next list some useful properties connected to the semigroup property of the HJB equation, where we use the indicator function and the support:

Definition 8.3.3 (Indicator function). *The indicator function $\mathbb{1}_S : G \rightarrow \{0, 1\}$ of some set $S \subset G$ is defined as*

$$\mathbb{1}_S(x) := \begin{cases} 1 & \text{if } x \in S, \\ 0 & \text{else.} \end{cases}$$

Definition 8.3.4 (Support of a function). *The support of the function $f : G \rightarrow \mathbb{R}$ is given by the closure:*

$$\text{supp}(f) := \overline{\{g \in G \mid f(g) \neq 0\}}.$$

Lemma 8.3.1 (Properties morphological dilations). *Let $\alpha \geq 1$. Let $\frac{1}{\alpha} + \frac{1}{\beta} = 1$. For a given closed set $A \subset G$ and a function $f : G \rightarrow [0, 1]$, one has*

$$\varphi_t^\alpha(f)(g) \in [0, 1], \quad \forall g \in G, \quad (8.15)$$

$$A_{\varepsilon(t, \alpha)} = \text{supp}(\varphi_t^\alpha(\mathbb{1}_A)), \quad \forall t \geq 0, \quad (8.16)$$

$$\mathbb{1}_{A_t} = (\varphi_t^1(\mathbb{1}_A)), \quad (8.17)$$

with

$$\varepsilon(t, \alpha) = t^\beta \sqrt{\frac{\beta}{t}}, \quad (8.18)$$

and where $A_t, A_{\varepsilon(t, \alpha)}$ are t - and $\varepsilon(t, \alpha)$ -thickened sets of A , as defined in Definition 8.2.8.

Proof. We prove the statements one by one, starting with Eq. (8.15): Let $f : G \rightarrow [0, 1]$. Then, as the kernels are positive (8.12), we have for all $g \in G$

$$\begin{aligned} \varphi_t^\alpha(f)(g) &\stackrel{(8.13)}{=} -(k_t^\alpha \square - f)(g) = \sup_{h \in G} \{f(h) - k_t^\alpha(h^{-1}g)\} \\ &\leq \|f\|_\infty = 1, \end{aligned}$$

and

$$\begin{aligned} \varphi_t^\alpha(f)(g) &\stackrel{(8.13)}{=} -(k_t^\alpha \square - f)(g) = \sup_{h \in G} \{f(h) - k_t^\alpha(h^{-1}g)\} \\ &\geq \sup_{h \in G} \{-k_t^\alpha(h^{-1}g)\} = 0. \end{aligned}$$

The two estimates above prove Eq. (8.15).

We next prove Eq. (8.16): We are interested in the support of $\varphi_t^\alpha(\mathbb{1}_A)$. From Eq. (8.15), we know

$$\varphi_t^\alpha(\mathbb{1}_A) = -(k_t^\alpha \square - \mathbb{1}_A) \in [0, 1].$$

By definition of the support and the positivity of $\varphi_t^\alpha(\mathbb{1}_A)$, we have $\text{supp}(\varphi_t^\alpha(\mathbb{1}_A)) = \overline{\{g \in G \mid \varphi_t^\alpha(\mathbb{1}_A)(g) > 0\}}$, so we aim to find the set B , such that for all $g \in B$, we have

$$0 < \varphi_t^\alpha(\mathbb{1}_A)(g) = \sup_{h \in G} \left\{ \mathbb{1}_A(h) - \frac{t}{\beta} \left(\frac{d(h^{-1}g, e)}{t} \right)^\beta \right\} \leq 1, \quad (8.19)$$

Note that $\overline{B} = \text{supp}(\varphi_t^\alpha(\mathbb{1}_A))$ and that the upper boundary is always satisfied as the kernel k_t^α , recall Eq. (8.12), is positive and $\mathbb{1}_A \leq 1$. The supremum is only

strictly larger than zero if the supremum is reached for $h \in A$, so $A \subset \overline{B}$. Then, Eq. (8.19) gives

$$0 < 1 - \frac{t}{\beta} \left(\frac{d(g, A)}{t} \right)^\beta \Leftrightarrow d(g, A) < t^{\beta} \sqrt{\frac{\beta}{t}} \text{ for } g \in B.$$

Taking $g \in \overline{B}$ gives the condition $d(g, A) \leq t^{\beta} \sqrt{\frac{\beta}{t}}$. Therefore, the set $\overline{B} = A_{\varepsilon(t, \alpha)}$, where $A_{\varepsilon(t, \alpha)}$ is a thickened set of A as defined in Definition 8.2.8. This proves Eq. (8.16).

The last statement in Eq. (8.17) follows immediately from Eq. (8.16) by taking the limit of $\alpha \downarrow 1$, combined with

$$\begin{aligned} k_t^1(g) &= \lim_{\alpha \downarrow 1} k_t^\alpha(g) = \lim_{\beta \rightarrow \infty} \frac{t}{\beta} \left(\frac{d(g, e)}{t} \right)^\beta \\ &= \begin{cases} 0 & \text{if } d(g, e) \leq t, \\ \infty & \text{else.} \end{cases} \end{aligned} \tag{8.20}$$

□

We conclude this section by connecting one last time to the main goal in this part of the chapter which is to explain how to create an ϵ -thickened set A_ϵ . The main message is that A_ϵ can be obtained through Eqs. (8.16) and (8.17), which crucially rely on morphological dilations. In practice, it is useful to rely on Eq. (8.17) to determine the ϵ -thickened set A_ϵ in the δ -connected component algorithm, as we will see in the next section.

8.4 COMPUTING δ -CONNECTED COMPONENTS USING ITERATIVE MORPHOLOGICAL CONVOLUTIONS

In Section 8.2.2, we explained the δ -connected component algorithm in a general setting. In Section 8.4.2, we will explain how the δ -connected component algorithm is constructed using morphological dilations, but first, we discuss the choice of the parameter α in the morphological dilation kernel k_t^α in Section 8.4.1. We finish with a convergence analysis for the algorithm in Section 8.4.3.

8.4.1 FIXING $\alpha = 1$ IN THE δ -CONNECTED COMPONENT ALGORITHM

We aim to construct a δ -connected component algorithm that relies on morphological dilations. Before we introduce the algorithm, we discuss the choice of the parameter $\alpha \geq 1$ in the morphological dilation kernel k_t^α (8.12).

We recall that identifying the δ -connected components is an iterative procedure as introduced in Section 8.2.2. The identification of one single δ -connected component is mainly described by Eq. (8.9). In every iteration step n , we create a δ -thickened set $C_\delta(g, n - 1)$ and take the intersection with a reference set $I \subset G$, resulting in an updated set $C(g, n)$. This is repeated until the updated set is the same as the initial set, i.e., $C(g, n) = C(g, n - 1)$.

Hence, the set $C(g, n)$ is thickened iteratively, which motivates the identification of the maximum radius of influence of one single point after n morphological dilations, as will be introduced in Definition 8.4.1. The maximum radius of influence follows immediately from the semigroup property in Remark 8.3.1 and the support after applying the flow to an indicator function in Eq. (8.16).

Definition 8.4.1 (Maximum radius of influence). *The maximum radius of the region of influence of a specific point after n applications of a morphological dilation of time t and kernel steepness α is denoted by*

$$d_{n,t}^\alpha := nt \left(\frac{\beta}{nt} \right)^{\frac{1}{\beta}} = \varepsilon(nt, \alpha),$$

where $\frac{1}{\alpha} + \frac{1}{\beta} = 1$ and $\varepsilon(t, \alpha)$ was defined in Eq. (8.18).

We recall that an ε -thickened set was produced with Eqs. (8.16) and (8.17). We aim to identify δ -connected components. Therefore, we want the support of the set to grow with rate δ in every iteration step while applying Eq. (8.16). That means we would like to have that $(A_{\varepsilon(t, \alpha)})_{\varepsilon(t, \alpha)} = A_{\varepsilon(2t, \alpha)}$ which is only satisfied if $\beta \rightarrow \infty \Leftrightarrow \alpha = 1$ as is clear from Eq. (8.18).

If $\alpha > 1$, the algorithm cannot always identify the full δ -connected component, while using the same kernel k_t^α in every set, as is visualized in Fig. G.1 in Appendix G. Therefore, in our computations of the δ -connected components, we set $\alpha = 1$ and apply morphological convolutions with k_t^1 , given by Eq. (8.20), with $\delta := t$.

8.4.2 δ -CONNECTED COMPONENT ALGORITHM

Next, we detail how we implement the general δ -connected component algorithm as discussed in Section 8.2.2 using the morphological dilations that were discussed in Section 8.3. More specifically, we explain how the δ -thickened set C_δ in `find_full_component` is determined using the properties of morphological dilations as stated in Lemma 8.3.1.

First, we discuss how to identify one single δ -connected component $[g_i]$ for a $g_i \in I$. We initialize

$$C(g_i, 0) = \{g_i\} = \text{supp}(\mathbb{1}_{\{g_i\}}).$$

During iteration step n of identifying one single δ -connected component $[g_i]$, the set $C(g_i, n - 1)$ from the previous iteration is expanded with a radius δ using morphological dilations and Eq. (8.17) in Lemma 8.3.1.

Therefore, we define a function $U : G \times \mathbb{N} \rightarrow \{0, 1\}$ which is the indicator function of the computed δ -connected component initialized in $g_i \in I$ at step n , i.e.,

$$U_{g_i}(\cdot, n - 1) := \mathbb{1}_{C(g_i, n - 1)}(\cdot).$$

This allows us to calculate the set $C(g_i, n)$ using morphological dilations on the function $U_{g_i}(\cdot, n - 1)$

$$U_{g_i}(\cdot, n) := \mathbb{1}_I(\cdot) \varphi_\delta^1(U_{g_i}(\cdot, n - 1)), \quad (8.21)$$

where

$$C(g_i, n) := \text{supp}(U_{g_i}(\cdot, n)). \quad (8.22)$$

These steps are repeated until $C(g_i, n) = C(g_i, n - 1)$. Then, we update $I = I \setminus C(g_i, n)$ and pick a new $g_{i+1} \in I$ to identify the next component as explained in `find_all_components` in Section 8.2.2. The complete algorithm is also stated in Algorithm 1: `find_δCC` and produces δ -connected components as stated in the next proposition.

Proposition 8.4.1. *Algorithm 1: `find_δCC` produces δ -connected components, i.e., $[g_1], \dots, [g_K]$ are all δ -connected.*

The result follows immediately from Eq. (8.21), using Eq. (8.17) in Lemma 8.3.1 where $t = \delta$.

Proposition 8.4.1 shows that `find_δCC` finds δ -connected components. It remains to be shown that the algorithm is indeed a stable algorithm which we will address next.

8.4.3 CONVERGENCE ANALYSIS

In this section, we prove in Theorem 8.4.1 that Algorithm 1: `find_δCC` computes the δ -connected components of any compact set I on any Lie group G in a finite number of calculation steps for $\alpha = 1$.

For this, we need to have an explicit non-recursive expression of $U_{g_0}(\cdot, n)$ because this describes the state of the δ -connected component algorithm in the n -th iteration step. In Proposition 8.4.2, we find such a non-recursive expression for $U_{g_i}(\cdot, n)$, where Definition 8.4.1 and Lemma 8.3.1 are key ingredients in proving the convergence of our δ -connected component algorithm (Theorem 8.4.1).

Algorithm 1 `find_δCC`

Input: Binary map $\mathbb{1}_I : G \rightarrow \{0, 1\}$, where $I \subset G$ the region of interest for the connected components, kernel k_δ^α with $\alpha = 1, \delta > 0$

set $k = 0$

set $\mathcal{J} = I$

while $\mathcal{J} \neq \emptyset$:

update $k = k + 1$

pick $g_k \in \mathcal{J}$

initialize $U_{g_k}(\cdot, 0) = \mathbb{1}_{\{g_k\}}$

initialize connected component sets $C(g_k, -1) := \emptyset$,

$C(g_k, 0) = \text{supp}(U_{g_k}(\cdot, 0)) = \{g_k\}$.

set $n := 0$

while n such that $C(g_k, n) \neq C(g_k, n - 1)$:

do a morphological convolution calculating $U_{g_k}(\cdot, n + 1)$ (see Eq. (8.21)).

create set $C(g_k, n + 1) = \text{supp}(U_{g_k}(\cdot, n + 1))$ (see Eq. (8.22))

update $n := n + 1$

end

set $[g_k] := C(g_k, n)$

update $\mathcal{J} = \mathcal{J} \setminus [g_k]$

end

$K = k$

Output: Connected components $[g_1], \dots, [g_K]$.

Proposition 8.4.2. Let $\alpha = 1$, $t = \delta > 0$, $g \in G$, $n \in \mathbb{N}$. Let $g_0 \in G$ be the reference point. Set $\delta = d_{1,t}^1$. Then, we have, for $U_{g_0}(g, n)$ given by Eq. (8.21), that $0 \leq U_{g_0}(g, n) \leq 1$ where

$$U_{g_0}(g, n) = \begin{cases} 1 & \text{if } g \sim g_0 \wedge m_\delta(g, g_0) \leq n \\ 0 & \text{else.} \end{cases},$$

where $m_\delta(g, g_0)$ was defined in Definition 8.2.4 and Remark 8.2.1.

For the details of the proof of Proposition 8.4.2, see Appendix A.5. Intuitively, the condition $m_\delta(g, g_0) \leq n$ ensures that the n steps of the connected component algorithm are sufficient to reach the point g starting from g_0 .

Knowing the algorithm's state after n iterations, allows us to confidently say that the algorithm can identify all δ -connected components in a finite number of steps, as proved in the next theorem.

Theorem 8.4.1 (Convergence of Algorithm 1: `find_delta_cc`). Let $\delta > 0$, $\alpha = 1$, and assume that I has K connected components $[g_1], \dots, [g_K]$ such that $I = [g_1] \cup \dots \cup [g_K] \subset G$ and $[g_i] \cap [g_j] = \emptyset$ for all $i \neq j \in \{1, \dots, K\}$. Then Algorithm 1: `find_delta_cc` correctly finds all the δ -connected components in a finite number of steps.

Proof. Consider $\alpha = 1$. Let $i \in \{1, \dots, K\}$ be given. Note that in this case, one has the dilated volume $U_{g_i}(\cdot, n) : G \rightarrow \{0, 1\}$ and the connected components

$$C(g_i, n) = \text{supp}(U_{g_i}(\cdot, n)) = \{g \in G \mid U_{g_i}(g, n) \neq 0\}.$$

In the proof of Proposition 8.4.2, we defined the set $\tilde{B}(g_i; n)$ by

$$\tilde{B}(g_i; n) := \{g \in I \mid m_\delta(g_i, g) \leq n\}. \quad (8.23)$$

Then, by Proposition 8.4.2 where we set g_i instead of g_0 , one has

$$U_{g_i}(g, n) = \mathbb{1}_{[g_i]}(g) \cdot \mathbb{1}_{\tilde{B}(g_i; n)}(g)$$

$$C(g_i, n) = \tilde{B}(g_i; n).$$

From the compactness of I and the fact that $I = \bigcup_i [g_i]$, $[g_i] \cap [g_j] = \emptyset$, one can deduce that $[g_i]$ is closed. Therefore, also $[g_i]$ is compact. Consequently, its covering number $n_\delta([g_i])$, as introduced in Eq. (8.8), is finite and thus $U_{g_i}(\cdot, n)$ converges in at most $n = n_\delta([g_i]) + 1 < \infty$ steps (see Corollary 8.4.1) to $[g_i]$ (Proposition 8.4.1), i.e., $U_{g_i}(\cdot, n_\delta([g_i]) + 1) = U_{g_i}(\cdot, n_\delta([g_i]) + 2)$ and thus $C(g_i, n_\delta([g_i]) + 1) = C(g_i, n_\delta([g_i]) + 2)$. \square

As a direct consequence of Theorem 8.4.1, we can give an upper bound for the maximum number of iterations that `find_δCC` needs to identify all δ -connected components.

Corollary 8.4.1 (Upper bound number of iterations). *Let $I = [g_1] \cup \dots \cup [g_K] \subset G$ compact, where the δ -connected components $[g_i] \cap [g_j] = \emptyset$ for all $i \neq j \in \{1, \dots, K\}$. The δ -connected component algorithm given in Algorithm 1: `find_δCC` finishes in at most $n = n_\delta(I) + K$ steps.*

Proof. We start by noticing that $n_\delta(I) = \sum_{i=1}^K n_\delta([g_i])$ because $d([g_i], [g_j]) > \delta$ for $i \neq j$.

To identify δ -connected component $[g_i]$, we need at most 1 step to get from the starting point $g_i \in I$ to a point $\tilde{g}_i \in C$, where C is the minimal δ -covering of I . Then, by Lemma 8.2.1, we need at most $n_\delta([g_i])$ steps to identify the full δ -connected component $[g_i]$.

Hence, identifying all δ -connected components in $I = \bigcup_{i=1}^K [g_i]$ is done in at most $\sum_{i=1}^K 1 + n_\delta([g_i]) = K + n_\delta(I)$ steps. \square

Remark 8.4.1 (Complexity). The presented method has a complexity of $O(nN_oK_oN_xN_yK_s^2)$ in SE(2) [99, Table 3.2], where $n := n_\delta + K$ denotes the maximum number of iterations of the algorithm, N_x , N_y , and N_o the dimensions of the orientation score in the x -, y - and θ -direction, and K_s , and K_o the spatial dimensions and the number of sampled orientations of the SE(2)-kernel, respectively. Note that in [99], they consider linear convolutions instead of morphological convolutions. However, the complexity directly carries over from the linear to the morphological setting by replacing the semifield $(\mathbb{R}, \cdot, +)$ by the semifield $(\mathbb{R}, +, \inf)$.

In a more general setting, consider a Lie group G of dimension $\dim(G)$. Then, the complexity of the morphological convolution is $O(\prod_{i=1}^{\dim(G)} N_i K_i)$, where the image data and the kernel have dimensions $N_1 \times \dots \times N_{\dim(G)}$ and $K_1 \times \dots \times K_{\dim(G)}$, respectively. At most $n := n_\delta + K$ morphological convolutions are calculated, leading to a complexity of $O(n \prod_{i=1}^{\dim(G)} N_i K_i)$.

8.5 CHOOSING δ WITH PERSISTENCE DIAGRAMS

In the previous section, we have discussed how to find δ -connected components using morphological convolutions. We chose a fixed δ at the start of the algorithm, and use `find_δCC` to identify all δ -connected components. Consequently, the algorithm's output heavily depends on the choice of δ (recall Fig. 8.2).

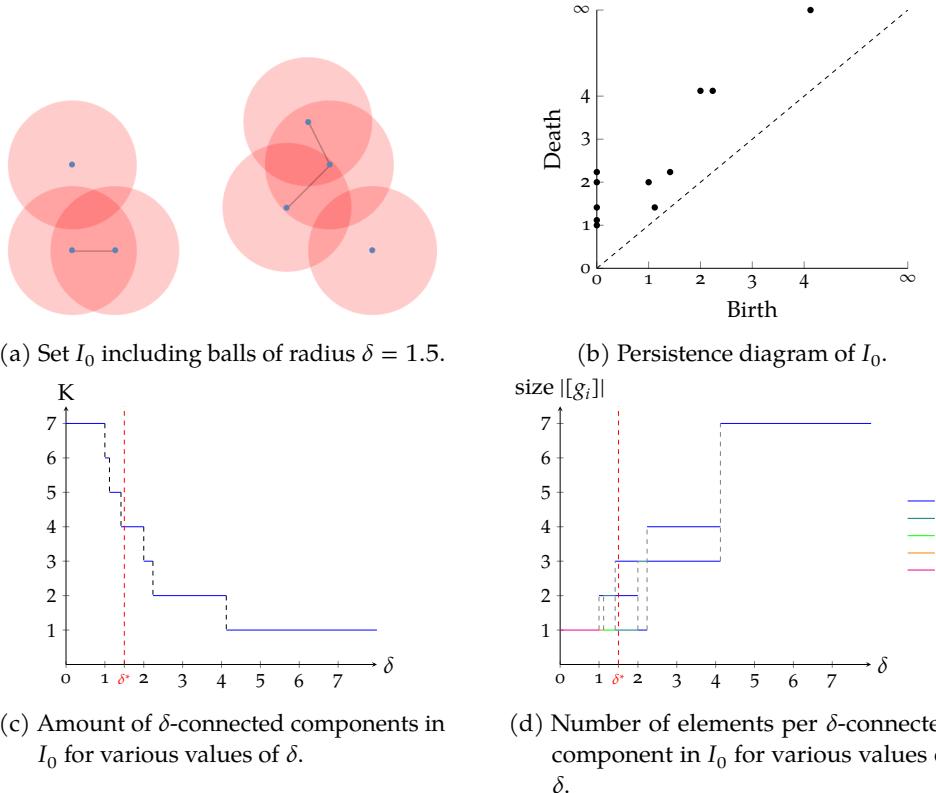


Figure 8.4: Persistence plots for set $I_0 \subset \mathbb{R}^2$. The longest lines (most values of δ) denote the most persistent connected components. Depending on the application, the optimal choice of δ is where the δ -connected components are the most stable/persistent (i.e., long lines in Figs. 8.4c and 8.4d) while aiming for the smallest possible choice of δ (to be as distinctive as possible).

To gain some insight into the behavior of the δ -connected components, it is common in Topological Data Analysis to create so-called *persistence diagrams* to study the stability of the components [46, 86, 183, 209]. Inspired by this approach, we visualize the persistence of components in three different ways. Firstly, we create the classical persistence diagrams from topological data analysis. Secondly, we create a figure where we plot the total number of δ -connected components for different values of δ . When the distance between two components is smaller than the considered δ , they merge and hence the number of connected components decreases. Lastly, we visualize, for sets of points, how many points are grouped in each δ -connected component. These plots (for instance in Figs. 8.4 and 8.5) allow us to verify the persistence of the δ -connected components. Depending on the application, we want them to be as persistent as possible, e.g., when looking at vascular images, or we want to choose δ small

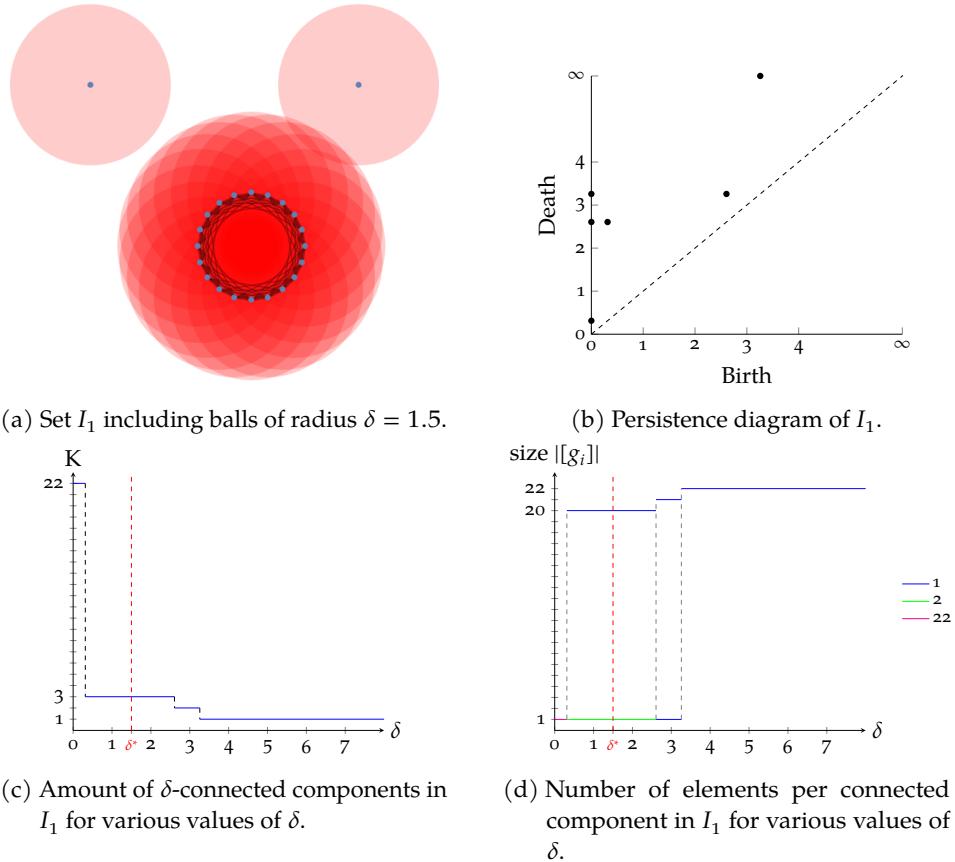


Figure 8.5: Persistence plots for set $I_1 \subset \mathbb{R}^2$. The longest lines (most values of δ) denote the most persistent connected components. Depending on the application, the optimal choice of δ is where the δ -connected components are the most stable/persistent (i.e., long lines in Figs. 8.5c and 8.5d) while aiming for the smallest possible choice of δ (to be as distinctive as possible).

enough to differentiate between the smallest possible δ -connected components. Note that using a very small value of δ results in a high number of δ -connected components due to noise being assigned its own δ -connected component.

In Figs. 8.4 and 8.5, we show the persistence diagrams for two examples in \mathbb{R}^2 . The first concerns a set of points (7 in total) that belong to our set I_0 , depicted in Fig. 8.4a. Then, we determine the δ -connected components for a range of values of δ . We plot the number of connected components per value of δ (cf. Fig. 8.4c) and the number of vertices that are part of every δ -connected component (cf. Fig. 8.4d). One can see from the plots that the most stable connected components (at least one connection, but not everything is connected) occur when $\delta \in (2.2, 4.1)$.

We do the same for the second example, a different set I_1 , where 20 points are placed in a circle, and with two outliers (see Fig. 8.5a). Then, the persistence graphs in Fig. 8.5c and Fig. 8.5d show a stabilization for $\delta \in (0.3, 2.6)$.

In Fig. 8.6, we do the same, but now on the Lie group $G = \text{SE}(2)$. Again, we see persistence of the δ -connected components for $\delta \in (0.57, 1.13)$ and $\delta \in (1.13, 1.92)$ with 6 and 4 δ -connected components respectively. Therefore, choosing δ in these intervals results in the most persistent δ -connected components.

8.6 AFFINITY MATRICES BETWEEN δ -CONNECTED COMPONENTS

Now that we have described a way to choose the parameter δ we can calculate the δ -connected components by Algorithm 1: `find_delta_cc`. Then we would like to quantify how well-aligned the δ -connected components are as a whole. For this we introduce and analyze affinities between connected components in this section. Intuitively, *affinity* measures the proximity and alignment of δ -connected components.

We define the affinity matrix $A = (a_{ij})$ by

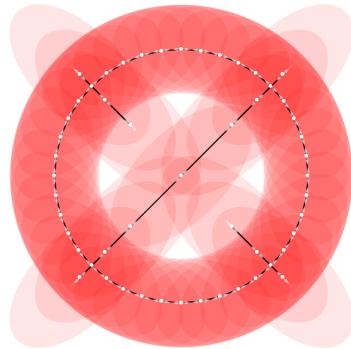
$$a_{ij} = \sup \left\{ \left(\frac{1}{\mu([g_i])} \int_{[g_i]} \varphi_t^\alpha \left(W_{[g_i]}^{(0)} \right) (h)^p dh \right)^{1/p}, \right. \\ \left. \left(\frac{1}{\mu([g_j])} \int_{[g_j]} \varphi_t^\alpha \left(W_{[g_j]}^{(0)} \right) (h)^p dh \right)^{1/p} \right\}, \quad (8.24)$$

with measure normalisation $\mu([g_i]) := \int_{[g_i]} dg$, and $\alpha > 1$, $t > 0$, $p \geq 1$ fixed, and where $W_{[g_i]}^{(0)}$ is given by

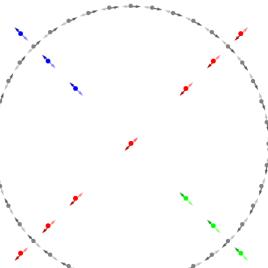
$$W_{[g_i]}^{(0)}(g) = \frac{D(g) \mathbb{1}_{[g_i]}(g)}{\sup_{h \in [g_i]} \{D(h)\}}, \quad g \in G, \quad (8.25)$$

where $D : G \rightarrow [0, 1]$ is a (non-zero) data term that can be freely chosen. In the experimental section, we chose the data term D to be the absolute value of the orientation score, defined in Eq. (7.1), rescaled to be between 0 and 1. Let us also recall that the flow operator φ_t^α was given by Eq. (8.13) and implemented by the morphological convolution in Eq. (8.11). We say a_{ij} denotes the affinity between δ -connected components $[g_i]$ and $[g_j]$.

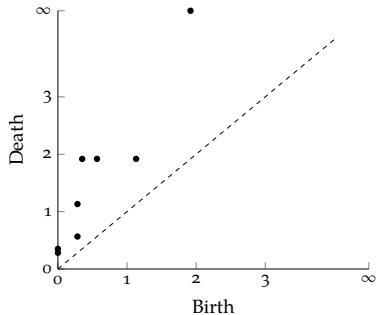
In the computation of the affinity matrices we fix the parameter t , based on the choice of α and the compact set I , such that the dilation of any δ -connected component $[g_i]$ has non-zero values for all $h \in I$. How this can be achieved is explained in the next proposition.



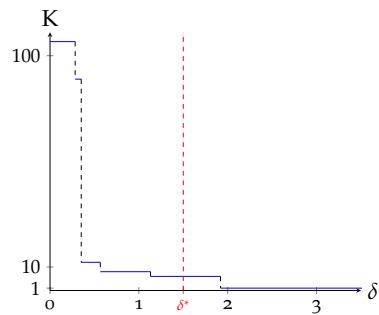
(a) Set $I_0 \subset \text{SE}(2)$ including balls of radius $\delta^* = 0.7$.



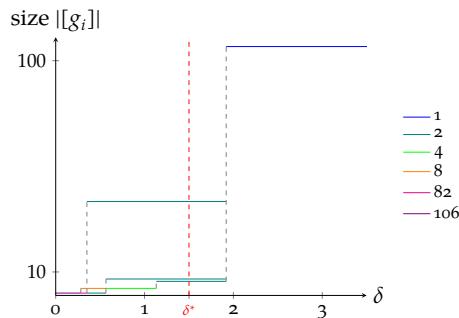
(b) Different δ -connected components identified with $\delta^* = 0.7$.



(c) Persistence diagram of I_0 .



(d) Amount of δ -connected components in I_0 for various values of δ .



(e) Number of elements per connected component in I_0 for various values of δ .

Figure 8.6: Persistence plots for an example set $I \subset G = \text{SE}(2)$ using metric tensor field parameters $(w_1, w_2, w_3) = (1, 2, 2)$. The longest lines (most values of δ) denote the most persistent connected components. Depending on the application, the optimal choice of δ is where the δ -connected components are the most stable/persistent (i.e., long lines in Figs. 8.6d and 8.6e) while aiming for the smallest possible choice of δ (to be as distinctive as possible).

Proposition 8.6.1 (Non-zero affinity between δ -connected components). *Let $\alpha > 1$ be given and $\frac{1}{\alpha} + \frac{1}{\beta} = 1$. Let $I = \bigcup_{k=1}^K [g_k] \subset G$ compact be given, $[g_i] \neq [g_j]$ for all $i \neq j$. Then*

$$\varphi_t^\alpha \left(W_{[g_i]}^{(0)} \right) (g) \neq 0 \text{ for all } g \in I, [g_i] \in \tilde{I}^\delta$$

at least when $t > \left(\sup_{q_1, q_2 \in I} d(q_1, q_2) \right)^\alpha \beta^{1-\alpha}$.

Proof. We need to show that one dilation step with morphological kernel k_t^α on $W_{[g_i]}^{(0)}(g)$ results in all non-zero values for all $g \in I$. That means that we start with calculating:

$$\begin{aligned} 0 \leq \varphi_t^\alpha \left(W_{[g_i]}^{(0)} \right) (g) &= - \left(k_t^\alpha \square - W_{[g_i]}^{(0)} \right) (g) \\ &= \sup_{h \in G} \left\{ W_{[g_i]}^{(0)}(h) - \frac{t}{\beta} \left(\frac{d(g, h)}{t} \right)^\beta \right\} \leq 1, \end{aligned}$$

where we find the inequalities by Lemma 8.3.1. Then, we calculate the value of $t > 0$ such that

$$0 < \sup_{h \in I} \left\{ W_{[g_i]}^{(0)}(h) - \frac{t}{\beta} \left(\frac{d(g, h)}{t} \right)^\beta \right\} \leq 1.$$

By definition of $W_{[g_i]}^{(0)}$, for any element in the connected component $q \in [g_i]$ where the data does not vanish ($D(q) \neq 0$), we have

$$W_{[g_i]}^{(0)}(q) - \frac{t}{\beta} \left(\frac{d(g, q)}{t} \right)^\beta \leq \sup_{h \in I} \left\{ W_{[g_i]}^{(0)}(h) - \frac{t}{\beta} \left(\frac{d(g, h)}{t} \right)^\beta \right\} \leq 1.$$

We are sure that all affinities are non-zero when

$$W_{[g_i]}^{(0)}(q) - \frac{t}{\beta} \left(\frac{d(g, q)}{t} \right)^\beta > 0, \text{ for all } g \in I.$$

By definition of $W_{[g_i]}^{(0)}$ in Eq. (8.25), there exists a $q \in [g_i]$ such that $W_{[g_i]}^{(0)}(q) = 1$, so that for all $g \in I$

$$\begin{aligned} W_{[g_i]}^{(0)}(q) - \frac{t}{\beta} \left(\frac{d(g, q)}{t} \right)^\beta &= 1 - \frac{t}{\beta} \left(\frac{d(g, q)}{t} \right)^\beta > 0 \\ \Leftrightarrow t > \left(\frac{d(g, q)^\beta}{\beta} \right)^{\frac{1}{\beta-1}} &= (d(g, q) \beta^{-1/\beta})^\alpha = (d(g, q))^\alpha \beta^{1-\alpha} \end{aligned}$$

for all $g \in I$, and where $\frac{1}{\alpha} + \frac{1}{\beta} = 1$ with $\alpha, \beta \geq 1$ and the last estimate holds because the distance $d(g, q)$ is positive. Then, since the inequality has to hold for all $q \in I$ and should be independent of the origin, we have

$$t > \left(\sup_{q_1, q_2 \in I} d(q_1, q_2) \right)^\alpha \beta^{1-\alpha} \geq \left(\sup_{\substack{q_1 \in I, \\ q_2 \in [g_i]}} d(q_1, q_2) \right)^\alpha \beta^{1-\alpha}.$$

□

Algorithm 2: `find_affinity` describes the algorithm to find the affinity matrices between the δ -connected components of any set $I \subset G$.

Algorithm 2 `find_affinity`

Input: δ -connected components $[g_1], \dots, [g_K]$ determined by `find_dcc`, $I = \bigcup_{k=1}^K [g_k] \subset G$ region of interest for the connected components, kernel k_t^α with $\alpha > 1$, $t > 0$, data $D : G \rightarrow [0, 1]$, parameter $p \geq 1$.

set $t > \left(\sup_{q_1, q_2 \in I} d(q_1, q_2) \right)^\alpha \beta^{1-\alpha}$ fixed

set $k = 1$

for $k \leq K$

initialize $W_{[g_k]}^{(0)}(g) := \frac{D(g) \cdot \mathbb{1}_{[g_k]}(g)}{\sup_{h \in [g_k]} \{D(h) \cdot \mathbb{1}_{[g_k]}(h)\}}$

update $W_{[g_k]}^{(1)}(g) = \varphi_t^\alpha \left(W_{[g_k]}^{(0)} \right)(g) = - \left(k_t^\alpha \square - W_{[g_k]}^{(0)} \right)(g)$

for $1 \leq l \leq K$

set $\tilde{a}_{kl} := \frac{\left(\int_{[g_l]} \varphi_t^\alpha \left(W_{[g_k]}^{(0)} \right)(h)^p dh \right)^{1/p}}{\left(\int_{[g_l]} dh \right)^{1/p}}$

update $k := k + 1$

end

for $i, j = 1, \dots, K$

set $a_{ij} = \sup \{ \tilde{a}_{ij}, \tilde{a}_{ji} \}$

end

Output: Affinity Matrix $A = (a_{ij})$ for all $i, j = 1, \dots, K$.

If the data $D(\cdot)$ is constant and non-zero in the connected components, we can give an upper and lower bound for the affinity of $[g_i]$ on $[g_j]$.

Proposition 8.6.2. *Let the data $D : G \rightarrow [0, 1]$ be such that $D(g) = c_{[g]} \neq 0$ for all $g \in I$, where $c_{[g]}$ is constant in the same δ -connected component. Let $\alpha > 1$ be given and $\frac{1}{\alpha} + \frac{1}{\beta} = 1$. Let $t > \sup_{q_1, q_2 \in I} d(q_1, q_2)^\alpha \beta^{1-\alpha}$ fixed.*

Then the affinities on and off-diagonal satisfy respectively

$$a_{ii} = 1 \quad \text{and} \quad 0 < 1 - \frac{t}{\beta} \left(\frac{r}{t} \right)^\beta \leq a_{ij} < 1 - \frac{t}{\beta} \left(\frac{\delta}{t} \right)^\beta < 1$$

for all $[g_i] \neq [g_j] \subset I$, with diameter $r = \sup_{q_1, q_2 \in I} d(q_1, q_2)$.

Proof. First, we note that since the data term $D(g) = c_{[g]} \neq 0$ for all $g \in I$, with $c_{[g]}$ constant in the same δ -connected component, we have

$$W_{[g_i]}^{(0)}(g) = \mathbb{1}_{[g_i]}(g).$$

Then, the morphological dilation of the initialization yields

$$W_{[g_i]}^{(1)}(g) := \varphi_\alpha^t(W_{[g_i]}^{(0)})(g) = 1 - \inf_{h \in [g_i]} \frac{t}{\beta} \left(\frac{d(g, h)}{t} \right)^\beta \quad (8.26)$$

where we used the relation $\frac{1}{\alpha} + \frac{1}{\beta} = 1$, with $\alpha > 1$. The distance satisfies two conditions:

1. If $g \in [g_i]$, then $\inf_{h \in [g_i]} d(g, h) = 0$.
2. If $g \notin [g_i]$, then $\delta < \inf_{h \in [g_i]} d(g, h) \leq r := \sup_{g, h \in I} d(g, h)$. The upper bound follows from $[g_i] \subset I$. The lower bound follows from g not being part of the δ -connected component $[g_i]$. Recall that by Definition 8.2.5, we know that the distance between two δ -connected components is larger than δ .

Using this information on the distances, we can further simplify the expression in Eq. (8.26) to

$$W_{[g_i]}^{(1)}(g) = 1 \text{ if } g \in [g_i], \quad (8.27)$$

and

$$1 - \frac{t}{\beta} \left(\frac{r}{t} \right)^\beta \leq W_{[g_i]}^{(1)}(g) < 1 - \frac{t}{\beta} \left(\frac{\delta}{t} \right)^\beta \text{ if } g \notin [g_i], \quad (8.28)$$

using the distance estimates in items 1 and 2. Then, the affinity (8.24) equals $a_{ij} = \sup\{\tilde{a}_{ij}, \tilde{a}_{ji}\}$ with

$$\tilde{a}_{ij} := \left(\frac{1}{\mu([g_j])} \int_{[g_j]} (W_{[g_i]}^{(1)}(h))^p dh \right)^{1/p}.$$

Clearly if $i = j$ then $a_{ij} = a_{ii} = 1$ by Eq. (8.27). For $i \neq j$, Eq. (8.28) provides an upper bound

$$\tilde{a}_{ij} < \left(\frac{1}{\mu([g_j])} \int_{[g_j]} \left(1 - \frac{t}{\beta} \left(\frac{\delta}{t} \right)^\beta \right)^p dh \right)^{1/p} = 1 - \frac{t}{\beta} \left(\frac{\delta}{t} \right)^\beta < 1, \quad (8.29)$$

and a lower bound

$$\tilde{a}_{ij} \geq \left(\frac{1}{\mu([g_j])} \int_{[g_j]} \left(1 - \frac{t}{\beta} \left(\frac{r}{t} \right)^\beta \right)^p dh \right)^{1/p} = 1 - \frac{t}{\beta} \left(\frac{r}{t} \right)^\beta > 0, \quad (8.30)$$

where $1 - \frac{t}{\beta} \left(\frac{r}{t} \right)^\beta > 0$ follows from the choice of t as was explained and proven in Proposition 8.6.1. Thereby one has

$$0 < 1 - \frac{t}{\beta} \left(\frac{r}{t} \right)^\beta \leq a_{ij} = \max\{\tilde{a}_{ij}, \tilde{a}_{ji}\} < 1 - \frac{t}{\beta} \left(\frac{\delta}{t} \right)^\beta < 1,$$

and the result follows. \square

8.7 EXPERIMENTS

In the previous section, we introduced several concepts and algorithms, such as a formal algorithm to identify connected components ([find_dcc](#)) and an algorithm to calculate affinity matrices ([find_affinity](#)). Here, we will show the results of these algorithms applied to several images of the STAR dataset [[1](#), [205](#)]. All experiments are performed on the Lie group $G = \text{SE}(2)$, and the Mathematica notebooks are available via [[27](#)].

We start with discussing the experiment setup in Section 8.7.1 and our approach to identifying the δ -connected components and the corresponding results in Section 8.7.2. Then, we discuss the results of the affinity matrices on some of the δ -connected component experiments in Section 8.7.3.

8.7.1 EXPERIMENTAL SET-UP

In the experiments, we will identify the δ -connected components (flowchart in Fig. 8.7) in retinal images from the STAR dataset [[1](#), [205](#)] on the Lie group $G = \text{SE}(2)$. The retinal images in this dataset are standard 2D-images. Therefore, we first explain how we prepare the images for processing in the Lie group $\text{SE}(2)$.

We consider an input image $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. We lift it to the space of positions and orientations $\text{SE}(2)$ (by creating an orientation score) to disentangle crossing

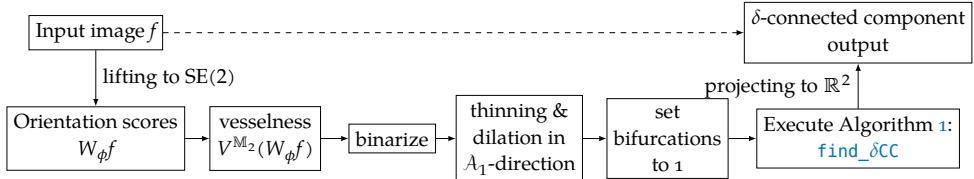


Figure 8.7: Flowchart of the executed steps in the connected component experiments shown in Figs. 8.9 to 8.13.

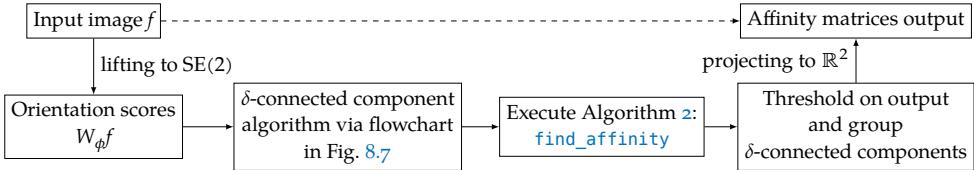


Figure 8.8: Flowchart of the executed steps in the affinity matrices experiments shown in Figs. 8.11 to 8.13.

structures. The orientation score $W_{\phi}f$ is calculated by a convolution with a rotating anisotropic wavelet ϕ and is given by Eq. (7.1). For ϕ , we use real-valued cake wavelets (see [75, 78]). We use 32 orientations for all experiments.

From the orientation score $W_{\phi}f$, we calculate a crossing-preserving vesselness $V^{\mathbb{M}_2}(W_{\phi}f)$, which is done as described in [30, Appendix D]. In all experiments, we use parameter settings $\xi = 1$, $\zeta = 1$, $s = \{1.5, 2, 2.5, 3\}$, $\sigma_{s,Ext} = 0$ and $\sigma_{a,Ext} = 0$.

From this crossing-preserving vesselness, we compute a cost function $C = 1/(1 + \lambda V^{\mathbb{M}_2}(W_{\phi}f)^p)$ with parameters $\lambda > 0$ and $p > 0$. Then, we binarize the cost function using Otsu's method to identify the binarization threshold value. Because the width of the structures varies within the same image, we choose to identify the centerline of the binarized structure, using the approach proposed in [125]. Since this is done in an isotropic way, we compensate for that by slightly dilating the structures in the A_1 -direction; the principal direction of the local structure (see Eq. (8.6) for the explicit formula of A_1). The support of the resulting binarized function on $\text{SE}(2)$ together with the lifted bifurcation information (for all orientations at bifurcation position equal to 1) is then used for the reference set I in the δ -connected component algorithm, more specifically, we identified $\mathbb{1}_I$.

Then, we use the connected component algorithm as introduced in `find_delta_cc` to identify all connected components in the lifted image. To visualize the

output of the δ -connected component algorithm, we define the function $f^{CC} : \text{SE}(2) \rightarrow \mathbb{N}$ by

$$f^{CC}(x, y, \theta) = \begin{cases} i & \text{if } (x, y, \theta) \in [g_i] \\ 0 & \text{else.} \end{cases}$$

The δ -connected components are visualized by projecting them back onto \mathbb{R}^2 (taking per location the maximum over all orientations), i.e.,

$$f_{\text{out}}^{CC}(x, y) = \max_{\theta \in [-\pi, \pi]} f^{CC}(x, y, \theta).$$

Moreover, we also calculate the affinity matrices (flowchart in Fig. 8.8). The algorithm used to compute them, uses the δ -connected components as input. Therefore, one first follows the steps for computing the δ -connected components described in the previous paragraph. The projection step to visualize the results can be skipped as the identification of the affinity matrices also happens in $\text{SE}(2)$. Instead, we perform the affinity matrices algorithm as given in [find_affinity](#) to identify all affinity matrices. Here, we use the orientation score data $W_\phi f$ as the data term $D = |W_\phi f| / \sup |W_\phi f|$. To visualize the results of the output of the affinity matrices algorithm, we determine a threshold value of T and group the δ -connected components having an affinity higher than this threshold value T , i.e.,

$$\tilde{C}_i = \bigcup_{j \in \mathbb{I}_i} [g_j],$$

with \mathbb{I}_i the set containing the vertices that belong to the same connected component of the adjacency graph $A > T$, i.e.,

$$\mathbb{I}_i = \{j \in \{1, \dots, K\} \mid \exists k_1, \dots, k_n; k_1 = i, k_n = j \text{ s.t. } a_{k_l k_{l+1}} > T\}.$$

Consequently, multiple δ -connected components will be grouped in the new visualization

$$f^{AM}(x, y, \theta) = \begin{cases} i & \text{if } (x, y, \theta) \in \tilde{C}_i \\ 0 & \text{else.} \end{cases}$$

To visualize the results, we again project the affinity matrices results back onto \mathbb{R}^2 by taking per location the maximum over all orientations

$$f_{\text{out}}^{AM}(x, y) = \max_{\theta \in [-\pi, \pi]} f^{AM}(x, y, \theta)$$

The additional parameter settings used to process all images can be found in Table 8.1. The *Mathematica* notebooks are publicly available via [\[27\]](#).

Figure	Image	Cost		Dilation		δ -connected Components			Affinity Matrices			
		λ	p	(w_1, w_2, w_3)	α	$(\tilde{w}_1, \tilde{w}_2, \tilde{w}_3), w_i = k\tilde{w}_i$	α	δ	(w_1, w_2, w_3)	α	T	p
Fig. 8.9	STAR48	100	3	(0.2, 1.5, 50)	1.3	(0.06, 0.7, 2)	1	1	-	-	-	-
Fig. 8.10	STAR13	50	3.5	(0.1, 1.5, 50)	1.3	(0.1, 0.7, 4)	1	1	-	-	-	-
Fig. 8.11	STAR34	200	3	(0.2, 1.5, 50)	1.3	(0.2, 1, 2)	1	1	(0.1, 4, 4)	2	0.99957	2
Fig. 8.12	STAR37	100	3	(0.2, 1.5, 50)	1.3	(0.08, 0.7, 4)	1	1	(0.5, 2, 0.5)	2	0.9985	2
Fig. 8.13	STAR38	100	3	(0.2, 1.5, 50)	1.3	(0.2, 1, 2)	1	1	(0.5, 2, 0.1)	2	0.993	2

Table 8.1: Parameter settings for preprocessing of data and calculation of δ -connected components and affinity matrices. The parameter k in the δ -connected components is given by $k = (13/3)^{6/13}$.

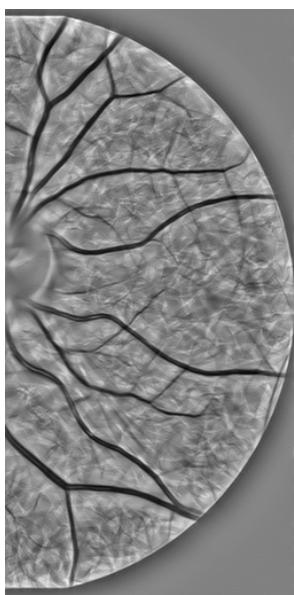
Remark 8.7.1. The output of the δ -connected component algorithm primarily depends on the metric tensor weights, which directly affect the left-invariant Riemannian distance used to compute point-to-point distances. By selecting weight parameters based on image resolution, one can ensure that the resulting δ -connected components remain consistent across different discretizations – up to rounding errors.

8.7.2 δ -CONNECTED COMPONENTS

We identify the δ -connected components using the method described in Section 8.7.1 and visualized in the flowchart in Fig. 8.7. We identify the components for five different images from the STAR dataset [1, 205], and show the results in Figs. 8.9 to 8.13.

We start with the retinal image STAR48 in Fig. 8.9. Due to the binarization, we see that not all vascular structures are nicely connected. However, the δ -connected component algorithm can compensate for interrupted vascular structures by choosing the right threshold value for δ , and the correct distance parameters w_1 , w_2 and w_3 . We see that the algorithm has correctly grouped most of the segments that belong to the same vessel in the underlying image.

The δ -connected components of the retinal image STAR13 are visualized in Fig. 8.10. The vascular structure in this image does not contain a lot of bifurcations or crossing structures. The algorithm is good at identifying components that correspond to the structures in the underlying image. We also see that the choice of the threshold in the binarization (found with Otsu’s method) has a big influence on the input in the δ -connected component algorithm; many vascular structures are interrupted, and therefore the algorithm needs to compensate for that as well. The results are good as long as the gaps are smaller than a certain threshold value. If the gaps are bigger, the δ -connected component algorithm is not able to identify them as belonging to the same vascular structure.

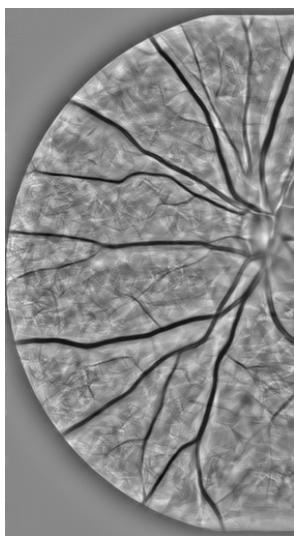


(a) Underlying image (STAR48).

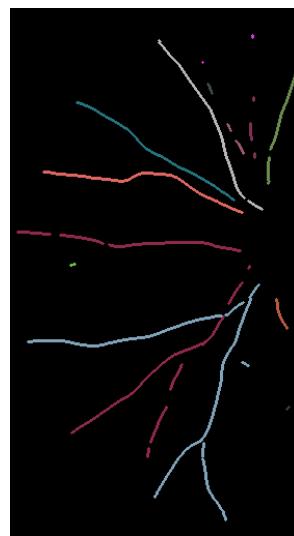


(b) Connected components.

Figure 8.9: The output of the δ -connected component algorithm executed on a thinning of the calculated vesselness of the given image. The parameter settings can be found in Table 8.1.



(a) Underlying image (STAR13).



(b) Connected components.

Figure 8.10: The output of the δ -connected component algorithm executed on a thinning of the calculated vesselness of the given image. The parameter settings can be found in Table 8.1.

In Fig. 8.11, we applied the algorithm to STAR34. The results are shown in Fig. 8.11b. The vessels in the image are more sinuous. We see that this causes some challenges if the vascular structure gets interrupted in the binarization. This is because we need to choose the metric parameters in the δ -connected component algorithm. We chose these parameters, cf. column 7&8 in Table 8.1, such that forward movement is allowed, but sideways movement and changing orientation are not, to avoid crossing structures being connected. However, one does need to change orientation and forward movement at interrupted tortuous structures. Consequently, the δ -connected component algorithm has trouble connecting the vessel segments that are interrupted at highly tortuous parts.

In Fig. 8.12, we performed the algorithm on the retinal image STAR37. The algorithm groups the vessel segments correctly using the chosen parameters. However, it does not identify full vascular trees, but only parts of it. This is due to the relatively large spatial gaps between vessel parts, often also changing orientation.

Lastly, we look at STAR38 in Fig. 8.13b. The δ -connected component algorithm can identify the large vascular structures correctly. Some small vessels are not correctly connected to the main vessel at the bifurcations, but all vessel segments are correctly connected.

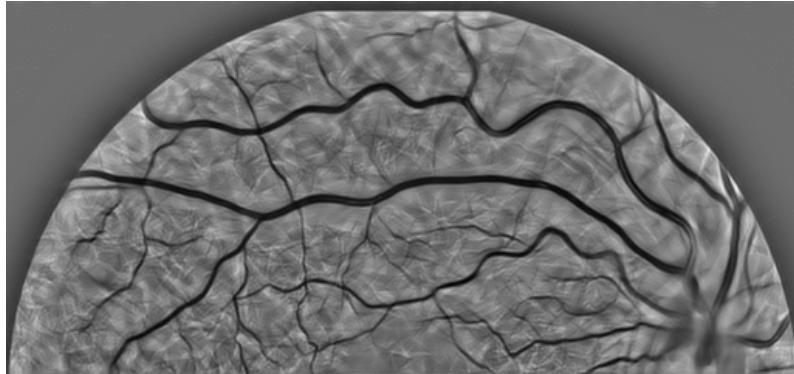
We conclude that this δ -connected component algorithm allows us to identify parts of vascular trees. Additionally, it can differentiate between different structures when they are crossing. However, when the vascular structure is interrupted at a very tortuous part, the algorithm can have difficulties connecting the right parts, depending on the chosen metric parameters.

8.7.2.1 Runtime and Accuracy

We compare two different models: a) our δ -connected component algorithm in $SE(2)$, and b) the classical connected components on \mathbb{R}^2 where voxels are connected if they share at least a corner. We use the STAR dataset to compare both methods.

We report the calculation times of both a) and b). For b), we show the calculation times both for the standard Mathematica implementation and our own (non-optimized) method to identify δ -connected components applied to \mathbb{R}^2 , using only 1 orientation layer.

Lastly, we report the accuracy of the identified components compared to the ground truth for all images in the STAR dataset. We do this using two different measures E_{split} and E_{merge} , where E_{split} measures how many components one



(a) Underlying image (STAR34).



(b) Connected components.

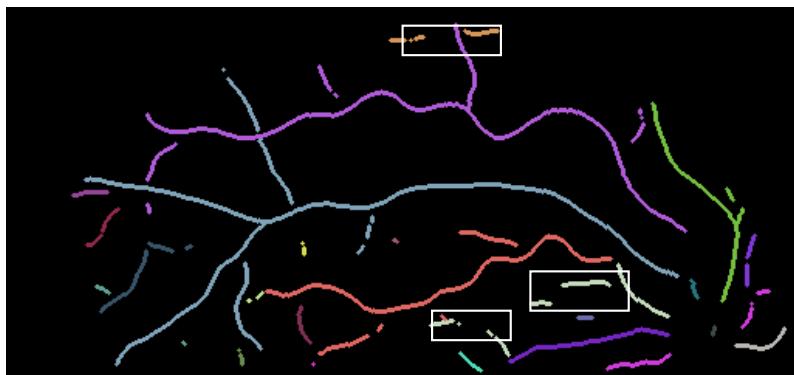
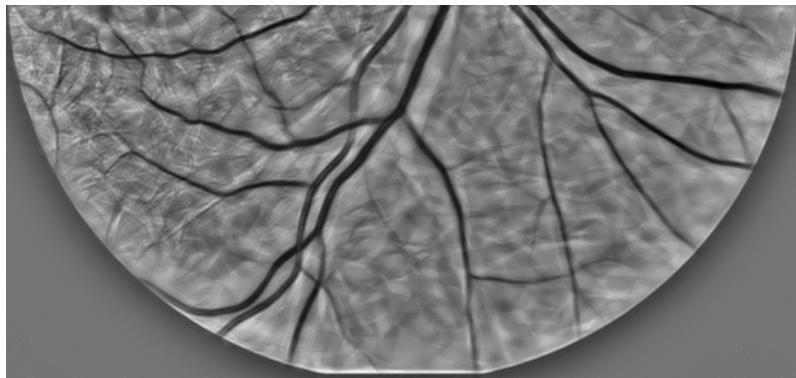
(c) Affinity Matrices calculated on the connected components in Fig. 8.11b. Newly grouped δ -connected components are indicated by a white box.

Figure 8.11: The output of the δ -connected component algorithm and affinity matrices algorithm executed on a thinning of the calculated vesselness of the given image. The parameter settings can be found in Table 8.1.



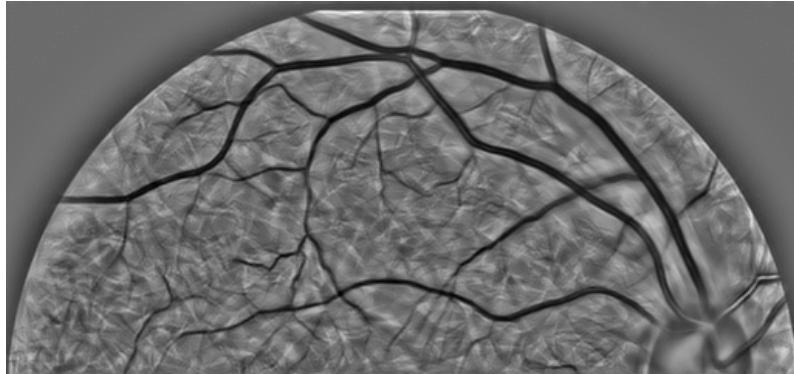
(a) Underlying image (STAR37).



(b) Connected components.

(c) Affinity Matrices calculated on the connected components in Fig. 8.12b. Newly grouped δ -connected components are indicated by a white box.

Figure 8.12: The output of the δ -connected component algorithm and affinity matrices algorithm executed on a thinning of the calculated vesselness of the given image. The parameter settings can be found in Table 8.1.



(a) Underlying image (STAR38).



(b) Connected components.

(c) Affinity Matrices calculated on the connected components in Fig. 8.13b. Newly grouped δ -connected components are indicated by a white box.

Figure 8.13: The output of the δ -connected component algorithm and affinity matrices algorithm executed on a thinning of the calculated vesselness of the given image. The parameter settings can be found in Table 8.1.

vascular tree of the ground truth is divided on average, and E_{merge} measures how many vascular trees are covered by a single component on average, i.e.,

$$E_{split} = \frac{\sum_{i=1}^N \left(\sum_{k=1}^K |[g_k] \cap T_i| \right)}{N}; \quad (8.31)$$

$$E_{merge} = \frac{\sum_{k=1}^K \left(\sum_{i=1}^N |[g_k] \cap T_i| \right)}{K}, \quad (8.32)$$

where T_i represents one of the N vascular trees, and $[g_k]$ one of the identified K δ -connected components.

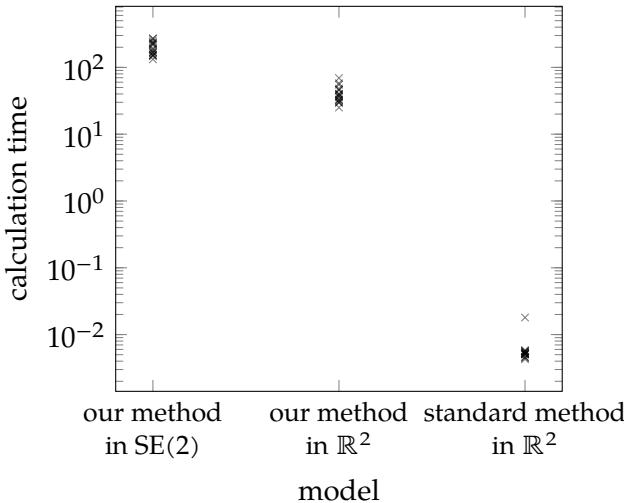
The calculation times of the δ -connected component algorithm are the longest (cf. Fig. 8.14a). This is as we expect, as we have not optimized the algorithm. The calculation times for (non-)optimized \mathbb{R}^2 components suggest that one can significantly improve calculation times by relying on more sophisticated methods originally designed for the classical connected component algorithm.

In the accuracy plots in Figs. 8.14b and 8.14c, each point represents a different image. The points indicated by an ‘o’ show improved results using the δ -connected component algorithm, whereas those indicated by an ‘x’ performed better using the classical connected component algorithm. We see that in a significant majority of the cases, the δ -connected component algorithm outperforms the classical connected components.

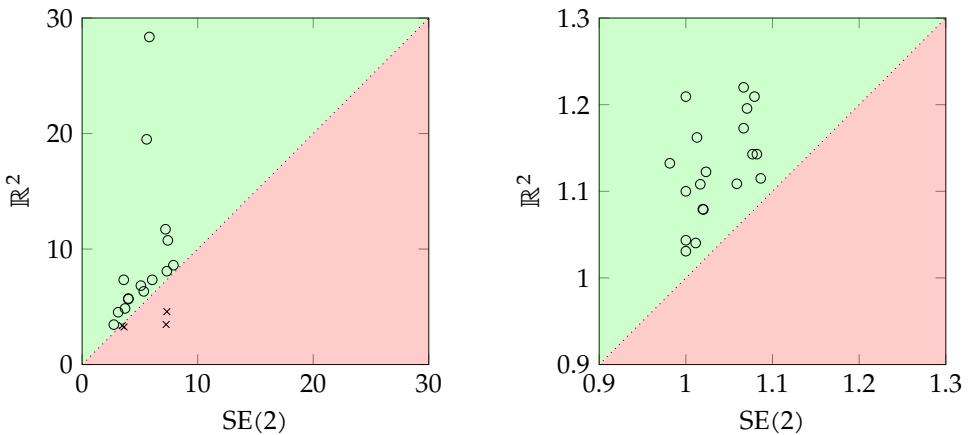
8.7.3 AFFINITY MATRICES

In the previous section, we discussed the experimental results of the δ -connected component algorithm. We demonstrated that the algorithm can identify parts of vessels belonging to the same vascular structure, but often does not identify full vascular trees. We will use the affinity matrices to group different δ -connected components that are most likely to belong to the same vascular structure based on their local alignment and proximity. We have performed the affinity matrices algorithm on the δ -connected component results of Figs. 8.11 to 8.13. We use the orientation score data $|W_{\phi f}|$ as the data term in the affinity matrices algorithm given in [find_affinity](#).

First, we calculated the affinity matrices on the δ -connected component output in Fig. 8.11c. Thresholding on the affinities allows us to group different components, which results in slightly more complete vascular trees. We cannot pick parameter settings such that all parts that belong to the same vessel are connected without connecting them to other structures. However, the output in Fig. 8.11c is a more complete vascular tree classification than the δ -connected



(a) Calculation times for different methods to identify connected components.



(b) # found components covering one real one: E_{split} . (c) # real components covered by one found: E_{merge} .

Figure 8.14: Visualizations of the runtime and accuracy measures E_{split} and E_{merge} . In the green upper-triangle the δ -connected component outperforms the classical algorithm (indicated by an 'o'), whereas in the red lower-triangle the classical algorithm is more accurate than the presented δ -connected component algorithm (points indicated by an 'x').

component output in Fig. 8.11b. We have marked the correctly grouped vascular structures with a white box.

In Fig. 8.12c, we have also applied the affinity matrices algorithm to the output of the δ -connected component algorithm in Fig. 8.12b. The thresholded output of the affinity matrices has correctly grouped different δ -connected components that belong to the same vascular structure. The newly grouped structures have been indicated with a white box. We note that the output is closer to the actual underlying vascular structure. Still, some cases have not been grouped correctly (due to the imperfect data term D).

Last, we have calculated the affinity matrices for the retinal image of STAR38 (cf. Fig. 8.13). The δ -connected component output in Fig. 8.13b has been used as input for [find_affinity](#). The output of this algorithm has been thresholded and the δ -connected components having an affinity higher than the threshold are grouped. The output is visualized in Fig. 8.13c. Here, full vascular trees are identified completely and correctly. The white boxes indicate the changes compared to the δ -connected component output.

We found that thresholding the affinity matrices improves the results; δ -connected components are grouped such that more complete vascular trees are identified. It is important to note that the choice of the metric parameters of Table 8.1 and threshold value T are important to the output.

8.8 CONCLUSION AND FUTURE WORK

In this chapter, we have introduced a way to identify so-called δ -connected components on a Lie group G . These components consist of sets (of points) with a maximum distance δ from each other. First, we introduced the general idea behind the algorithm and some theoretical background on morphological dilations. Then, we connected this theory to the general algorithm, resulting in the δ -connected component algorithm, stated in [find_δCC](#).

We studied the convergence of the δ -connected component algorithm in Theorem 8.4.1. We proved that the algorithm always finishes in a finite number of iteration steps. Subsequently, we discussed the choice of the parameter δ . We suggested using persistence diagrams to choose the optimal value for δ and illustrated it with a few examples.

Once the δ -connected components are introduced and calculated, we aim to determine a hierarchy between the different components. Therefore, we propose to use specific affinity matrices. They describe a way to group components based on their proximity and local alignment. To account for subtleties in the data, we include a data term in the initialization of the affinity matrices. The full algorithm can be found in [find_affinity](#).

To show the performance of the δ -connected component algorithm and the affinity matrices algorithm, we have tested both algorithms on several 2D images of the retina. All experiments show that the δ -connected component algorithm can distinguish different structures at crossings. Additionally, the δ -connected components group well-aligned structures, resulting in more complete vessels in the output. However, it cannot always identify full vascular trees. Therefore, we calculated the affinity matrices for several results of the δ -connected component algorithm. We see that this leads to more complete vascular trees, where different δ -connected components are grouped that belong to the same vascular tree.

The algorithms are challenged by gaps at highly bending parts in the vascular structure. The algorithm cannot connect these structures without connecting different vascular trees at crossings. Large spatial gaps also form challenges: choosing δ too high results in connecting vessel segments belonging to different vascular trees.

For future work, it would be interesting to train the metric tensor weights (w_1, w_2, w_3) via a PDE-G-CNN on $G = \text{SE}(2)$, cf. [184], to improve results further, and to see if optimal transport on $\text{SE}(2)$ [34] would be a viable alternative to determine the affinity between different δ -connected components.

Part IV

GEOMETRIC LEARNING FOR ARTERY-VEIN CLASSIFICATION

9

ARTERY-VEIN CLASSIFICATION

9.1 INTRODUCTION

The blood vessels in the human body can be divided in two categories: arteries and veins. Arteries generally carry oxygen-rich blood from the heart to all organs, while veins return the oxygen-poor blood back to the heart [194]. The only exception is the pulmonary vein that brings oxygen-rich blood to the heart and the pulmonary artery that brings back oxygen-poor blood to the lungs [194].

Research has shown that the behavior of retinal arteries and veins serves as a risk-predictor for various diseases, such as Schizophrenia, bipolar disorder, dementia, and diabetes [11, 39, 96, 109, 113, 124, 142, 162, 195]. For patients, it is an easy, non-invasive procedure to capture the retinal vessels in a picture. Then doctors can use this picture to evaluate the individual's health. As the behavior of arteries and veins is distinctive, it is important to assign all vessels to one of these two categories.

Typically, experts manually segment (parts of) the retinal image. They differentiate between arteries and veins based on the following criteria [160]:

1. Arteries do not cross arteries, and veins do not cross veins.
2. In a bifurcation, the angle between the parent vessel and its respective child is not more than 90 degrees. Note that this is false in 4% of the cases, but can still be used as a reliable indication [70].
3. Arteries are thinner than their neighboring veins [123].
4. The width of a blood vessel decreases (or stays approximately the same) after a bifurcation [70].
5. Parallel vessel segments are usually of opposite classes [123].
6. Arteries are brighter than veins [123].
7. The central light reflex is brighter and wider in the arteries than in the veins [123].

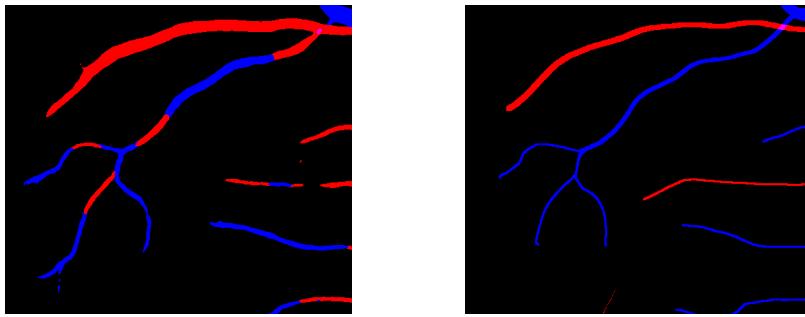


Figure 9.1: Typically, artery-vein classification by neural networks is inconsistent along the vasculature. In this image, red and blue denote artery and vein, respectively.

8. Occasionally, a blood vessel splits into three parts. These points are often treated as two consecutive bifurcations [71].

To automate this tedious process of classifying arteries and veins, various methods have been proposed. In recent years, deep learning techniques have gained popularity and significantly improved the accuracy of classifications [4, 147]. However, a variety of challenges remain in developing accurate artery-vein classification models. In this last chapter of the thesis, we will focus on one aspect: ensuring consistency in classification along a single vessel. Neural networks typically lack awareness of the underlying anatomical structures, allowing them to inconsistently label one blood vessel, as can be seen in Fig. 9.1.

In this chapter, we explore the possibilities of making artery-vein classification consistent and present preliminary work on combining the inconsistent output of the neural network with the δ -connected components on $SE(2)$ introduced in Chapter 8. The technique of δ -connected components identifies patches that are spatially close and well aligned. Additionally, the method ensures that crossing structures (such as crossing vessels) are not merged, but treated as individual segments. We assign each δ -connected component to a class based on the majority vote of the neural network output.

In this chapter we

1. Designed and tested a model to make the artery-vein classification consistent on a network trained with a limited amount of data, which led to improved results. Unfortunately, these results did not transfer to neural networks trained with sufficient data.

2. Investigated the precision of the segmentation by the vesselness filter [30, App. C] and compared it to the precision of the segmentation of a U-net.
3. Investigated the precision of various ways to lift a binary image before applying the δ -connected component algorithm (to separate crossing line structures). We determined the best (simple) way to preprocess the image to get the most accurate results.
4. Made variations of our model to improve the precision of the artery-vein classification. Their success varied: removing more than only the optic disk was not useful, but lifting the artery and vein output of the network separately before applying our method showed promising results.
5. Investigated the transferability of the results of our neural network to other datasets of retinal images. We compared the dice scores of three differently trained U-nets (training dataset was augmented using different techniques) on two test datasets: one belonging to the training dataset and one independent dataset. The more data augmentation steps were included in the training phase of the neural networks, the better the transferability to other datasets. However, the data augmentation steps also slightly worsened the performance on the test dataset belonging to the training dataset.

9.2 RELATED WORK

Many researchers have worked on developing strategies for automatic artery-vein classification in recent decades. These methods rely on feature analysis, neural networks, or a combination of both. Typically, the oldest methods rely on feature analysis, whereas neural networks have gained popularity over the past decade. Manual input has steadily decreased as methods became more and more automated.

The first methods used for artery-vein classification heavily relied on manual input. These methods often automatically identified vascular trees, which were then manually categorized [9, 91, 133].

One of the first automatic methods for artery-vein classification was proposed in [103]. Here, they classify the vessels in a concentric zone around the optic disc. They consider the region $\{x \in \mathbb{R}^2 \mid r_{OD} \leq \|x_{OD} - x\|_2 \leq 4 \cdot r_{OD}\}$, where $x_{OD} \in \mathbb{R}^2$ is the center of the optic disc and $r_{OD} > 0$ the radius. In this region, it is easier to differentiate between the arteries and the veins. The differentiation between the two classes is based on (average) intensity levels. After labeling the vessels in the concentric zone, the labels are propagated to the outer regions of the image using tracking techniques.

More graph-based approaches were proposed [64, 166]. These methods propagate labels through the graph of a vascular tree in a rule-based fashion. Where [166] requires manual input for the segmentation and the labeling of some vessel segments, [64] automatically determines the artery-vein classification based on a combination of the graph-based labeling results and a set of intensity features.

Many other methods also rely on color intensity to differentiate between arteries and veins. In [126], they describe the cross-profile with piecewise Gaussians. They leverage that the cross-profile of arteries and veins differs, as the central light reflex is more apparent in arteries. They extract the Gaussian parameters from the cross-profiles and use them to differentiate. Using the Mahalanobis distance in a minimum distance classifier, they construct the final classification. In [164], they use a Gaussian Mixture Model - Expectation Maximization clustering method to classify the centerline pixels. This classification method takes as input the mean of red, green, and hue, and the variance of red in a circular neighborhood, after which the pixel is assigned to one of the following categories: artery, vein, or not labeled. Finally, the label of a vessel segment is determined by a majority vote. In [190], they use cross-profiles of vessels and an k -means algorithm to assign classes, and use (local) tracking methods to make the labeling consistent along vascular trees.

Among the first neural network strategies was the method presented by [123]. Here, they extract features from the image, reducing dimensionality with principal component analysis. Then, they combine a support vector machine and a neural network to determine the classification. Later works often used Linear Discriminant Analysis [63, 64, 115, 141], k -nearest neighbors [64, 190, 202], or support vector machines [8, 114, 192] to differentiate between arteries and veins.

In recent years, machine learning approaches have become increasingly more popular for artery-vein classification. In [147], an overview of a variety of artery-vein classification methods is presented. Each method employs a distinct network design, classifier (softmax/sigmoid), and image pre- and post-processing techniques [101, 102, 112, 118, 131, 198, 203, 204]. The network designs include generative adversarial networks [204], convolutional neural networks [101, 198], or as U-nets [112, 131]. Many models incorporate image preprocessing techniques to enhance the performance of the neural network. Commonly used methods include histogram matching, linear intensity transformations, illumination corrections, and mean/median filtering. On the other hand, post-processing techniques are used to obtain more consistent classifications along vascular trees (cf. Fig. 9.1). Here, it is customary to identify connected components (parts of the vessel tree that do not branch or cross) and give them a consistent label [101, 102, 198].

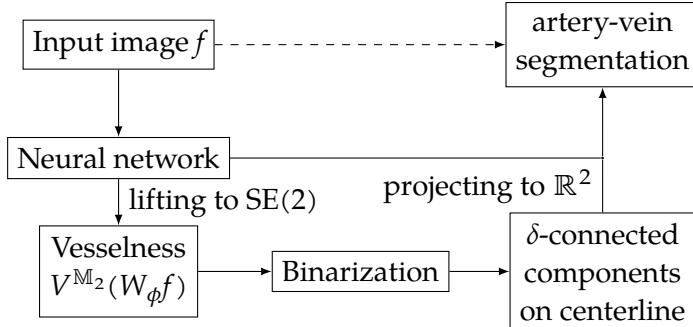


Figure 9.2: Flowchart of the artery-vein classification method

In this chapter, we investigate whether we can use the δ -connected components as a post-processing technique to ensure consistent labeling along the vasculature. The main advantage of this method is that it automatically differentiates between structures at crossings, by relabeling only the vessel segments between bifurcations.

9.3 METHOD

Our method relies on a combination of neural networks and geometric image processing methods. We intend to use the latter to post-process the outcome of the neural network and resolve common issues with the consistency of the classification.

The post-processing step consists of the δ -connected component algorithm, which identify well-aligned, spatially close parts of the image. In Fig. 9.2, we present a flowchart of the method. In this chapter, we will explain the details of the different steps.

9.3.1 DATASET AND DATA AUGMENTATION

We use the Leuven-Haifa High-Resolution Fundus Image Dataset [87]. This dataset consists of 240 disc-centered fundus images from patients with various eye diseases (image dimensions: 1444x1444). For all images, an annotation for arterioles and venules is provided. The dataset is divided into a training and test set consisting of 156 and 56 images, respectively. Additionally, a validation set of 28 images is provided.

To train the neural network, we apply data augmentation to create a more robust model. The model is trained with patches of size 512x512 from the input image. These patches are randomly cropped from the input image of size

1444×1444 . The patches cover between 30 and 40% of the input image, and have an aspect ratio between 9/10 and 10/9. These patches are flipped horizontally with a probability of 0.5, and can undergo a random rotation over an angle $\theta \in [-\pi, \pi]$, where the interpolated pixels are determined by the nearest pixel in the original image. The image is padded with black pixels (0,0,0) and the corresponding mask with (1,0,0,0), where the first to fourth entries contain probabilities of the pixel being background, artery, vein and overlap of artery and vein.

Networks with Limited and Sufficient Training Data

We begin by testing our method on a fast, but with limited data, trained network. Here, we use only 8% of the dataset. This corresponds to just 10 images (of the 156). This network does not perform great, but allows us to check the compatibility of our method.

Once we have seen that our method performs as expected on this simple network, we create a network with sufficient training data, trained on 25% of the dataset (for memory purposes). That means that we train our model using only 39 images.

In the test phase, we do use all 56 test images for both networks.

Transferability to other Datasets

In Section 9.4.7, we check the transferability of the performance of our model to another dataset: STAR [1, 205]. Here, we add one or two extra data augmentation steps to improve the classification results.

The main difference between the two datasets is the color. In Fig. 9.3, we see two typical examples of images in the Leuven-Haifa High-Resolution Fundus Image Dataset and the STAR dataset. We see that the images in the Leuven-Haifa dataset are a lot darker, whereas the image from the STAR dataset is brighter and has a warmer color.

The first additional data augmentation step is a function that randomly changes the brightness, contrast, saturation, and hue of an image. This function takes an image as input and returns an image with adapted brightness (up to 20%), contrast (up to 20%), saturation (up to 20%), and hue (up to 10%).

The second extra data augmentation step adapts the color temperature of the image. Typically, the images from different datasets are more blueish or reddish. We designed a function that adapts the red and blue channel in an image. For more details, we refer to Appendix I. We used the parameter setting $r = 0.5$ while training the models, leading to visually similar images as in the STAR dataset [1, 205], cf. Figs. 9.3b and I.1.

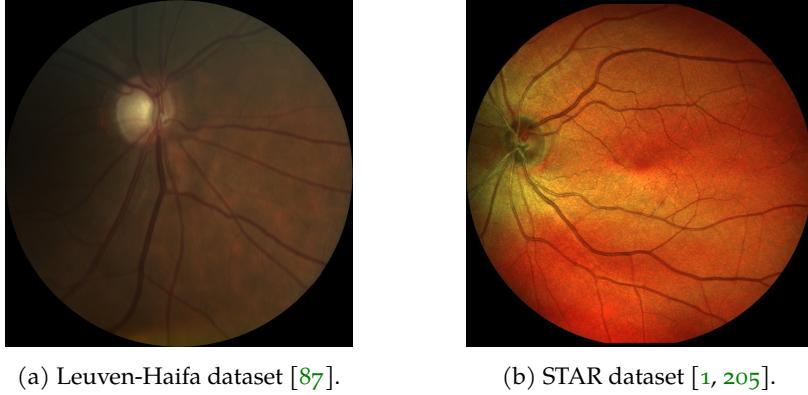


Figure 9.3: Examples of images from the Leuven-Haifa dataset and the STAR dataset.

9.3.2 NETWORK ARCHITECTURE

We use a U-net structure for the artery-vein classification. We use pre-trained decoder channels 256, 128, 64, 32, and 16 of “resnet50”. The model $m^\theta : \mathbb{L}_2(\mathbb{R}^2)^3 \rightarrow \mathbb{L}_2(\mathbb{R}^2)^4$ takes in an RGB-image $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ and outputs a prediction $m^\theta(f) : \mathbb{R}^2 \rightarrow \mathbb{R}^4$, where the background, arteries, veins, and overlap are represented by the first to fourth entry of the output, respectively.

The models are trained for 1000 epochs and use a batch-size of 8. We use a cross-entropy loss function, with weights $w = (w_1, w_2, w_3, w_4) = (0.055, 1.025, 0.96, 19.42)$ and no label smoothing. This loss is given by

$$l(x, y) = L = \begin{pmatrix} l_1 \\ \vdots \\ l_N \end{pmatrix}, \quad l_i = -w_{y_i} \log \frac{\exp(x_{i,y_i})}{\sum_{c=1}^4 \exp(x_{i,c})},$$

where x is the input, y is the target, w is the weight, and N is the minibatch size.

We use AdamW as an optimizer with a learning rate of 10^{-3} and weight decay 0.01.

To accelerate the training speed, we use gradient accumulation of 2 batches; only optimizing the model every other batch iteration.

9.3.3 POST-PROCESSING

Normalization of Network Output

To interpret the output of the trained network as probabilities of background, artery, vein, and overlap, we calculate the softmax of the model output, i.e.,

$$\sigma(\mathbf{z}) = \frac{e^{z_i}}{\sum_{j=1}^4 e^{z_j}},$$

where \mathbf{z} is the model output $m^\theta(f)(\mathbf{x}) =: \mathbf{z}$.

Visualization of Network Output

When visualizing the output of the neural network, we show an image in which each pixel's color is determined by its most likely class. Hence, we visualize $f^\theta : \mathbb{R}^2 \rightarrow \{0, 1, 2, 3\}$, given by

$$f^\theta(\mathbf{x}) = \arg \max_{i \in \{1, 2, 3, 4\}} \{\sigma_i(m^\theta(f)(\mathbf{x}))\} - 1,$$

where

$$f^\theta(\mathbf{x}) = \begin{cases} 0 & \text{if } f(\mathbf{x}) \text{ resembles background;} \\ 1 & \text{if } f(\mathbf{x}) \text{ resembles artery;} \\ 2 & \text{if } f(\mathbf{x}) \text{ resembles vein;} \\ 3 & \text{if } f(\mathbf{x}) \text{ resembles overlap.} \end{cases}$$

We plot the background, arteries, veins, and overlap (of arteries and veins) in black, red, blue, and purple, respectively.

9.3.4 δ -CONNECTED COMPONENTS FOR CONSISTENT CLASSIFICATION

The output of a trained neural network typically results in inconsistent labeling along the vasculature, cf. Fig. 9.1. Therefore, we do post-processing with δ -connected components [28]. This method groups pixels in components based on their spatial proximity and alignment.

We use the input image enhanced with illumination enhancement and crossing-preserving denoising [185, 207] as the basis for the algorithm. First, we determine the orientation score of this image [73], after which we identify the vessels with the crossing-preserving vesselness filter [30, App. C]. This

filter identifies tubular structures in the lifted data (orientation scores) that often correspond to blood vessels in the image.

We remove the optic disk from the lifted data, because here, blood vessels are clustered. Currently, this is done manually, but could be automated using methods such as [167, 188].

We binarize the vesselness, and identify the centerlines using [125]. To ensure face connectivity of the centerline, we dilate the found structures with 1 voxel in each adjacent direction, after which we can calculate the δ -connected components. This method groups pixels according to their local proximity and alignment [28], and extends the concept of regular connected components, which only take into account local proximity.

Once all components in the image have been identified (denoted by f^{CC}), we assign each component to the class that is the most frequent in the neural network output, i.e., for δ -connected component $i \in \{1, \dots, N\}$, the new label λ_i is denoted by

$$\lambda_i = \arg \max_{l \in \{2,3,4\}} \|\mathbb{1}_{f^{CC}=i} \cdot \sigma_l(m^\theta(f))\|_1, -1$$

where labels $l = 2, 3, 4$ represent the probabilities for artery, vein, and overlap, respectively, and $\|\cdot\|_1$ denotes the \mathbb{L}_1 -norm. Then, the final classification $f^{out} : \mathbb{R}^2 \rightarrow \{0, 1, 2, 3\}$ is given by

$$f^{out} = \sum_{i=1}^N \lambda_i \cdot \mathbb{1}_{f^{CC}=i},$$

where $\mathbb{1}_{f^{CC}=i}$ denotes the indicator function with support on the set determined by the i th δ -connected component.

9.3.5 EVALUATION OF THE METHOD

Lastly, we aim to compare the results of different classification methods. To evaluate performance, we use an accuracy metric to compare two classes: the dice score. This is defined as

$$DSC = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} = \frac{2 \cdot |A \cap B|}{|A| + |B|},$$

where TP, FP, FN denote “true positive”, “false positive” and “false negative” respectively. Alternatively, it can be interpreted using the cardinality of the set A , the set of pixels assigned to a specific class, and set B the corresponding set of target pixels. We calculate the dice score per class, i.e.,

$$DSC_i = \frac{2|A_i \cap B_i|}{|A_i| + |B_i|}, \quad (9.1)$$

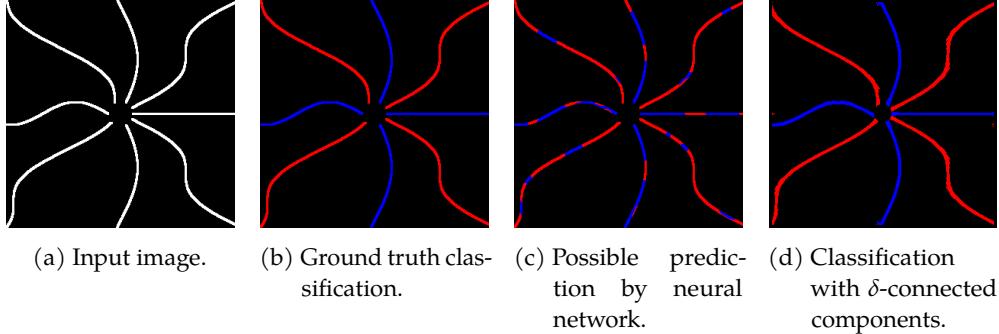


Figure 9.4: Synthetic example containing curved lines. Often, the neural network produces a non-consistent output, as in Fig. 9.4c. In Fig. 9.4d, we classify δ -connected components (calculated on the vesselness, binarized with automatic Otsu threshold), and do majority voting for the label, based on the neural network output in Fig. 9.4c.

where $A_i := \{\mathbf{x} \in \mathbb{R}^2 \mid g(\mathbf{x}) = i\}$ and $B_i = \{\mathbf{x} \in \mathbb{R}^2 \mid f^{GT}(\mathbf{x}) = i\}$, with g the (post-processed) output of the network and f^{GT} the target output. We consider the classes background, artery and vein, where the overlapping regions count toward both artery and vein predictions.

9.4 EXPERIMENTS

In this section, we begin by testing our strategy on some synthetic examples. Then, we show the results of our method applied to networks with both a limited and sufficient amount of training. Furthermore, we discuss the accuracy of the vesselness filter and the accuracy of lifting the binary vesselmap. We try to improve our own method by slightly modifying the model lifting the artery and vein map separately. Finally, we discuss the transferability of the performance of our network to other datasets.

9.4.1 SYNTHETIC EXAMPLES

First, we aimed to validate our approach. Therefore, we designed a synthetic example. We considered abstract (black-white) images that contain lines.

We begin with the synthetic example shown in Fig. 9.4. Here, we needed to assign the curved white lines to a class. Typically, the neural network produced an inconsistent classification, such as in Fig. 9.4c. We aimed to prevent this by combining the neural network output with our δ -connected component algorithm. Then, the labeling became consistent, as in Fig. 9.4d.

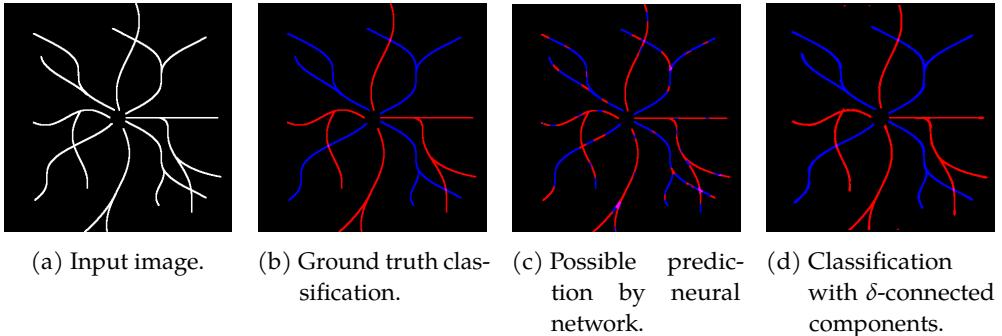


Figure 9.5: Synthetic example containing curved lines with crossings and bifurcations. Often, the neural network produces a non-consistent output, as in Fig. 9.5c. In Fig. 9.5d, we classify δ -connected components (calculated on the vesselness, binarized with automatic Otsu threshold), and do majority voting for the label, based on the neural network output in Fig. 9.5c.

However, in real retinal images, blood vessels are crossing and have bifurcations. Therefore, we also considered a more advanced synthetic example, cf. Fig. 9.5. We created a ground truth (Fig. 9.5b) and a possible prediction by a neural network (Fig. 9.5c). Then, we applied our method to the input image. We saw that our method automatically differentiated between the crossing structures. It identified the vascular trees correctly and then assigned each tree to the correct category (artery/vein).

These results suggest that our proposed method is promising. In the next sections, we applied the method to real data and evaluated its performance.

9.4.2 NETWORK WITH LIMITED TRAINING DATA

We began our experiments on real data with a neural network with a limited amount of training data (U-net as described in Section 9.3 trained with only 10 images). We compared the dice scores of the neural network to the results found by our proposed method with post-processing. Fig. 9.6 shows the dice scores for 10 different (trained) U-nets. In Fig. 9.7, we show an example of the output of all different methods for one trained U-net.

As expected, the network that was trained with a very limited amount of data did not perform very well. The dice scores for the neural network were between 0.4 and 0.6 for both the arteries and the veins.

We saw that the network sometimes classified background as artery or vein, cf. Fig. 9.7d. Therefore, we enhanced the image as in [31] using illumination enhancement [207] and denoising [185] and calculated the vasculature in the enhanced image using the vesselness filter [30, App. C], cf. Fig. 9.7b. We man-

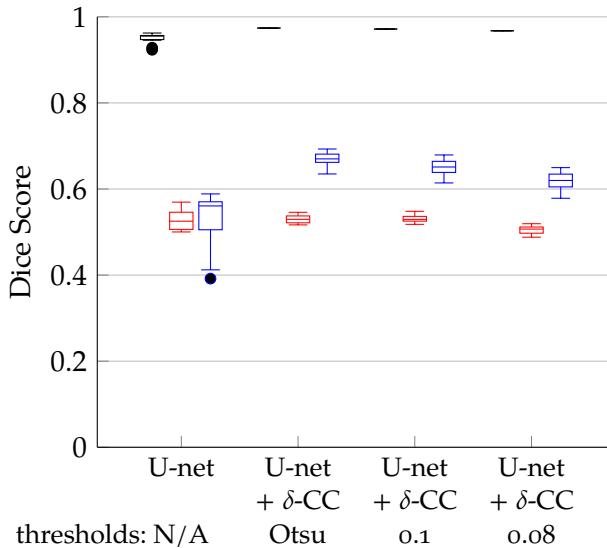


Figure 9.6: Dice scores of a U-net with a limited amount of training data (trained on the training data of the Leuven-Haifa dataset) tested on the Leuven-Haifa dataset with different post-processing methods. The dice scores for the categories background, artery, and vein are visualized in black, red, and blue, respectively.

ually removed the optic disk after binarization. We identified the δ -connected components on this vasculature mask and assigned each component to a class (artery/vein) based on the most frequent label in the neural network output.

Identifying the δ -connected components requires binarizing the vesselness. First, we used Otsu's method to determine the threshold. This method automatically determines the value where the variance of the intensity values between different classes is maximal. The automatically determined threshold failed to detect all vasculatur structures (see Fig. 9.7e). Therefore, we relaxed the parameter, manually setting it to 0.1 and 0.08, after which we picked up on the majority of the vasculature, cf. Fig. 9.7f.

The dice scores of our method were between 0.5 and 0.6 for the arteries and between 0.55 and 0.7 for the veins. We saw that our method improved the dice score for the veins, but not for the arteries. This is most likely due to the veins being picked up better by the vesselness filter, and the veins being slightly wider than the arteries, leading to components being more likely labeled as a vein. If there is no majority in the prediction of the U-net, the segment is assigned the label "artery".

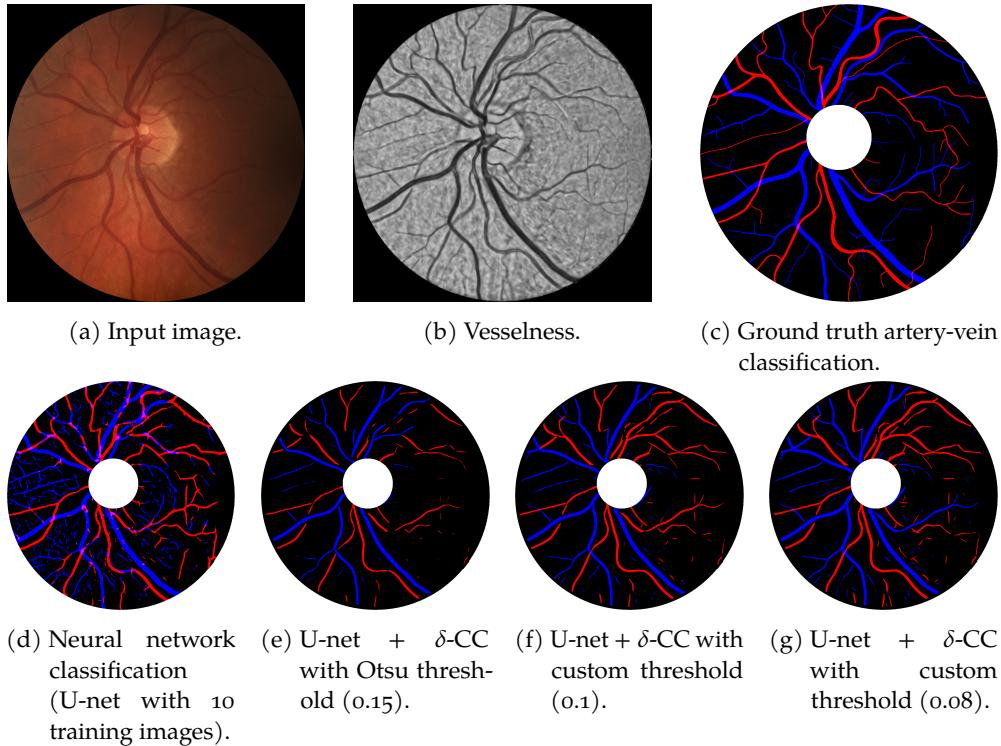


Figure 9.7: Example of the artery-vein classification in one image for different methods building on a neural network with limited training data (10 training images). The artery-vein classifications in Figs. 9.7e to 9.7g are constructed by lifting the (enhanced) input image, binarizing the vesselness using the threshold value, center-line identification, optic disk removal, and then determining the δ -connected components. Labels are assigned using majority voting based on the corresponding neural network output in Fig. 9.7d.

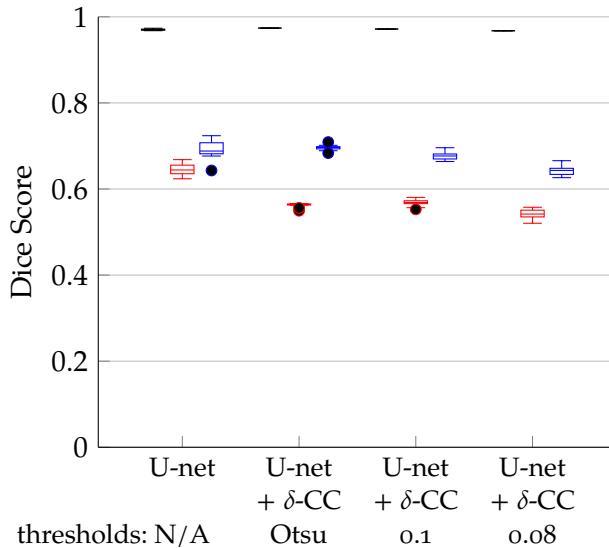


Figure 9.8: Dice Scores of a U-net with sufficient training data (trained on 25% of the training data of the Leuven-Haifa dataset) tested on the Leuven-Haifa dataset with different post-processing methods. The dice scores for the categories background, artery, and vein are visualized in black, red, and blue, respectively.

Although the visual results improved when setting a manual threshold, the dice scores did not improve accordingly. Most likely, this is a consequence of the SE(2)-vesselness not picking up on all structures.

9.4.3 USAGE OF A NEURAL NETWORK WITH SUFFICIENT TRAINING DATA

Next, we extended our model to a neural network with sufficient training data. This time, we trained the same neural network with 38 training images instead of only 10.

We applied the same strategy as in Section 9.4.2 and saw that the accuracy of the U-net improved significantly (cf. Fig. 9.8): the dice scores for arteries and veins were both between 0.6 and 0.75. The improvement was substantial, and our method could not match the dice scores of the U-net. Although our model's classifications improved, they were still outperformed by the neural network for the arteries. The dice scores for the veins were comparable in both the U-net and our method.

When we took a closer look at the output of the classification, cf. Fig. 9.9, it was immediately evident that the neural network produced a more accurate classification. It identified the majority of the vessels correctly, but the labeling is

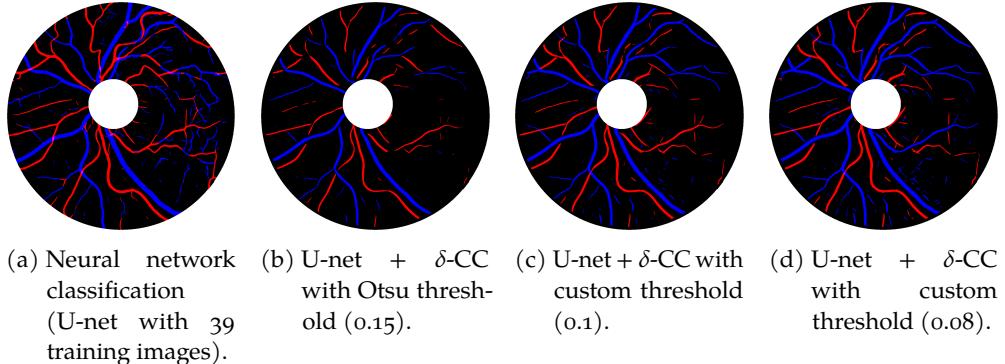


Figure 9.9: Example of the artery-vein classification in one image for different methods building on a neural network with sufficient training data (39 training images). The artery-vein classification in Figs. 9.9b to 9.9d are constructed by lifting the (enhanced) input image in Fig. 9.7b, binarizing the vesselness, identifying the centerlines, removing the optic disk, and then determining the δ -connected components. Labels are assigned using majority voting based on the neural network output in Fig. 9.9a.

not always consistent along the blood vessels. In contrast, our method still relied on the vesselness calculated on the enhanced image. Here, the segmentation (background-vasculature) is unchanged, but the classification (artery-vein) has slightly improved. Nonetheless, some vascular structures remained undetected.

9.4.4 ACCURACY OF THE VESSELNESS FILTER

This led us to take a closer look at the accuracy of the vesselness filter. Each method produces a distinct segmentation map and, consequently, the corresponding artery-vein classification were also unique.

The trained neural network automatically calculated a segmentation mask. On the other hand, we can also calculate the so-called vesselness, which we use in the experiments described in Sections 9.4.2 and 9.4.3. There, we determined the location of the blood vessels using geometric methods. We calculated the orientation scores of the enhanced image. Then, we calculated the vesselness [30, App. C], and we binarized the output. If the value was higher than the threshold value (either hand-picked or calculated with Otsu's method), the pixel is considered to be on the vasculature and if the value is lower, it is not.

We aimed to compare the accuracy of both approaches. In Fig. 9.10, we visualized the dice scores of the vasculature-accuracy for the trained U-nets and vesselness binarized with either Otsu's threshold, or the manually picked threshold 0.1. We see that the dice scores for U-net with limited training data

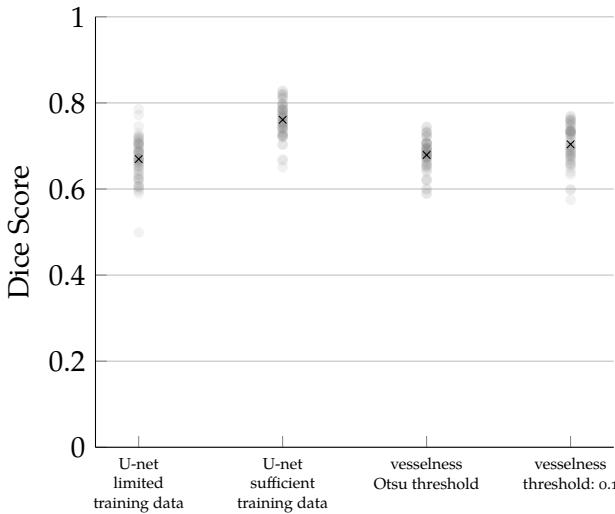


Figure 9.10: Dice scores of the segmentation by different methods (neural networks and vesselness).

were comparable to the dice score of the vesselness binarized with Otsu's threshold (0.67 and 0.68 respectively). The dice score of the vesselness binarized with fixed threshold 0.1 was slightly better: 0.70. However, they were all outperformed by the U-net trained with sufficient data with a dice score of around 0.76.

In Fig. 9.11, we analyzed the influence of a fixed threshold value for the vesselness on the dice score. The dice score rapidly increases at 0, reaches an optimum at 0.1, and then slowly decreases towards 1. Note that this optimal threshold value still had a significantly lower dice score than the U-net trained with more training data.

Remark 9.4.1. Note that, in the experiments in Sections 9.4.2 and 9.4.3, we only assigned labels (artery and/or vein) to the pixels on the vasculature. Therefore, if the initial segmentation was inaccurate, it influenced the classification results. Pixels that did not belong to the vasculature but were in the segmentation were classified as artery or vein. Similarly, pixels that belong to the vasculature but are excluded by the segmentation are not classified. This influenced the obtained dice scores.

Lifting the (Binary) Vesselmap?

In the previous section, we learned that the dice score of the vessel segmentation by the U-net with sufficient training data was significantly higher than the dice scores produced by our vesselness segmentation. Therefore, we were interested

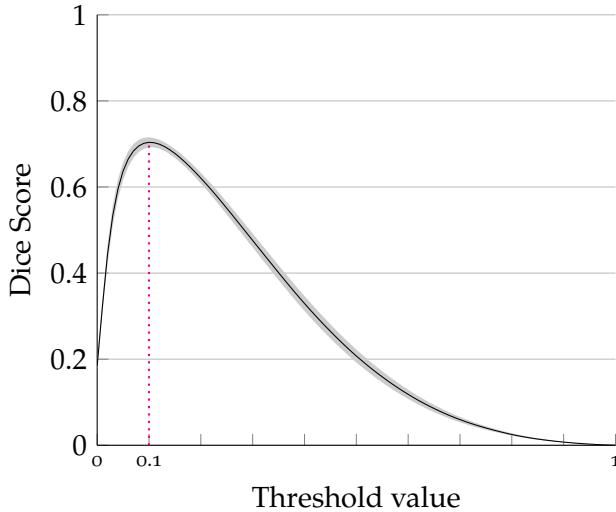


Figure 9.11: Dice scores for different binarization thresholds of the vesselness with a 95% confidence interval. Optimal performance is achieved for threshold value 0.1.

in the accuracy of lifting a binary map, for example the segmentation map of a network, and the corresponding thresholds.

We used two basic strategies to quickly create a lifted, binary image from the binary segmentation $f : \mathbb{R}^2 \rightarrow \mathbb{R}$:

1. We calculated the orientation scores and took the positive part, i.e., we considered

$$(W_\psi f(\mathbf{x}, \theta))_+, \quad (9.2)$$

where W_ψ is the orientation score transform and $(a)_+ = \max\{0, a\}$. We then binarized by thresholding with value t :

$$\mathbb{1}_{(W_\psi f)_+ > t}(\mathbf{x}, \theta) = \begin{cases} 1 & \text{if } (W_\psi f)(\mathbf{x}, \theta) > t \\ 0 & \text{else.} \end{cases}$$

2. We calculated the multi-scale vesselness on the orientation score, as described in [30, App. C], i.e.,

$$V^{\mathbb{M}_2}(W_\psi f(\mathbf{x}, \theta)). \quad (9.3)$$

Then, we binarized by thresholding with value t :

$$\mathbb{1}_{V^{\mathbb{M}_2}(W_\psi f) > t}(\mathbf{x}, \theta) = \begin{cases} 1 & \text{if } V^{\mathbb{M}_2}(W_\psi f)(\mathbf{x}, \theta) > t \\ 0 & \text{else.} \end{cases}$$

We compared the lifted binary data to the binary segmentation f by projecting the lifted data back to \mathbb{R}^2 via a maximum intensity projection, i.e.,

$$\tilde{f}(\mathbf{x}) := \sup_{\theta} \mathbb{1}_{A>t}(\mathbf{x}, \theta),$$

where $A : \text{SE}(2) \rightarrow \mathbb{R}^+$ denotes the functions given in Eqs. (9.2) and (9.3) for the lifting strategy presented in Items 1 and 2 respectively.

For both methods, we determined the dice score for possible threshold values between 0 and 1. For the second method, we used different spatial scales: $\sigma_s \in \{5, 10\}$ and $\sigma_s \in \{1, 5, 10\}$. We visualized all dice scores in Fig. 9.12. We saw that the first method reached the highest dice score at threshold value 0.1. On the other hand, the optimal dice score for the vesselness (calculated with two different scales) was slightly lower, but less sensitive to the choice of the threshold value.

In practice, it can be challenging to pick the optimal value. Therefore, depending on the scenario, it might be useful to stick to the second method using the vesselness. However, in the case of lifting a binary map, the dice score variance is minimal. Therefore, we will continue with the method presented in Item 1.

It is important to note that applying Otsu's thresholding method to binarize the positive part of the orientation score, denoted as $(W_\psi f)_+$, is not ideal. The corresponding dice scores exhibit a sharp rise followed by a rapid decline around the optimal threshold, indicating high sensibility to the selected threshold. In contrast, the dice scores derived from the vesselness maps demonstrate greater stability across a range of thresholds, making the use of Otsu's method comparatively less risky.

In Fig. 9.13, we show some examples of the segmentations obtained by the two presented methods. Visually, we don't see much difference. Additionally, the centerlines (calculated in the lifted space) of both segmentations are shown, as we apply the δ -connected component algorithm to this centerline.

9.4.5 ADAPTING THE MODEL: LIFTING ARTERIES AND VEINS SEPARATELY

Based on our findings in Section 9.4.4, we decided that it was better to adapt our model: we no longer calculated the vesselness, but instead lifted the segmentation of the neural network. However, this did not solve all our problems: Our method still struggled with so-called *kissing structures*, i.e., (almost) overlapping structures with the same orientation. In this section, we lifted the artery- and vein-prediction separately, to avoid as much entanglement as possible. We calculated the centerlines of the positive part of the orientation scores of the arteries and the veins, and applied the δ -connected component algorithm to a volume containing both. Then, we labeled the data in the lifted orientation

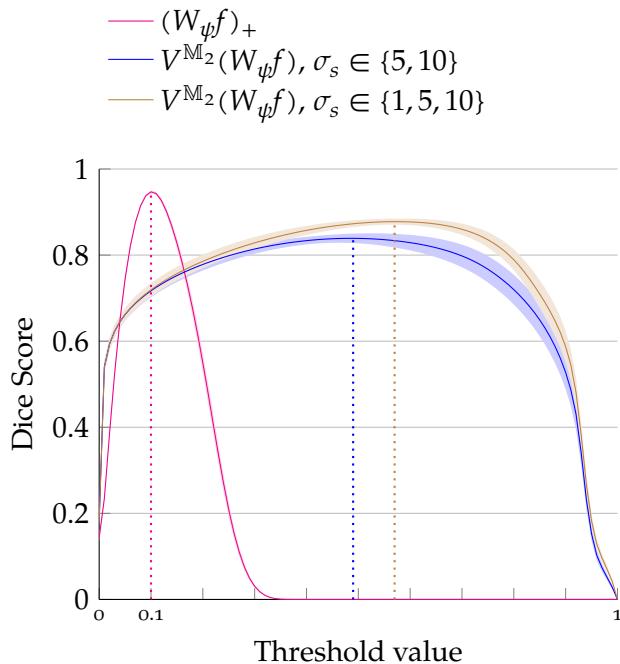


Figure 9.12: Dice scores for different binarization thresholds of the vesselness with a 95% confidence interval on binary images of vasculature. Optimal performance is achieved for the positive part of the orientation score at threshold value 0.1.

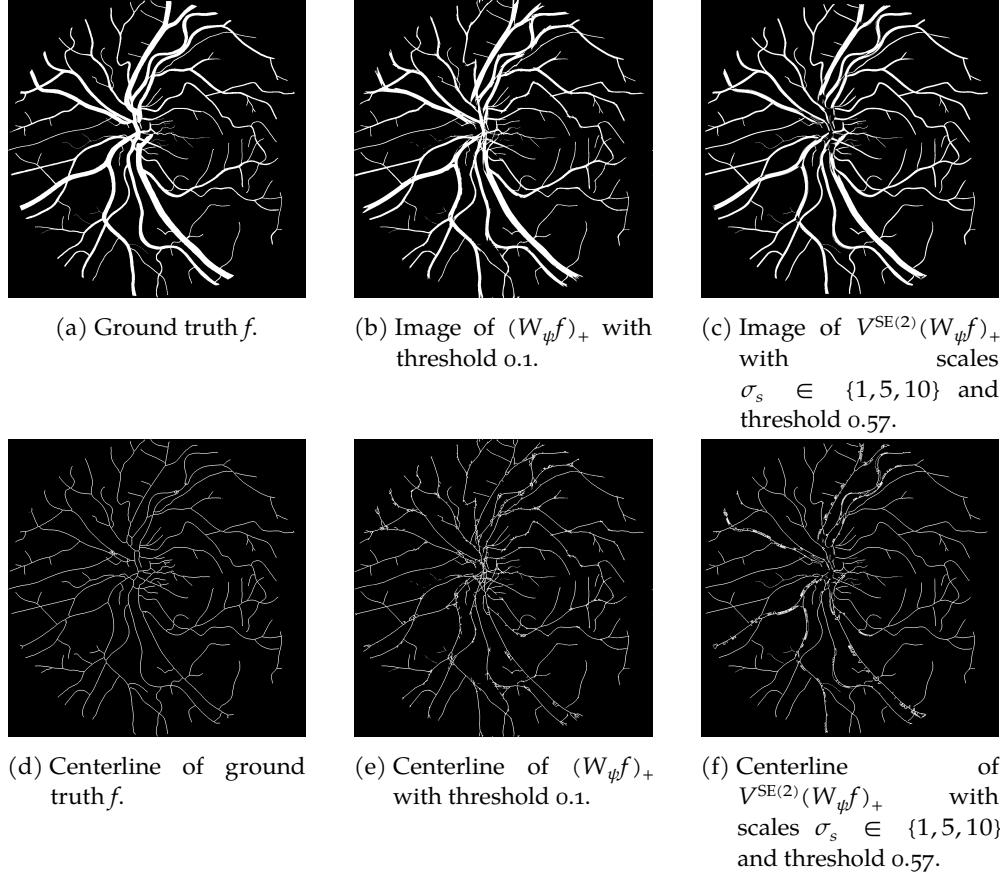


Figure 9.13: Visualization of the results of binarizing a lifted binary image for the optimal binarization threshold values according to Fig. 9.12. On average, the results in Fig. 9.13b have a higher dice score than in Fig. 9.13c. In Figs. 9.13e and 9.13f, we show the (projected) centerline of both lifted images.

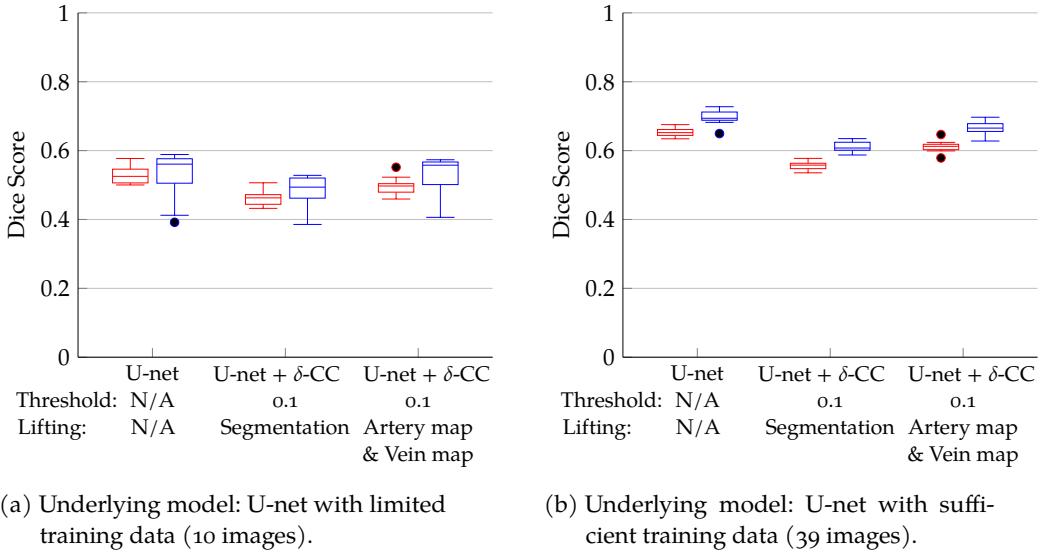


Figure 9.14: The dice scores of neural networks (with limited training data in Fig. 9.14a and sufficient training data in Fig. 9.14b), and our method applied to them, where we either lift the segmentation predicted by the network, or the artery- and vein-map predicted by the network. Red and blue represent arteries and veins, respectively.

scores of the arteries and the veins separately, after which we assigned each component a label using majority voting.

We looked at the dice scores for this new method presented in Fig. 9.14. We saw that they are an improvement on our previous results, where we applied the δ -connected component algorithm to the lifted segmentation (given by the network). Although the results were very promising, we did not improve on the output of the neural network. In Fig. 9.15, we show two situations in which our new method fails where it previously succeeded and succeeds where it first failed. In Fig. 9.15a, we saw that lifting the artery and vein map separately can lead to preservation of fragmentation along the width of a blood vessel. On the other hand, lifting the artery and vein map separately can significantly improve the classification results when arteries and veins are spatially close and well-aligned. In Fig. 9.15b, we were able to prevent the merging of an artery and a vein by lifting the artery and vein map separately.

9.4.6 CHOOSING THE METRIC PARAMETERS OF δ -CONNECTED COMPONENTS

We also investigated the parameter choice in the δ -connected component algorithm, as this choice significantly affects the final artery-vein classification.

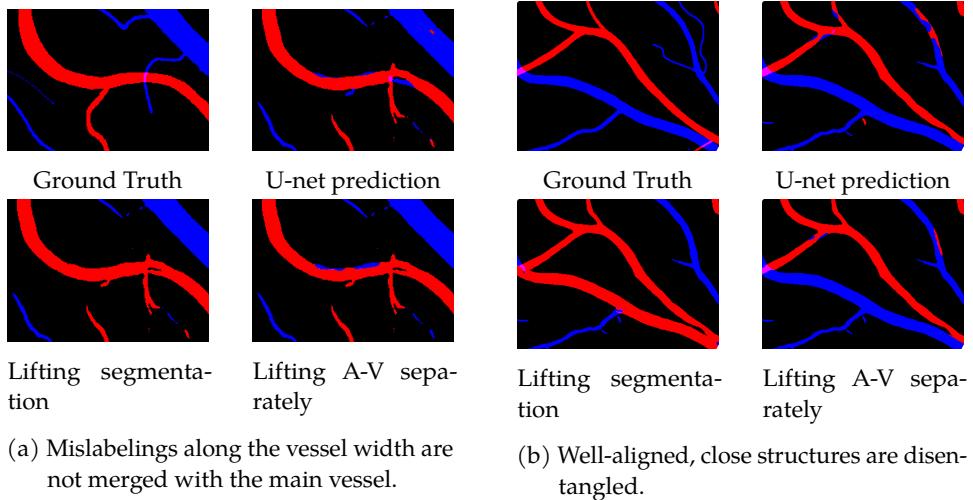


Figure 9.15: Visualization of the problems we do and do not fix by lifting the artery- and vein-prediction separately.

	Background	Artery	Vein
Fig. 9.16b	0.9666	0.6674	0.7266
Fig. 9.16c	0.9666	0.6043	0.7064
Fig. 9.16d	0.9666	0.5980	0.7030

Table 9.1: Dice scores of artery-vein classifications in Fig. 9.16.

In Fig. 9.16, we show the results for one of our experiments using different metric parameters. We lifted the artery- and vein-map separately, calculated their vesselness, and applied the δ -connected component algorithm to the centerline of the combined, binarized vesselness with two different metric tensor weights. In Figs. 9.16c and 9.16d, we used metric tensor weights (0.2,1,4) and (0.25,1,4) respectively. We observed that the choice of metric tensor weights has a substantial influence on the outcome of our method. In the yellow boxes, we highlighted the main changes between both results.

In Fig. 9.16c, we see that the metric parameters fixed the labeling issues at the bifurcation (in the lower box), but it mislabeled a big artery as a vein (in the upper box). On the other hand, in Fig. 9.16d, the artery in the upper box was not mislabeled as a vein, but the bifurcation in lower box is not uniformly labeled. This led us to conclude that the parameters of the δ -connected component algorithm should be position-orientation-dependent: good parameter settings in one place lead to mistakes in another.

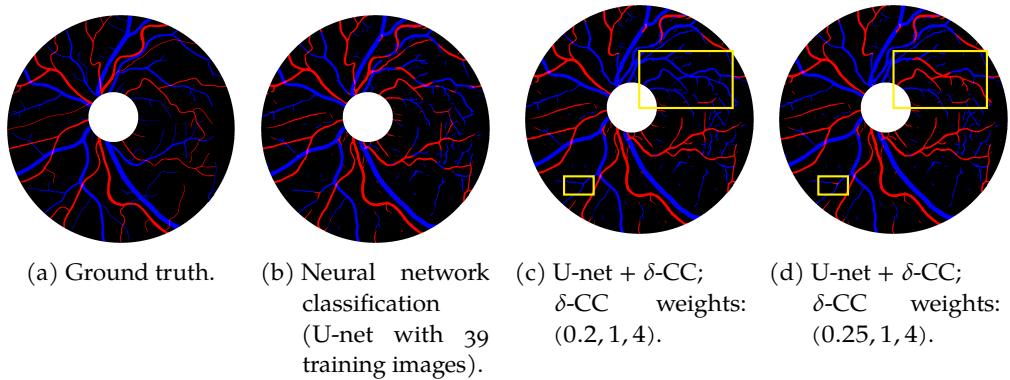
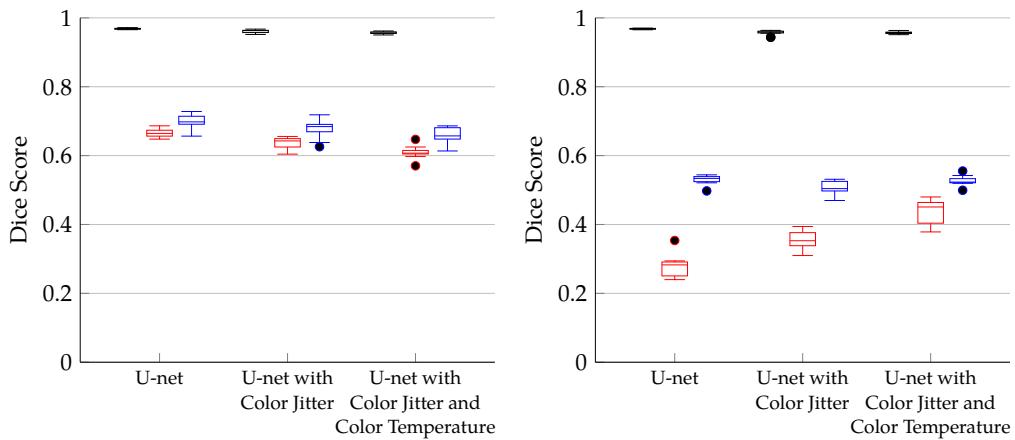


Figure 9.16: Example of the artery-vein classification in one image for different methods building on a neural network with sufficient training data. For Figs. 9.16c and 9.16d, we lift the artery- and vein-map of the neural network separately, calculate the vesselness, binarize using Otsu’s threshold, and extract the centerline. We remove the optic disk. Then, we find the δ -connected components with given parameter settings and do majority voting per component to determine the corresponding label. Only pixels classified as artery or vein are relabeled. The dice scores can be found in 9.1. Global parameters for the δ -connected components are suboptimal: good parameters at one position, lead to problems in another (see yellow boxes). The parameters for the δ -connected components should be position-orientation-dependent and data-driven.



(a) Test dataset: 100% of the test data of the Leuven-Haifa dataset. (b) Test dataset: 100% of the test data of the STAR dataset.

Figure 9.17: Dice scores of our U-nets (trained on 25% of the training data of the Leuven-Haifa dataset) applied to different test datasets. The dice scores for background, artery, and vein are indicated by colors black, red, and blue, respectively. The visualized results are the average performance of 10 models (per type) over the complete test dataset.

9.4.7 TRANSFERABILITY OF THE PERFORMANCE TO OTHER DATASETS

Finally, we considered the transferability of our findings to other datasets. We trained our networks on (subsets of) the Leuven-Haifa dataset. We used different data augmentation techniques on the images from the Leuven-Haifa dataset in an effort to achieve better performance on other datasets. For more details on the data augmentation techniques used, see Section 9.3.1. In this section, we report how these trained networks performed on the STAR dataset.

First, we trained and tested our networks on the Leuven-Haifa dataset. In Fig. 9.17a, we visualized the performance. We noticed that the U-net without additional data augmentation seemed to perform the best, followed closely by the U-nets trained with “color jitter”, and “color jitter” and “color temperature” augmentations. The differences are small.

Then, we tested the same networks on the STAR dataset. The results are shown in Fig. 9.17b. Here we see a reverse trend: the more data augmentation, the better the dice scores.

A possible explanation would be that the network becomes less sensitive to camera-specific artifacts. Additionally, the network can better deal with variations in brightness, contrast and saturation, which can be influenced by both the camera-settings and the environment. This emphasizes that it is important

to use a training dataset which is as diverse as possible. This ranges from the hardware (used cameras) to the inclusion of different pathologies, e.g., different diseases including final stages of diabetic retinopathy and different populations [88].

9.5 CONCLUSION AND FUTURE WORK

We have developed a method to make the output of a neural network more consistent using δ -connected components. This method can lead to improved consistency in classification results, as shown in the synthetic examples in Figs. 9.4 and 9.5. However, when applied to real data with sufficient labeled training data, the method did not result in the expected better dice scores.

We tried to improve the results by lifting the artery and vein prediction maps separately, aiming to disentangle near and well-aligned vascular segments. Here, we clearly saw the strong dependence on the parameter settings for the lifting and the δ -connected components: optimal parameter settings for one region may cause errors in another (cf. Fig. 9.16). Therefore, we strongly advise to use a local, position-orientation-dependent and data-driven way to identify the δ -connected components in the future.

Alternatively, it would be interesting to use crossing-preserving tracking methods to identify vascular trees, and the connectivity of components with different labels. The tracts could be used to target specific regions to switch classes. Especially when a part of a vascular tree has a different label, this method could allow for targeted local interventions. For this, we need to automatically identify the endpoints of each δ -connected component.

Lastly, one could think about designing a grouping method that inherently satisfies several rules of thumb, applicable to healthy human individuals. Examples are that arteries typically do not cross other arteries, veins do not cross other veins, and often, the order of blood vessels leaving the optic disk is alternating. Implementing this requires fundamental changes to our current strategy, but is expected to improve the classification results.

We also analyzed the dice scores of different segmentation maps. We saw that our initial segmentation (using the vesselness filter) was not as accurate as the segmentation of a well-trained network. Additionally, we identified the optimal threshold for the vesselness, calculated on a binary image.

After that, we lifted a binary map using different methods. Here, we found that binarization of the positive part of the orientation scores, using threshold 0.1, gave the best dice score. However, vesselness followed shortly after with optimal threshold 0.57, less sensitivity to the selected threshold value.

Finally, we considered the transferability of trained networks from one dataset to another. Here, we found that, even though data augmentation slightly worsened performance on the dataset on which it was trained (Fig. 9.17a), significantly improved performance on datasets it was not trained on (Fig. 9.17b), especially for the arteries in the images. We recommend training the network on a dataset that is as diverse as possible. This ranges from using different cameras to including different pathologies. All influence the performance of the network.

DISCUSSION AND OUTLOOK

In this thesis, we have developed techniques to automate retinal image analysis. We have focused on three areas of research: 1) Tracking, 2) Grouping, and 3) Classifying blood vessels. Most of the methods (tracking, grouping) can also be used for various other applications, such as crack detection in bridges [122]. Below, we summarize the developments in each area.

TRACKING OF COMPLEX VASCULAR TREES IN RETINAL IMAGES

We have improved the existing tracking models by designing an extension built on the underlying image data in Chapter 4. This ensures that the tracking results better follow the local structures. This results in tracts that can follow highly sinuous structures. This is important to track highly curved vessels, which often occur in retinal images of, for example, diabetic patients (cf. Fig. 1). We have added a data term (norm of the Hessian) to the classical construction of the Reeds-Shepp car model (Eq. (4.4)). This ensures that moving in the direction of local structures results in small distances, even if these structures are not perfectly aligned with the local orientation, or are curved and thus change orientation. We refer to the diagonalization of the new data-driven left-invariant model as the data-driven left-invariant frame. We can use this data-driven frame to define a data-driven Cartan connection. In Theorem 4.3.1, we have stated and proved how “straight” and “short” curves are described with respect to the data-driven Cartan connection. We additionally explained that this can be extended to the asymmetric Finsler model that automatically places in-place rotations at bifurcations avoiding the formation of cusps.

We have used a new implementation of anisotropic fast-marching that supports the data-driven left-invariant frame. In Section 4.4, we elaborated on the implementation of this algorithm and proved the necessary extensions and generalizations of existing theory. The experimental results show significant improvements in the model’s ability to follow curved vessels (cf. Figs. 4.3 to 4.5). Full vascular tree tracking needs to be handled with care at difficult crossings, which is done in the mixed model Eq. (5.13). In general, the tracking results perform very well in the discussed two-step approach (cf. Figs. 4.10 and 4.11), by first connecting the tips to the nearest bifurcation, and then connecting these bifurcations to the corresponding seeds. After prior classification of seeds and

tips belonging to the same vascular tree, the tracking results follow the vessels perfectly (cf. Fig. 4.12).

Additionally, we have looked at the influence of optical and TV-flow enhancements while tracking; specifically on retinal images. We have developed a new asymmetric, data-driven left-invariant Finsler geometric model that includes contextual contrast enhancement via TV-flows on $SE(2)$ in Chapter 5. We use our knowledge of the physics behind the retinal image to retrieve the real retina. The recovered image suffers less from uneven illumination (and haze), as shown in [206]. The recovered image still suffers from noise. Therefore, we apply TV-flow enhancement on $SE(2)$, a crossing-preserving denoiser. We have applied our new model to realistic, challenging retinal images where full vascular trees are computed from a *single* asymmetric Finslerian distance map. Then, we showed in the experiments that the tracking results significantly improve with each processing step. There are still exceptional cases where vessel tracts go wrong, and take the wrong exit. This occurs at places where crossings and bifurcations are very close to each other (cf. the red circles in Fig. 5.2c). Additionally, we showed that the more prior knowledge on connectivity is known, the more accurate the tracking results are. However, this requires too costly user-knowledge, so we have worked on developing automatic grouping methods in Parts III and IV.

Lastly, we have introduced a model on the lifted space of spherical positions and orientation \mathbb{W}_2 that accounts for local angular information in Chapter 6. We have extended the model introduced in [135] to deal with wide-field retinal images and to induce cusp-free geodesics. Additionally, we identified the coordinate mapping between \mathbb{W}_2 and the space of planar positions and orientations \mathbb{M}_2 . This mapping is constructed in such a way that horizontal curves in \mathbb{W}_2 map to horizontal curves in \mathbb{M}_2 , using the same spatial coordinate mapping introduced in [135]. The coordinate mapping allows us to pull back the cost function from $C^{\mathbb{M}_2}$ to $C^{\mathbb{W}_2}$. We have validated the effectiveness of the extension of the coordinate mapping in the experimental section (Section 6.4). We conclude that tracking models in the space of spherical positions and orientations \mathbb{W}_2 perform better when using a crossing-preserving cost function, which differentiates between structures at crossings of blood vessels. Additionally, we found that the tracking results (on wide-field images) in \mathbb{W}_2 can improve upon those calculated in the space of planar positions and orientations \mathbb{M}_2 .

For future work, it would be interesting to look into the possibilities to train the cost function C using PDE-G-CNNs [184], which is now geometrically computed as explained in Appendix C. In the past, this method had promising results for vessel segmentation. Additionally, it would be interesting to extend the model that does crossing-preserving tracking on the sphere to a data-driven spherical tracking model, in which we assure that the forward direction \mathcal{B}_1

is better aligned with the underlying image data, as we also did for the left-invariant tracking on planar images in Chapter 4. Lastly, it would be interesting to develop methods that can automatically identify the width of a vessel and of the vessel wall, as they also contain important information on an individual’s health. The methods in this thesis mainly focus on tracking, where often the centerline is identified. We believe that these methods can be extended to include information about vessel and vessel wall width in a straightforward way.

IDENTIFYING CONNECTED COMPONENTS ON NON-FUZZY SEGMENTATIONS OF VASCULATURE

We have designed a new connected component method that identifies δ -connected components in any Lie group G , i.e., points at most distance δ apart belong to the same component. We have developed the corresponding theory and proved several properties such as convergence (Theorem 8.4.1). We discussed the choice of the parameter δ and suggested using persistence homology arguments to choose the optimal value for δ , illustrated with a few examples.

Once the δ -connected components are introduced and calculated, we determine a hierarchy between the different components. For this, we propose to use specific affinity matrices, which group components based on their proximity and local alignment. To account for subtleties in the data, we include a data term in the initialization of the affinity matrices (Algorithm 2).

To show the benefits and performance of the δ -connected component and the affinity matrices, we have tested both on several 2D images of the retina. In the experiments, we used the Lie group $G = \text{SE}(2)$, where we can differentiate between structures at crossings. All experiments show that the δ -connected component algorithm differentiates between structures at crossings, and additionally groups well-aligned structures. This results in a more complete and more correct vascular tree in the output of the algorithm. However, it cannot always identify full vascular trees. Therefore, we calculated the affinity matrices for several experimental results of the δ -connected component algorithm. We see that this leads to more complete vascular trees, where different δ -connected components are grouped belonging to the same vascular tree.

Both algorithms are challenged by gaps at highly curved parts in the vascular structure. It cannot connect these structures without connecting different vascular trees at crossings. Large spatial gaps also form challenges: choosing a value that is too high for δ results in connecting vessel segments belonging to different vascular trees.

For future work, it would be interesting to train the metric tensor weights (w_1, w_2, w_3) via a PDE-G-CNN on $G = \text{SE}(2)$, cf. [184], to improve results further, and to explore whether optimal transport on $\text{SE}(2)$ [34] would be a viable alternative to determine the affinity between different δ -connected components. Additionally, it would be interesting to develop a data-driven version of the δ -connected component algorithm. By including a cost function in the calculation of the distances, we expect that the grouping of components will be beneficial for effectively grouping structures, particularly in retinal images. The data-driven distance will be smaller on well-aligned, similar structures, whereas before, only alignment and proximity were of importance to the δ -connected components.

CLASSIFYING VESSELS AS ARTERIES OR VEINS

We designed a new method to ensure consistent classification along the vasculature relying on the δ -connected components. We showed that this method worked well on synthetic examples. The generalization to real data was more challenging. The performance on networks trained with limited data was better or similar to the underlying networks, but was outperformed by networks trained on larger datasets.

We adapted our method from calculating a segmentation map itself, to using the segmentation map of the network. Additionally, we used the neural network output by lifting the arteries and veins separately before applying our post-processing technique. This fixed some of the issues, but not all, as visible in Fig. 9.15.

We also saw that our method is very sensitive to the right choice of parameters (see Fig. 9.16), where the right settings in one area lead to problems in another and vice versa. Therefore, we advise using a local, position-orientation-dependent and data-driven way to identify the δ -connected components in the future.

Alternatively, using crossing-preserving tracking methods to identify full vascular trees would be a valuable addition making the classification more consistent. One could use these tracking techniques to target specific regions that make the labeling inconsistent. Consequently, one can effectively modify the prediction, ensuring that the labeling along vascular trees is (more) consistent.

Finally, one could consider building a method that ensures that a set of rules is automatically satisfied. Examples of these rules include: arteries do not cross other arteries, veins do not cross other veins, and vessel trees leaving the optic disk have alternating labels. Adding these rules to our current framework would require significant adaptations, but is expected to improve the classification

result. However, it is important to realize that adapting the model such that a given set of rules is satisfied, can lead to more mistakes when dealing with specific diseases, typically not satisfying this set of rules.

Besides the method that we introduced, we also compared the segmentations produced by trained networks to those of the vesselness. The vesselness performed better than the U-nets with limited training data, but was outperformed by the networks with sufficient training data.

We also looked into the transferability of the trained networks to other datasets. Here, we realized that it is very important to do data-augmentation. Although this led to a slightly worse classification on the test dataset belonging to the training dataset (Leuven-Haifa [87]), it resulted in a significantly better classification on the STAR dataset [1, 205], especially for the arteries in the images. Hence, we advise training the network on a dataset that is as diverse as possible. This ranges from datasets produced by different cameras and in different setups to including different pathologies, since all these factors influence network performance.

BIBLIOGRAPHY

- [1] S. ABBASI-SURESHJANI, I. SMIT-OCKELOEN, J. ZHANG, and B. M. TER HAAR ROMENY. "Biologically-Inspired Supervised Vasculature Segmentation in SLO Retinal Fundus Images." In: *Image Analysis and Recognition*. Ed. by M. KAMEL and A. CAMPILHO. Cham: Springer International Publishing, 2015, pp. 325–334.
- [2] S. ABBASI-SURESHJANI, J. ZHANG, R. DUIT, and B. M. TER HAAR ROMENY. "Retrieving challenging vessel connections in retinal images by line co-occurrence statistics." In: *Biological Cybernetics* 111.3 (2017), pp. 237–247.
- [3] S. ABBASI-SURESHJANI, J. ZHANG, G. R. SANGUINETTI, R. DUIT, and B. M. TER HAAR ROMENY. "Geometric Connectivity Analysis Based on Edge Co-Occurrences in Retinal Images." In: *Proceedings of the Ophthalmic Medical Image Analysis International Workshop* 3(2016). 2016, pp. 154–155.
- [4] A. A. ABDULSAHIB, M. A. MAHMOUD, M. A. MOHAMMED, H. H. RASHEED, S. A. MOSTAFA, and M. S. MAASHI. "Comprehensive review of retinal blood vessel segmentation and classification techniques: intelligent solutions for green computing in medical images, current challenges, open issues, and knowledge gaps in fundus medical images." In: *Network Modeling Analysis in Health Informatics and Bioinformatics* 10 (Mar. 2021).
- [5] P.-A. ABSIL, R. E. MAHONY, and R. SEPULCHRE. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.
- [6] A. A. AGRACHEV, D. BARILARI, and U. BOSCAIN. *A Comprehensive Introduction to Sub-Riemannian Geometry*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2019.
- [7] A. A. AGRACHEV and Y. L. SACHKOV. *Control Theory from the Geometrical Viewpoint*. Vol. 87. Springer-Verlag, 2004.
- [8] S. AKBAR, M. USMAN AKRAM, M. SHARIF, A. TARIQ, and U. U. YASIN. "Arteriovenous ratio and papilledema based hybrid decision support system for detection and grading of hypertensive retinopathy." In: *Computer Methods and Programs in Biomedicine* 154 (2018), pp. 123–141.

- [9] K. AKITA and H. KUGA. "Pattern Recognition Of Blood Vessel Networks In Ocular Fundus Images." In: *Medical Imaging and Image Interpretation*. Ed. by J. M. S. PREWITT. Vol. 0375. International Society for Optics and Photonics. SPIE, 1982, pp. 436–443.
- [10] J. ANGULO and S. VELASCO-FORERO. "Riemannian Mathematical Morphology." In: *Pattern Recognition Letters* 47 (2014). Advances in Mathematical Morphology, pp. 93–101.
- [11] A. APPAJI et al. "Relation between retinal vascular abnormalities and working memory impairment in patients with schizophrenia and bipolar disorder." In: *Asian Journal of Psychiatry* 49 (2020).
- [12] D. ATTALI, H. DAL POZ KOŘIMSKÁ, C. FILLMORE, I. GHOSH, A. LIEUTIER, E. STEPHENSON, and M. WINTRAECKEN. "Tight Bounds for the Learning of Homotopy à la Niyogi, Smale, and Weinberger for Subsets of Euclidean Spaces and of Riemannian Manifolds." In: *40th International Symposium on Computational Geometry (SoCG 2024)*. Ed. by W. MULZER and J. M. PHILLIPS. Vol. 293. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 11:1–11:19.
- [13] D. AZAGRA, J. FERRERA, and F. LÓPEZ-MESAS. "Nonsmooth analysis and Hamilton–Jacobi equations on Riemannian manifolds." In: *Journal of Functional Analysis* 220.2 (2005), pp. 304–361.
- [14] D. BAO, S.-S. CHERN, and Z. SHEN. *An Introduction to Riemann-Finsler Geometry*. Springer New York, 2000.
- [15] D. BARBIERI, G. CITTI, G. COCCI, and A. SARTI. "A Cortical-Inspired Geometry for Contour Perception and Motion Integration." In: *JMIV* 49 (2014), pp. 511–529.
- [16] E. J. BEKKERS. "Retinal Image Analysis using Sub-Riemannian Geometry in $SE(2)$." PhD Thesis. TU Eindhoven, 2017.
- [17] E. J. BEKKERS, D. CHEN, and J. M. PORTEGIES. "Nilpotent approximations of sub-Riemannian distances for fast perceptual grouping of blood vessels in 2D and 3D." In: *Journal of Mathematical Imaging and Vision* (2018), pp. 1–18.
- [18] E. J. BEKKERS, R. DUTTS, T. T. BERENDSCHOT, and B. M. TER HAAR ROMENY. "A multi-orientation analysis approach to retinal vessel tracking." In: *Journal of Mathematical Imaging and Vision* 49.3 (2014), pp. 583–610.
- [19] E. J. BEKKERS, R. DUTTS, A. MASHTAKOV, and G. R. SANGUINETTI. "A PDE approach to data-driven sub-Riemannian geodesics in $SE(2)$." In: *SIAM SIMS* 8.4 (2015), pp. 2740–2770.

- [20] E. J. BEKKERS, M. W. LAFARGE, M. VETA, K. A. J. EPPENHOF, J. P. W. PLUIM, and R. DUTTS. "Roto-Translation Covariant Convolutional Networks for Medical Image Analysis." In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Cham: Springer International Publishing, 2018, pp. 440–448.
- [21] E. J. BEKKERS, J. ZHANG, R. DUTTS, and B. M. TER HAAR ROMENY. "Curvature based biomarkers for diabetic retinopathy via exponential curve fits in $SE(2)$." In: *Proceedings of the Ophthalmic Medical Image Analysis Second International Workshop, OMIA*. 2015, pp. 113–120.
- [22] W. J. BEKSI and N. PAPANIKOLOPOULOS. "3D point cloud segmentation using topological persistence." In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 5046–5051.
- [23] G. BELLAARD, D. L. J. BON, G. PAI, B. M. N. SMETS, and R. DUTTS. "Analysis of (sub-)Riemannian PDE-G-CNNs." In: *Journal of Mathematical Imaging and Vision* (2023).
- [24] F. BENMANSOUR and L. D. COHEN. "Tubular structure segmentation based on minimal path method and anisotropic enhancement." In: *International Journal of Computer Vision* 92.2 (2011), pp. 192–210.
- [25] V. N. BERESTOVSKIĬ and I. A. ZUBAREVA. "Sub-Riemannian distance in the Lie groups $SU(2)$ and $SO(3)$." In: *Siberian Advances in Mathematics* 26.2 (2016), pp. 77–89.
- [26] N. J. VAN DEN BERG. *Data-driven left-invariant tracking on optically enhanced images with total variation flow in Mathematica*. https://github.com/NickyvdBerg/tracking_enhancement_total_variation_flow. 2023.
- [27] N. J. VAN DEN BERG. *Image Analysis Applications of Connected Components on Lie Groups in $SE(2)$ in Mathematica*. <https://github.com/NickyvdBerg/ConnectedComponentsOnLieGroups>. 2024.
- [28] N. J. VAN DEN BERG, O. MULA, L. VIS, and R. DUTTS. *Connected Components on Lie Groups and Applications to Multi-Orientation Image Analysis*. arXiv preprint arXiv:2409.18002. 2024.
- [29] N. J. VAN DEN BERG, F. M. SHERRY, T. T. J. M. BERENDSCHOT, and R. DUTTS. "Crossing-Preserving Geodesic Tracking on Spherical Images." In: *Scale Space and Variational Methods in Computer Vision*. Ed. by T. A. BUBBA, R. GABURRO, S. GAZZOLA, K. PAPAFITSOROS, M. PEREYRA, and C.-B. SCHÖNLIEB. Cham: Springer Nature Switzerland, 2025, pp. 192–204.
- [30] N. J. VAN DEN BERG, B. M. N. SMETS, G. PAI, J.-M. MIREBEAU, and R. DUTTS. "Geodesic Tracking via New Data-Driven Connections of Cartan Type for Vascular Tree Tracking." In: *JMIV* 66 (2024), pp. 198–230.

- [31] N. J. VAN DEN BERG, S. ZHANG, B. M. N. SMETS, T. T. J. M. BERENDSCHOT, and R. Duits. "Geodesic Tracking of Retinal Vascular Trees with Optical and TV-Flow Enhancement in $SE(2)$." In: *Scale Space and Variational Methods in Computer Vision*. Ed. by L. CALATRONI, M. DONATELLI, S. MORIGI, M. PRATO, and M. SANTACESARIA. Cham: Springer International Publishing, 2023, pp. 525–537.
- [32] I. Y. BESCHASTNYI and Y. L. SACHKOV. "Sub-Riemannian and almost-Riemannian geodesics on $SO(3)$ and S^2 ." In: *preprint* (2014).
- [33] J.-D. BOISSONNAT, F. CHAZAL, and M. YVINEC. *Geometric and topological inference*. Vol. 57. Cambridge University Press, 2018.
- [34] D. L. J. BON, G. PAI, G. BELLAARD, O. MULA, and R. Duits. "Optimal Transport on the Lie Group of Roto-translations." In: *SIAM Journal on Imaging Sciences* 18.2 (2025), pp. 789–821.
- [35] R. VAN DEN BOOMGAARD and R. VAN BALEN. "Methods for Fast Morphological Image Transforms Using Bitmapped Binary Images." In: *CVGIP: Graphical Models and Image Processing* 54.3 (1992), pp. 252–258.
- [36] U. BOSCAIN, R. A. CHERTOVSKIY, J. P. GAUTHIER, and A. O. REMIZOV. "Hyperelliptic Diffusion and Human Vision: A Semidiscrete New Twist." In: *SIAM Journal on Imaging Sciences* 7.2 (2014), pp. 669–695.
- [37] U. BOSCAIN and F. ROSSI. "Projective Reeds-Shepp Car on S^2 with Quadratic Cost." In: *COCV* 16 (2010), pp. 275–297.
- [38] BOSCAIN, UGO AND CHARLOT, GRÉGOIRE AND ROSSI, FRANCESCO. "Existence of planar curves minimizing length and curvature." In: *Proceedings of the Steklov Institute of Mathematics* 270 (2010), pp. 43–56.
- [39] L. BRAZIONIS et al. "Review and comparison of retinal vessel calibre and geometry software and their application to diabetes, cardiovascular disease, and dementia." In: *Graefe's archive for clinical and experimental ophthalmology* 261 (8 2023), pp. 2117–2133.
- [40] T. BRENDL, J. SCHWANKE, P. F. JENSCH, and R. MEGNET. "Knowledge-based object recognition for different morphological classes of plants." In: *Optics in Agriculture, Forestry, and Biological Processing*. Vol. 2345. SPIE. 1995, pp. 277–284.
- [41] J. L. BROOKS. "57 Traditional and new principles of perceptual grouping." In: *The Oxford Handbook of Perceptual Organization*. Oxford University Press, Aug. 2015. eprint: https://academic.oup.com/book/0/chapter/333038172/chapter-ag-pdf/44459793/book_38175_section_333038172.ag.pdf.

- [42] B. BURGETH, M. WELK, C. FEDDERN, and J. WEICKERT. "Morphological Operations on Matrix-Valued Images." In: *Computer Vision - ECCV 2004*. Ed. by T. PAJDLA and J. MATAS. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 155–167.
- [43] Y. CAO, P. LEUNG, and A. MONOD. "k-means clustering for persistent homology." In: *Advances in Data Analysis and Classification* 19 (1 Mar. 2025), pp. 95–119.
- [44] V. CASELLES, R. KIMMEL, and G. SAPIRO. "Geodesic Active Contours." In: *International Journal of Computer Vision* 22 (Feb. 1997), pp. 61–79.
- [45] V. CASELLES, R. KIMMEL, and G. SAPIRO. "Geodesic Active Contours." In: *International Journal of Computer Vision* 22 (Feb. 1997), pp. 61–79.
- [46] F. CHAZAL, L. J. GUIBAS, S. Y. OUDOT, and P. SKRABA. "Persistence-Based Clustering in Riemannian Manifolds." In: *J. ACM* 60.6 (Nov. 2013).
- [47] D. CHEN. "New Minimal Paths Models for Tubular Structure Extraction and Image Segmentation." PhD thesis. University Paris-Dauphine, 2016.
- [48] D. CHEN and L. D. COHEN. "A New Dynamic Minimal Path Model for Tubular Structure Centerline Delineation." In: *2018 24th International Conference on Pattern Recognition (ICPR)*. 2018, pp. 3001–3006.
- [49] D. CHEN and L. D. COHEN. "Fast asymmetric fronts propagation for image segmentation." In: *Journal of Mathematical Imaging and Vision* 60.6 (2018), pp. 766–783.
- [50] D. CHEN, J.-M. MIREBEAU, and L. D. COHEN. "Vessel tree extraction using radius-lifted keypoints searching scheme and anisotropic fast marching method." In: *Journal of Algorithms & Computational Technology* 10.4 (2016), pp. 224–234.
- [51] D. CHEN, J. SPENCER, J.-M. MIREBEAU, K. CHEN, M. SHU, and L. D. COHEN. "A Generalized Asymmetric Dual-Front Model for Active Contours and Image Segmentation." In: *IEEE Transactions on Image Processing* 30 (2021), pp. 5056–5071.
- [52] W. CHEN, S. YU, K. MA, W. JI, C. BIAN, C. CHU, L. SHEN, and Y. ZHENG. "TW-GAN: Topology and width aware GAN for retinal artery/vein classification." In: *Medical Image Analysis* 77 (2022), p. 102340.
- [53] C. Y.-l. CHEUNG, E. LAMOUREUX, M. K. IKRAM, M. B. SASONGKO, J. DING, Y. ZHENG, P. MITCHELL, J. J. WANG, and T. Y. WONG. "Retinal vascular geometry in Asian persons with diabetes and retinopathy." In: *J Diabetes Sci Technol* 6.3 (2012), pp. 595–605.

- [54] D. CHUDASAMA, T. PATEL, S. JOSHI, and G. PRAJAPATI. "Image Segmentation using Morphological Operations." In: *International Journal of Computer Applications* 117 (May 2015), pp. 16–19.
- [55] G. CITTI, B. FRANCESCHIELLO, G. SANGUINETTI, and A. SARTI. "Sub-Riemannian Mean Curvature Flow for Image Processing." In: *SIAM Journal on Imaging Sciences* 9 (1 2016), pp. 212–237.
- [56] G. CITTI and A. SARTI. "A cortical based model of perceptual completion in the roto-translation space." In: *Journal of Mathematical Imaging and Vision* 24.3 (2006), pp. 307–326.
- [57] A. COGLIATI and P. MASTROLIA. "Cartan, Schouten and the search for connection." In: *Historia Mathematica* 45 (2017), pp. 39–74.
- [58] L. D. COHEN. "Multiple contour finding and perceptual grouping using minimal paths." In: *Journal of mathematical imaging and vision* 14.3 (2001), pp. 225–236.
- [59] T. COHEN and M. WELLING. "Group equivariant convolutional networks." In: *International conference on machine learning*. PMLR. 2016, pp. 2990–2999.
- [60] P. COLLIGRIS, M. J. PEREZ DE LARA, B. COLLIGRIS, and J. PINTOR. "Ocular manifestations of Alzheimer's and other neurodegenerative diseases: the prospect of the eye as a tool for the early diagnosis of Alzheimer's disease." In: *Journal of ophthal.* 2018 (2018).
- [61] T. F. COOTES, C. J. TAYLOR, D. H. COOPER, and J. GRAHAM. "Active shape models-their training and application." In: *Computer vision and image understanding* 61.1 (1995), pp. 38–59.
- [62] D. J. CREEL. *Clinical Electrophysiology*. Accessed: 2021-05-03. July 2007.
- [63] B. DASHTBOZORG, A. M. MENDONÇA, and A. CAMPILHO. "Automatic Estimation of the Arteriolar-to-Venular Ratio in Retinal Images Using a Graph-Based Approach for Artery/Vein Classification." In: *Image Analysis and Recognition*. Ed. by M. KAMEL and A. CAMPILHO. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 530–538.
- [64] B. DASHTBOZORG, A. M. MENDONÇA, and A. CAMPILHO. "An Automatic Graph-Based Approach for Artery/Vein Classification in Retinal Images." In: *IEEE Transactions on Image Processing* 23.3 (2014), pp. 1073–1083.
- [65] F. G. B. DE NATALE and G. BOATO. "Detecting Morphological Filtering of Binary Images." In: *IEEE Transactions on Information Forensics and Security* 12.5 (2017), pp. 1207–1217.

- [66] T. S. DEEPTHI MURTHY and G. SADASHIVAPPA. "Brain tumor segmentation using thresholding, morphological operations and extraction of features of tumor." In: *2014 International Conference on Advances in Electronics Computers and Communications*. 2014, pp. 1–6.
- [67] T. K. DEY and Y. WANG. *Computational topology for data analysis*. Cambridge University Press, 2022.
- [68] R. DIESTEL. *Graph Theory*. Springer-Verlag New York, 1997.
- [69] E. H. S. DIOP, A. MBENGUE, B. MANGA, and D. SECK. "Extension of Mathematical Morphology in Riemannian Spaces." In: *Scale Space and Variational Methods in Computer Vision*. Cham: Springer International Publishing, 2021, pp. 100–111.
- [70] B. AL-DIRI and A. HUNTER. "Automated Measurements of Retinal Bifurcations." In: *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany*. Ed. by O. DÖSSEL and W. C. SCHLEGEL. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 205–208.
- [71] B. AL-DIRI, A. HUNTER, D. STEEL, and M. HABIB. "Manual measurement of retinal bifurcation features." In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. 2010, pp. 4760–4764.
- [72] J. DUGUNDJI. *Topology*. Allyn and Bacon, inc., 1966.
- [73] R. DUITS. "Perceptual organization in image analysis: a mathematical approach based on scale, orientation and curvature." PhD thesis. Eindhoven University of Technology, 2005.
- [74] R. DUITS, U. BOSCAIN, F. ROSSI, and Y. L. SACHKOV. "Association fields via cuspless sub-Riemannian geodesics in $SE(2)$." In: *Journal of mathematical imaging and vision* 49.2 (2014), pp. 384–417.
- [75] R. DUITS, M. DUITS, M. VAN ALMSICK, and B. M. TER HAAR ROMENY. "Invertible orientation scores as an application of generalized wavelet theory." In: *Pattern Recognition and Image Analysis* 17.1 (2007), pp. 42–75.
- [76] R. DUITS, M. FELSBERG, G. GRANLUND, and B. M. TER HAAR ROMENY. "Image Analysis and Reconstruction using a Wavelet Transform Constructed from a Reducible Representation of the Euclidean Motion Group." In: *International Journal of Computer Vision* 72 (1 Apr. 2007), pp. 79–102.

- [77] R. Duits and E. M. FRANKEN. "Left Invariant Parabolic Evolution Equations on $SE(2)$ and Contour Enhancement via Invertible Orientation Scores, Part II: Nonlinear Left-Invariant Diffusion Equations on Invertible Orientation Scores." In: *Quarterly of Applied mathematics*, AMS 68 (June 2010), pp. 293–331.
- [78] R. Duits and E. M. FRANKEN. "Left-Invariant Parabolic Evolutions on $SE(2)$ and Contour Enhancement via Invertible Orientation Scores Part I: Linear Left-Invariant Diffusion Equations on $SE(2)$." In: *Quarterly of Applied Mathematics* 68 (2 June 2010), pp. 255–292.
- [79] R. Duits and E. M. FRANKEN. "Left-Invariant Diffusions on the Space of Positions and Orientations and their Application to Crossing-Preserving Smoothing of HARDI images." In: *International Journal of Computer Vision* 92 (3 May 2011), pp. 231–264.
- [80] R. Duits, H. FUEHR, B. . JANSSEN, M. C. BRUURMIJN, L. M. FLORACK, and H. A. VAN ASSEN. "Evolution equations on Gabor transforms and their applications." In: *ACHA* 35.3 (2013), pp. 483–526.
- [81] R. Duits, A. GHOSH, T. C. J. DELA HAIJE, and A. MASHTAKOV. "On sub-Riemannian geodesics in $SE(3)$ whose spatial projections do not have cusps." In: *Journal of dynamical and control systems* 22.4 (2016), pp. 771–805.
- [82] R. Duits, M. H. JANSSEN, J. HANNINK, and G. R. SANGUINETTI. "Locally adaptive frames in the roto-translation group and their applications in medical imaging." In: *Journal of Mathematical Imaging and Vision* 56.3 (2016), pp. 367–402.
- [83] R. Duits, S. P. L. MEESTERS, J.-M. MIREBEAU, and J. M. PORTEGIES. "Optimal paths for variants of the 2D and 3D Reeds–Shepp car with applications in image analysis." In: *Journal of Mathematical Imaging and Vision* (2018), pp. 1–33.
- [84] R. Duits, B. M. N. SMETS, A. J. WEMMENHOVE, J. W. PORTEGIES, and E. J. BEKKERS. "Springer Handbook of Mathematical Imaging 2021." In: Springer, June 2021. Chap. Recent Geometric Flows in Multi-Orientation Image Processing via a Cartan Connection, pp. 1–60.
- [85] H. EDELSBRUNNER and J. HARER. *Computational topology: an introduction*. American Mathematical Soc., 2010.
- [86] H. EDELSBRUNNER, D. LETSCHER, and A. ZOMORODIAN. "Topological Persistence and Simplification." In: *Discrete & Computational Geometry* 28 (4 Nov. 2002), pp. 511–533.

- [87] J. VAN EIJGEN, J. FHIMA, M.-I. BILLEN MOULIN ROMSÉE, J. A. BEHAR, E. CHRISTINAKI, and I. STALMANS. "Leuven-Haifa High-Resolution Fundus Image Dataset for Retinal Blood Vessel Segmentation and Glaucoma Diagnosis." In: *Scientific Data* 11 (1 2024), p. 257.
- [88] J. VAN EIJGEN, L. VAN WINCKEL, H. HANSSEN, K. KOTLIAR, T. VANASSCHE, E. M. VAN CRAENENBROECK, V. CORNELISSEN, A. H. VAN CRAENENBROECK, E. JONES, and I. STALMANS. "Retinal vessel analysis to assess microvascular function in the healthy eye: A systematic review on the response to acute physiological and pathological stressors." In: *Survey of Ophthalmology* 70.2 (2025), pp. 200–214.
- [89] A. F. M. TER ELST and D. W. ROBINSON. "Weighted Subcoercive Operators on Lie Groups." In: *Journal of Functional Analysis* 157 (1998), pp. 88–163.
- [90] L. C. EVANS. "Partial differential equations." In: *Providence, RI* (1998).
- [91] B. FANG, W. HSU, and M. LEE. "On the Detection of Retinal Vessels in Fundus Images." <http://hdl.handle.net/1721.1/3675>. Nov. 2003.
- [92] A. FATHI and E. MADERNA. "Weak KAM theorem on non compact manifolds." In: *Nonlinear Differential Equations and Applications NoDEA* 14.1-2 (Aug. 2007), pp. 1–27.
- [93] P. J. FIGUEROA, N. J. LEITE, and R. M. L. BARROS. "A flexible software for tracking of markers used in human motion analysis." In: *Computer Methods and Programs in Biomedicine* 72.2 (2003), pp. 155–165.
- [94] G. L. FORESTI. "Object recognition and tracking for remote video surveillance." In: *IEEE Transactions on circuits and systems for video technology* 9.7 (1999), pp. 1045–1062.
- [95] P.-E. FORSSÉN. "Sparse representations for medium level vision." PhD thesis. Linköping University, 2001.
- [96] S. I. FRADKIN, D. BANNAI, P. LIZANO, A. LAI, C. CROSTA, J. L. THOMPSON, and S. M. SILVERSTEIN. "Deep retinal layer microvasculature alterations in schizophrenia." In: *Biomarkers in Neuropsychiatry* 10 (2024), p. 100084.
- [97] B. FRANCESCHIELLO, A. MASHTAKOV, G. CITTI, and A. SARTI. "Geometrical optical illusion via sub-Riemannian geodesics in the roto-translation group." In: *Differential Geometry and its Applications* 65 (2019), pp. 55–77.
- [98] A. F. FRANGI, W. J. NIJSEN, K. L. VINCKEN, and M. A. VIERGEVER. "Multi-scale Vessel Enhancement Filtering." In: *Med. Image Comput. Comput. Assist. Interv.* 1496 (Feb. 2000), pp. 130–137.
- [99] E. M. FRANKEN. "Enhancement of Crossing Elongated Structures in Images." PhD Thesis. TU Eindhoven, 2008.

- [100] E. M. FRANKEN and R. DUTTS. "Crossing-preserving coherence-enhancing diffusion on invertible orientation scores." In: *International Journal of Computer Vision* 85.3 (2009), p. 253.
- [101] F. GIRARD and F. CHERIET. "Artery/vein classification in fundus images using CNN and likelihood score propagation." In: *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2017, pp. 720–724.
- [102] F. GIRARD, C. KAVALEC, and F. CHERIET. "Joint segmentation and classification of retinal arteries/veins from fundus images." In: *Artificial Intelligence in Medicine* 94 (2019), pp. 96–109.
- [103] E. GRISAN and A. RUGGERI. "A divide et impera strategy for automatic classification of retinal vessels into arteries and veins." In: *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings* 1 (Oct. 2003), 890–893 Vol.1.
- [104] J. J. VAN DE GRONDE, M. LYSenko, and J. B. T. M. ROERDINK. "Path-Based Mathematical Morphology on Tensor Fields." In: *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*. Ed. by I. Hotz and T. Schultz. Cham: Springer International Publishing, 2015, pp. 109–127.
- [105] C. GU and M.-C. LEE. "Semiautomatic segmentation and tracking of semantic video objects." In: *IEEE Transactions on Circuits and Systems for Video Technology* 8.5 (1998), pp. 572–584.
- [106] B. M. TER HAAR ROMENY. *Front-End Vision and Multi-Scale Image Analysis*. Springer Netherlands, 2003.
- [107] A. HAAS, G. MATHERON, and J. SERRA. "Morphologie mathématique et granulométries en place." In: *Annales des mines*. Vol. 11. 1967, pp. 736–753.
- [108] J. HANNINK, R. DUTTS, and E. J. BEKKERS. "Crossing-preserving multi-scale vesselness." In: *Proc. MICCAI LNCS 8674*. Springer. 2014, pp. 603–610.
- [109] H. HANSSEN, L. STREESE, and W. VILSER. "Retinal vessel diameters and function in cardiovascular risk and disease." In: *Progress in Retinal and Eye Research* 91 (2022), p. 101095.
- [110] H. J. A. M. HEIJMANS. "Theoretical aspects of gray-level morphology." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.6 (1991), pp. 568–582.

- [111] H. J. A. M. HEIJMANS. "Connected Morphological Operators for Binary Images." In: *Computer Vision and Image Understanding* 73.1 (1999), pp. 99–120.
- [112] R. HEMELINGS, B. ELEN, I. STALMANS, K. VAN KEER, P. DE BOEVER, and M. B. BLASCHKO. "Artery–vein segmentation in fundus images using a fully convolutional network." In: *Computerized Medical Imaging and Graphics* 76 (2019), p. 101636.
- [113] L. Hosák et al. "Retinal arteriolar and venular diameters are widened in patients with schizophrenia." In: *Psychiatry and Clinical Neurosciences* 74 (11 2020), pp. 619–621.
- [114] Q. Hu, M. D. ABRÀMOFF, and M. K. K. GARVIN. "Automated construction of arterial and venous trees in retinal images." In: *Journal of Medical Imaging* 2.4 (2015), p. 044001.
- [115] F. HUANG, B. DASHTBOZORG, and B. M. t. H. ROMENY. "Artery/vein classification using reflection features in retina fundus images." In: *Machine Vision and Applications* 29.1 (2018), pp. 23–34.
- [116] A. A. KALITZEOS, G. Y. H. LIP, and R. HEITMAR. "Retinal vessel tortuosity measures and their applications." In: *Experimental eye research* 106.0 (2013), pp. 40–46.
- [117] H. J. KAPLAN. *Retina Image Bank; Severe NPDR*. <https://imagebank.asrs.org/file/5343/severe-npdr>. Accessed: 2021-05-03. Mar. 2013.
- [118] R. A. KARLSSON and S. H. HARDARSON. "Artery vein classification in fundus images using serially connected U-Nets." In: *Computer Methods and Programs in Biomedicine* 216 (2022), p. 106650.
- [119] J. KIM, J. SHIN, F. CHAZAL, A. RINALDO, and L. WASSERMAN. "Homotopy Reconstruction via the Cech Complex and the Vietoris-Rips Complex." In: *SoCG 2020- 36th International Symposium on Computational Geometry*. 2020.
- [120] A. KIRILLOV et al. "Segment Anything." In: *arXiv:2304.02643* (2023).
- [121] S. KOBAYASHI and K. NOMIZU. *Foundations of differential geometry*. Vol. 45. John Wiley & Sons, 1963.
- [122] A. KOMPANETS, R. DUIT, D. LEONETTI, N. J. VAN DEN BERG, and H. H. SNIJDER. "Segmentation Tool for Images of Cracks." In: *Advances in Information Technology in Civil and Building Engineering*. Ed. by S. SKATULLA and H. BEUSHAUSEN. Cham: Springer International Publishing, 2024, pp. 93–110.

- [123] C. KONDERMANN, D. KONDERMANN, and M. YAN. "Blood vessel classification into arteries and veins in retinal images." In: *Medical Imaging 2007: Image Processing*. Ed. by J. P. W. PLUIM and J. M. REINHARDT. Vol. 6512. International Society for Optics and Photonics. SPIE, 2007, p. 651247.
- [124] V. KORANN et al. "Association between retinal vascular caliber and brain structure in schizophrenia." In: *Asian Journal of Psychiatry* 70 (Apr. 2022), p. 103042.
- [125] T. C. LEE, R. L. KASHYAP, and C. N. CHU. "Building Skeleton Models via 3-D Medial Surface Axis Thinning Algorithms." In: *CVGIP: Graphical Models and Image Processing* 56.6 (1994), pp. 462–478.
- [126] H. LI, W. HSU, M. L. LEE, and H. WANG. "A piecewise Gaussian model for profiling and differentiating retinal vessels." In: *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*. Vol. 1. 2003, pp. I-1069.
- [127] M. LICHTENSTEIN, G. PAI, and R. KIMMEL. "Deep Eikonal Solvers." In: *Scale Space and Variational Methods in Computer Vision*. Ed. by J. LELLMANN, M. BURGER, and J. MODERSITZKI. Cham: Springer International Publishing, 2019, pp. 38–50.
- [128] L. LIU, D. CHEN, L. D. COHEN, H. SHU, and M. PÂQUES. "Vessel Extraction Using Crossing-Adaptive Minimal Path Model With Anisotropic Enhancement And Curvature Constraint." In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. 2019, pp. 20–23.
- [129] L. Liu, D. Chen, M. Shu, H. Shu, and L. D. Cohen. "A New Tubular Structure Tracking Algorithm Based On Curvature-Penalized Perceptual Grouping." In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 2195–2199.
- [130] L. LIU, M. WANG, S. ZHOU, M. SHU, L. D. COHEN, and D. CHEN. "Curvilinear Structure Tracking Based on Dynamic Curvature-penalized Geodesics." In: *Pattern Recognition* 134 (2023), p. 109079.
- [131] W. MA, S. YU, K. MA, J. WANG, X. DING, and Y. ZHENG. "Multi-task Neural Networks with Spatial Activation for Retinal Vessel Segmentation and Artery/Vein Classification." In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. Ed. by D. SHEN, T. LIU, T. M. PETERS, L. H. STAIB, C. ESSERT, S. ZHOU, P.-T. YAP, and A. KHAN. Cham: Springer International Publishing, 2019, pp. 769–778.
- [132] J. E. MARSDEN and T. S. RATIU. *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*. Vol. 17. Springer Science & Business Media, 2013.

- [133] M. E. MARTINEZ-PEREZ, A. D. HIGHES, A. V. STANTON, S. A. THORN, N. CHAPMAN, A. A. BHARATH, and K. H. PARKER. "Retinal vascular tree morphology: a semi-automatic quantification." In: *IEEE Transactions on Biomedical Engineering* 49.8 (2002), pp. 912–917.
- [134] A. MASHTAKOV. "Extremal Controls for the Duits Car." In: *Geometric Science of Information*. Ed. by F. NIELSEN and F. BARBARESCO. Cham: Springer International Publishing, 2021, pp. 73–81.
- [135] A. MASHTAKOV, R. DUTTS, Y. L. SACHKOV, E. J. BEKKERS, and I. Y. BESCHASTNYI. "Tracking of Lines in Spherical Images via Sub-Riemannian Geodesics in $SO(3)$." In: *JMIV* 58 (2 2017), pp. 239–264.
- [136] F. MEYER and S. BEUCHER. "Morphological segmentation." In: *Journal of Visual Communication and Image Representation* 1.1 (1990), pp. 21–46.
- [137] J.-M. MIREBEAU. "Fast-marching methods for curvature penalized shortest paths." In: *Journal of Mathematical Imaging and Vision* 60.6 (2018), pp. 784–815.
- [138] J.-M. MIREBEAU. "Riemannian Fast-Marching on Cartesian Grids, Using Voronoi's First Reduction of Quadratic Forms." In: *SIAM Journal on Numerical Analysis* 57.6 (2019), pp. 2608–2655.
- [139] J.-M. MIREBEAU. *Adaptive grid discretization - A set of tools for discretizing anisotropic PDEs on cartesian grids*. https://nbviewer.jupyter.org/github/Mirebeau/AdaptiveGridDiscretizations_showcase/blob/master/Summary.ipynb. Accessed: 2021-03-17.
- [140] J.-M. MIREBEAU and J. PORTEGIES. "Hamiltonian fast marching: a numerical solver for anisotropic and non-holonomic eikonal PDEs." In: *Image Processing On Line* 9 (2019), pp. 47–93.
- [141] Q. MIRSHARIF, F. TAJERIPOUR, and H. POURREZA. "Automated characterization of blood vessels as arteries and veins in retinal images." In: *Computerized Medical Imaging and Graphics* 37.7 (2013), pp. 607–617.
- [142] J. MITCHELL et al. "Choroidal and retinal vascular changes in adults with Down syndrome: Insights into the Alzheimer's disease continuum." In: *Alzheimer's & Dementia* 21.5 (May 2025).
- [143] A. MITRA, S. Roy, S. Roy, and S. K. SETUA. "Enhancement and restoration of non-uniform illuminated Fundus Image of Retina obtained through thin layer of cataract." In: *Computer Methods and Programs in Biomedicine* 156 (2018), pp. 169–178.
- [144] I. MOISEEV and Y. L. SACHKOV. "Maxwell strata in sub-Riemannian problem on the group of motions of a plane." In: *ESAIM: Control, Optimisation and Calculus of Variations* 16.2 (2010), pp. 380–399.

- [145] P. MOMAYYEZ and K. SIDDIQI. “3D stochastic completion fields for fiber tractography.” In: *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2009, pp. 178–185.
- [146] R. MONTGOMERY. *A tour of subriemannian geometries, their geodesics and applications*. 91. American Mathematical Soc., 2002.
- [147] M. R. K. MOOKIAH, S. HOGG, T. J. MACGILLIVRAY, V. PRATHIBA, R. PRADEEP, V. MOHAN, R. M. ANJANA, A. S. DONEY, C. N. A. PALMER, and E. TRUCCO. “A review of machine learning methods for retinal blood vessel segmentation and artery/vein classification.” In: *Medical Image Analysis* 68 (2021), p. 101905.
- [148] D. MURRAY and A. BASU. “Motion tracking with an active camera.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.5 (1994), pp. 449–459.
- [149] P. NRYOGI, S. SMALE, and S. WEINBERGER. “Finding the homology of submanifolds with high confidence from random samples.” In: *Discrete & Computational Geometry* 39 (2008), pp. 419–441.
- [150] G. K. OUZOUNIS. “Generalized Connected Morphological Operators for Robust Shape Extraction.” PhD thesis. Rijksuniversiteit Groningen, 2009.
- [151] N. R. PAL and S. K. PAL. “A review on image segmentation techniques.” In: *Pattern Recognition* 26.9 (1993), pp. 1277–1294.
- [152] B. PAWAR, V. T. HUMBE, and L. KUNDNANI. “Morphology based moving vehicle detection.” In: *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*. 2017, pp. 217–223.
- [153] M. PÉCHAUD, M. DESCOTEAUX, and R. KERIVEN. “Brain connectivity using geodesics in HARDI.” In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2009, pp. 482–489.
- [154] J. PETITOT. *Elements of Neurogeometry. Functional Architectures of Vision*. Springer Cham, 2017.
- [155] G. PEYRÉ, M. PÉCHAUD, R. KERIVEN, and L. D. COHEN. “Geodesic methods in computer vision and graphics.” In: *Foundations and Trends® in Computer Graphics and Vision* 5.3–4 (2010), pp. 197–397.
- [156] E. PIUZE, J. SPORRING, and K. SIDDIQI. “Maurer-Cartan Forms for Fields on Surfaces: Application to Heart Fiber Geometry.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.12 (2015), pp. 2492–2504.

- [157] J. M. PORTEGIES, R. H. J. FICK, G. R. SANGUINETTI, S. P. MEESTERS, G. GIRARD, and R. DUIT. "Improving fiber alignment in HARDI by combining contextual PDE flow with constrained spherical deconvolution." In: *Plos one* 10.10 (2015), e0138122.
- [158] J. M. PORTEGIES, S. P. L. MEESTERS, P. OSSENBLOK, A. FUSTER, L. M. J. FLORACK, and R. DUIT. "Brain Connectivity Measures via Direct Sub-Finslerian Front Propagation on the 5D Sphere Bundle of Positions and Directions." In: *Computational Diffusion MRI*. Ed. by E. BONET-CARNE, F. GRUSSU, L. NING, F. SEPEHRBAND, and C. M. W. TAX. Cham: Springer International Publishing, 2019, pp. 309–321.
- [159] Z. YU-QIAN, G. WEI-HUA, C. ZHEN-CHENG, T. JING-TIAN, and L. LING-YUN. "Medical Images Edge Detection Based on Mathematical Morphology." In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. 2005, pp. 6492–6495.
- [160] T. A. QURESHI, M. HABIB, A. HUNTER, and B. AL-DIRI. "A manually-labeled, artery/vein classified benchmark for the DRIVE dataset." In: *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*. 2013, pp. 485–488.
- [161] N. RAVI et al. "SAM 2: Segment Anything in Images and Videos." In: *arXiv preprint arXiv:2408.00714* (2024).
- [162] S. REBOUÇAS et al. "Retinal microvasculature and incident dementia over 10 years: results from the Three-City-Alienor cohort." In: *Alzheimer's & Dementia* 15 (4 Oct. 2023).
- [163] J. REEDS and L. SHEPP. "Optimal paths for a car that goes both forwards and backwards." In: *Pacific journal of mathematics* 145.2 (1990), pp. 367–393.
- [164] D. RELAN, T. J. MACGILLIVRAY, L. BALLERINI, and E. TRUCCO. "Retinal vessel classification: Sorting arteries and veins." In: *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2013, pp. 7396–7399.
- [165] J. B. T. M. ROERDINK and A. MEIJSTER. "The Watershed Transform: Definitions, Algorithms and Parallelization Strategies." In: *Fundamenta informaticae* 41.1-2 (2000), pp. 187–228.
- [166] K. ROTHHAUS, X. JIANG, and P. RHIEM. "Separation of the retinal vascular graph in arteries and veins based upon structural knowledge." In: *Image and Vision Computing* 27.7 (2009). 7th IAPR-TC15 Workshop on Graph-based Representations (GbR 2007), pp. 864–875.

- [167] C. RUST, S. HÄGER, N. TRAULSEN, and J. MODERSITZKI. "A robust algorithm for optic disc segmentation and fovea detection in retinal fundus images." In: *Current Directions in Biomedical Engineering* 3.2 (2017), pp. 533–537.
- [168] Y. L. SACHKOV. "Cut Locus and Optimal Synthesis in the Sub-Riemannian Problem on the Group of Motions of a Plane." In: *ESAIM: Control, Optimization and Calculus of Variations* 17 (2011), pp. 293–321.
- [169] P. SALEMBIER, P. BRIGGER, J. R. CASAS, and M. PARDAS. "Morphological operators for image and video compression." In: *IEEE Transactions on Image Processing* 5.6 (1996), pp. 881–898.
- [170] P. SALEMBIER and M. H. F. WILKINSON. "Connected operators." In: *IEEE Signal Processing Magazine* 26.6 (2009), pp. 136–157.
- [171] G. R. SANGUINETTI. "Invariant models of vision between phenomenology, image statistics and neurosciences." PhD thesis. Universidad de la Republica, Montevideo, Uruguay, 2011.
- [172] G. SAPIRO. *Geometric partial differential equations and image analysis*. Cambridge university press, 2006.
- [173] M. B. SASONGKO, T. Y. WONG, T. T. NGUYEN, C. Y. CHEUNG, J. E. SHAW, R. KAWASAKI, E. L. LAMOUREUX, and J. J. WANG. "Retinal vessel tortuosity and its relation to traditional and novel vascular risk markers in persons with diabetes." In: *Current eye research* 41.4 (2016), pp. 551–557.
- [174] M. B. SASONGKO, T. Y. WONG, T. T. NGUYEN, C. Y. CHEUNG, J. E. SHAW, and J. J. WANG. "Retinal vascular tortuosity in persons with diabetes and diabetic retinopathy." In: *Diabetologia* 54.9 (2011), pp. 2409–2416.
- [175] P. SAVADJIEV, G. J. STRIKERS, A. J. BAKERMAN, E. PIUZE, S. W. ZUCKER, and K. SIDDIQI. "Heart wall myofibers are arranged in minimal surfaces to optimize organ function." In: *PNAS* 109.24 (2012), pp. 9248–9253.
- [176] M. SCHMIDT and J. WEICKERT. "Morphological Counterparts of Linear Shift-Invariant Scale Spaces." In: *Journal of Mathematical Imaging and Vision* 56 (2 Oct. 2016), pp. 352–366.
- [177] M. T. SCHRAM, S. J. SEP, C. J. VAN DER KALLEN, P. C. DAGNELIE, A. KOSTER, N. SCHAPER, R. M. A. HENRY, and C. D. A. STEHOUWER. "The Maastricht Study: an extensive phenotyping study on determinants of type 2 diabetes, its complications and its comorbidities." In: *European Journal of Epidemiology* 29.1 (2014), pp. 439–451.
- [178] E. SELLING. "Ueber die binären und ternären quadratischen Formen." In: *Journal für die reine und angewandte Mathematik* 77 (1874), pp. 143–229.

- [179] J. A. SETHIAN. "Fast marching methods." In: *SIAM review* 41.2 (1999), pp. 199–235.
- [180] F. SHERRY. *Solve Eikonal PDEs using an iterative method*. <https://github.com/finnsherry/IterativeEikonal>. 2024.
- [181] F. M. SHERRY, K. SCHAEFER, and R. Duits. "Diffusion-Shock Filtering on the Space of Positions and Orientations." In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2025, pp. 205–217.
- [182] F. Y. SHIH. "Object representation and recognition using mathematical morphology model." In: *Journal of Systems Integration* 1.2 (1991), pp. 235–256.
- [183] P. SKRABA, M. OVSJANIKOV, F. CHAZAL, and L. GIBAS. "Persistence-based segmentation of deformable shapes." In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*. 2010, pp. 45–52.
- [184] B. M. N. SMETS, J. W. PORTEGIES, E. J. BEKKERS, and R. Duits. "PDE-Based Group Equivariant Convolutional Neural Networks." In: *Journal of Mathematical Imaging and Vision* (2023).
- [185] B. M. N. SMETS, J. W. PORTEGIES, E. ST-ONGE, and R. Duits. "Total Variation and Mean Curvature PDEs on the Homogeneous Space of Positions and Orientations." In: *JMIV* 63.2 (2021), pp. 237–262.
- [186] B. M. SMETS. "Geometric Partial Differential Equations in Deep Learning & Image Processing." PhD thesis. Eindhoven University of Technology, 2024.
- [187] R. TINARRAGE. "Recovering the Homology of Immersed Manifolds." In: *Discrete & Computational Geometry* 69.3 (Feb. 2023), pp. 659–744.
- [188] M. USMAN AKRAM, A. KHAN, K. IQBAL, and W. H. BUTT. "Retinal Images: Optic Disk Localization and Detection." In: *Image Analysis and Recognition*. Ed. by A. CAMPILHO and M. KAMEL. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 40–49.
- [189] S. R. S. VADAMARAN. "On the behavior of the fundamental solution of the heat equation with variable coefficients." In: *Communications on Pure and Applied Mathematics* 20 (2 May 1967), pp. 431–455.
- [190] S. G. VÁZQUEZ, B. CANCELA, N. BARREIRA, M. G. PENEDO, M. RODRÍGUEZ-BLANCO, M. PENA SEJO, G. COLL TUERO, M. A. BARCELÓ, and M. SAEZ. "Improving retinal artery and vein classification by means of a minimal path approach." In: *Machine Vision and Applications* 24.5 (2013), pp. 919–930.

- [191] S. VELASCO-FORERO and J. ANGULO. "On Nonlocal Mathematical Morphology." In: *Mathematical Morphology and Its Applications to Signal and Image Processing*. Ed. by C. L. L. HENDRIKS, G. BORGEFORS, and R. STRAND. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 219–230.
- [192] V. VIJAYAKUMAR, D. D. KOOZEKANANI, R. WHITE, J. KOHLER, S. ROYCHOWDHURY, and K. K. PARHI. "Artery/vein classification of retinal blood vessels using feature selection." In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2016, pp. 1320–1323.
- [193] P. VISHNU PRIYA, A. SRINIVASA RAO, and J. SHARMA. "Diabetic Retinopathy - Can Lead to Complete Blindness." In: *International Journal of Science Inventions Today* 2 (Jan. 2013), pp. 254–265.
- [194] R. WALKER. *Human Body*. Dorling Kindersley Limited, 2014.
- [195] X. WEI, W. C. IAO, Y. ZHANG, Z. LIN, and H. LIN. "Retinal Microvasculature Causally Affects the Brain Cortical Structure: A Mendelian Randomization Study." In: *Ophthalmology Science* 4.6 (2024), p. 100465.
- [196] J. WEICKERT. "Efficient image segmentation using partial differential equations and morphology." In: *Pattern Recognition* 34.9 (2001), pp. 1813–1824.
- [197] D. L. WEILER, C. B. ENGELKE, A. L. O. MOORE, and W. W. HARRISON. "Arteriole tortuosity associated with diabetic retinopathy and cholesterol." In: *OVS* 92.3 (2015), pp. 384–391.
- [198] R. A. WELIKALA, P. J. FOSTER, P. H. WHINCUP, A. R. RUDNICKA, C. G. OWEN, D. P. STRACHAN, and S. A. BARMAN. "Automated arteriole and venule classification using deep learning for retinal images from the UK Biobank cohort." In: *Computers in Biology and Medicine* 90 (2017), pp. 23–32.
- [199] C. S. WON. "A block-based MAP segmentation for image compressions." In: *IEEE Transactions on Circuits and Systems for Video Technology* 8.5 (1998), pp. 592–601.
- [200] Y. WU, X. PENG, K. RUAN, and Z. HU. "Improved image segmentation method based on morphological reconstruction." In: *Multimedia Tools and Applications* 76.19 (2017), pp. 19781–19793.
- [201] L. XIONG, H. LI, and L. XU. "An Approach to Evaluate Blurriness in Retinal Images with Vitreous Opacity for Cataract Diagnosis." In: *Journ. of Healthcare Eng.* (2017).

- [202] X. XU, W. DING, M. D. ABRÀMOFF, and R. CAO. "An improved arteriovenous classification method for the early diagnostics of various diseases in retinal image." In: *Computer Methods and Programs in Biomedicine* 141 (2017), pp. 3–9.
- [203] X. XU, T. TAN, and F. XU. "An Improved U-Net Architecture for Simultaneous Arteriole and Venule Segmentation in Fundus Image." In: *Medical Image Understanding and Analysis*. Ed. by M. NIXON, S. MAHMOODI, and R. ZWIGGELAAR. Cham: Springer International Publishing, 2018, pp. 333–340.
- [204] J. YANG, X. DONG, Y. HU, Q. PENG, G. TAO, Y. OU, H. CAI, and X. YANG. "Fully Automatic Arteriovenous Segmentation in Retinal Images via Topology-Aware Generative Adversarial Networks." In: *Interdisciplinary Sciences: Computational Life Sciences* 12.3 (2020), pp. 323–334.
- [205] J. ZHANG, B. DASHTBOZORG, E. J. BEKKERS, J. P. W. PLUIM, R. DUIT, and B. M. TER HAAR ROMENY. "Robust Retinal Vessel Segmentation via Locally Adaptive Derivative Frames in Orientation Scores." In: *IEEE Transactions on medical imaging* 35.12 (2016), pp. 2631–2644.
- [206] S. ZHANG, T. T. J. M. BERENDSCHOT, and C. A. B. WEBERS. "Luminosity Rectified Blind Richardson-Lucy Deconvolution for Single Retinal Image Restoration." In: *SSRN* (2023).
- [207] S. ZHANG, C. A. B. WEBERS, and T. T. J. M. BERENDSCHOT. "A double-pass fundus reflection model for efficient single retinal image enhancement." In: *Signal Processing* 192 (2022).
- [208] M. ZHOU, K. JIN, S. WANG, J. YE, and D. QIAN. "Color Retinal Image Enhancement Based on Luminosity and Contrast Adjustment." In: *IEEE TBE* 65.3 (2018), pp. 521–527.
- [209] A. ZOMORODIAN and G. CARLSSON. "Computing Persistent Homology." In: *Discrete & Computational Geometry* 33 (2 Feb. 2005), pp. 249–274.

Part V
APPENDIX

A

PROOFS OF VARIOUS THEOREM'S AND LEMMA'S

A.1 PROOF OF LEMMA 4.3.1

Lemma 4.3.1. *When expressing Eqs. (4.15) and (4.16) more explicitly in data-driven left-invariant frame components (gauge frame components for short), one finds*

$$(\nabla^U)_X Y = \sum_{k=1}^n \left(\dot{\tilde{y}}^k + \sum_{i,j=1}^n \tilde{c}_{ij}^k(\cdot) \tilde{x}^i \tilde{y}^j \right) \mathcal{A}_k^U, \quad (4.18)$$

and for the dual connection

$$(\nabla^U)_X^* \lambda = \sum_{i=1}^n \left(\dot{\tilde{\lambda}}_i + \sum_{k,j=1}^n (\tilde{x}^j \tilde{\lambda}_k) \tilde{c}_{ij}^k(\cdot) \right) \omega_U^i, \quad (4.19)$$

where $X = \sum_{i=1}^n \tilde{x}^i \mathcal{A}_i^U|_\gamma$, $Y = \sum_{i=1}^n \tilde{y}^i \mathcal{A}_i^U|_\gamma$ and $\lambda = \sum_{i=1}^n \tilde{\lambda}_i \omega_U^i$, and where derivations of the components of Y and λ equal

$$\begin{aligned} \dot{\tilde{y}}^k(t) &:= \frac{d}{dt} \tilde{y}^k(\gamma(t)) = (X(\dot{\tilde{y}}^k))(\gamma(t)), \\ \dot{\tilde{\lambda}}_i(t) &:= \frac{d}{dt} \tilde{\lambda}_i(\gamma(t)) = (X(\dot{\tilde{\lambda}}_i))(\gamma(t)), \end{aligned}$$

along a flow-line¹ $\gamma : [0, 1] \rightarrow \mathbb{M}_2$ of smooth vector field X .

Proof. Expanding Eq. (4.15), we obtain

$$\begin{aligned} (\nabla^U)_X Y &= \sum_{k=1}^n \left(\sum_{i=1}^n \omega_U^i(X) \mathcal{A}_i^U \omega_U^k(Y) \right. \\ &\quad \left. + \sum_{i,j=1}^n \omega_U^i(X) \omega_U^j(Y) \tilde{c}_{ij}^k(\cdot) \right) \mathcal{A}_k^U \\ &= \sum_{k=1}^n \left(\sum_{i=1}^n \tilde{x}^i \mathcal{A}_i^U \tilde{y}^k + \sum_{i,j=1}^n \tilde{x}^i \tilde{y}^j \tilde{c}_{ij}^k(\cdot) \right) \mathcal{A}_k^U \\ &= \sum_{k=1}^n \left(\dot{\tilde{y}}^k + \sum_{i,j=1}^n \tilde{c}_{ij}^k(\cdot) \tilde{x}^i \tilde{y}^j \right) \mathcal{A}_k^U, \end{aligned}$$

¹ A curve γ satisfying $\dot{\gamma}(t) = X_{\gamma(t)}$.

where the last equality holds since

$$\begin{aligned}\dot{\tilde{y}}^k(t) &= \frac{d}{dt}\tilde{y}^k(\gamma(t)) = \sum_{i=1}^n \dot{\tilde{\gamma}}^i (\mathcal{A}_i^U \tilde{y}^k)(\gamma(t)) \\ &= \sum_{i=1}^n \tilde{x}^i (\mathcal{A}_i^U \tilde{y}^k)(\gamma(t)) = X(\tilde{y}^k)(\gamma(t)).\end{aligned}$$

Similarly, we have for Eq. (4.16)

$$\begin{aligned}(\nabla^U)_X^* \lambda &= \sum_{i=1}^n \left(\sum_{j=1}^n \omega_U^j(X) \mathcal{A}_j^U \mathcal{A}_i^U(\lambda) \right. \\ &\quad \left. + \sum_{j,k=1}^n \omega_U^j(X) \mathcal{A}_k^U(\lambda) \tilde{c}_{ij}^k(\cdot) \right) \omega_U^i \\ &= \sum_{i=1}^n \left(\sum_{j=1}^n \tilde{x}^j \mathcal{A}_j^U \tilde{\lambda}_i + \sum_{j,k=1}^n \tilde{x}^j \tilde{\lambda}_k \tilde{c}_{ij}^k(\cdot) \right) \omega_U^i \\ &= \sum_{i=1}^n \left(\dot{\tilde{\lambda}}_i + \sum_{j,k=1}^n \tilde{c}_{ij}^k \tilde{x}^j \tilde{\lambda}_k \right) \omega_U^i,\end{aligned}$$

where the last equality holds since

$$\begin{aligned}\dot{\tilde{\lambda}}_i(t) &= \frac{d}{dt}\tilde{\lambda}_i(\gamma(t)) = \sum_{j=1}^n \dot{\tilde{\gamma}}^j (\mathcal{A}_j^U \tilde{\lambda}_i)(\gamma(t)) \\ &= \sum_{j=1}^n \tilde{x}^j (\mathcal{A}_j^U \tilde{\lambda}_i)(\gamma(t)) = X(\tilde{\lambda}_i)(\gamma(t)).\end{aligned}$$

□

A.2 PROOF OF THEOREM 4.3.1

Theorem 4.3.1 (Straight and shortest curves: parallel velocity and momentum w.r.t. connection ∇^U). *In a Riemannian manifold (G, \mathcal{G}^U) with data-driven left-invariant metric tensor field \mathcal{G}^U satisfying Eq. (4.3), and with induced Riemannian metric $d_{\mathcal{G}^U}$ Eq. (4.9), we have:*

- γ is a straight curve with respect to $\nabla^U \stackrel{\text{def.}}{\Leftrightarrow} \nabla_{\dot{\gamma}}^U \dot{\gamma} = 0$

$$\Leftrightarrow \exists (c^1, \dots, c^n) \in \mathbb{R}^n \text{ constant s.t. } \dot{\gamma} = \sum_{i=1}^n c^i \mathcal{A}_i^U|_{\gamma}.$$

- γ is a shortest curve with respect to $\nabla^U \Rightarrow$ the curve-momentum pair $\nu = (\gamma, \lambda) : [0, 1] \rightarrow T^*(G)$ satisfies the Hamiltonian flow

$$\dot{\nu} = \vec{h}(\nu) \Leftrightarrow \begin{cases} (\nabla^U)_{\dot{\gamma}}^* \lambda = 0 \\ \mathcal{G}^U \dot{\gamma} = \lambda, \end{cases} \quad (4.22)$$

where one has the following pull-back symmetry² of the data-driven Cartan connection

$$(L_q)^* (\nabla^{\mathcal{L}_q U})^* = (\nabla^U)^* \text{ for all } q \in G, \quad (4.23)$$

with left actions L and \mathcal{L} given by Eqs. (1.2) and (1.18) respectively.

The shortest curve $\gamma : [0, 1] \rightarrow G$ with $\gamma(0) = g$ and $\gamma(1) = g_0$ may be computed by steepest descent backtracking on the distance map $W(g) = d_{\mathcal{G}^U}(g, g_0)$

$$\gamma(t) := \gamma_{g, g_0}^U(t) = \text{Exp}_g(t v(W)) \quad t \in [0, 1], \quad (4.24)$$

where Exp integrates the following vector field on G :

$v(W) := -W(g)(\mathcal{G}^U)^{-1} dW = -W(g) \sum_{k=1}^n |\alpha_k|^{-1} \mathcal{A}_k^U(W) \mathcal{A}_k^U$ and where W is the viscosity solution of the eikonal PDE system

$$\begin{cases} \| \text{grad}_{\mathcal{G}^U} W \| = 1, \\ W(g_0) = 0, \end{cases} \quad (4.25)$$

where we assume g is neither a 1st Maxwell point nor a conjugate point. As $v(W)$ is data-driven left-invariant, the geodesics carry the symmetry

$$\gamma_{qg, qg_0}^U(t) = q \gamma_{g, g_0}^U(t) \text{ for all } q, g, g_0 \in G, t \in [0, 1]. \quad (4.26)$$

Proof. Firstly, we address the characterisation of straight curves.

$$(\nabla^U)_{\dot{\gamma}} \dot{\gamma} = 0 \Leftrightarrow \ddot{\gamma}^k + \sum_{i,j=1}^n c_{ij}^k(\gamma(t)) \dot{\gamma}^i \dot{\gamma}^j = 0 \text{ for } k = 1, \dots, n.$$

Since $[\mathcal{A}_i^U, \mathcal{A}_j^U] = -[\mathcal{A}_j^U, \mathcal{A}_i^U]$ due to Eqs. (4.12) and (4.13), we have $\tilde{c}_{ij}^k = -\tilde{c}_{ji}^k$ and $\tilde{c}_{ii}^k = 0$.

Consequently, we are left with $\ddot{\gamma}^k = 0 \Rightarrow \dot{\gamma}^k - c^k = 0$. Summarizing we have

$$(\nabla^U)_{\dot{\gamma}} \dot{\gamma} = 0 \Leftrightarrow \dot{\gamma} = \sum_{k=1}^n \dot{\gamma}^k \mathcal{A}_k^U = \sum_{k=1}^n c^k \mathcal{A}_k^U.$$

² For the definition of pullback of a dual connection, see Remark A.2.4 in Appendix A.2.

Secondly, we address the characterisation of shortest curves.

The Pontryagin Maximum Principle is given by (Hamiltonian flow on co-tangent bundle $T^*(G)$)

$$\begin{cases} \dot{\nu} = \vec{h}(\nu) \\ \nu(0) = (\gamma(0), \lambda(0)), \end{cases} \quad (\text{A.1})$$

where $\nu(t) = (\gamma(t), \lambda(t)) \in T^*(G), \lambda(t) \in T_{\gamma(t)}^*(G)$.

For details see [7].

Remark A.2.1 (Liouville's theorem). By the chain rule, for any smooth function $f : T^*(G) \rightarrow \mathbb{R}$, one has

$$\begin{aligned} \frac{d}{dt} f(\gamma(t), \lambda(t)) &= \nabla_\gamma f(v(t)) \cdot \nabla_\lambda h(v(t)) \\ &\quad - \nabla_\lambda f(v(t)) \cdot \nabla_\gamma h(v(t)) \\ &= \{h, f\}|_{\gamma(t)}, \end{aligned} \quad (\text{A.2})$$

where h denotes the Hamiltonian flow, and where ∇_γ and ∇_λ denote the gradient with respect to γ and λ respectively.

Eq. (A.1) can be expressed in coordinates as follows

$$\begin{cases} \gamma(t) = (x(t), y(t), \theta(t)) \\ \lambda(t) = \sum_{i=1}^n \tilde{\lambda}_i \omega_U^i|_{\gamma(t)} \\ \dot{\gamma}(t) = \sum_{i=1}^n \dot{\gamma}(t) \mathcal{A}_i^U(\gamma(t)). \end{cases}$$

The horizontal part is given by

$$\dot{\gamma}^i = \langle \omega_U^i, \dot{\gamma} \rangle = \tilde{\lambda}^i \quad i = 1, \dots, n. \quad (\text{A.3})$$

This follows from the computation of the Hamiltonian via the Fenchel transform.

The vertical part is given by

$$\begin{aligned} \dot{\tilde{\lambda}}_i &\stackrel{(\text{A.2})}{=} \{h, \tilde{\lambda}_i\} = \left\{ \frac{1}{2} \sum_{j=1}^n \tilde{\lambda}_j \tilde{\lambda}^j, \tilde{\lambda}_i \right\} = \left\{ \frac{1}{2} \sum_{j=1}^n \alpha^j(\cdot) \tilde{\lambda}_j^2, \tilde{\lambda}_i \right\} \\ &= \sum_{j=1}^n \tilde{\lambda}_j \alpha^j(\cdot) \{ \tilde{\lambda}_j, \tilde{\lambda}_i \} + \frac{1}{2} \sum_{j=1}^n \tilde{\lambda}_j^2 \{ \alpha^j(\cdot), \tilde{\lambda}_i \} \\ &= \sum_{j=1}^n \sum_{k=1}^n \tilde{c}_{ji}^k \alpha^j(\cdot) \tilde{\lambda}_j \tilde{\lambda}_k = \sum_{j,k=1}^n \tilde{c}_{ji}^k \tilde{\lambda}_k \tilde{\lambda}^j. \end{aligned}$$

In the above derivation, we have used that

$$\{\tilde{\lambda}_i, \tilde{\lambda}_j\} = \sum_{k=1}^n \tilde{c}_{ij}^k(\cdot) \tilde{\lambda}_k.$$

Additionally, it is important to note that

$$\begin{aligned} \{\alpha^j(\cdot), \tilde{\lambda}_i\} &= \sum_{k=1}^n \mathcal{A}_k^U \alpha^j(\cdot) \frac{\partial \tilde{\lambda}_i}{\partial \tilde{\lambda}_k} - \frac{\partial \alpha^j(\cdot)}{\partial \lambda_k} \mathcal{A}_k^U \tilde{\lambda}_i \\ &= \mathcal{A}_i^U \alpha^j(\cdot), \end{aligned} \quad (\text{A.4})$$

since $\mathcal{A}_k^U \tilde{\lambda}_i = 0$. Consequently, we find

$$\begin{aligned} \sum_{j=1}^n \frac{1}{2} \tilde{\lambda}_j^2 \{\alpha^j(\cdot), \tilde{\lambda}_i\} &\stackrel{(\text{A.4})}{=} \sum_{j=1}^n \frac{1}{2} \tilde{\lambda}_j^2 \mathcal{A}_i^U \alpha^j(\cdot) \\ &\stackrel{(4.11)}{=} \mathcal{A}_i^U \mathfrak{h}(\lambda) = 0, \end{aligned}$$

since the Hamiltonian is constant. So in total we have

$$\begin{cases} \dot{\tilde{\gamma}}^i(t) = \tilde{\lambda}_i^i(t) \\ \dot{\tilde{\lambda}}_i(t) = \sum_{j=1}^n \sum_{k=1}^n \tilde{c}_{ji}^k(\gamma(t)) \tilde{\lambda}_k(t) \tilde{\lambda}_j^i(t). \end{cases}$$

This is equivalent to

$$\begin{aligned} &\begin{cases} \dot{\tilde{\gamma}}^i(t) = \left((\mathcal{G}_{\gamma(t)}^U)^{-1} \tilde{\lambda}(t) \right)^i \\ \dot{\tilde{\lambda}}_i(t) = \sum_{j,k=1}^n \tilde{c}_{ji}^k(\gamma(t)) \tilde{\lambda}_k(t) \tilde{\lambda}_j^i(t) = 0 \end{cases} \\ &\Leftrightarrow \begin{cases} \dot{\gamma}(t) = (\mathcal{G}_{\gamma(t)}^U)^{-1} \lambda(t) \\ (\nabla^U)_{\dot{\gamma}}^* \lambda = 0 \end{cases} \\ &\Leftrightarrow \begin{cases} \lambda(t) = (\mathcal{G}_{\gamma(t)}^U) \dot{\gamma}(t) \\ (\nabla^U)_{\dot{\gamma}}^* \lambda = 0. \end{cases} \end{aligned}$$

Remark A.2.2 (Justification of Eq. (A.3)). We have Lagrangian

$$\mathcal{L}(\gamma, \dot{\gamma}) = \frac{1}{2} \sum_{i=1}^n \alpha_i(\cdot) \left(\dot{\tilde{\gamma}}^i \right)^2.$$

From this expression, we derive the Hamiltonian \mathfrak{h} :

$$\begin{aligned}\mathfrak{h}(\gamma, \lambda) &= \sup_{\dot{\gamma} \in T_\gamma(G)} \{ \langle \lambda, \dot{\gamma} \rangle - \mathcal{L}(\gamma, \dot{\gamma}) \} \\ &= \max_{(v^1, \dots, v^n)} \left\{ \sum_{i=1}^n \tilde{\lambda}_i v^i - \frac{1}{2} \alpha_i(\cdot) (v^i)^2 \right\}.\end{aligned}$$

To find the maximum, we differentiate with respect to v :

$$\begin{pmatrix} \tilde{\lambda}_1 - \alpha_1(\cdot) v_{\max}^1 \\ \vdots \\ \tilde{\lambda}_n - \alpha_n(\cdot) v_{\max}^n \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \Leftrightarrow \begin{pmatrix} \tilde{\lambda}_1 \\ \vdots \\ \tilde{\lambda}_n \end{pmatrix} = \begin{pmatrix} v_1^{\max} \\ \vdots \\ v_n^{\max} \end{pmatrix},$$

where

$$\tilde{\lambda}_i = \alpha_i(\cdot) \tilde{\lambda}^i \text{ and } v_i^{\max} = \alpha_i(\cdot) v_{\max}^i.$$

Consequently, we find $\tilde{\lambda}^i = v_{\max}^i$ for $i = 1, \dots, n$, and the Hamiltonian is given by

$$\begin{aligned}\mathfrak{h} &= \sum_{i=1}^n \tilde{\lambda}_i \tilde{\lambda}^i - \frac{1}{2} \alpha_i(\cdot) (\tilde{\lambda}^i)^2 \\ &= \sum_{i=1}^n \left(\tilde{\lambda}_i \tilde{\lambda}^i - \frac{1}{2} \tilde{\lambda}_i \tilde{\lambda}^i \right) = \frac{1}{2} \sum_{i=1}^n \tilde{\lambda}_i \tilde{\lambda}^i.\end{aligned}$$

Hence, we find that $\tilde{\lambda}_i = \dot{\gamma}_i$ and $\tilde{\lambda}^i = \dot{\gamma}^i$, which we aimed to show. Note that reformulation in a coordinate-free manner yields

$$\forall_{i \in \{1, \dots, n\}} \dot{\gamma}^i = \tilde{\lambda}^i \Leftrightarrow \lambda|_{\gamma(t)} = (\mathcal{G}_{\gamma(t)}^U)^{-1} \dot{\gamma}(t),$$

for all $t \in [0, W(g)]$.

Remark A.2.3. In Theorem 4.3.1 we give a backtracking formulation (where geodesics descend to the origin via steepest descent) where we rescaled time back to the interval $t \in [0, 1]$ (as this is more practical). Similar to [83, Thm.4, Eq.28] this is done as follows: in the ODE backtracking for geodesics, cf. Eq. (4.24), we included an extra negative scaling factor $-W(g)$ in comparison to all the canonical ODEs above.

Next, we address the symmetry, cf. Eq. (4.23), of the data-driven Cartan connection.

By construction of Eqs. (4.4) and (4.15), we have the correct symmetry in Eq. (4.23). Indeed, from Eq. (4.4), it follows that

$$\mathcal{A}_i^{\mathcal{L}_q U} \Big|_{g\mathbf{p}} = (L_g)_* \mathcal{A}_i^U \Big|_{\mathbf{p}} \text{ and } \omega_{\mathcal{L}_q U}^i \Big|_{g\mathbf{p}} = (L_g)^* \omega_U^i \Big|_{\mathbf{p}}$$

and via Eq. (4.15) we get $(L_g)^* \left(\nabla^{\mathcal{L}_g U} \right)^* = (\nabla^U)^*$, where we use the fact that \mathcal{G}^U is diagonal with respect to the basis $\{\mathcal{A}_i^U\}$ and where we respectively applied the pushforward of a vector field, the pullback of a co-vector field, and the pullback of a dual connection.

Remark A.2.4. In general the pullback $\Phi^*\nabla^*$ of a dual connection ∇^* on manifold G under a smooth mapping $\Phi : G \rightarrow G$ is given by $(\Phi^*\nabla_X^*)(\Phi^*\lambda) = \Phi^*(\nabla_{\Phi_* X}^* \lambda)$ with $\lambda \in T^*(G)$ and $X \in T(G)$.

Finally, we address the integration of geodesics and their symmetry.

Eq. (4.24)) follows by Eq. (4.22). Here $\lambda = dW$ implies we must indeed set the following momentum components:

$$\tilde{\lambda}^k = \mathcal{A}_k^U W(\gamma(\cdot)), \text{ for all } k = 1, \dots, n.$$

Furthermore in Eq. (4.22) we invert the data-driven left-invariant metric tensor field \mathcal{G}^U (recall Eq. (4.7)) and express the velocities as $\dot{\gamma}^k = g_{kk}^{-1} \lambda_k = g^{kk} \lambda_k$. Then via Remark 4.3.6 we obtain that the geodesics indeed follow by integration of the vector field $v(W)$ on G . Clearly, this vector field is data-driven left-invariant (as all the vector fields \mathcal{A}_i^U are) and thereby one has:

$$\frac{d}{dt} \left(\gamma_{qg, qg_0}^{\mathcal{L}_g U} \right) (t) = (L_q)_* \frac{d}{dt} \gamma_{g, g_0}^U (t),$$

for all $q, g, g_0 \in G, t \in [0, 1]$, from which the symmetry, cf. Eq. (4.26), follows by integration. \square

A.3 PROOF OF LEMMA 4.4.2

Lemma 4.4.2 (Dual Finsler functions). *With our choice (4.29) of Finsler function \mathcal{F}_ϵ used in Eq. (4.31), the dual Finsler function \mathcal{F}_ϵ^* is given for all $\hat{p} \in T_p^*(\mathbb{M}_2) \equiv \mathbb{R}^3$ by*

$$\mathcal{F}_\epsilon^*(p, \hat{p})^2 = \langle \hat{p}, D_\epsilon \hat{p} \rangle + \langle \hat{p}, \eta_\epsilon \rangle_+^2, \text{ with} \quad (4.34)$$

$$D_\epsilon = \frac{\mathcal{A} \mathcal{A}^\top}{\mathcal{A}^\top M^0 \mathcal{A}} + O(\epsilon^2), \quad (4.35)$$

$$\eta_\epsilon = \frac{M^{-1}(\omega_U^1 - \alpha \omega_U^2)}{\sqrt{(\omega_U^1)^\top M^{-1}(\omega_U^1 - \alpha \omega_U^2)}} + O(\epsilon^2), \quad (4.36)$$

where we used shorthand notation $M^{-1} := (M^0)^{-1}$, the cross product $\mathcal{A} := \omega_U^1 \times \omega_U^2$, and the orthogonalization coefficient $\alpha := (\omega_U^2)^\top M^{-1} \omega_U^1 / (\omega_U^2)^\top M^{-1} \omega_U^2$.

Proof. For brevity in this proof, we write $\omega^i := \omega_U^i$ for $i = 1, 2, 3$.

Let P be a $3 \times k$ matrix of rank k ($1 \leq k \leq 3$). Then, by the Woodbury formula, one has

$$\begin{aligned} (\text{Id}_3 + \epsilon^{-2}PP^\top)^{-1} &= \text{Id}_3 - \epsilon^{-2}P(\text{Id}_k + \epsilon^{-2}P^\top P)^{-1}P^\top \\ &= \text{Id}_3 - P(P^\top P)^{-1}P^\top + O(\epsilon^2), \end{aligned}$$

which is up to $O(\epsilon^2)$ error equal to the orthogonal projection $\text{Id}_3 - P(P^\top P)P^\top$ onto $\text{Span}(P)^\perp$. We calculate D_ϵ using the Woodbury formula and a Taylor expansion:

$$D_\epsilon := (M^0 + \epsilon\tilde{P}\tilde{P}^\top)^{-1} = (M^0)^{-\frac{1}{2}} (\text{Id}_3 + \epsilon^{-2}PP^\top)^{-1} (M^0)^{-\frac{1}{2}},$$

where $P := (M^0)^{-1/2}\tilde{P}$ and $\tilde{P} = [\omega^1 \quad \omega^2]$. Note that $\text{Span}\{P\}^\perp = \text{Span}\{\tilde{\omega}\}$, with $\tilde{\omega} := (\sqrt{M^0}^{-1}\omega^1) \times (\sqrt{M^0}^{-1}\omega^2)$. Then, one has

$$D_\epsilon = (M^0)^{-1/2} \frac{\tilde{\omega}\tilde{\omega}^\top}{\tilde{\omega}^\top M^0 \tilde{\omega}} (M^0)^{-1/2} + O(\epsilon^2).$$

Simplification yields $D = \frac{\omega\omega^\top}{\omega^\top M^0 \omega} + O(\epsilon^2)$, with $\omega := \omega^1 \times \omega^2$ as stated in Eq. (4.35).

Likewise, via Lemma 4.4.1, the Woodbury formula, and applying a Taylor expansion, we obtain that

$$\begin{aligned} M_\epsilon^{-1} &= (M^0 + \epsilon^{-2}\omega^2(\omega^2)^\top)^{-1} \\ &= (M^0)^{-1/2}(\text{Id}_3 - \tilde{\omega}^2((\tilde{\omega}^2)^\top \tilde{\omega}^2)^{-1}(\tilde{\omega}^2)^\top)(M^0)^{-1/2} \\ &\quad + O(\epsilon^2), \end{aligned}$$

where $\tilde{\omega}^2 := (M^0)^{-1/2}\omega^2$ and

$$(\text{Id}_3 - (M^0)^{-1/2}\omega^2((\omega^2)^\top(M^0)^{-1}\omega^2)^{-1}(\omega^2)^\top(M^0)^{-1/2})$$

is up to $O(\epsilon^2)$ error the orthogonal projection onto $\text{Span}\{(M^0)^{-1/2}\omega^2\}^\perp$. The further simplification of

$$\eta_\epsilon = \frac{\epsilon^{-1}M_\epsilon^{-1}\omega^1}{\sqrt{1 + \epsilon^{-2}(\omega^1)^\top M_\epsilon^{-1}\omega^1}}$$

boils down to Eq. (4.36). □

A.4 MULTIPLE CONSECUTIVE MORPHOLOGICAL DILATIONS

We analyze the effect of multiple consecutive morphological dilations in Lemma A.4.1. In the statement and proof, we use the morphological delta which we define first.

Definition A.4.1 (Morphological delta). *We define the morphological delta $\delta_e^M : G \rightarrow \mathbb{R} \cup \{\infty\}$ as*

$$\delta_e^M(g) = \begin{cases} 0 & \text{if } g = e \\ \infty & \text{else.} \end{cases} \quad (\text{A.5})$$

Lemma A.4.1. *Let $1 \leq \alpha \leq \infty$. Consider the morphological kernel k_t^α . Let $g \in G$ be arbitrary. When defining³ $k_0^\alpha = \delta_e^M$ for $\alpha < \infty$, and $k_0^\infty = d(\cdot, e)$, we have:*

$$\forall t, s \geq 0 : (k_t^\alpha \square k_s^\alpha)(g) = k_{t+s}^\alpha(g) = (\kappa_t^\alpha \square_{\mathbb{R}} \kappa_s^\alpha)(d(g, e)),$$

where $\kappa_t^\alpha : \mathbb{R}^+ \rightarrow \mathbb{R}^+$, is defined as $\kappa_t^\alpha(d(g, e)) = k_t^\alpha(g)$.

Proof. For $t = 0, \alpha < \infty$, one has $k_0^\alpha = \delta_e^M$, and $\delta_e^M \square f = f$, so let us examine $1 \leq \alpha < \infty, t, s > 0$. Let $\kappa_t^\alpha(r) := \kappa_t^\alpha(|r|)$ for all $r \in \mathbb{R}$. Then, one can write

$$(k_t^\alpha \square k_s^\alpha)(g) = \inf_{h \in G} k_t^\alpha(h^{-1}g) + k_s^\alpha(h) \quad (\text{A.6a})$$

$$= \inf_{h \in G} \kappa_t^\alpha(d(h^{-1}g, e)) + \kappa_s^\alpha(d(h, e)) \quad (\text{A.6b})$$

$$= \inf_{h \in \gamma_{e,g}^{\min}} \kappa_t^\alpha(d(g, e) - d(h, e)) + \kappa_s^\alpha(d(h, e)) \quad (\text{A.6c})$$

$$= \inf_{0 \leq r \leq d(g,e)} \kappa_t^\alpha(d(g, e) - r) + \kappa_s^\alpha(r) \quad (\text{A.6d})$$

$$= \inf_{r \in \mathbb{R}} \kappa_t^\alpha(d(g, e) - r) + \kappa_s^\alpha(r) \quad (\text{A.6e})$$

$$= (\kappa_t^\alpha \square_{\mathbb{R}} \kappa_s^\alpha)(d(g, e)) \quad (\text{A.6f})$$

$$= \kappa_{t+s}^\alpha(d(g, e)) = k_{t+s}^\alpha(g), \quad (\text{A.6g})$$

where the third equality (A.6c) holds because one has \geq due to the triangle inequality and κ_t^α being monotonous. The \leq holds because $\gamma_{e,g}^{\min} \subset G$, where $\gamma_{e,g}^{\min}$ denotes the minimizing geodesic connecting e and g . To justify the fifth equality (A.6e), we define $G(r) := \kappa_t^\alpha(d(g, e) - r) + \kappa_s^\alpha(r) = \frac{t}{\beta} \left(\frac{d-r}{t} \right)^\beta + \frac{s}{\beta} \left(\frac{r}{s} \right)^\beta$, where we omit the dependence on the elements g and e in the notation for the distance $d = d(g, e)$. Then differentiating G with respect to r yields:

$$G'(r) = - \left(\frac{d-r}{t} \right)^{\beta-1} + \left(\frac{r}{s} \right)^{\beta-1} = 0 \Leftrightarrow r = \frac{s}{t+s}d.$$

³ These pointwise limits arise from Eq. (8.12).

Since $s, t > 0$, r attains a global minimum between 0 and distance $d = d(g, e)$. The sixth equality (A.6f) is by definition of the morphological convolutions on \mathbb{R} . The seventh equality (A.6g) follows by well-known results for morphological PDEs on \mathbb{R} [176].

Note that the case $\alpha = \infty, t = 0, s > 0$ is the same as $\alpha = \infty, t, s > 0$ since $k_0^\infty(g) = k_t^\infty(g)$ for all $t \geq 0$, and where

$$(k_t^\infty \square k_s^\infty)(g) = \inf_{h \in G} d(h^{-1}g, e) + d(h, e) = d(g, e).$$

□

A.5 PROOF OF PROPOSITION 8.4.2

Proposition 8.4.2. Let $\alpha = 1, t = \delta > 0, g \in G, n \in \mathbb{N}$. Let $g_0 \in G$ be the reference point. Set $\delta = d_{1,t}^1$. Then, we have, for $U_{g_0}(g, n)$ given by Eq. (8.21), that $0 \leq U_{g_0}(g, n) \leq 1$ where

$$U_{g_0}(g, n) = \begin{cases} 1 & \text{if } g \not\sim g_0 \wedge m_\delta(g, g_0) \leq n \\ 0 & \text{else.} \end{cases},$$

where $m_\delta(g, g_0)$ was defined in Definition 8.2.4 and Remark 8.2.1.

Proof. First of all, note that $m_\delta(g, g_0) \leq n$ implies that $d(g, g_0) \leq n\delta$ by applying the triangular inequality to the sequence of intermediate points in Definition 8.2.4.

Then, note that the inequalities follow directly from Lemma 8.3.1, Eq. (8.15). We will prove the rest of this statement by induction with respect to n . Without loss of generality in view of left invariance, we prove the statement for $g_0 = e$. This is analogous to applying a roto-translation with g_0^{-1} of the given binary map $\mathbb{1}_I$, and afterward pushing forward the output $U_e(n, \cdot)$ to $U_{g_0}(n, \cdot)$. First, we show that the statement is true for the first step of the algorithm. We recall Eq. (8.20), which stated

$$k_t^1(g) := \begin{cases} 0 & \text{if } d(g, e) < t \\ \infty & \text{else.} \end{cases}$$

We can easily express the first step of the connected component algorithm $n = 1$ by calculating $U_e(1, \cdot)$:

$$\begin{aligned} U_e(1, g) &= (- (k_\delta^1 \square - U_e(0, \cdot)) (g)) \mathbb{1}_I(g) \\ &= \begin{cases} \sup_{h \in G} \{U_e(0, h) - k_\delta^1(h^{-1}g)\} & \text{if } g \in I, \\ 0 & \text{else;} \end{cases} \\ &= \begin{cases} 1 & \text{if } g \in I \text{ and } d(e, g) < \delta \\ 0 & \text{else,} \end{cases} \end{aligned}$$

where $g \in I$ and $d(e, g) < \delta$ is equivalent to $g \not\sim e$ and $m_\delta(e, g) \leq 1$. Note that $g \not\sim e$ automatically implies $g \in I$.

Next, we assume that the statement is true for a given $n \in \mathbb{N}$, meaning that one has

$$\begin{aligned} U_e(n, g) &= \begin{cases} 1 & \text{if } g \not\sim e \wedge m_\delta(e, g) \leq n \\ 0 & \text{else.} \end{cases} \\ &= \mathbb{1}_{I_{n,\delta}^e}(g), \end{aligned}$$

where

$$I_{n,\delta}^e = [e] \cap \tilde{B}(e; n),$$

where $[e]$ denotes the set of points that are δ -connected to e and where $\tilde{B}(e; n)$ is defined by

$$\tilde{B}(e; n) := \{g \in I \mid m_\delta(e, g) \leq n\}. \quad (\text{A.7})$$

We now aim to show that the statement is also true for iteration $n + 1$. That means that one investigates whether one can write $U_e(n + 1, \cdot)$ in the same form as $U_e(n, \cdot)$. Let $g \in I$, then

$$\begin{aligned} U_e(n + 1, g) &= - (- (k_\delta^1 \square - U_e(n, \cdot)) (g)) \\ &= - (- (k_\delta^1 \square - \mathbb{1}_{I_{n,\delta}^e})(g)) \\ &= \sup_{h \in G} \left\{ \mathbb{1}_{I_{n,\delta}^e}(h) - k_\delta^1(h^{-1}g) \right\}. \end{aligned} \quad (\text{A.8})$$

Then, in order to reach the supremum in Eq. (A.8), we can simplify the expression to

$$U_e(n + 1, g) = \begin{cases} 1 & \text{if } g \in I_{n,\delta}^e \text{ or if } \exists h \in I_{n,\delta}^e \text{ s.t. } d(h, g) < \delta \\ 0 & \text{else.} \end{cases}$$

We will reformulate the condition for $U_e(n+1, g)$ to equal 1. First, it is important to note that $I_{n,\delta}^e \subseteq I_{n+1,\delta}^e$. Let us assume $g \notin I_{n,\delta}^e$, but there exists a $h \in I_{n,\delta}^e$ such that $d(h, g) < \delta$. Then, the first condition gives us

$$m_\delta(e, g) > n. \quad (\text{A.9})$$

Simultaneously, there exists a $h \in I_{n,\delta}^e$ such that $d(h, g) < \delta$, so

$$m_\delta(e, h) \leq n \quad \text{and} \quad d(h, g) < \delta, \quad \text{so} \quad m_\delta(e, g) \leq n + 1. \quad (\text{A.10})$$

Combining Eqs. (A.9) and (A.10) tells us that $m_\delta(e, g) = n + 1$. Additionally, the value of $U_e(n + 1, g)$ is equal to 1 when $g \in I_{n,\delta}^e$, i.e., $m_\delta(e, g) \leq n$. Hence, $U_e(n + 1, g)$ can be reformulated to

$$U_e(n + 1, g) = \begin{cases} 1 & \text{if } g \in I_{n+1,\delta}^e, \text{ i.e., } m_\delta(e, g) \leq n + 1 \text{ and } g \not\sim e \\ 0 & \text{else.} \end{cases}$$

This is the same expression as was assumed to be true for some $n \in \mathbb{N}$ but now for $n + 1 \in \mathbb{N}$, and thus, the statement is proven.

□

THE USED METRIC TENSOR FIELD IS INDEED A DATA-DRIVEN LEFT-INVARIANT METRIC TENSOR FIELD

B

We first rely on a convenient standard formula of the Hessian of smooth function on a manifold relative to a connection on that manifold in Lemma B.1. Then we provide an alternative formulation of such a Hessian in Lemma B.2 (via the notion of parallel transport).

Finally, we prove that \mathcal{G}^U , that heavily relies Eq. (4.4) on a Hessian HU of a sufficiently smooth orientation score $U : \text{SE}(2) \rightarrow \mathbb{R}$, is indeed a data-driven left-invariant metric tensor field in Lemma B.3.

Lemma B.1. *The Hessian $HU = \nabla^* dU$ of a smooth function $U : M \rightarrow \mathbb{R}$ relative to connection ∇ on manifold M satisfies*

$$HU(X, Y) = X(YU) - \nabla_X YU. \quad (\text{B.1})$$

Proof. One can easily see that

$$\begin{aligned} HU(X, Y) &= \nabla^* dU(X, Y) = \langle \nabla_X^* dU, Y \rangle \\ &\stackrel{(4.17)}{=} X \langle dU, Y \rangle - \langle dU, \nabla_X Y \rangle \\ &= X(YU) - (\nabla_X Y)U. \end{aligned}$$

□

Remark B.1 (Alternative formulation Hessian). Let M be a smooth manifold with connection ∇ . Let $\mathbf{p} \in M$ and $X_p, Y_p \in T_p(M)$, i.e., two tangent vectors not necessarily associated to a vector field. Let $f \in C^\infty(M, \mathbb{R})$.

Let $\mathcal{X} : [-\delta, \delta] \rightarrow M$, with $\delta > 0$, such that

$$\begin{cases} \mathcal{X}(0) = \mathbf{p} \\ \dot{\mathcal{X}}(0) = X_p \\ \nabla_{\dot{\mathcal{X}}(t)} \dot{\mathcal{X}} = 0 \quad \forall t \in [-\delta, \delta], \end{cases}$$

i.e., \mathcal{X} is the unique autoparallel curve through \mathbf{p} with tangent vector X_p . For all $s, t \in [-\delta, \delta]$ let $P_{s,t}^{\mathcal{X}} : T_{\mathcal{X}(s)} M \rightarrow T_{\mathcal{X}(t)} M$ be the parallel transport operator along the curve \mathcal{X} , which is uniquely defined by the following properties

1. $P_{t,t}^{\mathcal{X}} = id \quad \forall t \in [-\delta, \delta],$
2. $P_{t_2,t_3}^{\mathcal{X}} \circ P_{t_1,t_2}^{\mathcal{X}} = P_{t_1,t_3}^{\mathcal{X}}, \quad \forall t_1, t_2, t_3 \in [-\delta, \delta],$
3. smooth with respect to \mathcal{X}, t and s .

Then the map $t \mapsto P_{0,t}^{\mathcal{X}} Y_p \in T_{\gamma(t)} M$ gives a smooth vector field along the curve \mathcal{X} that is unique parallel transport of Y_p along that curve, i.e., with the property $\nabla_{\dot{\mathcal{X}}(t)} (P_{0,t}^{\mathcal{X}} Y_p) = 0$.

Lemma B.2. *We can now define the Hessian of a (sufficiently) smooth function $f : M \rightarrow \mathbb{R}$ also as follows*

$$\begin{aligned} Hf_p(X_p, Y_p) &:= \partial_t ((P_{0,t}^{\mathcal{X}} Y_p) f)(0) \\ &:= \lim_{t \downarrow 0} \frac{(P_{0,t}^{\mathcal{X}} Y_p) f - Y_p f}{t}. \end{aligned} \tag{B.2}$$

Proof. If X and Y are smooth vector fields then

$$(\nabla_X Y)|_p = \nabla_{X_p} Y = \lim_{t \rightarrow 0} \frac{P_{t,0}^{\mathcal{X}} Y_{\mathcal{X}(t)} - Y_p}{t}.$$

Note that $\lim_{t \rightarrow 0} P_{t,0}^{\mathcal{X}} = id$, so

$$\begin{aligned} \lim_{t \rightarrow 0} \frac{Y_{\mathcal{X}(t)} - P_{0,t}^{\mathcal{X}} Y_p}{t} &= \lim_{t \rightarrow 0} P_{t,0}^{\mathcal{X}} \frac{Y_{\mathcal{X}(t)} - P_{0,t}^{\mathcal{X}} Y_p}{t} \\ &= \lim_{t \rightarrow 0} \frac{P_{t,0}^{\mathcal{X}} Y_{\mathcal{X}(t)} - Y_p}{t} = \nabla_{X_p} Y. \end{aligned} \tag{B.3}$$

Now, we have

$$\begin{aligned} Hf(X_p, Y_p) &\stackrel{(B.2)}{=} \lim_{t \rightarrow 0} \frac{P_{0,t}^{\mathcal{X}} Y_p f - Y_p f}{t} \\ &\stackrel{(B.3)}{=} \lim_{t \rightarrow 0} \frac{(Yf)(\mathcal{X}(t)) - (Yf)(\mathcal{X}(0))}{t} - \nabla_{X_p} Y f \\ &= X_p(Yf) - \nabla_{X_p} Y f, \end{aligned}$$

which is the same as Eq. (B.1). \square

Lemma B.3. *The metric tensor field \mathcal{G}^U introduced in Eq. (4.4) is a data-driven left invariant metric tensor field.*

Proof. We recall that the dual norm used in the definition of the data-driven metric tensor field \mathcal{G}^U is given by

$$\|HU|_p(\dot{\mathbf{p}}, \cdot)\|_* = \sup_{\substack{Y \in T_p(\mathbb{M}_2) \\ \|Y\|=1}} |HU|_p(\dot{\mathbf{p}}, Y)|.$$

In order to prove that \mathcal{G}^U is a data-driven left invariant metric tensor field, we need the following identities:

$$\left((L_g)_* Y \right)_{g\mathbf{p}} (\mathcal{L}_g U) = Y_{\mathbf{p}}(U), \quad (\text{B.4})$$

$$\nabla_{(L_g)_* X} (L_g)_* Y = (L_g)_* \nabla_X Y, \quad (\text{B.5})$$

where Eq. (B.4) is the definition of the pushforward, and Eq. (B.5) is the equivariance of the Cartan connection $\nabla = \nabla^{U=1}$. In addition, it is important that

$$Y_{\mathbf{p}} \mapsto (L_g)_* Y_{\mathbf{p}} \text{ is an isometry } T_{\mathbf{p}}(\mathbb{M}_2) \rightarrow T_{g\mathbf{p}}(\mathbb{M}_2), \quad (\text{B.6})$$

so $\|Y\| = \|\tilde{Y}\|$, where $Y \in T_{\mathbf{p}}(\mathbb{M}_2)$ and $\tilde{Y} \in T_{g\mathbf{p}}(\mathbb{M}_2)$.

We verify the data-driven left invariance for each term of the metric tensor field, starting with $\mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}})$:

$$\begin{aligned} \mathcal{G}_{g\mathbf{p}}((L_g)_* \dot{\mathbf{p}}, (L_g)_* \dot{\mathbf{p}}) &= \sum_{i,j=1}^n g_{ij} \omega^i|_{g\mathbf{p}} ((L_g)_* \dot{\mathbf{p}}) \omega^j|_{g\mathbf{p}} ((L_g)_* \dot{\mathbf{p}}) \\ &= \sum_{i,j=1}^n g_{ij} \omega^i|_{\mathbf{p}} (\dot{\mathbf{p}}) \omega^j|_{\mathbf{p}} (\dot{\mathbf{p}}) = \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}}). \end{aligned}$$

Next, we examine the data-driven left invariance of the term

$$\begin{aligned} \|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2 &= \sup_{\substack{Y \in T_{\mathbf{p}}(\mathbb{M}_2) \\ \|Y\|=1}} |HU|_{\mathbf{p}}(\dot{\mathbf{p}}, Y)|^2 \\ &= \sup_{\substack{Y \in T_{\mathbf{p}}(\mathbb{M}_2) \\ \|Y\|=1}} |\dot{\mathbf{p}}(YU) - dU\nabla_{\dot{\mathbf{p}}} Y|^2. \end{aligned}$$

This holds because (set $\tilde{Y} = (L_g)_* Y$)

$$\begin{aligned} &\left\| H(\mathcal{L}_g U)|_{g\mathbf{p}}((L_g)_* \dot{\mathbf{p}}, \cdot) \right\|_*^2 \\ &= \sup_{\substack{\tilde{Y} \in T_{g\mathbf{p}}(\mathbb{M}_2) \\ \|\tilde{Y}\|=1}} \left| (L_g)_* \dot{\mathbf{p}}(\tilde{Y} \mathcal{L}_g U) - d\mathcal{L}_g U \nabla_{(L_g)_* \dot{\mathbf{p}}} \tilde{Y} \right|^2 \\ &= \sup_{\substack{Y \in T_{\mathbf{p}}(\mathbb{M}_2) \\ \|Y\|=1}} \left| \dot{\mathbf{p}}(YU) - dU\nabla_{\dot{\mathbf{p}}} Y \right|^2 = \|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2. \end{aligned}$$

Since $\|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2$ satisfies the data-driven left invariant property, we also have

$$\frac{\|H(\mathcal{L}_g U)|_{g\mathbf{p}}((L_g)_*\dot{\mathbf{p}}, \cdot)\|_*^2}{\max_{\substack{\dot{\mathbf{q}} \in T_{\mathbf{q}}(\mathbb{M}_2) \\ \|\dot{\mathbf{q}}\|=1}} \|H(\mathcal{L}_g U)|_{g\mathbf{p}}((L_g)_*\dot{\mathbf{q}}, \cdot)\|_*^2} = \frac{\|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2}{\max_{\substack{\dot{\mathbf{q}} \in T_{\mathbf{p}}(\mathbb{M}_2) \\ \|\dot{\mathbf{q}}\|=1}} \|HU|_{\mathbf{p}}(\dot{\mathbf{q}}, \cdot)\|_*^2}.$$

Therefore, \mathcal{G}^U is data-driven left-invariant:

$$\begin{aligned} \mathcal{G}_{g\mathbf{p}}^{\mathcal{L}_g U}((L_g)_*\dot{\mathbf{p}}, (L_g)_*\dot{\mathbf{p}}) &= \mathcal{G}_{g\mathbf{p}}((L_g)_*\dot{\mathbf{p}}, (L_g)_*\dot{\mathbf{p}}) \\ &\quad + \lambda \frac{\|H(\mathcal{L}_g U)|_{g\mathbf{p}}((L_g)_*\dot{\mathbf{p}}, \cdot)\|_*^2}{\max_{\substack{\dot{\mathbf{q}} \in T_{\mathbf{p}}(\mathbb{M}_2) \\ \|\dot{\mathbf{q}}\|=1}} \|H(\mathcal{L}_g U)|_{g\mathbf{p}}((L_g)_*\dot{\mathbf{q}}, \cdot)\|_*^2} \\ &= \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}}) + \lambda \frac{\|HU|_{\mathbf{p}}(\dot{\mathbf{p}}, \cdot)\|_*^2}{\max_{\substack{\dot{\mathbf{q}} \in T_{\mathbf{p}}(\mathbb{M}_2) \\ \|\dot{\mathbf{q}}\|=1}} \|HU|_{\mathbf{p}}(\dot{\mathbf{q}}, \cdot)\|_*^2} \\ &= \mathcal{G}_{\mathbf{p}}^U(\dot{\mathbf{p}}, \dot{\mathbf{p}}). \end{aligned}$$

□

COST FUNCTION C: A MULTI-SCALE CROSSING-PRESERVING VESSELNESS FILTER VARIANT



The differentiable cost function $C : \mathbb{M}_2 \rightarrow [\delta, 1]$ is an important tool used to encode where vascular structures are located. Costs are high (“1”) outside the blood vessels, and low (δ given in experiments) at the center of the blood vessels, stimulating the geodesic to move along the vascular structure. Many approaches to automatically calculate the vessel locations have been proposed over the years [19, 108, 205]. In the calculation of the tracking results, we use a cost function inspired by [108]. The exact relationship between the cost and the vesselness filter is defined in Eq. (C.2). The vesselness expression $V^{\mathbb{M}_2}(U_f^a) : \mathbb{M}_2 \rightarrow \mathbb{R}^+$ is, as in [98, 108]

$$V^{\mathbb{M}_2}(U_f^a) = \begin{cases} 0 & \text{if } Q \leq 0, \\ \exp\left(\frac{-\mathcal{R}^2}{2\sigma_1^2}\right)\left(1 - \exp\left(\frac{-\mathcal{S}^2}{2\sigma_2^2}\right)\right) & \text{if } Q > 0, \end{cases} \quad (\text{C.1})$$

where $U_f^a(\mathbf{x}, \theta)$, $a > 0$ fixed, is a single layer of a multilayer wavelet transform. In all experiments, we set $\sigma_1 = 0.5$ and $\sigma_2 = 0.5 \|\mathcal{S}\|_\infty$. In Eq. (C.1), the anisotropy measure \mathcal{R} , structure measure \mathcal{S} and convexity criterion Q are given by

$$\begin{aligned} \mathcal{R} &= \left| \frac{\left(\mathcal{A}_1^2 U_f^a \right)^{s, \beta, \sigma_{s, Ext}, \sigma_{a, Ext}}}{\left(\mathcal{A}_2^2 U_f^a \right)^{s, \beta, \sigma_{s, Ext}, \sigma_{a, Ext}}} \right|, \quad Q = \left(\mathcal{A}_2^2 U_f^a \right)^{s, \beta, \sigma_{s, Ext}, \sigma_{a, Ext}}, \\ \mathcal{S} &= \sqrt{\left(\left(\mathcal{A}_1^2 U_f^a \right)^{s, \beta, \sigma_{s, Ext}, \sigma_{a, Ext}} \right)^2 + \left(\left(\mathcal{A}_2^2 U_f^a \right)^{s, \beta, \sigma_{s, Ext}, \sigma_{a, Ext}} \right)^2}, \end{aligned}$$

where $\mathcal{A}_i^2 U_f^a := \mathcal{A}_i \mathcal{A}_i U_f^a$, and where the superscripts s, β denote Gaussian derivatives at spatial scale $s = 0.5\sigma_s^2$ and angular scale $0.5\beta^2$, where $\beta = 0.75$, and where the superscripts $\sigma_{s, Ext}, \sigma_{a, Ext}$ denote external regularization with spatial scale $\sigma_{s, Ext} = \sigma_s = a$ and angular scale $\sigma_{a, Ext}$.

Here, we implement the dual norm $\|HU|_{\mathbf{p}}(\mathbf{\hat{p}}, \cdot)\|_*^2$ is computed by Eq. (4.6) using Gaussian derivatives with scales $\sigma_{s, Ext}$ and $\sigma_{a, Ext}$.

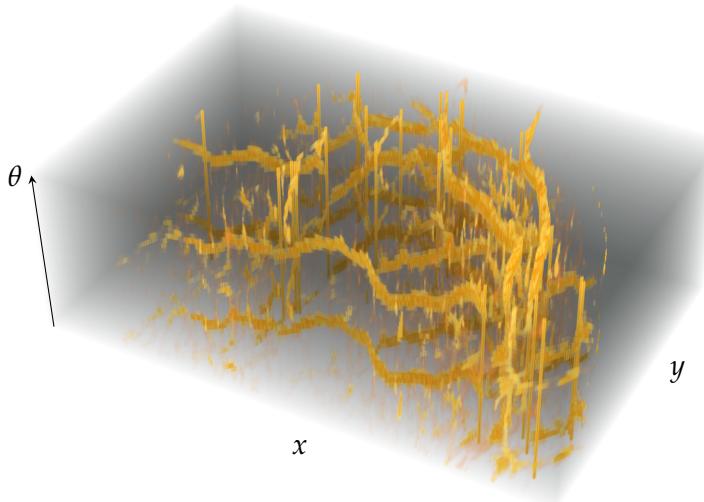


Figure C.1: 3D visualisation of a cost function $(x, y, \theta) \mapsto C(x, y, \theta)$ calculated with the introduced multi-scale crossing-preserving vesselness variant, of a retinal image f , with $\sigma_s \in \{1, 2\}$ and $\sigma_{a,Ext} = 0$.

Last, we apply erosion with scale s_e on $V^{\mathbb{M}_2}$, denoting the result by $V_{s_e}^{\mathbb{M}_2}$. Then, the cost function $C : \mathbb{M}_2 \rightarrow \mathbb{R}^+$ is defined as (similar to [19])

$$\begin{aligned} (V^{\mathbb{M}_2}(U_f))(\mathbf{x}, \theta) &:= \mu_\infty^{-1} \sum_{l=1}^{N_s} (V_{s_e}^{\mathbb{M}_2}(U_f^{a_l}))(\mathbf{x}, \theta), \\ C(\mathbf{x}, \theta) &:= \left(1 + \lambda \left((V^{\mathbb{M}_2}(U_f))(\mathbf{x}, \theta)\right)^p\right)^{-1} \end{aligned} \quad (\text{C.2})$$

where N_s denotes the number of scales, and a_l denotes the different scales that are considered. The scaling parameter μ_∞ is defined as $\mu_\infty := \left\| \sum_{l=1}^{N_s} V_{s_2}^{\mathbb{M}_2}(U_f^{a_l}) \right\|_\infty$. In all experiments, the values of parameters $\sigma_{a,Ext}, \lambda \geq 0$ and $p > 0$ are chosen to be 0, 1000 and 2 respectively. In Fig. C.1, a cost function constructed by the above formulation is visualized. Here, the vertical structures at bifurcations allow for in-place rotations, as depicted in Fig. 1.4.

D

A DAPTATION TO ASYMMETRIC DATA-DRIVEN FINSLER FUNCTIONS

A generalization of [83, Thm. 1, 2, 4] to go from the symmetric model $(\mathbb{M}_2, \mathcal{G}^U)$ to the asymmetric model $(\mathbb{M}_2, \mathcal{F}^U)$, means in practice that we have to adapt the Eikonal equation, cf. Eq. (4.25), to

$$(\alpha_1^U(\cdot))^{-1} ((\mathcal{A}_1^U W)_+)^2 + \sum_{j=2}^3 (\alpha_j^U(\cdot))^{-1} (\mathcal{A}_j^U W)^2 = 1,$$

where $(x)_+ = \max\{x, 0\}$, and backtracking, cf. Eq. (4.27) to

$$\begin{aligned}\dot{\tilde{\gamma}}^1 &= \frac{1}{W(g)} |\alpha_1^U|^{-1} ((\mathcal{A}_1^U W)_+) (\tilde{\gamma}) \\ \dot{\tilde{\gamma}}^k &= \frac{1}{W(g)} |\alpha_k^U|^{-1} (\mathcal{A}_k^U W) (\tilde{\gamma}), k = 2, 3.\end{aligned}$$

This adapted model performs reasonably well in practice. However, cusps may still appear in projected geodesics of this adapted model since the required (*no “reverse gear”*) condition

$$\dot{\gamma}^1 = \dot{x} \cos \theta + \dot{y} \sin \theta = \dot{\mathbf{x}} \cdot \mathbf{n}(\theta) \geq 0 \quad (\text{D.1})$$

differs from the actually applied condition $\dot{\tilde{\gamma}}^1 \geq 0$. If the angle between $\mathcal{A}_1^U|_{\tilde{\gamma}}$ and $\mathcal{A}_1|_{\tilde{\gamma}}$ is not too large (which is often the case when geodesics pass locations with low cost), projected geodesics usually do not exhibit cusps.

A more robust and less intuitive solution -*that does exclude cusps altogether*- in spatially projected data-driven geodesics, is given in Lemma 4.4.2 in the Numerical Section with computational scheme in Eq. (4.39) and backtracking in Eq. (4.41).

Essentially, in this approach (used in our experiments!) one takes the positive part of $\langle M^{-1}(\omega_U^1 - \alpha \omega_U^2)|_{\tilde{\gamma}}, \dot{\tilde{\gamma}} \rangle$ to ensure (D.1), rather than taking the positive part of $\dot{\tilde{\gamma}}^1 = \langle \omega_U^1|_{\tilde{\gamma}}, \dot{\tilde{\gamma}} \rangle$.

If U is constant, then $\alpha = 0$, $\omega_U^1 = \omega^1$ and then we have $\dot{\tilde{\gamma}}^1 = \dot{\gamma}^1 \geq 0$, thereby if U is constant we indeed end up in the standard Reeds-Shepp car model without reverse gear [83].

E

LOGARITHMIC NORM APPROXIMATION

Computing exact distances in a Lie group G is computationally expensive. Therefore, we approximate the norm with a so-called *logarithmic norm approximation*. This chapter elaborates on how this is done, as in [23, 184].

Let us consider any Lie group G , equipped with a left-invariant metric tensor field \mathcal{G} , Lie group elements $g \in G$ and identity element $e \in G$. We consider an exponential curve $\gamma : [0, 1] \rightarrow G$, connecting e to g where $\|\dot{\gamma}(0)\|$ is minimal, i.e., $\gamma(0) = e$, $\gamma(1) = g$ with the property $\gamma(s + t) = \gamma(s) \cdot \gamma(t)$. We want to be able to calculate distances between different elements in the Lie group, e.g., the distance between e and g , denoted by $d_{\mathcal{G}}(e, g)$. This operation is computationally intensive. Therefore, we approximate the exact distance with the distance of a logarithm, as explained in this chapter:

$$d_{\mathcal{G}}(e, g) \leq \text{Length}_{\mathcal{G}}(\gamma) \quad (\text{E.1a})$$

$$= \int_0^1 \|\dot{\gamma}(t)\|_{\mathcal{G}} dt \quad (\text{E.1b})$$

$$= \int_0^1 \|(L_{\dot{\gamma}(t)})_* \dot{\gamma}(0)\|_{\mathcal{G}} dt \quad (\text{E.1c})$$

$$= \int_0^1 \|\dot{\gamma}(0)\|_{\mathcal{G}} dt \quad (\text{E.1d})$$

$$= \|\dot{\gamma}(0)\|_{\mathcal{G}} \quad (\text{E.1e})$$

$$= \|\log g\|_{\mathcal{G}}, \quad (\text{E.1f})$$

where the equality in Eq. (E.1f) holds because by definition $\log g := \dot{\gamma}(0) \in T_e(G)$, Eq. (E.1e) holds because the metric tensor field is left-invariant, and where Eq. (E.1d) holds because

$$\begin{aligned} \dot{\gamma}(s) &= \frac{d}{dt} \gamma(s + t) \Big|_{t=0} = \frac{d}{dt} (\gamma(s) \cdot \gamma(t)) \Big|_{t=0} \\ &= \left(\frac{d}{dt} L_{\gamma(s)} \gamma(t) \right) \Big|_{t=0} = (L_{\gamma(s)})_* \dot{\gamma}(0), \end{aligned}$$

with $L_g : G \rightarrow G$ the left action defined by $L_g h = g \cdot h$ for all $h \in G$, and where the last equality follows directly by the definition of the pushforward. Hence,

we have found an upper bound for the distance from $e \in G$ to a point $g \in G$ in any Lie group G .

In this thesis, we consider two Lie groups; the special Euclidean group $G = \text{SE}(2)$ and the special orthogonal group $G = \text{SO}(3)$. For both, we will calculate $\log g$ for $g \in G$. Let $R_G(g)$ denote the matrix representation of the element $g \in G$, such that $R_G(g_1)R_G(g_2) = R_G(g_1g_2)$. Then, the associated Lie algebra is denoted by $\text{span}(A_i)$, where

$$A_i = \left. \frac{\partial R_G(g)}{\partial g^i} \right|_e, \quad (\text{E.2})$$

with $e \in G$ the identity element. Here, $\{A_i\}$ denotes the left-invariant frame, with dual basis $\{\omega^i\}$ satisfying $\langle \omega^i, A_j \rangle = \delta_j^i$. Then,

$$\sum_{i=1}^n g^i A_i \leftrightarrow g \in G.$$

E.1 LOGARITHMIC NORM APPROXIMATION IN $\text{SE}(2)$

We begin by calculating $\log g$ for the special Euclidean group $G = \text{SE}(2)$, where $g = (x, y, \theta) \in G$. The generating matrix $R_{\text{SE}(2)}$ is given by

$$R_{\text{SE}(2)}(x, y, \theta) = \begin{pmatrix} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{pmatrix}.$$

Then, the associated Lie algebra $se(2) = \text{span}(A_1, A_2, A_3)$ is spanned by

$$A_1 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

calculated by Eq. (E.2) using identity element $e = (0, 0, 0)$. Hence,

$$\begin{aligned} \text{SE}(2) \ni (x, y, \theta) &\leftrightarrow \\ &\begin{pmatrix} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{pmatrix} = \exp(xA_1) \exp(yA_2) \exp(\theta A_3) \\ &= \exp(c^1 A_1 + c^2 A_2 + c^3 A_3) \\ &= \begin{pmatrix} \cos c^3 & -\sin c^3 & \left(c^1 \cos \frac{c^3}{2} - c^2 \sin \frac{c^3}{2}\right) \text{sinc} \frac{c^3}{2} \\ \sin c^3 & \cos c^3 & \left(c^1 \sin \frac{c^3}{2} + c^2 \cos \frac{c^3}{2}\right) \text{sinc} \frac{c^3}{2} \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned}$$

Hence, one has the following relations:

$$\begin{cases} x = \left(c^1 \cos \frac{c^3}{2} - c^2 \sin \frac{c^3}{2} \right) \operatorname{sinc} \frac{c^3}{2} \\ y = \left(c^1 \sin \frac{c^3}{2} + c^2 \cos \frac{c^3}{2} \right) \operatorname{sinc} \frac{c^3}{2} \\ \theta = c^3 \mod 2\pi \end{cases}, \quad \begin{cases} c^1 = \frac{x \cos \frac{\theta}{2} + y \sin \frac{\theta}{2}}{\operatorname{sinc} \frac{\theta}{2}} \\ c^2 = \frac{-x \sin \frac{\theta}{2} + y \cos \frac{\theta}{2}}{\operatorname{sinc} \frac{\theta}{2}} \\ c^3 = \theta, \end{cases} \quad (\text{E.3})$$

where $c^1 \partial_x|_e + c^2 \partial_y|_e + c^3 \partial_\theta|_e \in T_e(\text{SE}(2))$ represent the logarithmic coordinates.

E.2 LOGARITHMIC NORM APPROXIMATION IN SO(3)

We follow the same approach as above for the special orthogonal group $G = \text{SO}(3)$, with group elements $(\alpha, \beta, \varphi) \in \text{SO}(3)$. The generating matrix $R_{\text{SO}(3)}$ is given by

$$R_{\text{SO}(3)}(\alpha, \beta, \varphi) = \begin{pmatrix} c\alpha \ c\beta & s\beta \ c\varphi & -s\alpha \ c\beta \ s\varphi & -s\alpha \ c\beta \ c\varphi & -s\beta \ s\varphi \\ -c\alpha \ s\beta & s\alpha \ s\beta \ s\varphi + c\beta \ c\varphi & s\alpha \ s\beta \ c\varphi - c\beta \ s\varphi & & \\ s\alpha & & c\alpha \ s\varphi & & c\alpha \ c\varphi \end{pmatrix},$$

where $c\alpha = \cos \alpha$, $c\beta = \cos \beta$, $c\varphi = \cos \varphi$, $s\alpha = \sin \alpha$, $s\beta = \sin \beta$ and $s\varphi = \sin \varphi$. The associated Lie algebra $\text{so}(3)$ is spanned by

$$A_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

calculated by Eq. (E.2) using identity element $e = (0, 0, 0)$. Hence,

$$\begin{aligned} \text{SO}(3) \ni (\alpha, \beta, \varphi) &\leftrightarrow R_{\text{SO}(3)}(\alpha, \beta, \varphi) \\ &= \exp(-\beta A_3) \exp(-\alpha A_2) \exp(\varphi A_1) \\ &= \exp(c^1 A_1 + c^2 A_2 + c^3 A_3). \end{aligned} \quad (\text{E.4})$$

We aim to determine the relationship between (α, β, φ) and (c^1, c^2, c^3) such that Eq. (E.4) holds. To this end we note that we can interpret the rotation described by $R_{\text{SO}(3)}$ as a counterclockwise rotation around an axis \mathbf{a} with an angle φ . Both can be identified from the matrix representation of the group elements of $\text{SO}(3)$ given by $R_{\text{SO}(3)}$. The rotation axis, denoted by the vector \mathbf{a} , is the eigenvector corresponding to the eigenvalue 1. If the rotation matrix $R_{\text{SO}(3)}$

is not symmetric, the eigenvector \mathbf{a} and the corresponding rotation angle ϕ are given by

$$\mathbf{a} = \frac{1}{2 \sin \phi} \begin{pmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{pmatrix}, \quad \phi = \arccos \left(\frac{\text{Tr}(R_{\text{SO}(3)}) - 1}{2} \right). \quad (\text{E.5})$$

The expression for the rotation angle ϕ is readily verified, as for a rotation matrix \tilde{R} rotating a vector with an angle $\tilde{\phi}$ around the z -axis. Then, using the conjugation invariance of the trace operator and the fact that there exists a matrix P such that $P^{-1}\tilde{R}P = R_{\text{SO}(3)}$, Eq. (E.5) holds for the rotation with angle ϕ around the axis \mathbf{a} .

Then, one has, by Rodrigues' rotation formula and the well-known Lie group exp formula in $\text{SO}(3)$ that

$$\begin{aligned} \exp(\phi(a^1 A_1 + a^2 A_2 + a^3 A_3))(\mathbf{x}) &= \exp(\phi \cdot \mathbf{a})(\mathbf{x}) \\ &= \mathbf{x} \cos \phi + (\mathbf{a} \times \mathbf{x}) \sin \phi + \mathbf{a} (\mathbf{a} \cdot \mathbf{x}) (1 - \cos \phi) = R_{\text{SO}(3)}(\mathbf{x}). \end{aligned}$$

Thus, we conclude that the logarithmic coordinates in $\text{SO}(3)$ are given by

$$\begin{aligned} (c^1, c^2, c^3)^T &= \phi \cdot \mathbf{a} = \frac{1}{2 \operatorname{sinc} \phi} \begin{pmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{pmatrix} \\ \Leftrightarrow \Omega_{\mathbf{a}} &= a^1 A_1 + a^2 A_2 + a^3 A_3 = \frac{1}{2 \operatorname{sinc} \phi} (R - R^T), \end{aligned} \quad (\text{E.6})$$

where $\Omega_{\mathbf{a}} \in \text{so}(3) = T_e(\text{SO}(3))$ is the antisymmetric matrix associated to $\Omega_{\mathbf{a}}(\mathbf{x}) = \mathbf{a} \times \mathbf{x}$. The formula has a removable singularity at $\phi = \pm\pi$ where the matrix R becomes symmetric (with 1,-1,-1 on the diagonal after diagonalisation). At $\phi = 0$ we have $\operatorname{sinc}(0) = 1$.

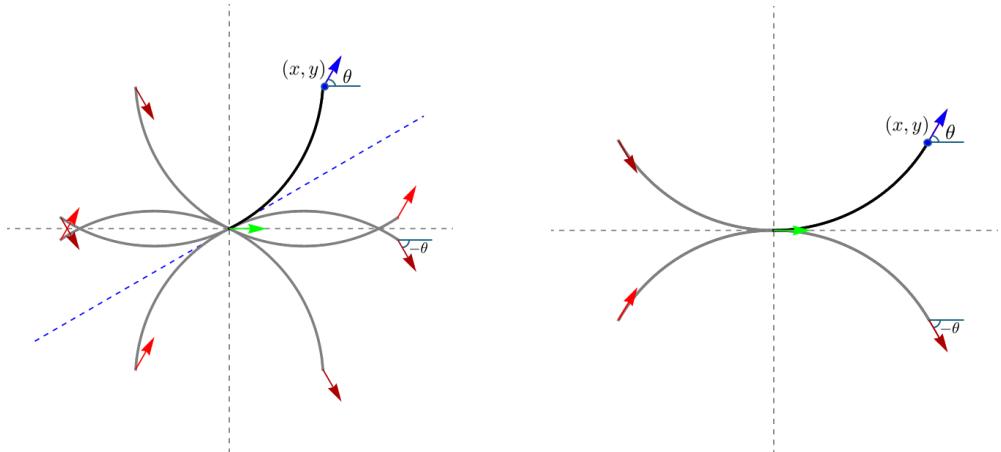
F

REFLECTIONAL SYMMETRIES

As shown in Eq. (8.7), the distance function is left-invariant $d_{\mathcal{G}}(g, h) = d_{\mathcal{G}}(h^{-1}g, e) \approx \|\log h^{-1}g\|_{\mathcal{G}}$. Furthermore, this distance's (reflectional) symmetries carry over to the δ -connected component algorithm as we will show.

Lemma F.1. Set $q = h^{-1}g$. Then $\log q$ can be expressed as a linear combination of basis vectors, i.e., $\log q = \sum_{i=1}^n c^i(q) A_i|_e \in T_e(G)$. Now its norm $\|\log q\|_{\mathcal{G}}$ is invariant under $c^i \mapsto \pm c^i$, $i = 1, \dots, n$, which gives rise to 2^n symmetries.

Proof. We recall from Appendix E that $\|\log q\|_{\mathcal{G}} = \sqrt{\sum_{i=1}^n w_i(c^i)^2}$. Then, its invariance under $c^i \mapsto \pm c^i$ follows immediately. \square



(a) Reflectional symmetries with c^2 unconstrained (general Riemannian case [23]).

(b) Reflectional symmetries with $c^2 = 0$ (sub-Riemannian case [144]).

Figure F.1: Reflectional symmetries of the blue point $(x, y, \theta) \equiv (x, y, R_\theta) \in \text{SE}(2)$, reflected in the three reflection axes. Depending on the axis, the orientation is updated to $-\theta$ or stays the same. The blue reflection axis is only for the endpoints of the geodesics.

Fig. F.1 illustrates this concept for $\text{SE}(2)$ where $n = \dim \text{SE}(2) = 3$. For further details, see [23, 144].

Corollary F.1. *If the connected component algorithm is applied to a point cloud $\{q_1, \dots, q_n\} \subset \text{SE}(2)$, then it produces the same reflected connected components as when it is applied to $\{\tilde{\varepsilon}^i(q_1), \dots, \tilde{\varepsilon}^i(q_n)\}$, where $\tilde{\varepsilon}^i(q) = \exp(\varepsilon^i \log(q))$, $i = 0, \dots, 2^n - 1$, denotes the symmetry given by the reflection of the logarithmic coordinates as denoted in [23, Table 5].*

Proof. The statement follows immediately from Lemma F.1 applied to $\|\log \tilde{\varepsilon}^i(g)\|$:

$$\|\log \tilde{\varepsilon}^i(q)\|_{\mathcal{G}} = \|\varepsilon^i(\log q)\|_{\mathcal{G}} \stackrel{\text{Lemma F.1}}{=} \|\log q\|_{\mathcal{G}}.$$

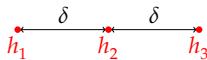
Thus, the distances between elements are the same, and consequently, the δ -connected components are also the same. \square

The precise formulas of these reflections can be found in [23, 144]. For a quick intuition, see Fig. F.1. Essentially, they are found by line reflections in the half-angle axis, the x - and y -axes, possibly negating the angle. Intuitively, they are depicted in Fig. F.1a.

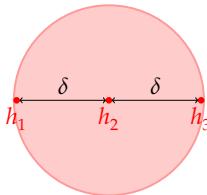
G

FOR $\alpha > 1$ THE EQUIVALENCE RELATION BREAKS

An example illustrating the breakdown of the equivalence relation \mathcal{E} for $\alpha > 1$ is shown in Fig. G.1. It motivates why in connected component algorithms we choose $\alpha \downarrow 1$ (or $\alpha = 1$) and why we used the evolutions for $\alpha > 1$ only for the

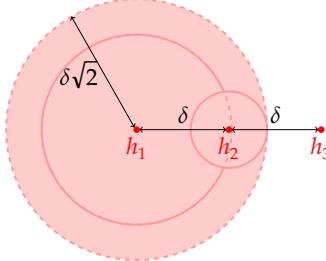


- (a) Example setting of one connected component $[h] = \bigcup_{i=1}^3 h_i$ with $d(h_1, h_3) = 2\delta$ and $d(h_1, h_2) = d(h_2, h_3) = \delta$.



According to `find_deltaCC`
with $\alpha = 2$ and $g_0 = h_2$:
 $h_1 \mathcal{E} h_2 \mathcal{E} h_3$

- (b) Identification of δ -connected component with the algorithm starting at $g_0 = h_2$, identifying the entire component in a single step, independent of the value of α .



According to `find_deltaCC`
with $\alpha = 2$ and $g_0 = h_1$:
 $h_1 \mathcal{E} h_2, h_2 \mathcal{E} h_3,$
but $h_1 \not\mathcal{E} h_3$

- (c) Identification of δ -connected component with $\alpha = 2$ and the algorithm starting at $g_0 = h_1$ converging after 1 step, but fails to identify h_3 as part of the δ -connected component $[h_1]$.

Figure G.1: The δ -connected component $[h]$ (in red in Fig. G.1a) is not always completely identified with `find_deltaCC` when not choosing $\alpha = 1$. In the example, we have used $\alpha = 2$ to show that the algorithm produces different δ -connected components depending on the starting point that may differ from the full δ -connected component. The example uses starting points h_2 and h_1 for results in Figs. G.1b and G.1c respectively.

affinity measure experiments (where $\alpha > 1$ shrinks the wavefront propagation and softens the max-pooling). Some theoretical results on viscosity solutions for HJB equations [13, 23, 69, 92], do require $\alpha > 1$. In the algorithms for the experiments (e.g., `find_δCC`) one can either take $\alpha = 1$ or $\alpha \downarrow 1$ which boils down to the same when taking the pointwise limit in the morphological convolutions (Eq. (8.11)) that solve the HJB PDE system (Eq. (8.10)) almost everywhere.

COMPARISON OF δ -CONNECTED COMPONENT METHOD TO EXISTING SEGMENTATION AND CLUSTERING METHODS

H.1 SEGMENT ANYTHING MODEL 2

The Segment Anything Model 2, trained on the SA-V dataset [161], is an improvement of the Segment Anything Model [120]. The updated model can segment both videos and images.

The model can be used in two different ways: 1) an automatic segmentation method that does not require user input, and 2) an interactive segmentation method that requires the user to select (or deselect) pixels that need to be included (or excluded) from the segmentation. Fig. H.1 shows the output of the two different methods applied to the reference image in Fig. H.1a. The automatic method automatically samples its starting point. This results in segments of the background (black) and of the foreground (white). The density of the sample points can be adjusted. An increased number of sample points slows the model down, but increases the chance of correctly segmenting smaller features, such as thin blood vessels. For example, in Fig. H.1b, the model missed the segment

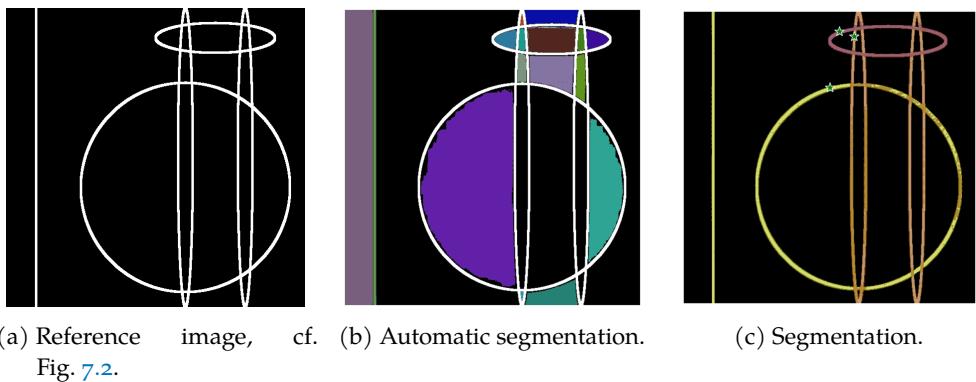


Figure H.1: The output of the SAM 2 [161] applied to the reference image using two different versions of the model; the auto masklet generator in Fig. H.1b and interactive segmentation that requires the (de)selection of pixels by the user in Fig. H.1c. The three stars were manually selected by the user (see Remark H.1.1). See Table H.1 for the prediction scores of the segments in Fig. H.1c.

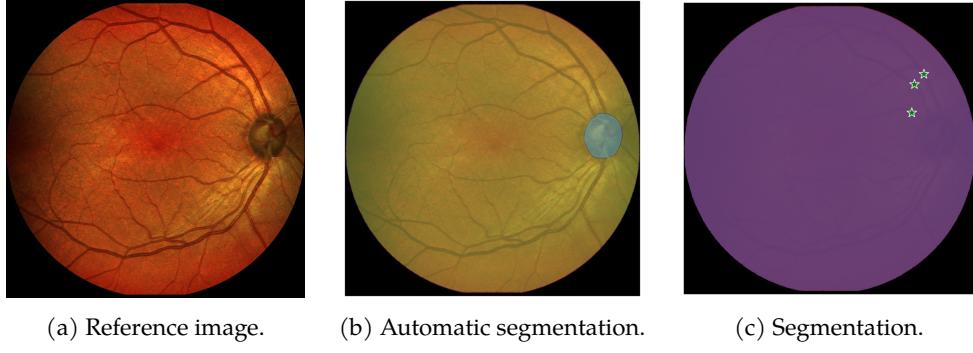


Figure H.2: The output of the SAM 2 [161] applied to a retinal image using two different versions of the model; the auto masklet generator (cf. Fig. H.2b) and interactive segmentation that requires the (de)selection of pixels by the user in Fig. H.2c. Three points (stars) were manually selected by the user (see Remark H.1.1). Note that all three stars gave the same segmentation as output, visualized by the purple circle. See Table H.1 for the prediction scores of the segments in Fig. H.2c.

of the circles and ellipses. With the interactive method, the user selects the pixels that they want to be included in the segmentation.

Fig. H.1c shows the output of the three segmentations corresponding to the three selected pixels (visualized with a green star). Note that the line on the left is included in the segmentation of the large circle, the vertically oriented ellipses are grouped together, and there is overlap between the segmentations.

Fig. H.2 shows the output of the automatic and interactive model applied to a retinal image. Note that in both cases the model did not succeed in segmenting any blood vessels. We got the same results when trying the model on an enhanced version of the retinal image.

Remark H.1.1. The interactive version of SAM 2 produces multiple segmentations per user input. These segmentations are ranked by a quality prediction score generated by the model. We noted cases where the lower-ranked segmentations were closer to the desired output. However, this was not consistently the case. Fig. H.3 shows the difference between segmentations of higher and lower rank according to the model applied to different retinal images. Nevertheless, the output of SAM 2 could potentially be improved by adjusting the quality prediction score to our specific use case (i.e., segmentation of blood vessels).

H.2 TOPOLOGICAL MODE ANALYSIS TOOL (TOMATO)

ToMATo [46] is a useful tool from topological data analysis that allows for the identification of clusters in images. In Fig. H.4, we present some results

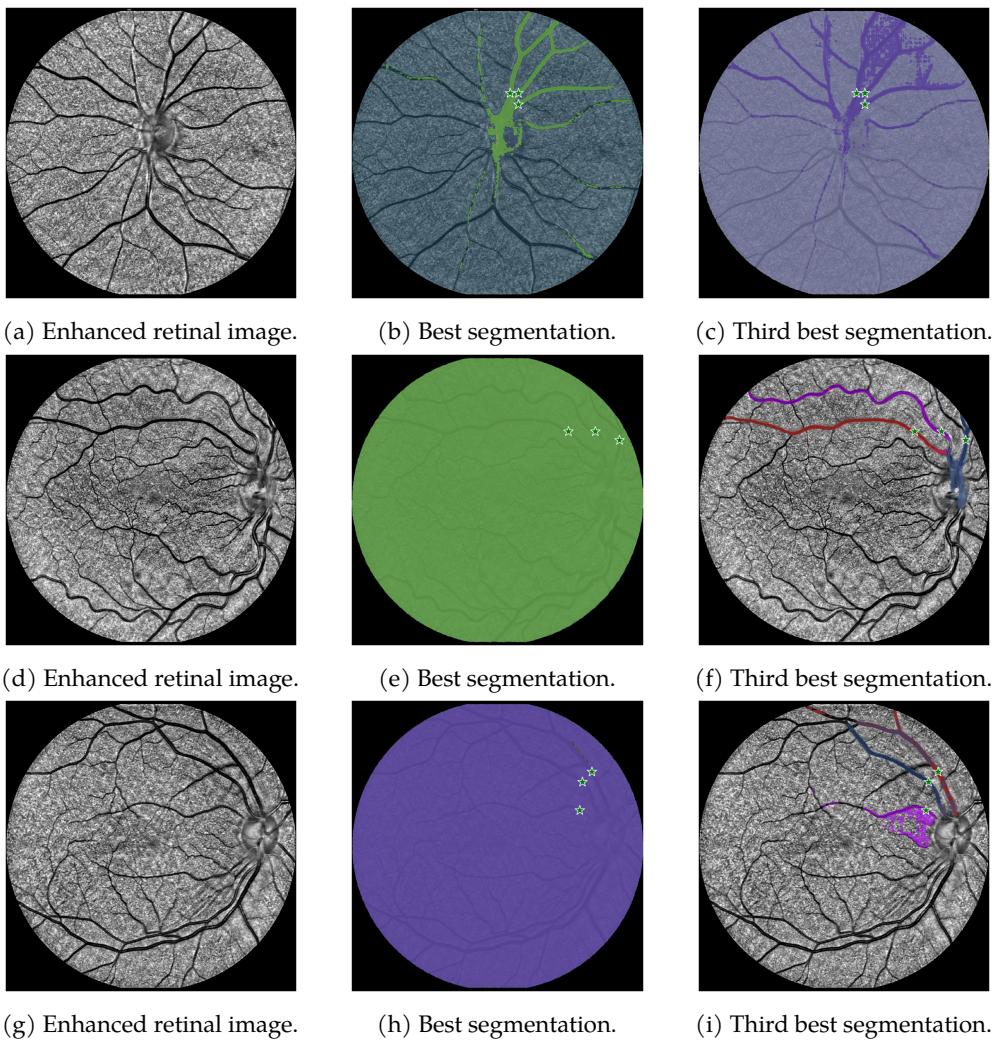


Figure H.3: The output of interactive version of the SAM z applied to three enhanced retinal images (STAR48, STAR34 and STAR38 in Figs. H.3a, H.3d and H.3g, respectively, to be compared to the δ -connected component output in Figs. 8.9, 8.11 and 8.13, respectively). The middle images show the segmentation with the best quality prediction score generated by the model, often generating the same output for all stars. The left images show the segmentation with the third best score. Note that there are situations where the third best segmentation is actually better than the best segmentation according to the model.

	Segment Prediction scores		
	Top/left	Middle	Bottom/right
Fig. H.1c	0.789	0.590	0.746
Fig. H.2c	0.988	0.988	0.992
Fig. H.3b	0.186	0.145	0.143
Fig. H.3c	0.053	0.093	0.028
Fig. H.3e	0.311	0.914	0.391
Fig. H.3f	0.042	0.017	0.047
Fig. H.3h	0.193	0.938	0.945
Fig. H.3i	0.079	0.032	0.030

Table H.1: The prediction scores for the segmentations of different images. For more information, see Remark H.1.1.

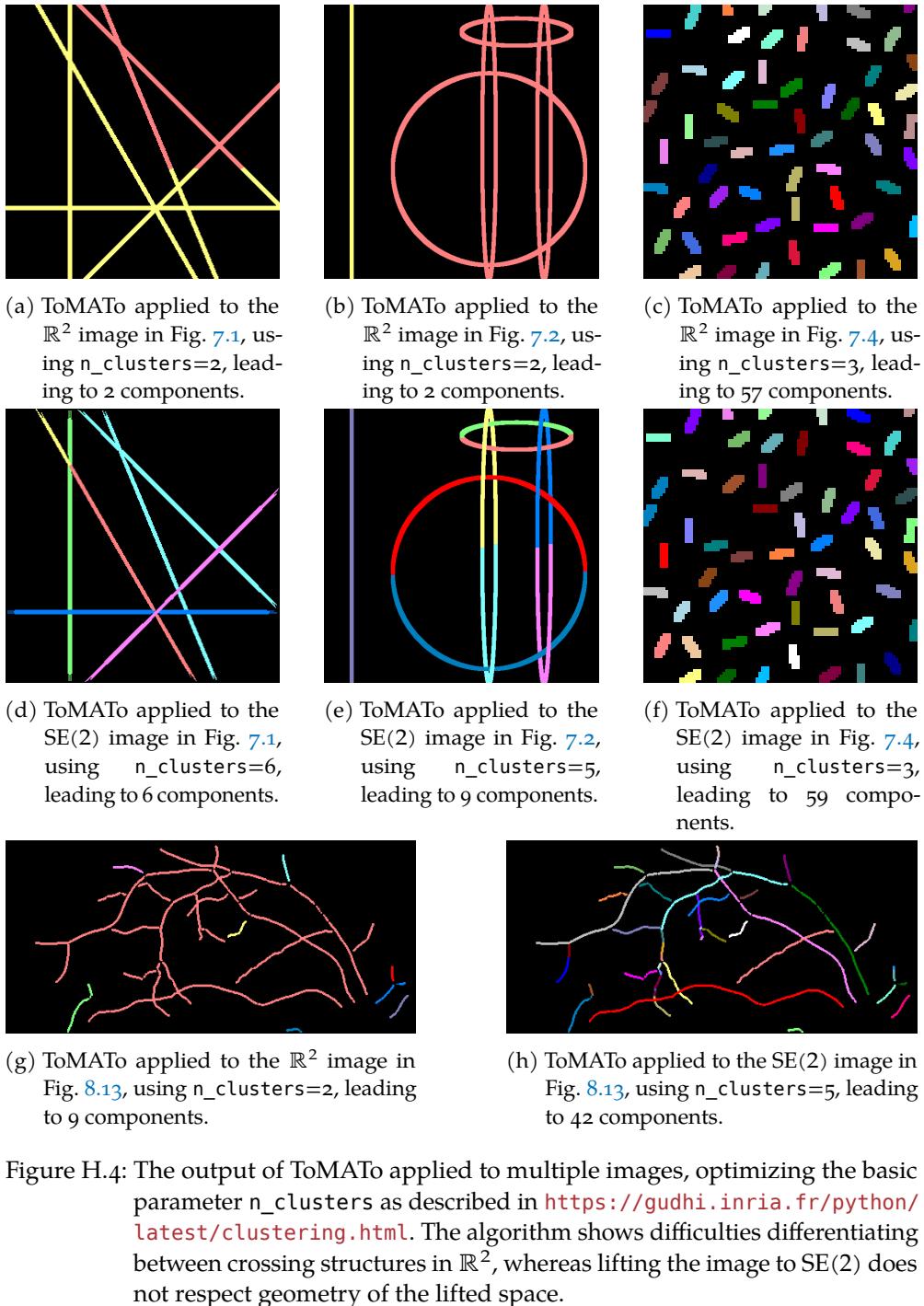
where we applied this tool to some of the images in our introduction and experimental section. We carefully followed the parameter settings and the basic optimization steps described in <https://gudhi.inria.fr/python/latest/clustering.html>, always manually choosing the optimal number of clusters in the application of ToMATo.

The method provides reasonable (sub-)segments, but does struggle to differentiate between different crossing structures in \mathbb{R}^2 , as can be seen in Figs. H.4a and H.4b compared to Fig. 7.1 in Chapter 8. Lifting the data to $SE(2)$, and applying the algorithms to this lifted data, does not respect (nor take advantage of) the underlying geometry of the space.

For instance, the periodicity in the angular direction is not respected, see Fig. H.4e. More importantly, akin to the results by Bekkers et al. [19, Fig.11], we see that (left-invariant) sub-Riemannian geometry (or anisotropic Riemannian geometry [83, Fig.15, Thm.2]) outperforms isotropic Riemannian geometry, in terms of perceptual organization and tracking of line elements. Our δ -connected component algorithm takes advantage of such sub-Riemannian geometry on Lie groups and thereby provides well-aligned elements by design.

ToMATo does provide a reasonable grouping of elements, but as it is not designed on the Lie group $SE(2)$ (and does not include neurogeometrical Lie group models [15, 23, 55, 74, 154]). Here, it struggles with perceptual grouping of lines as visible in Figs. H.4c and H.4f.

Lastly, we applied ToMATo to one of the retinal images in the experimental section. Again, applying the algorithm to the 2-dimensional image data leads to merging of separate vascular trees (cf. Fig. H.4g). Lifting the data to $SE(2)$ results in the splitting of vessel segments (Fig. H.4h).



As ToMATo is a mathematically well-underpinned grouping algorithm to identify clusters in data, it could be very interesting to extend the methods to Lie groups such as SE(2). This requires a serious redesign of the existing ToMATo algorithm, which is beyond the scope of this thesis.

I

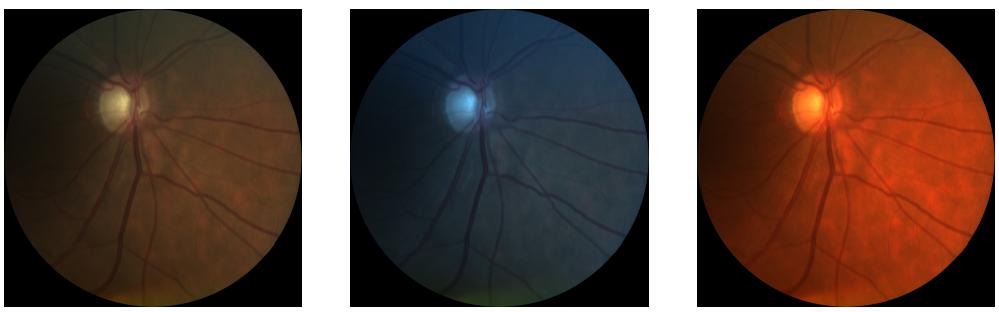
COLOR TEMPERATURE

We define a color temperature transformation $T_p : \mathbb{L}_2(\mathbb{R}^2) \rightarrow \mathbb{L}_2(\mathbb{R}^2)$ that changes the color temperature of an image $f : \mathbb{R}^2 \rightarrow [0, 1]^3$ with a factor $p > 0$. Let $f(\mathbf{x}) = (r(\mathbf{x}), g(\mathbf{x}), b(\mathbf{x}))$, where $r(\cdot), g(\cdot), b(\cdot)$ denote the red, green and blue channel of the image. The transformation is defined as

$$(T_p f)(\mathbf{x}) = (\max\{r(\mathbf{x}) \cdot p^{-1}, 1\}, g(\mathbf{x}), \max\{b(\mathbf{x}) \cdot p, 1\}).$$

In Fig. I.1, one can see the effect of this transformation on an image from the Leuven-Haifa dataset with parameters $p = 0.5$ and $p = 2$.

The parameter p is sampled from the density distribution $f_X(x) = \frac{1}{x \log(r^{-2})}$ where $x \in [r, 1/r]$ with user input $r \in (0, 1]$. Sampling from this density distribution ensures that $\mathbb{P}(X < p) = \mathbb{P}(X > 1/p)$. Hence, the probability of the image appearing more red than the input image is equal to that of the image appearing more blue.



(a) Input image from Leuven-Haifa dataset. (b) Changed color temperature with $p = 0.5$. (c) Changed color temperature with $p = 2$.

Figure I.1: Example of image from the Leuven-Haifa dataset [87].

SUMMARY

The retina contains rich information about an individual's health, enabling non-invasive diagnostics through retinal imaging. This thesis focuses on improving automated methods for analyzing these images, specifically in tracking blood vessels, identifying connected components, and classifying vessels as arteries or veins.

TRACKING BLOOD VESSELS

We present three advances in vessel tracking:

- **Data-driven Cartan connections:** We introduce a new geodesic tracking model on the space of positions and orientations $\mathbb{M}_2 := \mathbb{R}^2 \times S^1$, which allows globally optimal tracking of complex vasculature. The model automatically adapts to curvature and misalignment and is computed via a modified anisotropic fast-marching method. We formulate a theorem that describes all shortest and straight curves with respect to the new data-driven connection. The experiments confirm that the geodesic tracking method uses a single, flexible, transparent, data-driven geodesic model, providing globally optimal curves that correctly follow highly complex vascular structures in retinal images.
- **Optical and TV-flow enhancements:** We provide a single, new, more effective Finsler function on \mathbb{M}_2 that combines PDE-based tracking with PDE-based enhancements and optical corrections (e.g., dehazing and illumination), improving performance on challenging clinical images.
- **Crossing-preserving tracking on spherical images:** We recognize that the retina is not flat, but spherical. We propose a method to compute cusp-free, crossing-preserving geodesics in the space of spherical positions and orientations \mathbb{W}_2 on wide-field images. The results show clear advantages compared to the non-crossing-preserving tracking in \mathbb{W}_2 and are comparable to the tracking results in \mathbb{M}_2 .

IDENTIFYING CONNECTED COMPONENTS

We develop a new algorithm to extract connected components from a compact set I in Lie groups with a left-invariant Riemannian distance. For a given $\delta > 0$,

the algorithm finds the largest cover of I such that all sets in the cover are separated by at least distance δ . Using morphological dilations, the presented method groups well-aligned structures and distinguishes at crossings. We prove convergence in a finite number of steps. Furthermore, we propose to use specific affinity matrices, post-processing the δ -connected components by grouping them based on their local proximity and alignment. Applying the δ -connected component method to retinal images via the orientation scores on $SE(2)$ outperforms standard \mathbb{R}^2 approaches in identifying vascular branches.

CLASSIFYING VESSELS AS ARTERIES AND VEINS

We enhance artery-vein classification by combining convolutional neural networks with our δ -connected component framework. This ensures consistent labeling along entire vascular trees, improving reliability in clinical applications.

ABOUT THE AUTHOR

Nicky van den Berg started her studies in Applied Mathematics at Eindhoven University of Technology in 2014. After graduating with her bachelor's degree, she spent a year as the secretary of the study association, GEWIS. In 2018, she continued her studies with a Master's in Industrial and Applied Mathematics in Eindhoven. During her master's degree, she spent six months at the University of Western Ontario under the supervision of prof.dr. Mikko Karttunen. After returning to the Netherlands, she did her graduation project titled '*Improved Computational Models and Algorithms for Geodesic Vessel Tracking in Orientation Scores*', supervised by Remco Duits and Bart M.N. Smets. She graduated cum laude in 2021.



During her studies, Nicky was actively involved in co-determination, serving on both the program committee and the department council. Towards the end of her studies, she co-founded an alumni association for graduates of Computer Science and Applied Mathematics at Eindhoven University of Technology, initially serving as secretary and currently as treasurer.

Following her graduation, she began her Ph.D. in the Geometric Learning and Differential Geometry group in 2021, where she worked on the content presented in this thesis. For this work, she was supervised by dr.ir. Remco Duits, dr. Tos T.J.M. Berendschot, and prof.dr. Olga Mula. She visited Emory University for one month in October 2024, hosted by dr. Lars Ruthotto. During the last year of Nicky's Ph.D. trajectory, she spent one day a week in the ophthalmology department of Maastricht UMC+.

ACKNOWLEDGMENTS

This brings us to the final part of my thesis: the acknowledgments – probably one of the most-read parts of this dissertation. I am deeply grateful for all the support I have received over the past four years. Colleagues, friends, and family have helped me both directly and indirectly with this research and the writing of this thesis. Thank you to all!

First of all, I would like to thank my promoter. Remco, without you, this thesis would have taken a very different form. I am very grateful that you found me to be the best candidate for this Ph.D. position. You stimulated me to grow. Your energy and never-ending enthusiasm for research are infectious and have definitely inspired me. I look back at five years of working together, where I have learned a lot from you. I am especially thankful for your compassion and understanding, which allowed me the space to emotionally process difficult situations in my personal life.

Tos, your involvement as co-promoter has been invaluable, and I deeply appreciate your support. You taught me a lot about the medical aspects discussed in this thesis. Your critical eye helped me refine my methods to better suit medical applications. Especially during the last year, the biweekly ophthalmology research talks broadened my horizon and inspired possible research directions.

Olga, I am truly grateful for your guidance and support as my co-promoter. I feel fortunate that Remco invited you to collaborate on my project. I enjoyed our discussions about my research, and your thoughtful feedback significantly improved the quality of my work.

I am also sincerely grateful for the members of my defense committee: Giovanna Citti, Barry Koren, Jan Modersitzki, Clarisa Sánchez Gutiérrez, and Mitko Veta. Thank you for reading my thesis and providing valuable feedback that helped enhance the final result.

My CASA colleagues helped me feel at home every day. First of all, my direct colleagues: Bart, Finn, Gautam, Gijs, Lars, Leanne, Rick, and Sjors. I truly enjoyed sharing an office in the first few years and the many (non-)scientific discussions we had. I also look back fondly on our many fun office outings, which helped us connect beyond work. I also fondly remember the dinners I shared with the 'Data-Driven Group'. Benjamin, Daan, Jeroen, and Rafael: thank you for the wonderful dinners and fun drinks together.

I also really enjoyed the weekly bouldering sessions with Daan, Francesco, Joop, Pierre, and Robert. Those Tuesday nights helped me unwind and balance the challenges of work with relaxation and sport.

I also look back fondly on many fun social drinks and activities, like karaoke, together with colleagues. I enjoyed spending time together. Special thanks to our secretaries: Diane, Enna, and Gea, for their support and the co-organization of the CASA outing, and to Antonio and Teun for co-organizing the CASA days together.

I am grateful to have had the opportunity to collaborate with excellent researchers from other universities: Erik Bekkers, Alejandro García-Castellanos, Andrii Kompanets, Jean-Marie Mirebeau, and Shuhe Zhang. Our interactions were invaluable to my understanding of the topics we worked on together. The time I spent in Atlanta at Emory University also shaped me as a researcher. Thank you to Lars Ruthotto for hosting Gijs, Pascal and me. I enjoyed the interactions with your group and the other visiting researchers from NTNU. Elena Celledoni, Marta Gherardelli, Martine Hansen, and Brynjulf Owren, thank you for the engaging discussions about our research. They helped me broaden my perspective. Lastly, I am grateful for the Fridays I spent in Maastricht. Interacting with non-mathematicians helped me view my research from new angles and communicate it in a more accessible way.

Without my friends, this thesis would have been very different. I am grateful for their support during difficult times, as well as for the many joyful moments we have shared over the years. Anne, Bram, Floris, Jeffrey, Laura, Niekje, Niels, Roy, and Wesley: our monthly Ada Alumni meetings were a highlight. Together, we built a wonderful network that helps alumni stay connected to the university and to each other. Laura, thank you for our many shared sports sessions. They helped me relax. Merel, thank you for the evenings we spent talking and crafting together. Floris, thank you for our movie nights. Puck, your loyalty and kindness make you a truly special friend. Our tea breaks, during which we rarely drank tea, always lifted my spirits. Your empathy and perspective helped me see things more clearly. Jim, your patience and kindness are remarkable. Thank you for always showing up, listening, and reminding me of my strengths. Huib, thank you for becoming a part of my life. I look forward to the future.

That brings me to my family, who cheered me on and reminded me what truly matters. Mom, thank you for always being there for me. Dad, your belief in me helped me through the challenging times. Lisa, I am very proud of who you have become. It was not always easy, but together with Loekie, you can do anything. Anke, your kindness and wisdom are unlimited. You have the world at your feet. I am proud to be your daughter and sister. I love you.