

MASTER

Improved Computational Models and Algorithms for Geodesic Vessel Tracking in Orientation Scores

van den Berg, Nicky

Award date:
2021

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

MASTER THESIS

Improved Computational Models and
Algorithms for Geodesic Vessel Tracking in
Orientation Scores

Author:

Nicky VAN DEN BERG
n.j.v.d.berg@student.tue.nl
0923424

Graduation Committee:

Remco DUITTS (supervisor)
r.duits@tue.nl

Bart SMETS (supervisor)
b.m.n.smets@tue.nl

Barry KOREN (committee member)

July 13, 2021

Contents

1	Theoretical Background	6
1.1	Main Concepts	6
1.1.1	Lie Groups	6
1.1.2	Group Representations	8
1.1.3	Images	9
1.1.4	Orientation Score	9
1.1.5	Vessel Tracking	10
1.1.6	(Pseudo)distances	11
1.2	Static Frame	14
1.3	Left Invariant Frame	14
1.4	Gauge Frame	17
1.4.1	Construction of the Gauge Frame	18
1.4.2	External Regularization	19
1.4.3	Iterated Stabilization	19
2	Metric Tensor Fields	21
3	Numerical Approach	23
3.1	General Approach	23
3.2	Solving the Eikonal Equation in Riemannian Setting	23
3.2.1	Finite Differences	24
3.2.2	(Hamiltonian) Fast Marching	28
3.3	Backtracking from Tip to Seed	32
3.3.1	Left Invariant Frame	33
3.3.2	Gauge Frame	34
3.4	Comparison of Results to Exact Solution	35
3.4.1	Comparison of Exact Solution and PDE-method	37
3.4.2	Comparison of Exact Solution and Fast Marching	38
3.4.3	Conclusion	39
4	Vessel Tracking	41
4.1	Experimental Settings	41
4.1.1	Cost Function	41
4.1.2	Parameter settings	42
4.2	Visualization Results	42
4.3	Experiments	42
4.3.1	Experiment 1	42
4.3.2	Experiment 2	44
4.3.3	Experiment 3 - Low Resolution	45
4.3.4	Experiment 4 - High Tortuosity	47
4.4	Conclusion	50
5	Tracking Blood Vessels in Vascular Trees	53
5.1	Experiment	55
5.2	Conclusion	60
6	Future Research	61
A	(Gauge Frame and Left Invariant) Metric Expressed with Respect to Static Frame	62
A.1	Conclusion	64
B	Control Sets	65
B.1	Left Invariant Frame	65
B.1.1	Visualization of the Control Sets	66
B.2	Gauge Frame	68
B.2.1	Visualization of the Control Sets	68
C	Cartan Connections	70
C.1	Left Invariant Frame	71
C.2	Gauge Frame	73

Introduction

Retinal images are frequently used to analyse the vascular system. This non-invasive method of observation of the circulatory system is often used to diagnose different kinds of diseases. Studying an image of the retina enables doctors amongst others to diagnose diabetes, as illustrated in Figure 1. Mathematical models can make it easier to keep track of the changes that occur over time with respect to the retina vessels. One of the ways to do so, is by constructing a tool that can automatically track the vessels and compare them with previously obtained images of the same blood vessels.

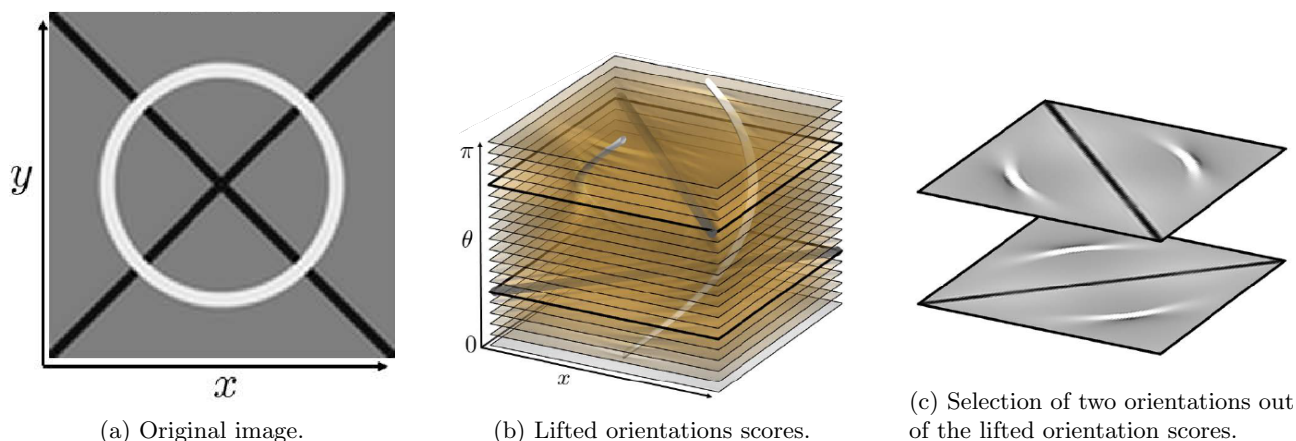


(a) Healthy retina. Taken from [1].

(b) Retina showing hemorrhage and microaneurysms; tortuous vessels. Taken from the Retina Image Bank [2].

Figure 1: The retina vessels of people suffering from diabetes may start to leak and cause bleeding [3]. In addition to that, high tortuosity is also an indicator of diabetes [4]. Eventually, this can lead to blindness.

Often, intersections pose a problem while applying automated tracking of blood vessels in retinal images. When two vessels overlap, the tracking algorithm may opt to continue its way on a different vessel, resulting in paths whose parts are incorrectly connected (e.g. paths that jump from one vessel to another, or paths that do not follow a vessel at all). To prevent this, we use so-called orientation scores. Here, we “lift” the image such that it consists of different layers. Every layer coincides with a different given orientation. For every location and orientation, we store the alignment of the line structure with the given orientation. In many cases, this will separate vessels that used to cross in the original image, as we can see in Figure 2.



(a) Original image.

(b) Lifted orientation scores.

(c) Selection of two orientations out of the lifted orientation scores.

Figure 2: Visualization of the orientation scores of a given image. Adapted from [5].

Every layer in the lifted orientation score has a frame that rotates according to the orientation that we are considering. This frame is called the left invariant frame. It allows a geometrical interpretation of the vessel tracking techniques. Research on discouraging (or disallowing) movement in the direction perpendicular to the vessel direction has been done in previous work by E.J. Bekkers, R. Duits, J.M. Mirebeau and others [6; 7; 8; 9; 10]. In Figure 3, we see a visualization of the alignment in the left invariant frame for different orientations for a given image.

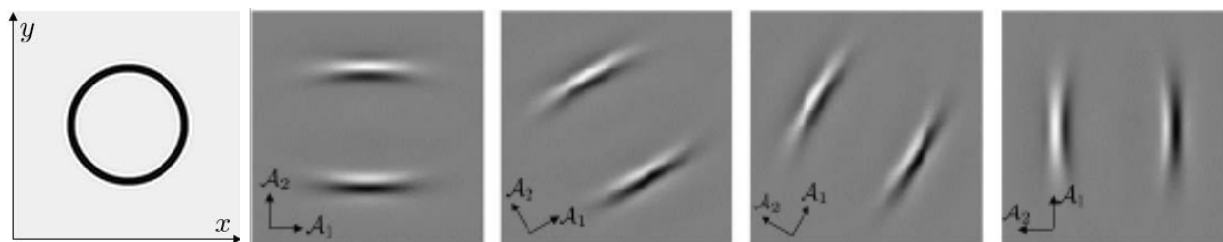


Figure 3: Original image and the alignment in the left invariant frame for different orientations. Adapted from [5; 11].

As we can see, the left invariant frame is not always able to align with the vessel structure in the correct way. This can be caused by:

1. Misestimation of the orientation of the vessel. It is possible that, due to the imperfectness of the orientation scores, a part of a vessel is assigned to have an angular coordinate which is not its true coordinate. Consequently, it is not able to align with the vessel data properly. This is called deviation from horizontality [12]. In Figure 4, a visualization of this phenomenon is given. In practice, we often use a low angular dimension (so we consider a limited number of orientations) in an effort to limit calculation times, which results in more extreme deviations, since a large range of vessel orientations is mapped to the same angular orientation. Accounting for deviation from horizontality is particularly important if one wants to reduce the number of orientation layers, for memory reduction and shorter computation times [12].
2. The left invariant frame lines up with the vessel structure in the orientation score. This works well for straight line structures, but when the orientation of the line structure changes, these curvatures are ignored.

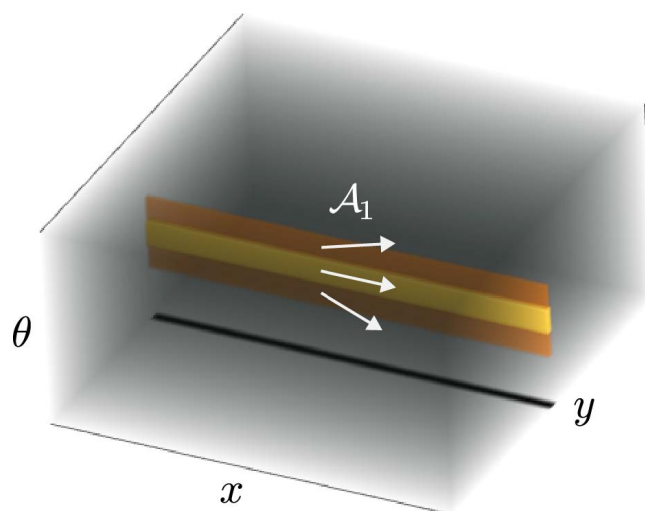


Figure 4: A straight line gets responses from multiple orientations. The directions which react are indicated by the white arrows. Not all of them are aligned with the given line structure. Taken from [13].

Deviation of horizontality may pose a problem when the frame steers the vessel tracking away from the actual vessel structure. An example of such a situation is given in Figure 5. In an effort to encourage the tracking to follow the vessel structure, the gauge frame (or locally adaptive frame) has been introduced in [12; 14; 15; 16]. This frame aligns better with the vessel structure, and stimulates moving in the correct direction, as can be seen in Figure 6. However, there is also a big disadvantage of these gauge frames; they are not smooth. They are vulnerable to changes, depending on the image, and the underlying vessel structure. In addition to that, noise in the image has an impact on the alignment of the frames, and therefore, the best fitting gauge frames can vary widely. There are different methods to align the gauge frames, and improve the alignment of the gauge frames. It is important to note that the gauge frames outside the vessel structures, are still randomly aligned, also after applying smoothing techniques.

Last, it is important to note that sometimes, tracking finds the shortest path by switching its moving direction. Considering the tracking as a car driving over a road (the vessel), it sometimes has to switch gears (hence switching from forward gear to reverse gear and vice-versa). However, vessels do not make sudden sharp turns, and therefore, it should not be necessary to switch gears. In the left invariant frame, a model has been

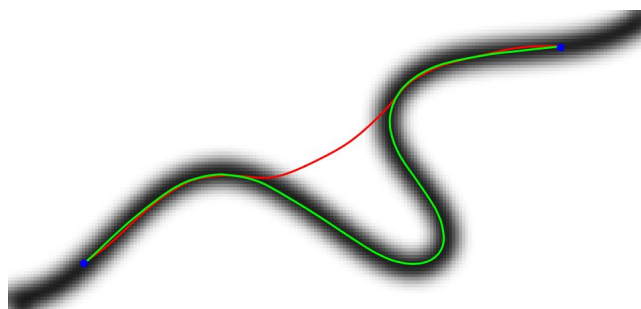


Figure 5: The tracking in the left invariant frame (red) takes a shortcut, where the tracking in the gauge frame (green) does not. The vessel structure is visible in black. The start and end point are indicated by blue points. Adapted from [13].

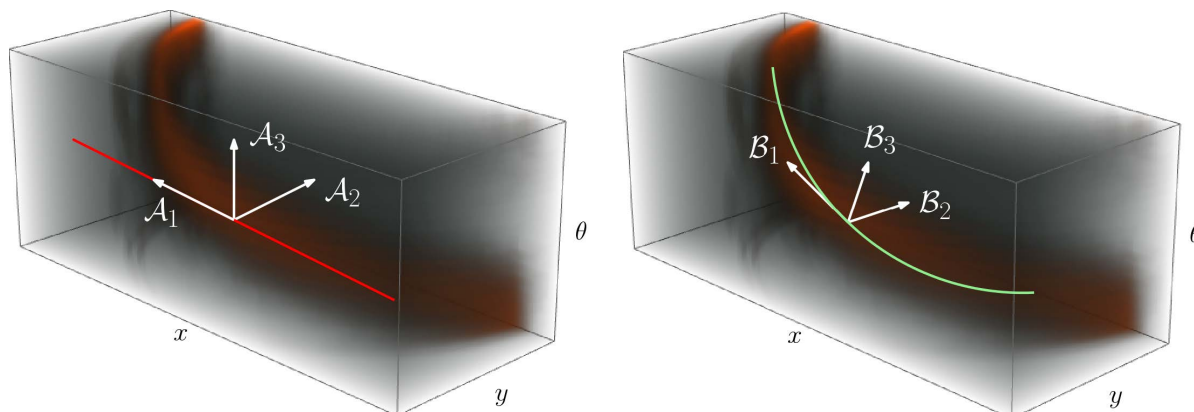


Figure 6: Visualization of left invariant frame (left) and gauge frame (right) on a curved vessel structure. Taken from [13].

introduced to prevent this backward motion by R. Duits, S.P.L. Meesters, J.-M. Mirebeau and J.M. Portegies [17]. In Figure 7, we see a visualization of the effect of disabling the reverse gear. Note that it is still possible for in place rotations to occur. That means that it is possible to stay in the same place and change the orientation slightly to the left or right, but not backward. In this report, we focus on the construction and implementation of the “forward gear” variant of the gauge frame. We investigate the results in simple settings, where we track a single vessel, and in more complex settings, where we track multiple vessels at once, which do not all originate from the same seed. We will consider both finite difference and fast marching solution strategies. We compare

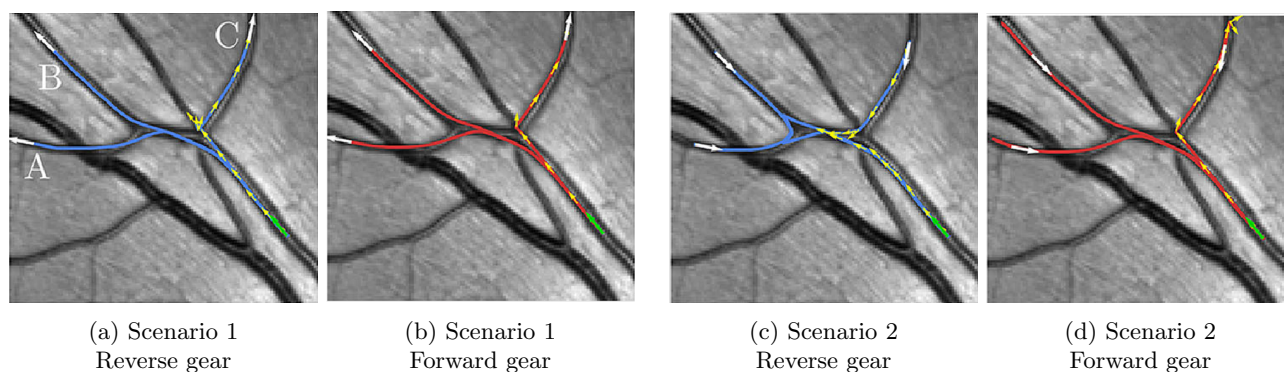


Figure 7: Tracking starts at the green arrow (in its given direction), and ends in the white arrow (in its given direction). The blue and red lines represent the reverse and forward gear model respectively. Adapted from [17].

the results and discuss how they relate to each other in Section 3. Next, we consider four experimental settings in which we calculate the geodesics, using the finite difference approach. We used different metric tensor fields; symmetric and forward gear variants of the (anisotropic) left invariant frame and the (anisotropic) gauge frame. In Section 4, we discuss the influence of the different parameters and differences between the models. The most interesting finding was that the gauge frame was able to follow a highly tortuous vessel, whereas the left

invariant frame did not succeed. The setting is similar to the setting in Figure 5. In Section 5, we present and discuss the tracking results when we track multiple vessels at once. We compare the different models, and discuss the flaws of the models. We find that the gauge frame always bends into the direction that gets the highest response. However, this is not always the direction in which the vessel should be moving, and problems occur at crossings. Solving these tracking problems and other open questions that are still standing, will be presented in Section 6 for future research purposes.

This report builds upon the master thesis of A.J. Wemmenhove [13]. In [13], emphasis is on data-driven versions of Cartan connections, whereas in this report the emphasis is on numerics, fast computational schemes, improved models and practical experiments.

1 Theoretical Background

In the first part of this report, we will discuss some theoretical concepts that are important in order to understand the geometrical methods applied in the mathematical imaging model. We start by introducing some geometrical concepts. Afterwards, we introduce different frames that can be used to do geometrical tracking.

1.1 Main Concepts

1.1.1 Lie Groups

First, we introduce several new concepts that are relevant for understanding the model with which we will be working. We start with the space on which we will be working and analysing our results.

Definition 1.1 (2D positions and orientations). The space of two-dimensional positions and orientations \mathbb{M}_2 is defined as a smooth manifold

$$\mathbb{M}_2 := \mathbb{R}^2 \rtimes S^1,$$

where $S^1 := \mathbb{R}/(2\pi\mathbb{Z})$.

This means that every point $(\mathbf{x}, \theta) \in \mathbb{M}_2$ consists of both a location $\mathbf{x} \in \mathbb{R}^2$ and an orientation $\theta \in S^1$. In Figure 8, an example is given on how the orientation is defined. The orientation θ indicates in which direction the local

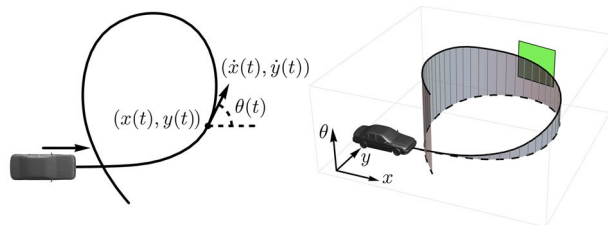


Figure 8: A trajectory of a car, at a given location $(x(t), y(t))$ has an orientation $\theta(t)$. The velocity vector is given by $(\dot{x}(t), \dot{y}(t))$. On the right image, we see the lifted trajectory (height corresponds to orientation $\theta(t)$). Taken from [17].

line-element (visualized as a car with a front in Figure 8) at location \mathbf{x} is pointing. An alternative illustrative example would be when we try to describe a group of people in this space. For each person, we write down their location \mathbf{x} and the direction θ that they are looking into (compared to a reference direction). This way, we can register the population in a more detailed way, since we also know the orientation of each single person, instead of only their locations. This can be of an added value when we want to make them do something, for instance walking in a specific direction or changing their orientation by rotation. Before we will explain how this works, we introduce the concept ‘‘Lie group’’. First, we recall the definition of a group.

Definition 1.2 (Group). A group G with group product \cdot is a structure (G, \cdot) with

- A closed group operation such that for every two elements $g, h \in G$, we have $g \cdot h \in G$.
- An associative multiplication such that for $g, h, i \in G$, we have $(g \cdot h) \cdot i = g \cdot (h \cdot i)$.
- An identity element $e \in G$ exists such that for all $g \in G$ holds $e \cdot g = g \cdot e = g$.
- For every element $g \in G$ an inverse $g^{-1} \in G$ such that $g \cdot g^{-1} = e = g^{-1} \cdot g$.

In our case, we are able to consider a specific type of groups, being the Lie group.

Definition 1.3 (Lie Group). A Lie group G is a differentiable manifold equipped with a group product such that the group operations are compatible with the differentiable structure on the manifold, i.e. multiplication $G \times G \rightarrow G$ and inversion $G \rightarrow G$ are smooth maps.

First, we note that \mathbb{M}_2 is a differentiable manifold. This is the case since \mathbb{M}_2 consists of the cartesian product of \mathbb{R}^2 and S^1 which are both smooth manifolds. In practice, this means that our manifold is locally similar enough to a linear space such that we can do calculus. More specifically, we are considering a group of rotations and translations of vectors in \mathbb{R}^2 ; the ‘‘roto-translation group’’.

Definition 1.4 (2D roto-translations). The space of two-dimensional roto-translations is a Lie group $SE(2) := \mathbb{R}^2 \rtimes SO(2)$. Consider $(\mathbf{x}_1, R_\alpha), (\mathbf{x}_2, R_\beta) \in SE(2)$. Here, $\mathbf{x}_i \in \mathbb{R}^2$ denotes the location and $R_\alpha \in SO(2)$ denotes the rotation matrix over an angle $\alpha \in S^1$ (in radians). Then, we define the semi-direct group product

$$(\mathbf{x}_1, R_1) \cdot (\mathbf{x}_2, R_2) = (\mathbf{x}_1 + R_1 \mathbf{x}_2, R_1 R_2). \quad (1)$$

Note that this is not the direct group product, since that would be given by $(\mathbf{x}_1, R_1)(\mathbf{x}_2, R_2) = (\mathbf{x}_1 + \mathbf{x}_2, R_1 R_2)$. Therefore, we wrote in Definition 1.1 “ \ast ” instead of “ \times ”.

Remark 1.1 (Commutativity). The 2D roto-translations are not commutative. This is easily checked:

$$\begin{aligned}(\mathbf{x}_1, R_1) \cdot (\mathbf{x}_2, R_2) &= (\mathbf{x}_1 + R_1 \mathbf{x}_2, R_1 R_2); \\ (\mathbf{x}_2, R_2) \cdot (\mathbf{x}_1, R_1) &= (\mathbf{x}_2 + R_2 \mathbf{x}_1, R_2 R_1).\end{aligned}$$

This means that the order of roto-translations cannot always be reversed, since $\mathbf{x}_1 + R_1 \mathbf{x}_2 \neq \mathbf{x}_2 + R_2 \mathbf{x}_1$ for most positions $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^2$ and rotation matrices $R_1, R_2 \in SO(2)$. Figure 9 confirms that the roto-translations are not commutative.

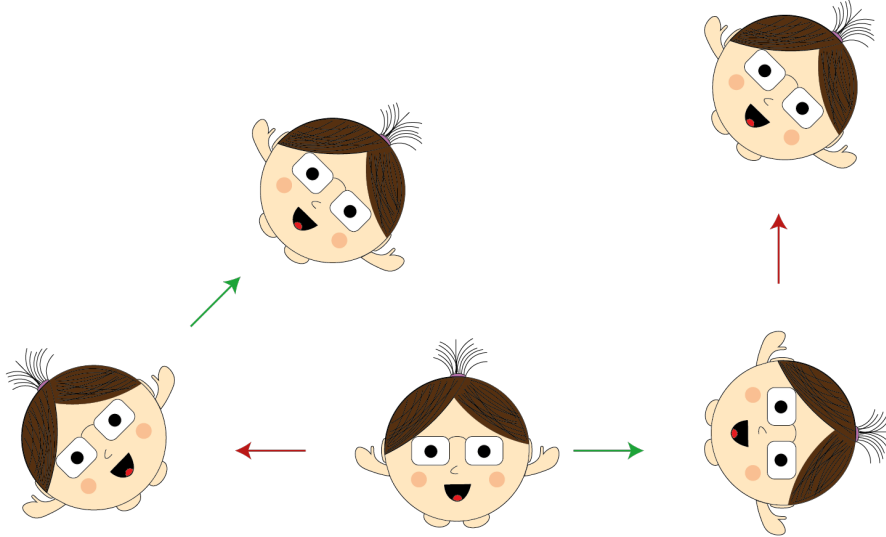


Figure 9: Composed roto-translations on a drawing of a person with a fixed coordinate frame. The green arrow indicates moving to the right and rotating with an angle $-\pi/2$ ($\pi/2$ clockwise), where the red arrow indicates moving to the left and rotating over an angle $\pi/4$ ($\pi/4$ counterclockwise). Switching the order of movements results in different end locations. Hence, the operations are not commutative.

Remark 1.2. Although there is a bijection between orientations $\theta \in S^1$ and rotations $R_\theta \in SO(2)$ in the two dimensional space, this is not the case for higher dimensions. Therefore, we keep the homogeneous space \mathbb{M}_2 and the Lie group $SE(2)$ (acting on the homogeneous space) separate.

Definition 1.5 (Unity element). The unity element of the Lie group is the unique element $e \in G$ such that the product $eg = g$ for all $g \in G$, as mentioned in Definition 1.2. The unity element of the Lie group of roto-translations is given by $e = (\mathbf{0}, I)$, where I is the identity map.

Definition 1.6 (Inverse element). The inverse element of an element $a \in G$ in the Lie group is the unique element $a^{-1} \in G$ such that the product $a^{-1}a = aa^{-1} = e$ where $e \in G$ is the unity element, as introduced in Definition 1.2. The inverse of a roto-translation (\mathbf{x}, R_α) is given by

$$(\mathbf{x}, R_\alpha)^{-1} := (-R_\alpha^{-1} \mathbf{x}, R_\alpha^{-1}).$$

We can easily check that the definition holds:

$$\begin{aligned}(\mathbf{x}, R_\alpha) \cdot (\mathbf{x}, R_\alpha)^{-1} &= (\mathbf{x} - R_\alpha R_\alpha^{-1} \mathbf{x}, R_\alpha R_\alpha^{-1}) = (\mathbf{0}, I) = e; \\ (\mathbf{x}, R_\alpha)^{-1} \cdot (\mathbf{x}, R_\alpha) &= (-R_\alpha^{-1} \mathbf{x} + R_\alpha \mathbf{x}, R_\alpha^{-1} R_\alpha) = (\mathbf{0}, I) = e.\end{aligned}$$

Definition 1.7 (Tangent vector). A tangent vector is considered to be a differential operator on locally defined functions f such that

$$\dot{\gamma}(0)f := \left. \frac{d}{dt} (f \circ \gamma)(t) \right|_{t=0}.$$

Definition 1.8 (Tangent space). The tangent space at the element g , denoted by $T_g(G)$, is the vector space of all tangent vectors of curves passing through the element g of the Lie group. In other words, that means that

$$T_g(G) = \{ \dot{\gamma}(0) | \gamma : \mathbb{R} \rightarrow G \text{ differentiable, with } \gamma(0) = g \}.$$

Definition 1.9 (Tangent bundle). The set of ordered pairs

$$\{(g, \dot{g}) \mid g \in G \text{ and } \dot{g} \in T_g(G)\}$$

is the tangent bundle $T(G)$.

1.1.2 Group Representations

In the previous section, we introduced the concepts of a Lie group, and the notion of “roto-translations”. In this section, we will introduce different group representations. We start with the general definition of group representations.

Definition 1.10 (Group representation). A representation \mathcal{R} of a group G onto a Banach space H is a homomorphism \mathcal{R} (preserves structure) between G and $B(H)$, the space of bounded linear operators on H . I.e. it must satisfy

$$\mathcal{R}_{gh} = \mathcal{R}_g \mathcal{R}_h \quad \text{for all } g, h \in G, \quad \text{and} \quad \mathcal{R}_e = I,$$

where I is the identity map.

Now, we introduce two mappings from \mathbb{R}^2 and \mathbb{M}_2 to \mathbb{R}^2 and \mathbb{M}_2 respectively.

Definition 1.11 (Mapping L_g). The mapping of $L_g : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is given by

$$L_g \mathbf{x} := (\mathbf{y}, \mathbf{R}_\alpha) \cdot \mathbf{x} = \mathbf{y} + \mathbf{R}_\alpha \mathbf{x}, \quad (2)$$

for a roto-translation $g = (\mathbf{y}, \mathbf{R}_\alpha) \in SE(2)$ and a position $\mathbf{x} \in \mathbb{R}^2$.

Definition 1.12 (Mapping U_g). The mapping $U_g : \mathbb{M}_2 \rightarrow \mathbb{M}_2$ is given by

$$U_g \mathbf{x} := (\mathbf{y}, \mathbf{R}_\alpha) \cdot (\mathbf{x}, \theta) = (\mathbf{y} + \mathbf{R}_\alpha \mathbf{x}, \alpha + \theta) \quad (3)$$

for a roto-translation $g = (\mathbf{y}, \mathbf{R}_\alpha) \in SE(2)$ and combined position and orientation $(\mathbf{x}, \theta) \in \mathbb{M}_2$.

Note that the above mappings are in fact group representations where $L : G \rightarrow \text{Aut}(\mathbb{R}^2)$ and $U : G \rightarrow \text{Aut}(\mathbb{M}_2)$, where $\text{Aut}(\mathbb{R}^2) := \{A : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \mid A \text{ linear and } A^{-1} \text{ exists}\}$.

Definition 1.13 (Left/right regular representations on Lie group G). Let G be a Lie group. Then its right regular representation $\mathcal{R} : G \rightarrow B(\mathbb{L}_2(G))$ is given by

$$\mathcal{R}_g U(h) = U(hg) \quad \text{for all } g, h \in G \text{ and all } U \in \mathbb{L}_2(G), \quad (4)$$

whereas its left regular representation $\mathcal{L} : G \rightarrow B(\mathbb{L}_2(G))$ is given by

$$\mathcal{L}_g U(h) = U(g^{-1}h) \quad \text{for all } g, h \in G \text{ and all } U \in \mathbb{L}_2(G). \quad (5)$$

Next, we introduce the definitions of (ir)reducible and unitary representations.

Definition 1.14 (Properties of representations). Let G be a group and H a Hilbert space.

A representation $\mathcal{R} : G \rightarrow B(H)$ is said to be irreducible if the only invariant closed subspaces of H are H and $\{\mathbf{0}\}$ and otherwise it is said to be reducible.

A representation $\mathcal{U} : G \rightarrow B(H)$ is called unitary if $(\mathcal{U}_g)^* = \mathcal{U}_g^{-1}$ for all $g \in G$.

In the definition of a unitary representation, we use the symbol for the adjoint operator. We recall its definition.

Definition 1.15 (Adjoint operator). Let $A : X \rightarrow Y$ be a bounded linear operator. Then, its adjoint operator $A^* : Y \rightarrow X$ satisfies

$$\langle Ax, y \rangle_Y = \langle x, A^* y \rangle_X,$$

where $x \in X$ and $y \in Y$, and where $\langle \cdot, \cdot \rangle_X$ denotes the inner product on space X .

Definition 1.16 (Left-regular representation on \mathbb{R}^2). Let the roto-translation operator be given by

$$(\mathcal{U}_{(\mathbf{x}_0, \theta)} f)(\mathbf{x}) = f(\mathbf{R}_\theta^{-1}(\mathbf{x} - \mathbf{x}_0)), \quad \mathbf{x} \in \mathbb{R}^2, (\mathbf{x}_0, \theta) \in \mathbb{R}^2 \times [0, 2\pi), \quad (6)$$

with translation vector $\mathbf{x}_0 \in \mathbb{R}^2$ and counter-clockwise rotation $\mathbf{R}_\theta \in SO(2)$ with angle θ . The matrix representation with respect to the standard basis is given by

$$\mathbf{R}_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}. \quad (7)$$

The left-regular representation on \mathbb{R}^2 first translates the function f over a distance \mathbf{x}_0 . Then it rotates the translated function over an angle θ .

Proposition 1.1. The operator $\mathcal{U} : G \rightarrow B(H)$ given by $g \mapsto \mathcal{U}_g$ is a unitary group representation.

Proof. First, we show that \mathcal{U}_g is a group representation. That means that we should have $\mathcal{U}_{g_1 g_2} = \mathcal{U}_{g_1} \mathcal{U}_{g_2}$ for all $g_1, g_2 \in G$. We can easily check this. Let $g_1 := (\mathbf{x}_1, \theta_1), g_2 := (\mathbf{x}_2, \theta_2) \in G, f \in \mathbb{L}_2$ given. Then,

$$\begin{aligned} (\mathcal{U}_{g_1} \mathcal{U}_{g_2} f)(\mathbf{x}) &= (\mathcal{U}_{g_2} f)(\mathbf{R}_{\theta_1}^{-1}(\mathbf{x} - \mathbf{x}_1)) = f(\mathbf{R}_{\theta_2}^{-1}(\mathbf{R}_{\theta_1}^{-1}(\mathbf{x} - \mathbf{x}_1) - \mathbf{x}_2)) \\ &= f((\mathbf{R}_{\theta_1} \mathbf{R}_{\theta_2})^{-1}(\mathbf{x} - (\mathbf{x}_1 + \mathbf{R}_{\theta_1} \mathbf{x}_2))) = (\mathcal{U}_{g_1 g_2} f)(\mathbf{x}), \end{aligned}$$

where we recall the definition of the roto-translation group product given in Eq. (1). We also need to show that \mathcal{U}_g is unitary. Let $f, \phi \in \mathbb{L}_2(\mathbb{R}^2)$ be given. Let $g := (\mathbf{x}_1, \theta_1) \in G$ be given. Then,

$$\begin{aligned} \langle \mathcal{U}_g f, \phi \rangle &= \int_{\mathbb{R}^2} \overline{\mathcal{U}_g f(\mathbf{x})} \phi(\mathbf{x}) d\mathbf{x} = \int_{\mathbb{R}^2} \overline{f(\mathbf{R}_{\theta_1}^{-1}(\mathbf{x} - \mathbf{x}_1))} \phi(\mathbf{x}) d\mathbf{x} \\ &= \int_{\mathbb{R}^2} \overline{f(\mathbf{x}_{new})} \phi(\mathbf{R}_{\theta_1} \mathbf{x}_{new} + \mathbf{x}_1) d\mathbf{x}_{new} = \int_{\mathbb{R}^2} \overline{f(\mathbf{x}_{new})} \mathcal{U}_{g^{-1}} \phi(\mathbf{x}_{new}) d\mathbf{x}_{new} \\ &= \langle f, \mathcal{U}_{g^{-1}} \phi \rangle. \end{aligned}$$

Now, showing that $\mathcal{U}_{g^{-1}} = \mathcal{U}_g^{-1}$ suffices. Let $g \in G$ be given. Since we know that \mathcal{U}_g is a group representation, we know that

$$\mathcal{U}_g \circ \mathcal{U}_{g^{-1}} = \mathcal{U}_{gg^{-1}} = \mathcal{U}_e = I,$$

so we have $\mathcal{U}_{g^{-1}} = (\mathcal{U}_g)^{-1} = \mathcal{U}_g^*$. □

1.1.3 Images

We use grey scale images f to assign grey-values to every location $(x, y) \in \mathbb{R}^2$. Each grey value $f(x, y)$ is in the interval $[0, 1]$. Here, 0 and 1 represent black and white respectively. The image is extended in such a way that the function takes on value 0 outside of the compact image domain Ω , i.e.

$$f(\mathbf{x}) := \begin{cases} \text{grey value at location } \mathbf{x} & \text{if } \mathbf{x} \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

That means that intuitively, this is the considered image against an (infinite) black background. This also means that we can consider the images $f \in \mathbb{L}_2(\mathbb{R}^2; [0, 1])$. Note that this holds since $\text{supp}(f) < \infty$ is compact. Hence,

$$\|f\|_{\mathbb{L}_2} = \left(\sum_{i,j} |f(x_{ij})|^2 \right)^{1/2} < \infty.$$

1.1.4 Orientation Score

At the beginning of this section, we introduced the space of 2D positions and orientations \mathbb{M}_2 . Here, we will explain how we intend to extend the domain of an image $f \in \mathbb{L}_2(\mathbb{R}^2; [0, 1])$ to the space of positions and orientations. Since we need our orientation score transform to be a bijective function, we first need to introduce the reproducing kernel Hilbert space.

Definition 1.17 (Reproducing Kernel Hilbert Space $\mathbb{C}_K^{\mathbb{I}}$). We denote the reproducing kernel Hilbert space consisting of functions on index set \mathbb{I} with reproducing kernel K by $\mathbb{C}_K^{\mathbb{I}}$.

Then, we define an orientation score transformation $W_\psi : \mathbb{L}_2(\mathbb{R}^2) \rightarrow \mathbb{C}_K^{SE(2)}$, where the reproducing kernel K is defined by $K(g_1, g_2) := (\mathcal{U}_{g_1}, \mathcal{U}_{g_2})_{\mathbb{L}_2(\mathbb{R}^2)}$ with $g_1, g_2 \in \mathbb{I}$, where index set $\mathbb{I} = SE(2)$. The corresponding orientation score $|W_\psi| \in [0, 1]$ indicates to what extent a line structure with orientation θ is present in the image f at position $\mathbf{x} \in \mathbb{R}^2$. The wavelet $\psi \in \mathbb{L}_2(\mathbb{R}^2) \cap \mathbb{L}_1(\mathbb{R}^2)$ measures the line structure at the origin, and has a horizontal orientation. In order to ensure that all positions and orientations are checked, we use the left-regular representation on \mathbb{R}^2 , given in Eq. (6). This allows us to calculate the alignment for all locations and all orientations. The explicit formulation can be found in Definition 1.18.

Definition 1.18 (Orientation score). Given an image $f \in \mathbb{L}_2(\mathbb{R}^2; [0, 1])$, its orientation score $W_\psi f : \mathbb{L}_2(\mathbb{R}^2) \rightarrow \mathbb{C}_K^{SE(2)}$ is given by

$$U_f(\mathbf{x}, \theta) := (W_\psi f)(\mathbf{x}, \theta) := (\mathcal{U}_{(\mathbf{x}, R_\theta)} \psi, f)_{\mathbb{L}_2(\mathbb{R}^2)} = \int_{\mathbb{R}^2} \psi(R_\theta^{-1}(\mathbf{y} - \mathbf{x})) f(\mathbf{y}) d\mathbf{y},$$

where $\psi \in \mathbb{L}_2(\mathbb{R}^2) \cap \mathbb{L}_1(\mathbb{R}^2)$ denotes a wavelet measuring a line structure at the origin with a horizontal orientation.

More information about the orientation score transform and its invertibility can be found in [18; 19; 20]. One can also find more information about different options for wavelets ψ here. We will be using cake wavelets ψ . They are called that way, since they resemble pieces of cake in the Fourier domain, as you can see in Figure 10.

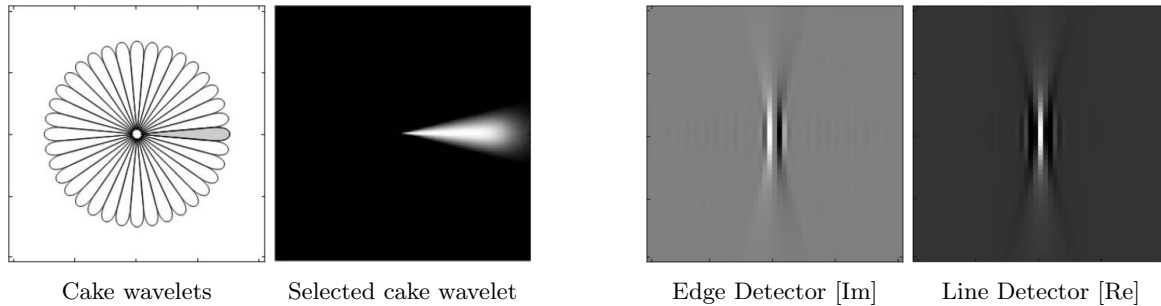


Figure 10: Visualization of the 2D cake-wavelets ψ . The edge and line detector are found after taking the inverse Fourier transform. Taken from [20].

1.1.5 Vessel Tracking

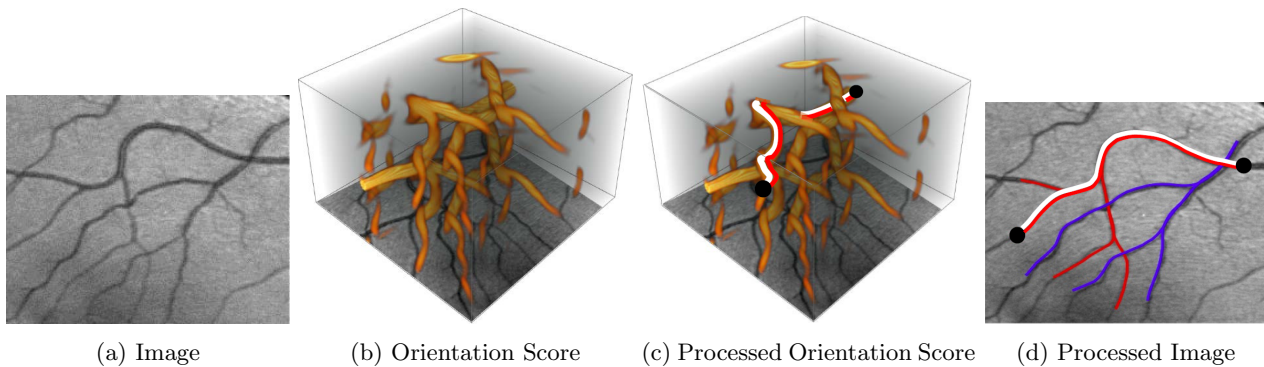
In the introduction, we already explained that in this report, we will be focusing on improving vessel tracking on images in \mathbb{R}^2 . For every tracking that is performed, starting points (seeds) and end points (tips) are provided. The vessel tracking connects every tip to a seed. In order to do so, the image goes through different steps. This section is specifically written for 2D images, but the theory also applies to higher dimensional images. Below, we will explain the different steps that the image goes through when it is processed. For every step, Figure 11 provides the corresponding visualization on an image.

1. We consider an image f in \mathbb{R}^2 . In our example, we consider the image in Figure 11a. This image satisfies the description of images given in Section 1.1.3.
2. We calculate the orientation score $W_\psi f$ in \mathbb{M}_2 , as explained in Section 1.1.4. Doing this allows us to identify the differences between crossings and bifurcation because it “lifts” the image. An extra (angular) dimension is added and therefore, we can recognize differences in orientations, as we can see in Figure 2. Continuing our example, the orientation scores are visualized in Figure 11b.
3. We process the image, such that we find a curve $\tilde{\gamma}$ in \mathbb{M}_2 connecting a tip to a seed. In Section 3, we will explain in full detail how these curves are calculated. As we can see in Figure 11c, this curve $\tilde{\gamma}$ follows the vessel structure in the space of positions and orientations.
4. The curves on the initial image are situated in \mathbb{R}^2 . Therefore, we project the curve $\tilde{\gamma}$ onto \mathbb{R}^2 , such that we find the curve γ . The projection $P : \mathbb{M}_2 \rightarrow \mathbb{R}^2$ boils down to only considering the positional data of the curve, so if $\tilde{\gamma}(t) := (\mathbf{x}(t), \theta(t))$, then $P\tilde{\gamma}(t) := \gamma(t) = \mathbf{x}(t)$, where $\mathbf{x}(t) \in \mathbb{R}^2$. This curve $\gamma(t)$ represents the tracking from tip to seed in \mathbb{R}^2 , as you can see in Figure 11d.

Schematically, this would be given by

$$\begin{array}{ccc} \text{Image } f \text{ in } \mathbb{R}^2 & \xrightarrow{W_\psi} & \text{Orientation Score } W_\psi f \text{ in } \mathbb{M}_2 \\ \Upsilon_\psi \downarrow & & \downarrow \Phi \\ \text{Curve } \gamma \text{ in } \mathbb{R}^2 & \xleftarrow{P} & \text{Curve } \tilde{\gamma} \text{ in } \mathbb{M}_2 \end{array}$$

Hence, we know that the operator Υ_ψ can be defined as $\Upsilon_\psi f := P \circ \Phi \circ W_\psi f$.



(a) Image (b) Orientation Score (c) Processed Orientation Score (d) Processed Image

Figure 11: Example of vessel-tracking in a 2D image via orientation scores. Taken from [20].

1.1.6 (Pseudo)distances

When we process the image, our aim is to find the shortest distance between a tip and (one of) the seeds. In this section, we will shed light over how these distances are defined and how they are computed.

In this example, we will be considering a Riemannian metric. The Riemannian metric tensor field is a family of innerproducts $\mathcal{G}_{\mathbf{p}} : T_{\mathbf{p}}(\mathbb{M}) \times T_{\mathbf{p}}(\mathbb{M}) \rightarrow \mathbb{R}$ for each $\mathbf{p} \in \mathbb{M}$. Therefore, it is also linear in its entries, so for all $\mathbf{p} \in \mathbb{M}$ and $\dot{\mathbf{p}}_0, \dot{\mathbf{p}}_1 \in T_{\mathbf{p}}(\mathbb{M})$, for all $\lambda \in \mathbb{R}$, we have

$$\mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}}_0 + \dot{\mathbf{p}}_1) = \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}}_0) + \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}}_1) \quad \text{and} \quad \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}_0 + \dot{\mathbf{p}}_1, \dot{\mathbf{p}}) = \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}_0, \dot{\mathbf{p}}) + \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}_1, \dot{\mathbf{p}}); \quad (8a)$$

$$\sqrt{\mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \lambda \dot{\mathbf{p}})} = \sqrt{\lambda \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}})} \quad \text{and} \quad \sqrt{\mathcal{G}_{\mathbf{p}}(\lambda \dot{\mathbf{p}}, \dot{\mathbf{p}})} = \sqrt{\lambda \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}})}. \quad (8b)$$

The distance from a point \mathbf{p} to a point \mathbf{q} is defined by

$$d_{\mathcal{G}}(\mathbf{p}, \mathbf{q}) = \inf \left\{ \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt \mid \gamma \in \Gamma_1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\}, \quad (9)$$

where $\Gamma_1 = \text{Lip}([0, 1], \mathbb{M})$ and where $\dot{\gamma}(t) := \frac{d}{dt} \gamma(t)$.

Remark 1.3. Note that the distance $d_{\mathcal{G}}$ does not have to be finite, continuous or symmetric in its arguments. We do not assume that the metric is smooth on $T(\mathbb{M})$, but instead, we make an assumption on the control sets, which is given in Definition 1.19.

Definition 1.19 (Control set). We define the control set to be given by

$$\mathcal{B}_{\mathcal{G}}(\mathbf{p}) := \left\{ \dot{\mathbf{p}} \in T_{\mathbf{p}}\mathbb{M} \mid \sqrt{\mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}})} \leq 1 \right\}. \quad (10)$$

The sets $\mathcal{B}_{\mathcal{G}}(\mathbf{p})$ are closed and vary continuously with respect to the point $\mathbf{p} \in \mathbb{M}$ in the sense of the Hausdorff distance [21; 22].

In Appendix B, we will explain how these control sets are calculated for the left invariant frame and the gauge frame respectively (which will be explained later on). In addition to that, we will also visualize them. We show that Definition 1.19 is satisfied in the left invariant frame.

Lemma 1.1. The distance $d_{\mathcal{G}}(\mathbf{p}, \mathbf{q})$ does not depend on the choice of (monotonic) parametrization of the curve $\gamma(\cdot)$.

Proof. Let $\gamma \in \text{Lip}([0, 1], \mathbb{M})$ be given. Define $\tilde{\gamma} := \gamma \circ A$ where $A : [0, 1] \rightarrow [0, 1]$ smooth on $[0, 1]$ and monotonic with $A > 0, A(0) = 0, A(1) = 1$. Then, using that \mathcal{G} satisfies Eq. (8b), we have

$$\int_0^1 \sqrt{\mathcal{G}_{\tilde{\gamma}(\tilde{t})} \left(\frac{d}{d\tilde{t}} \tilde{\gamma}(\tilde{t}), \frac{d}{d\tilde{t}} \tilde{\gamma}(\tilde{t}) \right)} d\tilde{t} = \int_0^1 \sqrt{\mathcal{G}_{\gamma(A(\tilde{t}))} \left(\dot{\gamma}(A(\tilde{t})) \frac{d}{d\tilde{t}} A(\tilde{t}), \dot{\gamma}(A(\tilde{t})) \frac{d}{d\tilde{t}} A(\tilde{t}) \right)} d\tilde{t} \quad (11a)$$

$$\stackrel{(8b)}{=} \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} \frac{dt}{d\tilde{t}} d\tilde{t} \quad (11b)$$

$$= \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt, \quad (11c)$$

where we substituted the expression for $\tilde{\gamma}$ in (11a). In (11b), we define $t := A(\tilde{t})$. Then $\frac{dt}{d\tilde{t}} = \frac{d}{d\tilde{t}} A(\tilde{t})$. Using integration by substitution, this expression yields Eq. (11c).

In order to show that the distance is independent of the parametrization, we also need to show that they satisfy the same conditions.

$$\begin{aligned}
 \gamma(0) = \mathbf{p} & \Rightarrow \tilde{\gamma}(0) = \gamma(A(0)) = \gamma(0) = \mathbf{p}; \\
 \gamma(1) = \mathbf{q} & \Rightarrow \tilde{\gamma}(1) = \gamma(A(1)) = \gamma(1) = \mathbf{q}; \\
 \gamma \in \text{Lip}([0, 1], \mathbb{M}) & \Rightarrow \tilde{\gamma} \in \text{Lip}([0, 1], \mathbb{M}).
 \end{aligned} \tag{12}$$

Eq. (12) holds since

$$\begin{aligned}
 \gamma \in \Gamma_1 & \Leftrightarrow \exists K > 0 \forall t_1 > t_2 \in [0, 1] : \gamma(t_1) - \gamma(t_2) \leq K|t_1 - t_2| \\
 & \Leftrightarrow \exists K > 0 \forall s_1 > s_2 \in [0, 1] : \tilde{\gamma}(s_1) - \tilde{\gamma}(s_2) \leq K|A(s_1) - A(s_2)| \\
 & \hspace{15em} = K|A'(\iota)| \cdot |s_1 - s_2| \\
 & \hspace{15em} \leq K_2|s_1 - s_2| \\
 & \Leftrightarrow \tilde{\gamma} \in \Gamma_1,
 \end{aligned} \tag{13}$$

where $A(s) := t$ and $K_2 := K|A'(\iota)|$ for some $\iota \in (s_1, s_2)$. Note that in the derivation above, Eq. (13) holds because of the mean value theorem. And hence, we can conclude that

$$d_{\mathcal{G}}(\mathbf{p}, \mathbf{q}) = \inf \left\{ \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt \mid \gamma \in \Gamma_1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\} \tag{14a}$$

$$= \inf \left\{ \int_0^1 \sqrt{\mathcal{G}_{\tilde{\gamma}(t)}\left(\frac{d}{dt}\tilde{\gamma}(t), \frac{d}{dt}\tilde{\gamma}(t)\right)} dt \mid \tilde{\gamma} \in \Gamma_1, \tilde{\gamma}(0) = \mathbf{p}, \tilde{\gamma}(1) = \mathbf{q} \right\}, \tag{14b}$$

so the distance does not depend on the choice of (monotonic) parametrization of the curve $\gamma(\cdot)$. \square

There are various equivalent expressions for the distance. In Lemma 1.2, we state them and prove that they are indeed equivalent.

Lemma 1.2. The following expressions for the distance $d_{\mathcal{G}}$ are equivalent:

$$d_{\mathcal{G}}(\mathbf{p}, \mathbf{q}) = \inf \left\{ \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt \mid \gamma \in \Gamma_1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\} \tag{15a}$$

$$= \inf \left\{ \left(\int_0^1 \left| \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} \right|^p dt \right)^{1/p} \mid \gamma \in \Gamma_1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\} \tag{15b}$$

$$= \inf \left\{ T \geq 0 \mid \exists \gamma \in \Gamma_T, \gamma(0) = \mathbf{p}, \gamma(T) = \mathbf{q}, \forall t \in [0, T] \dot{\gamma}(t) \in \mathcal{B}_{\mathcal{G}}(\gamma(t)) \right\}, \tag{15c}$$

where $\Gamma_T := \text{Lip}([0, T], \mathbb{M})$, and with $\mathcal{B}_{\mathcal{G}}(\gamma(t))$ as defined in (10).

Proof. We begin showing Eq. (15a) \Rightarrow Eq. (15b).

$$\int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt = \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t)) \cdot 1} dt \tag{16a}$$

$$\leq \left(\int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}^p dt \right)^{1/p} \cdot \|1\|_{\mathbb{L}_q([0,1])} \quad (\text{H\"older's inequality}) \tag{16b}$$

$$= \left(\int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}^p dt \right)^{1/p} \tag{16c}$$

Note that the inequality in Eq. (16b) is an equality when $p = 1$. We know that there exists a minimizer γ^* such that $\sqrt{\mathcal{G}_{\gamma^*(t)}(\dot{\gamma}^*(t), \dot{\gamma}^*(t))} = C \in \mathbb{R}$. Then, we have

$$\left(\int_0^1 \sqrt{\mathcal{G}_{\gamma^*(t)}(\dot{\gamma}^*(t), \dot{\gamma}^*(t))}^p dt \right)^{1/p} = (C^p)^{1/p} = C.$$

That means that we find

$$\begin{aligned}
 & \inf \left\{ \int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt \mid \gamma \in \Gamma_1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\} \\
 & = \inf \left\{ \left(\int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}^p dt \right)^{1/p} \mid \gamma \in \Gamma_1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\}.
 \end{aligned}$$

Next, we show that Eq. (15b) is equal to Eq. (15c). Since we know that Eq. (15b) holds for all p , we can consider

$$\liminf_{p \rightarrow \infty} \left\{ \left(\int_0^1 \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))}^p dt \right)^{1/p} \middle| \gamma \in \Gamma_1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\} \quad (17a)$$

$$= \inf \left\{ \left\| \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} \right\|_{L^\infty([0,1])} \middle| \gamma \in \Gamma_1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\} \quad (17b)$$

$$= \inf \left\{ T \geq 0 \middle| \exists \gamma \in \Gamma_1, \forall t \in [0, 1] : \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} \leq T, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\} \quad (17c)$$

$$= \inf \left\{ T \geq 0 \middle| \exists \gamma \in \Gamma_1, \forall t \in [0, 1] : \sqrt{\mathcal{G}_{\gamma(t)}(T^{-1}\dot{\gamma}(t), T^{-1}\dot{\gamma}(t))} \leq 1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\} \quad (17d)$$

$$= \inf \left\{ T \geq 0 \middle| \exists \tilde{\gamma} \in \Gamma_T, \forall \tilde{t} \in [0, T] : \sqrt{\mathcal{G}_{\tilde{\gamma}(\tilde{t})}(\frac{d}{d\tilde{t}}\tilde{\gamma}(\tilde{t}), \frac{d}{d\tilde{t}}\tilde{\gamma}(\tilde{t}))} \leq 1, \tilde{\gamma}(0) = \mathbf{p}, \tilde{\gamma}(T) = \mathbf{q} \right\} \quad (17e)$$

$$= \inf \left\{ T \geq 0 \middle| \exists \tilde{\gamma} \in \Gamma_T, \tilde{\gamma}(0) = \mathbf{p}, \tilde{\gamma}(T) = \mathbf{q}, \forall \tilde{t} \in [0, T] : \frac{d}{d\tilde{t}}\tilde{\gamma}(\tilde{t}) \in \mathcal{B}_{\mathcal{G}}(\tilde{\gamma}(\tilde{t})) \right\}. \quad (17f)$$

Here, Eq. (17d) holds because of the 1-homogeneity condition given in Eq. (8b). In addition to that, we define $\tilde{\gamma}$ such that $\tilde{\gamma}(\tilde{t}) := \gamma(t)$ for $\tilde{t} = Tt$. Consequently, we find that $\frac{d}{d\tilde{t}}\tilde{\gamma}(\tilde{t}) = T^{-1}\dot{\gamma}(t)$. Using the results of Lemma 1.1, we find that

$$\begin{aligned} & \inf \left\{ \left(\int_0^1 \left| \sqrt{\mathcal{G}_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} \right|^p dt \right)^{1/p} \middle| \gamma \in \Gamma_1, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \right\} \\ &= \inf \left\{ T \geq 0 \middle| \exists \gamma \in \Gamma_T, \gamma(0) = \mathbf{p}, \gamma(T) = \mathbf{q}, \forall t \in [0, T] \dot{\gamma}(t) \in \mathcal{B}_{\mathcal{G}}(\gamma(t)) \right\}. \end{aligned}$$

□

Last, we will show what the norm looks like for a specific metric tensor field. In order to do so, we introduce the definition for a metric tensor field.

Definition 1.20 (Metric tensor field). A metric tensor field \mathcal{G} is given by

$$\mathcal{G} = \sum_{i=1}^3 a_{ii}^2 \varrho^i \otimes \varrho^i, \quad (18)$$

where \mathcal{D} and ϱ^i for $i = 1, 2, 3$ represent the frame and its dual respectively. In this case, the coefficients $a_{ii} > 0$ represent the weight of the different directions.

Suppose we need to calculate $\|\text{grad } V\|_{\mathcal{G}}$. Then, using Remark 13 from [23], we have

$$\|\text{grad } V\|_{\mathcal{G}} = \sqrt{\mathcal{G}_{\mathbf{p}}(\text{grad } V(\mathbf{p}), \text{grad } V(\mathbf{p}))} = \sqrt{\mathcal{G}_{\mathbf{p}}(\mathcal{G}_{\mathbf{p}}^{-1}dV(\mathbf{p}), \mathcal{G}_{\mathbf{p}}^{-1}dV(\mathbf{p}))}, \quad (19)$$

where $\mathcal{G}_{\mathbf{p}}$ denotes an inner product defined on $T_{\mathbf{p}}(\mathbb{M}_2)$ where $\mathbf{p} \in \mathbb{M}_2$ such that

$$(\dot{\mathbf{p}}_1, \dot{\mathbf{p}}_2) \mapsto \mathcal{G}_{\mathbf{p}}(\dot{\mathbf{p}}_1, \dot{\mathbf{p}}_2) \text{ with } \dot{\mathbf{p}}_1, \dot{\mathbf{p}}_2 \in T_{\mathbf{p}}(\mathbb{M}_2),$$

and where $\mathcal{G}_{\mathbf{p}}^{-1}$ inverts the action of $\mathcal{G}_{\mathbf{p}}$.

We are interested to know $\mathcal{G}_{\mathbf{p}}^{-1}dV(\mathbf{p})$. Let us denote

$$dV(\mathbf{p}) = (\mathcal{D}_1|_{\mathbf{p}} V) \varrho^1 + (\mathcal{D}_2|_{\mathbf{p}} V) \varrho^2 + (\mathcal{D}_3|_{\mathbf{p}} V) \varrho^3,$$

where ϱ^i ($i = 1, 2, 3$) form the covector basis and $\mathcal{D}_i|_{\mathbf{p}} V$ ($i = 1, 2, 3$) are the coefficients in \mathbb{R} . From tensor calculus, we recall that $\varrho^i(\mathcal{D}_j) = \delta_j^i$ (result 2.1 from [24]). By the linearity of $\mathcal{G}_{\mathbf{p}}^{-1}$, we find

$$\mathcal{G}_{\mathbf{p}}^{-1} \left((\mathcal{D}_1|_{\mathbf{p}} V) \varrho^1 + (\mathcal{D}_2|_{\mathbf{p}} V) \varrho^2 + (\mathcal{D}_3|_{\mathbf{p}} V) \varrho^3 \right) = (\mathcal{D}_1|_{\mathbf{p}} V) \mathcal{G}_{\mathbf{p}}^{-1}(\varrho^1) + (\mathcal{D}_2|_{\mathbf{p}} V) \mathcal{G}_{\mathbf{p}}^{-1}(\varrho^2) + (\mathcal{D}_3|_{\mathbf{p}} V) \mathcal{G}_{\mathbf{p}}^{-1}(\varrho^3). \quad (20)$$

In order to find $\mathcal{G}_{\mathbf{p}}^{-1}(\varrho^i)$, we need to find $\mathcal{G}_{\mathbf{p}}(\mathcal{G}_{\mathbf{p}}^{-1}\varrho^i, \mathcal{D}_j|_{\mathbf{p}}) = \delta_j^i$. We know that $\mathcal{G}_{\mathbf{p}}$ is linear as well. Then, we have, for all $b^1, b^2, b^3 \in \mathbb{R}$:

$$\begin{aligned} \mathcal{G}_{\mathbf{p}} \left(\mathcal{G}_{\mathbf{p}}^{-1}\varrho^i, \sum_{j=1}^3 b^j \mathcal{D}_j|_{\mathbf{p}} \right) &= \sum_{j=1}^3 b^j \mathcal{G}_{\mathbf{p}}(\mathcal{G}_{\mathbf{p}}^{-1}\varrho^i, \mathcal{D}_j|_{\mathbf{p}}) \\ &= b^i a_{ii}^2 \varrho^i(\mathcal{G}_{\mathbf{p}}^{-1}\varrho^i) = b^i \\ \Leftrightarrow \varrho^i(\mathcal{G}_{\mathbf{p}}^{-1}\varrho^i) &= \frac{1}{a_{ii}^2} \Rightarrow \mathcal{G}_{\mathbf{p}}^{-1}\varrho^i = \frac{1}{a_{ii}^2} \mathcal{D}_i, \end{aligned}$$

where we use $\varrho^k(\mathcal{D}_j|_{\mathbf{p}}) = \delta_j^k$ and $\varrho^k(\mathcal{G}_{\mathbf{p}}^{-1}\varrho^i) = C\delta_i^k$ for some $C \in \mathbb{R}$. Consequently, the norm is given by

$$\begin{aligned} & \mathcal{G}_{\mathbf{p}}(\mathcal{G}_{\mathbf{p}}^{-1}((\mathcal{D}_1|_{\mathbf{p}}V)\varrho^1 + (\mathcal{D}_2|_{\mathbf{p}}V)\varrho^2 + (\mathcal{D}_3|_{\mathbf{p}}V)\varrho^3), \mathcal{G}_{\mathbf{p}}^{-1}((\mathcal{D}_1|_{\mathbf{p}}V)\varrho^1 + (\mathcal{D}_2|_{\mathbf{p}}V)\varrho^2 + (\mathcal{D}_3|_{\mathbf{p}}V)\varrho^3)) \\ &= \sum_{i=1}^3 \mathcal{G}_{\mathbf{p}}\left(\frac{1}{a_{ii}^2} \mathcal{D}_i|_{\mathbf{p}}V \cdot \mathcal{D}_i|_{\mathbf{p}}, \frac{1}{a_{ii}^2} \mathcal{D}_i|_{\mathbf{p}}V \cdot \mathcal{D}_i|_{\mathbf{p}}\right) \\ &= \sum_{i=1}^3 \frac{1}{a_{ii}^2} (\mathcal{D}_i|_{\mathbf{p}}V)^2. \end{aligned} \quad (21)$$

1.2 Static Frame

The first frame that we consider is the static frame denoted by $\{\partial_x, \partial_y, \partial_\theta\}$. This frame is induced by the coordinates x, y, θ for all points in \mathbb{M}_2 . Its dual frame, for the cotangent bundle $T^*\mathbb{M}_2$ is denoted by $\{dx, dy, d\theta\}$, where the symbol d denotes the exterior derivative.

1.3 Left Invariant Frame

One of the frames that we will be considering in this thesis, is the left invariant frame. The left invariant frame is a roto-translation of the pushforward of the static frame at the origin. We define the pushforward $(L_g)_* : T_h(G) \rightarrow T_{gh}(G)$ by

$$(L_g)_* \partial_x|_{\mathbf{e}} U = \partial_x|_{\mathbf{e}} (U \circ L_g), \quad (22)$$

where $\mathbf{e} = (0, 0, 0)$. Then, we define the left invariant frame $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ by

$$\mathcal{A}_i|_{(x,y,\theta)} = (L_{(x,y,\theta)})_* \mathcal{A}_i|_{\mathbf{e}} \quad (23a)$$

$$\mathcal{A}_1|_{\mathbf{e}} U = \partial_x U(\mathbf{e}) \quad (23b)$$

$$\mathcal{A}_2|_{\mathbf{e}} U = \partial_y U(\mathbf{e}) \quad (23c)$$

$$\mathcal{A}_3|_{\mathbf{e}} U = \partial_\theta U(\mathbf{e}). \quad (23d)$$

We denote the dual of the left invariant frame by $\{\omega^1, \omega^2, \omega^3\}$ such that $\langle \omega^i, \mathcal{A}_j \rangle = \delta_j^i$. We notice that we can calculate the left invariant frame at any point $gh \in SE(2)$, when we know the left invariant frame at the point $h \in SE(2)$, i.e. the frame is roto-translational invariant:

$$\mathcal{A}_i|_{gh} U = (L_{gh})_* \mathcal{A}_i|_{\mathbf{e}} U = \mathcal{A}_i|_{\mathbf{e}} (U \circ L_{gh}) = \mathcal{A}_i|_h (U \circ L_g) = \mathcal{A}_i|_{\mathbf{e}} (U \circ L_g \circ L_h) = (L_h)_* \mathcal{A}_i|_{\mathbf{e}} U = (L_g)_* \mathcal{A}_i|_h U. \quad (24)$$

That means that the left invariant frame at location $\mathbf{x} = (x, y)$ and orientation θ is given by

$$\begin{cases} \mathcal{A}_1|_{(x,y,\theta)} U = \cos \theta \partial_x U(x, y, \theta) + \sin \theta \partial_y U(x, y, \theta) \\ \mathcal{A}_2|_{(x,y,\theta)} U = -\sin \theta \partial_x U(x, y, \theta) + \cos \theta \partial_y U(x, y, \theta) \\ \mathcal{A}_3|_{(x,y,\theta)} U = \partial_\theta U(x, y, \theta). \end{cases} \quad (25)$$

Remark 1.4. We notice that the left invariant frame is a roto-translation of the static frame at the origin \mathbf{e} . Its base point is located at (x, y, θ) instead of $(0, 0, 0)$. In addition to that, it is rotated with the angle θ in the position space. In Figure 3 and Figure 12, a visualization of the frame can be found. The vector \mathcal{A}_1 lies parallel to line structures in the image. The vector \mathcal{A}_2 is orthogonal to the vector \mathcal{A}_1 , in the xy -plane. Therefore, we say that \mathcal{A}_1 points forward, and \mathcal{A}_2 points to the left. Therefore, when we do not allow backward movement in the frame, we ensure that $\mathcal{A}_1|_{(x,y,\theta)} \geq 0$. In practice this is done by replacing the constant $\mathcal{A}_1|_{(x,y,\theta)} W \in \mathbb{R}$ that indicates the length of the vector $\mathcal{A}_1|_{(x,y,\theta)}$ by $\max\{0, \mathcal{A}_1|_{(x,y,\theta)} W\}$ [17].

The frame introduced in Eq. (25) is left invariant, since it satisfies

$$\mathcal{A}_i \circ \mathcal{L}_g = \mathcal{L}_g \circ \mathcal{A}_i \text{ for all } g \in G. \quad (26)$$

For more information and the proof, see [25; 23].

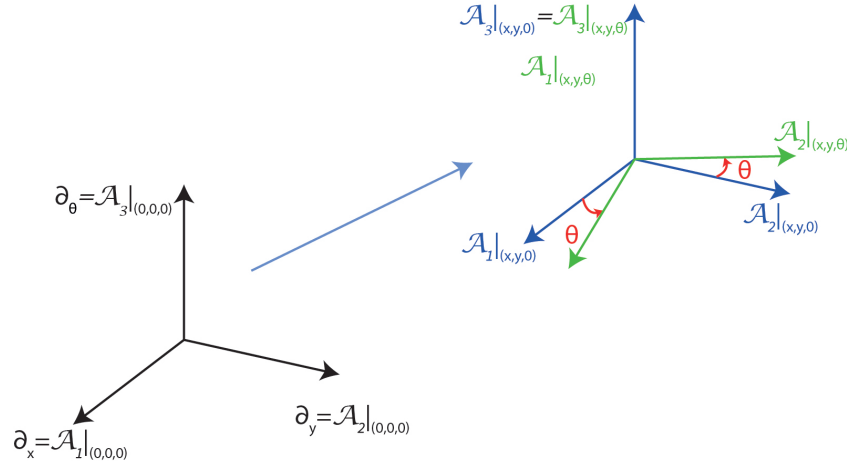


Figure 12: Visualization of left invariant frame. In black, we see the static frame. This frame is translated to location $\mathbf{x} = (x, y)$ (in blue). Then, it is rotated with an angle θ , which results in the green left invariant frame.

Exponential curves in $\mathbf{SE}(2)$ Next, we discuss the exponential curves $g \mapsto ge^{tA}$. These are the characteristics of the left invariant flow. First, we need to introduce the definition of generators of group representations, in specific of group representation \mathcal{R} .

Definition 1.21 (Generator of Group Representation \mathcal{R}). Let G be a Lie group. We consider the right regular representation $\mathcal{R} : G \rightarrow B(\mathbb{L}_2(G))$ defined in Definition 1.13. Let $A \in T_e(G)$, where $e \in G$ unity element. Then, the generator/derivative $d\mathcal{R}$ is defined as follows:

$$d\mathcal{R}(A)U(\mathbf{g}) := \lim_{t \rightarrow 0} \frac{(\mathcal{R}_{e^{tA}}U)(\mathbf{g}) - U(\mathbf{g})}{t} = \lim_{t \rightarrow 0} \frac{U(\mathbf{g}e^{tA}) - U(\mathbf{g})}{t}$$

for all $U \in \mathcal{D}_{\mathbb{L}_2(G)}$ where $\mathcal{D}_{\mathbb{L}_2(G)}$ is a domain of sufficiently regular $U \in \mathbb{L}_2(G)$ such that the limit $d\mathcal{R}(A)U$ exists in $\mathbb{L}_2(G)$.

Algebraically, the exponential curves are characterized by

$$e^{td\mathcal{R}(A)} = \mathcal{R}_{e^{tA}}.$$

Note that when we consider $W(\cdot, t) = e^{tQ}U(\cdot)$ for some Q , we find

$$\partial_t W(\cdot, t) = QW(\cdot, t) \quad \text{and} \quad W(\cdot, t)|_{t=0} = U(\cdot).$$

In this case, we have $U = \mathbf{e}$ and $Q = d\mathcal{R}(A)$.

Lemma 1.3. The following expressions are equivalent:

$$\mathcal{A}_i|_{\mathbf{g}} = (L_{\mathbf{g}})_* \mathcal{A}_i|_{\mathbf{e}} = d\mathcal{R}(\mathcal{A}_i|_{\mathbf{e}})$$

Proof. The first equivalence was already defined in Eq. (23a). That leaves us to prove that $(L_{\mathbf{g}})_* \mathcal{A}_i|_{\mathbf{e}} = d\mathcal{R}(\mathcal{A}_i|_{\mathbf{e}})$. First, we note that, for some function U and some point $\mathbf{p} = (\mathbf{x}, \theta)$, we have

$$(\mathcal{R}_{\mathbf{g}} \circ \mathcal{L}_{\mathbf{q}})U(\mathbf{p}) = U(\mathbf{q}^{-1}\mathbf{p}\mathbf{g}) = (\mathcal{L}_{\mathbf{q}} \circ \mathcal{R}_{\mathbf{g}})U(\mathbf{p}).$$

In addition to that, we find

$$\begin{aligned} (L_{\mathbf{q}})_* \mathcal{A}_i|_{\mathbf{e}} W &= (L_{\mathbf{q}})_* \partial_i W = \partial_i (W \circ L_{\mathbf{q}}) \\ &= \lim_{t \downarrow 0} \frac{W(\mathbf{q} \cdot (t\mathbf{e}_i)) - W(\mathbf{q})}{t} = \dots = \mathcal{A}_i|_{\mathbf{q}} \\ &= \lim_{t \downarrow 0} t^{-1} (W(\mathbf{q} \cdot e^{t\mathcal{A}_i|_{\mathbf{e}}}) - W(\mathbf{q})) \\ &= \lim_{t \downarrow 0} t^{-1} (\mathcal{R}_{e^{t\mathcal{A}_i|_{\mathbf{e}}}} - I) W(\mathbf{q}) = d\mathcal{R}(\mathcal{A}_i|_{\mathbf{e}}) W(\mathbf{q}), \end{aligned}$$

where \mathbf{e}_i is the i th unit vector and we use that

$$e^{t\mathcal{A}_i|_{\mathbf{e}}} = t\mathbf{e}_i.$$

□

Rewriting this expression for exponential curves, we find

$$\begin{cases} \frac{\partial W}{\partial t}(\mathbf{g}, t) = \sum_{i=1}^3 c^i d\mathcal{R}(\mathcal{A}|_{\mathbf{e}})W(\mathbf{g}, t) = (c^1(\cos\theta\partial_x + \sin\theta\partial_y) + c^2(-\sin\theta\partial_x + \cos\theta\partial_y) + c^3\partial_\theta)W(\mathbf{g}, t) \\ W(\cdot, t)|_{t=0} = \mathbf{e} \end{cases} \quad (27)$$

$$\Leftrightarrow \begin{cases} \frac{d}{dt}W(x(t), y(t), \theta(t)) = \frac{\partial W}{\partial x}|_{\gamma(t)}\dot{x}(t) + \frac{\partial W}{\partial y}|_{\gamma(t)}\dot{y}(t) + \frac{\partial W}{\partial \theta}|_{\gamma(t)}\dot{\theta}(t). \\ W(\cdot, 0) = \mathbf{e}. \end{cases} \quad (28)$$

Hence, the system can be rewritten to

$$\begin{cases} \dot{x} = c^1 \cos\theta - c^2 \sin\theta, & x(0) = 0; \\ \dot{y} = c^1 \sin\theta + c^2 \cos\theta, & y(0) = 0; \\ \dot{\theta} = c^3, & \theta(0) = 0. \end{cases} \quad (29)$$

Solving this system of ordinary, first order, equations results in

$$\begin{cases} x(t) = \frac{c^1}{c^3} \sin(c^3 t) + \frac{c^2}{c^3} (1 - \cos(c^3 t)) \\ y(t) = \frac{c^1}{c^3} (1 - \cos(c^3 t)) + \frac{c^2}{c^3} \sin(c^3 t) \\ \theta(t) = c^3 t \end{cases} \quad \text{if } c^3 \neq 0; \quad (30)$$

$$\begin{cases} x(t) = c^1 t \\ y(t) = c^2 t \\ \theta(t) = 0 \end{cases} \quad \text{otherwise.} \quad (31)$$

Lie Brackets Last, we introduce general concept of Lie brackets, and how they apply to the left invariant frame. In addition to that, we also calculate the structure constants.

Definition 1.22 (Lie bracket). Consider a Lie algebra \mathfrak{g} which is a vector space over a field \mathcal{F} equipped with a binary operation $[\cdot, \cdot]: \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$, which is defined such that for all $V, W \in \mathfrak{g}$ holds

$$[V, W] := V \circ W - W \circ V. \quad (32)$$

The binary operation given in Eq. (32) is called the Lie bracket of V and W , where we compose vector fields as

$$((V \circ W)f)(g) := V|_g(h \mapsto W|_h f).$$

The Lie brackets have a set of properties.

Proposition 1.2 (Properties Lie brackets). A Lie algebra \mathfrak{g} is a vector space over a field \mathcal{F} equipped with a binary operation (the so-called Lie bracket) $[\cdot, \cdot]: \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g}$ such that

$$\forall \alpha, \beta \in \mathcal{F} \quad \forall X, Y \in \mathfrak{g}: \quad [\alpha X + \beta Y, Z] = \alpha[X, Z] + \beta[Y, Z] \quad (33a)$$

$$\forall \alpha, \beta \in \mathcal{F} \quad \forall X, Y \in \mathfrak{g}: \quad [Z, \alpha X + \beta Y] = \alpha[Z, X] + \beta[Z, Y] \quad (33b)$$

$$\forall X \in \mathfrak{g}: \quad [X, X] = 0 \quad (33c)$$

$$\forall X, Y, Z \in \mathfrak{g}: \quad [X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0. \quad (33d)$$

Remark 1.5. A consequence of these properties is that we also have for the Lie bracket:

$$[X, Y] = -[Y, X] \text{ for all } X, Y \in \mathfrak{g}.$$

When we apply these Lie brackets specifically to the left invariant frame, we get

Lemma 1.4 (Lie brackets in the left invariant frame).

$$[\mathcal{A}_1, \mathcal{A}_2] = \mathbf{0}, \quad [\mathcal{A}_1, \mathcal{A}_3] = -\mathcal{A}_2, \quad [\mathcal{A}_2, \mathcal{A}_3] = \mathcal{A}_1.$$

Proof. Let $f \in C^\infty$ be given. Then we find

$$\begin{aligned}
 [\mathcal{A}_1, \mathcal{A}_2]f &= [\cos \theta \partial_x + \sin \theta \partial_y, -\sin \theta \partial_x + \cos \theta \partial_y]f \\
 &= (\cos \theta \partial_x + \sin \theta \partial_y) \circ (-\sin \theta \partial_x + \cos \theta \partial_y) f - (-\sin \theta \partial_x + \cos \theta \partial_y) \circ (\cos \theta \partial_x + \sin \theta \partial_y) f \\
 &= -\cos \theta \sin \theta \partial_{xx} f - \sin^2 \theta \partial_{xy} f + \cos^2 \theta \partial_{xy} f + \cos \theta \sin \theta \partial_{yy} f \\
 &\quad + \sin \theta \cos \theta \partial_{xx} f - \cos^2 \theta \partial_{xy} f + \sin^2 \theta \partial_{xy} f + \cos \theta \sin \theta \partial_{yy} f \\
 &= \mathbf{0}.
 \end{aligned}$$

$$\begin{aligned}
 [\mathcal{A}_1, \mathcal{A}_3]f &= [\cos \theta \partial_x + \sin \theta \partial_y, \partial_\theta]f \\
 &= \cos \theta \partial_{x\theta} f + \sin \theta \partial_{y\theta} f - \partial_\theta (\cos \theta \partial_x + \sin \theta \partial_y) f \\
 &= \cos \theta \partial_{x\theta} f + \sin \theta \partial_{y\theta} f + \sin \theta \partial_x f - \cos \theta \partial_{x\theta} f - \cos \theta \partial_y f - \sin \theta \partial_{y\theta} f \\
 &= \sin \theta \partial_x f - \cos \theta \partial_y f = -\mathcal{A}_2 f.
 \end{aligned}$$

$$\begin{aligned}
 [\mathcal{A}_2, \mathcal{A}_3]f &= [-\sin \theta \partial_x + \cos \theta \partial_y, \partial_\theta]f \\
 &= -\sin \theta \partial_{x\theta} f + \cos \theta \partial_{y\theta} f - \partial_\theta (-\sin \theta \partial_x + \cos \theta \partial_y) f \\
 &= -\sin \theta \partial_{x\theta} f + \cos \theta \partial_{y\theta} f + \cos \theta \partial_x f + \sin \theta \partial_{x\theta} f + \sin \theta \partial_y f - \cos \theta \partial_{y\theta} f \\
 &= \cos \theta \partial_x f + \sin \theta \partial_y f = \mathcal{A}_1 f.
 \end{aligned}$$

□

Definition 1.23 (Structure constants). There exist $c_{ij}^k \in \mathbb{R}$ such that

$$[\mathcal{A}_i, \mathcal{A}_j] = \sum_{k=1}^3 c_{ij}^k \mathcal{A}_k.$$

These constants c_{ij}^k are called the structure constants.

Remark 1.6. The non-zero structure constants of \mathbb{M}_2 are given by

$$\begin{aligned}
 c_{13}^2 &= -1, & c_{23}^1 &= 1, \\
 c_{31}^2 &= 1, & c_{32}^1 &= -1.
 \end{aligned}$$

1.4 Gauge Frame

We have just introduced the left invariant frame. Unfortunately, this frame is not always able to follow the line structures correctly. We recall (from the introduction) that the problems that arise are

1. Deviation of horizontality: The line structures are lifted and extended to the \mathbb{M}_2 domain by calculating the orientation scores. However, there will also be response for slightly different orientations than the orientation of the line structure. In other words, the orientation score contains line structure whose orientation is different from the orientation coordinate in the manifold. Since we use a low angular resolution (low number of orientations), the effects of deviating from the angular direction are bigger. In Figure 4, this problem is visualized.
2. Fixed orientations in \mathbb{M}_2 : The left invariant frame lines up with fixed orientations in the slices of \mathbb{M}_2 . When considering a curved line structure, there should be a smooth transaction between different slices. However, in practice, this is not the case since every slice lines up with exactly one orientation. When the line structure is very curved, it is possible that the orientation moves between multiple slices in a short period of time.

In an effort to avoid these problems, the gauge frame (also called locally adaptive frame) has been introduced. This frame and its dual are denoted by $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ and $\{\omega^1, \omega^2, \omega^3\}$ respectively. The gauge frame is able to align better to the line structure, as we can see in Figure 5.

It is important to note that this frame is fitted to the data, and therefore depending on the specific image that we are considering. It is common that there is some level of noise included in the image (and therefore also in the corresponding data). The effect of the noise is bigger when we are considering the frame at a different location than at one of the line structures. Therefore, it is important to apply some stabilization technique in order to obtain some level of smoothness of the gauge frame.

1.4.1 Construction of the Gauge Frame

First Gauge Vector

Definition 1.24 (First gauge vector). Given a differentiable function $f \in C^2(\mathbb{M}_2)$, $f : \mathbb{M}_2 \rightarrow \mathbb{R}$ with

$$\text{grad} f = \sum_{i=1}^n \xi_i^{-2} (\mathcal{A}_i f) \mathcal{A}_i,$$

the first gauge vector at the point $\mathbf{p} \in \mathbb{M}_2$ is given by the optimization problem

$$\mathcal{B}_1|_{\mathbf{p}} = \arg \min_{\substack{\mathbf{c} \in T_{\mathbf{p}}\mathbb{M}_2 \\ \|\mathbf{c}\|=1}} \left\| \sum_{i,j=1}^n (\xi_j)^{-2} \mathbf{c}^j \mathcal{A}_i \mathcal{A}_j f(\mathbf{p}) \mathcal{A}_i|_{\mathbf{p}} \right\| \quad (34)$$

where the norm is induced by a metric tensor field \mathcal{G} on \mathbb{M}_2 .

Lemma 1.5 (Solving first gauge vector). The first gauge vector is a coordinate vector $\mathbf{c} = (c^1, c^2, c^3)^\top \in \mathbb{R}^3$. It is the eigenvector of the matrix $M^2 H^\top M^2 H$ with the smallest eigenvalue and it has length $\|M^{-1}\mathbf{c}\|_{\mathbb{L}_2} = 1$. The matrices M and H are given by

$$M = \text{diag}(1/\xi, 1/\xi, 1) \text{ and } H_{ij} = \mathcal{A}_j \mathcal{A}_i(f)|_{\mathbf{p}}, \quad (35)$$

where i and j denote rows and columns respectively.

The proof of this lemma can be found in [13].

Remaining Gauge Vectors

Definition 1.25 (Remaining gauge vectors). Let the first gauge vector be given by

$$\mathcal{B}_1|_{\mathbf{p}} = \sum_{j=1}^3 c_1^j \mathcal{A}_j|_{\mathbf{p}}. \quad (36)$$

Let $P\mathcal{B}_1|_{\mathbf{p}}$ be the projection of the first gauge frame on the spatial part of $T_{\mathbf{p}}(\mathbb{M}_2)$, i.e. $\text{span}(\mathcal{A}_1, \mathcal{A}_2)$. The curvature of the line structure is expressed by the angle between the first gauge frame vector and its projection

$$\nu_{\mathbf{p}} := \angle(\mathcal{B}_1|_{\mathbf{p}}, P\mathcal{B}_1|_{\mathbf{p}}) = \arctan \left(\frac{c^3}{\sqrt{(c^1)^2 + (c^2)^2}} \right) \Big|_{\mathbf{p}},$$

and the deviation of horizontality is expressed by the angle between the projected first gauge vector and the forward direction of the left invariant frame:

$$\varphi_{\mathbf{p}} := \angle(P\mathcal{B}_1|_{\mathbf{p}}, \mathcal{A}_1|_{\mathbf{p}}) = \arctan \left(\frac{c^2}{c^1} \right) \Big|_{\mathbf{p}}.$$

Using these angles, the left invariant frame is adapted such that the forward vector lines up with the first gauge vector. That means that we do the following corrections to the left invariant frame.

1. Correct for curvature by an anticlockwise rotation over the angle $-\nu_{\mathbf{p}}$ around $\mathcal{A}_3|_{\mathbf{p}}$ in the plane spanned by $\mathcal{A}_1|_{\mathbf{p}}$ and $\mathcal{A}_2|_{\mathbf{p}}$.
2. Correct for deviation of horizontality by an anticlockwise rotation over the angle $-\varphi_{\mathbf{p}}$ around $\mathcal{A}_2|_{\mathbf{p}}$, in the plane spanned by the rotated vector $P\mathcal{B}_1|_{\mathbf{p}}$ and $\mathcal{A}_3|_{\mathbf{p}}$.
3. Normalize the resulting vectors w.r.t. the weighted Euclidean metric field.

The rotations are visualized in Figure 13. The procedure explained in Definition 1.25 gives the following two rotation matrices:

$$R_{\mathbf{e}_2, -\nu_{\mathbf{p}}} = \begin{pmatrix} \cos \nu_{\mathbf{p}} & 0 & -\sin \nu_{\mathbf{p}} \\ 0 & 1 & 0 \\ \sin \nu_{\mathbf{p}} & 0 & \cos \nu_{\mathbf{p}} \end{pmatrix}, \quad R_{\mathbf{e}_3, -\varphi_{\mathbf{p}}} = \begin{pmatrix} \cos \varphi_{\mathbf{p}} & -\sin \varphi_{\mathbf{p}} & 0 \\ \sin \varphi_{\mathbf{p}} & \cos \varphi_{\mathbf{p}} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The constants $c_2^1(\mathbf{p})$, $c_2^2(\mathbf{p})$, $c_2^3(\mathbf{p})$ and $c_3^1(\mathbf{p})$, $c_3^2(\mathbf{p})$, $c_3^3(\mathbf{p})$ belong to $\mathcal{B}_2|_{\mathbf{p}}$ and $\mathcal{B}_3|_{\mathbf{p}}$ respectively, and they are given by

$$\begin{pmatrix} c_2^1(\mathbf{p}) \\ c_2^2(\mathbf{p}) \\ c_2^3(\mathbf{p}) \end{pmatrix} := R_{\mathbf{e}_3, -\varphi_{\mathbf{p}}} R_{\mathbf{e}_2, -\nu_{\mathbf{p}}} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -\sin \varphi_{\mathbf{p}} \\ \cos \varphi_{\mathbf{p}} \\ 0 \end{pmatrix} \text{ and } \begin{pmatrix} c_3^1(\mathbf{p}) \\ c_3^2(\mathbf{p}) \\ c_3^3(\mathbf{p}) \end{pmatrix} := R_{\mathbf{e}_3, -\varphi_{\mathbf{p}}} R_{\mathbf{e}_2, -\nu_{\mathbf{p}}} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sin \nu_{\mathbf{p}} \cos \varphi_{\mathbf{p}} \\ -\sin \nu_{\mathbf{p}} \sin \varphi_{\mathbf{p}} \\ \cos \nu_{\mathbf{p}} \end{pmatrix}. \quad (37)$$

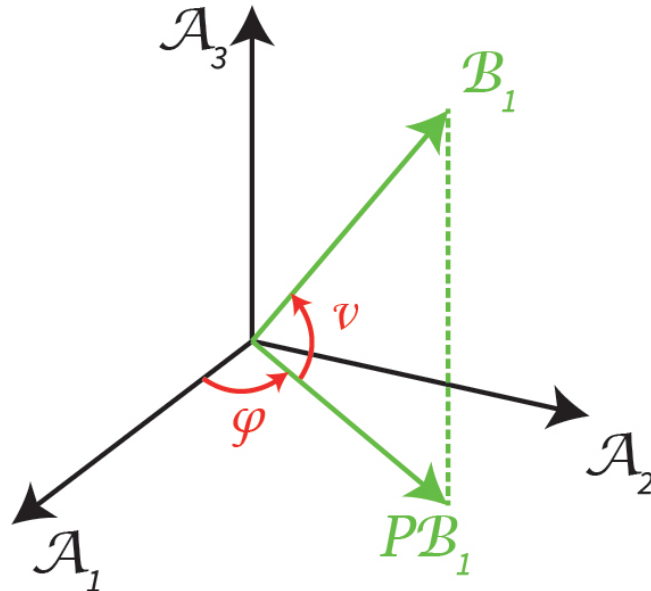


Figure 13: Visualization of the gauge frame compared to the left invariant frame.

Then, the (remaining) gauge vectors are given by

$$\mathcal{B}_i|_{\mathbf{p}} = \frac{\sum_{j=1}^3 c_i^j(\mathbf{p}) \mathcal{A}_j|_{\mathbf{p}}}{\left\| \sum_{j=1}^3 c_i^j(\mathbf{p}) \mathcal{A}_j|_{\mathbf{p}} \right\|}, \quad (38)$$

where $i = 2, 3$ and the norm is induced by a metric tensor field on \mathbb{M}_2 .

1.4.2 External Regularization

At the beginning of this section, we already mentioned that the gauge frame is not smooth. One way to smooth it, is by using external regularization. That means that the hessian is replaced by a locally averaged hessian such that the first gauge vectors are fitted, taking into account their surroundings.

Definition 1.26 (External regularization). Let the scaling matrix M and the hessian matrix $H|_{\mathbf{x}}$ be given as in Eq. (35). Consider the component-wise Gaussian smoothed Hessian given by

$$\tilde{H}|_{(\mathbf{x}, \theta)} := \int_{\mathbb{R}^2} \int_{S^1} G_{\rho_s}^{\mathbb{R}^2}(\mathbf{x} - \mathbf{y}) G_{\rho_a}^{S^1}(\theta - \phi) H|_{(\mathbf{y}, \phi)} d\sigma(\phi) d\mathbf{y}, \quad (39)$$

where σ is the normalized Borel measure on the sphere S^1 . The smoothing kernels G_{ρ}^M are given by

$$G_{\rho_s}^{\mathbb{R}^2}(\mathbf{x}) = \frac{1}{4\pi\rho_s} e^{-\frac{\|\mathbf{x}\|^2}{4\rho_s}},$$

$$G_{\rho_a}^{S^1}(\theta) = \frac{1}{2\sqrt{\pi\rho_a}} e^{-\frac{|\theta-a|^2}{4\rho_a}},$$

with $\rho_s, \rho_a > 0$, $a \in S^1$. For more details, see Section 2.7 of [16].

The externally regularized first gauge vector is the eigenvector of the smoothed matrix $M^2 \tilde{H}^\top M^2 \tilde{H}$ with the smallest eigenvalue.

1.4.3 Iterated Stabilization

Besides external regularization, the gauge frames can also be smoothed by iteratively fitting the gauge frames. The first time, the gauge frame is fitted to the orientation score. After the first iteration; we cannot use the orientation score again. Therefore, we introduce, the orientation confidence. This measure indicates the alignment of the first gauge vectors $\mathcal{B}_1|_{\mathbf{p}}$ to the underlying line structures.

Definition 1.27 (Orientation confidence). Given a gauge frame $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ fitted to an orientation score $V : \mathbb{M}_2 \rightarrow \mathbb{R}$, consider the Laplacian in the plane orthogonal to the principal direction $\mathcal{B}_1|_{\mathbf{p}}$:

$$(\Delta_{23}V)(\mathbf{p}) := (\mathcal{B}_2\mathcal{B}_2V)(\mathbf{p}) + (\mathcal{B}_3\mathcal{B}_3V)(\mathbf{p}). \quad (40)$$

The orientation confidence $S(V) : \mathbb{M}_2 \rightarrow [0, 1]$ is defined as

$$S(V)(\mathbf{p}) := \frac{(-(\Delta_{23}V)(\mathbf{p}))_+}{\|(-(\Delta_{23}V))_+\|_\infty}, \quad (41)$$

where $(a)_+ := \max\{0, a\}$.

To clarify this definition, we visualize an example as given in Figure 14. The vector $\mathcal{B}_1|_{\mathbf{p}}$ lines up with the line structure at point \mathbf{p} . Consequently, $\mathcal{B}_2|_{\mathbf{p}}$ and $\mathcal{B}_3|_{\mathbf{p}}$ are orthogonal to the line structure in the image. The presence of the curvature (in this case a bump created by the ridge) is measured by the negative Laplacian $-\Delta_{23}$. In order to understand why we only consider the positive part of $-\Delta_{23}$, we consider two parallel line structures as in Figure 15. The vector $\mathcal{B}_1|_{\mathbf{p}}$ aligns with its surrounding line structures for a point \mathbf{p} between the parallel lines. Consequently, a negative response to the negative Laplacian arises. This response does not correspond to a genuine line structure, and therefore we only consider the positive part $(-\Delta_{23})_+$. Last, we divide by the maximum to ensure that its values are in the interval $[0, 1]$.

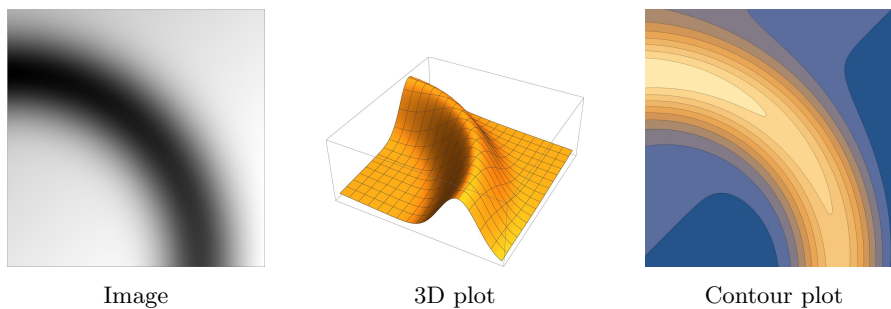


Figure 14: Visualization of a ridge. Taken from [13].

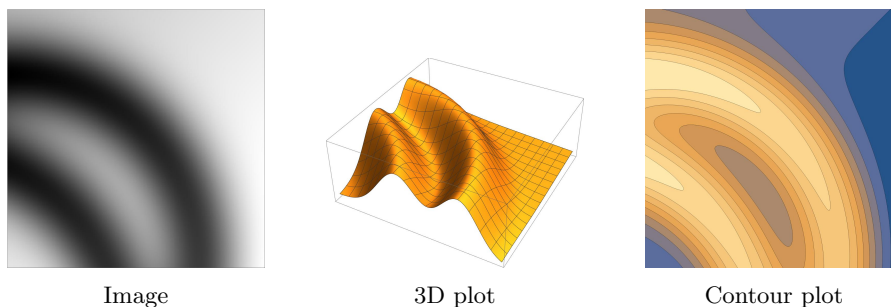


Figure 15: Two parallel line structures.

Remark 1.7. The calculation of $(\mathcal{B}_i\mathcal{B}_iV)(\mathbf{p})$ for $i = 1, 2, 3$ is fairly easy for an orientation score $V : \mathbb{M}_2 \rightarrow \mathbb{R}$. We know how to calculate the Hessian $H|_{\mathbf{p}}$ with respect to the left invariant frame as defined in Eq. (35). Then for $\mathcal{B}_i|_{\mathbf{p}} = \sum_{j=1}^3 c_i^j(\mathbf{p}) \mathcal{A}_j|_{\mathbf{p}}$, we have

$$(\mathcal{B}_i\mathcal{B}_iV)(\mathbf{p}) = (c_i^1(\mathbf{p}) \quad c_i^2(\mathbf{p}) \quad c_i^3(\mathbf{p})) H|_{\mathbf{p}} (c_i^1(\mathbf{p}) \quad c_i^2(\mathbf{p}) \quad c_i^3(\mathbf{p}))^\top. \quad (42)$$

The iterated stabilization of gauge frames starts with an orientation score $V : \mathbb{M}_2 \rightarrow \mathbb{R}$. An initial gauge frame is fitted to this orientation score, and results an orientation confidence $S(V)$. During the next iteration, the gauge frame is fitted to this orientation confidence $S(V)$ with resulting orientation confidence $S(S(V)) =: S^2(V)$. This process can have as many iterations as needed. In practice, the computation costs (and time) outweigh the benefits of the improved smoothness of the gauge frame after four iterations [13]. This method has been proposed in Section 11.3.2 and 11.3.3 in [5], but a formal proof of convergence $S^n \rightarrow S^*$ has not been published yet.

2 Metric Tensor Fields

In this section, we introduce the different metric tensor fields that were used for both the left invariant frame and the gauge frame. They describe how distances are defined on the manifold. We can visualize these distances with control sets. In Figure 16, we visualize the effect of such a metric tensor field. In the left figure, we see a situation in which the tracking is allowed to move backward, as you can see in the image in the left top corner. This model uses the control sets visualized in the left bottom corner. These control sets are symmetric in their spatial direction. On the right side, we see what happens when we restrict the model, and allow only forward movement. In this case, the model uses in-place rotations in order to take the shortest path, as we see in the right top corner. The control sets are visualized in the right bottom corner. We see that these only support movement in the forward direction.

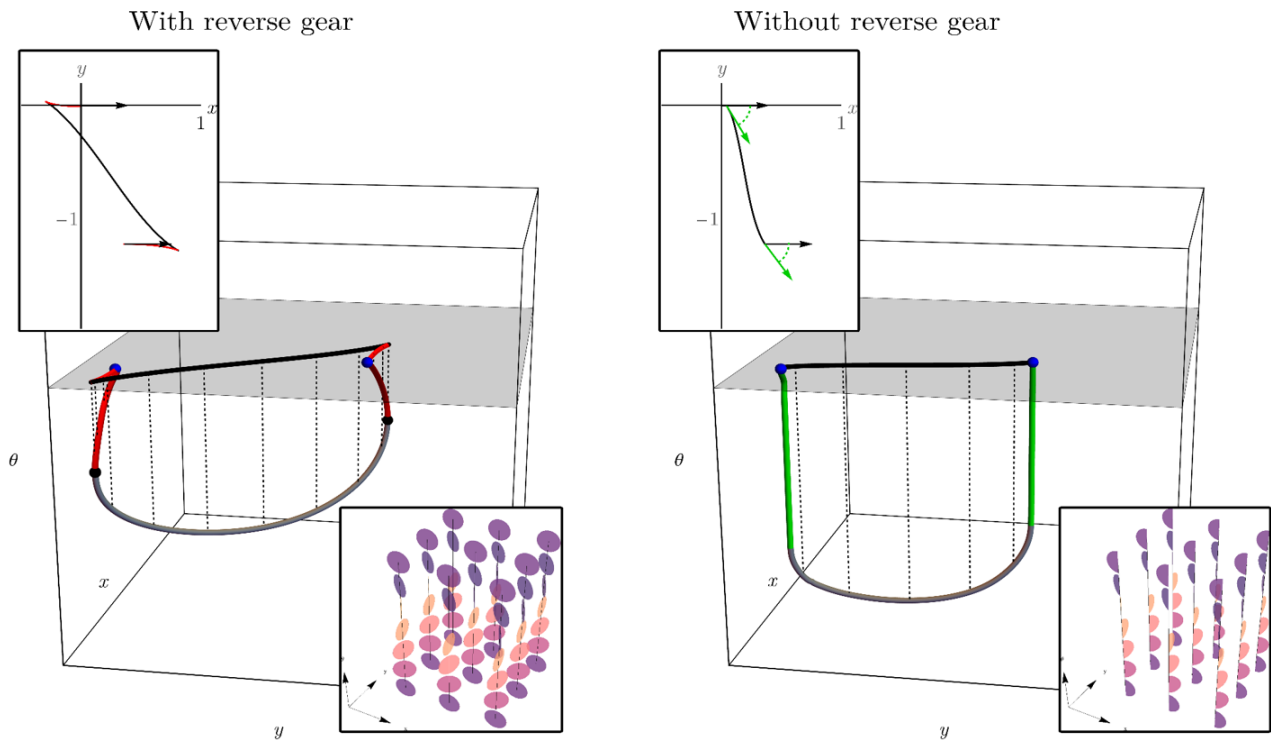


Figure 16: Visualization of the effect of control sets (symmetric and forward gear). Left: Model which is allowed to use reverse gear. The control sets are spatially symmetric and allow angular movement. Right: Model without reverse gear. The control sets are asymmetric, and allow only forward spatial (and angular) movement. Taken from [17].

In this section, we will define some specific metric sensor field, but first, we define $(a)_- := \min\{a, 0\}$ and $(a)_+ := \max\{a, 0\}$, which we use in the forward gear variants of the models.

- The *left invariant metric tensor field* \mathcal{G}_{LIF} is given by

$$\mathcal{G}_{\text{LIF}} := \frac{1}{(1 + \beta/C)^2} \left(\xi^2 \left(\omega^1 \otimes \omega^1 + \frac{1}{\zeta^2} \omega^2 \otimes \omega^2 \right) + \omega^3 \otimes \omega^3 \right). \quad (43)$$

In this expression, the cost function $C : \mathbb{M}_2 \rightarrow [\delta, 1]$ discourages the model from taking paths outside the vessels. The parameter β determines the influence of the cost function, where $\beta \geq 0$. When $\beta = 0$, the cost function does not influence the solution at all, whereas higher values of β make sure that the cost function plays a more important role towards distance calculations on the manifold. The anisotropy parameter ζ penalizes vectors with large sideways components, and thus stimulates forward movement. The last parameter is given by ξ and this is a weighting parameter which influences the flexibility of the calculated vessel, as we will see later on.

- The *forward gear left invariant metric tensor field* $\mathcal{G}_{\text{LIF}}^f$ is given by

$$\mathcal{G}_{\text{LIF}}^f := \frac{1}{(1 + \beta/C)^2} \left(\xi^2 \left(\omega^1 \otimes \omega^1 + (\varepsilon^{-2} - 1) (\omega^1 \otimes \omega^1)_- + \frac{1}{\zeta^2} \omega^2 \otimes \omega^2 \right) + \omega^3 \otimes \omega^3 \right). \quad (44)$$

In this expression, the parameters C , β , ζ and ξ have the same meaning and influence as explained in model (43). However, we have one extra parameter; $\varepsilon \in [0, 1]$. This parameter decides how strongly the

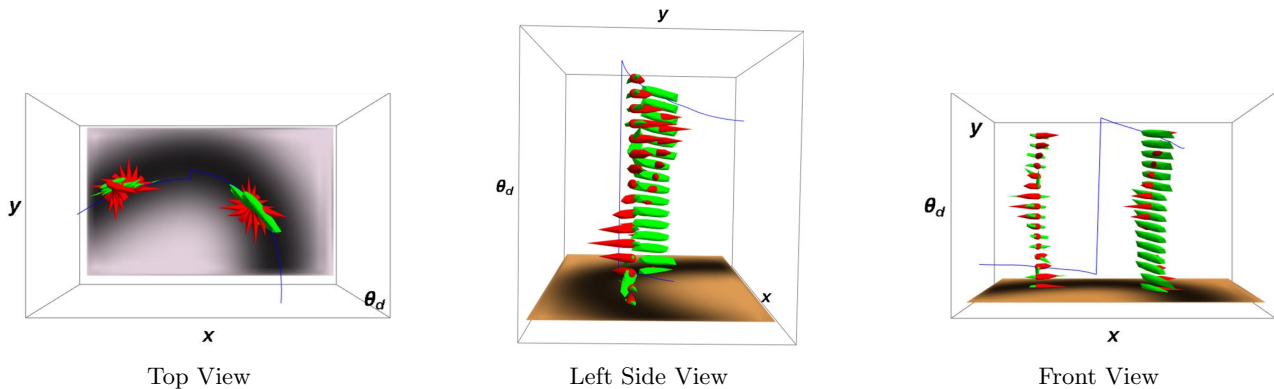


Figure 17: Local plots of the control sets (see Definition 1.19 and Appendix B; balls in the tangent space rather than the Lie group itself) on the curve (blue) acquired while backtracking. The underlying image is the same as in Figure 5, and the tracking is obtained using the gauge frame. The red and green control sets correspond to the left invariant and gauge frame respectively. The control sets are vertically plotted with respect to their discrete orientation $\theta_d = \frac{\theta}{2\pi/N_\theta} + 1$.

model has to adhere to the forward gear. Note that the case $\varepsilon = 1$ yields a Reeds-Shepp model which does allow backwards motions [17].

- The (*anisotropic*) gauge frame metric tensor field \mathcal{G}_{GF} is given by

$$\mathcal{G}_{\text{GF}} := \frac{1}{(1 + \beta/C)^2} \left(\chi^1 \otimes \chi^1 + \frac{1}{(1 - \alpha)^2} (\chi^2 \otimes \chi^2 + \chi^3 \otimes \chi^3) \right) = \sum_{i,j=1}^3 [\mathcal{G}_{\mathcal{B}}]_{ij} \chi^i \otimes \chi^j. \quad (45)$$

As explained in Section 1.4, the gauge frame is fitted to the line structures in the orientation score. Therefore, this model often gives a better approximation of the blood vessels, than the left invariant frame. This model has some parameters that are the same as for the left invariant frame, but also some that are different. The implementation of the cost function C is the same as it was for both left invariant frame metric tensor fields. The same holds for the parameter β . However, the anisotropy parameter has changed to $\alpha \in [0, 1)$. It penalizes vectors that are orthogonal to the line structure of the first gauge vector \mathcal{B}_1 . When $\alpha = 0$, there is no penalization of the vectors orthogonal to \mathcal{B}_1 . On the other hand, $\alpha \rightarrow 1$ results in high penalties for orthogonal vectors.

- Last, we introduce the (*anisotropic*) forward gear gauge frame metric tensor field $\mathcal{G}_{\text{GF}}^f$. It is important that we clearly define what “forward movement” is. In the left invariant frame, this was in the positive direction of \mathcal{A}_1 . However, in the gauge frame, we can both consider moving forward as moving in the positive direction of either \mathcal{A}_1 or \mathcal{B}_1 . We decide to use the convention that the forward direction, is the direction of \mathcal{B}_1 . Then, the metric tensor field $\mathcal{G}_{\text{GF}}^f$ is given by

$$\mathcal{G}_{\text{GF}}^f := \frac{1}{(1 + \beta/C)^2} \left(\chi^1 \otimes \chi^1 + (\varepsilon^{-2} - 1) (\chi^1 \otimes \chi^1)_- + \frac{1}{(1 - \alpha)^2} (\chi^2 \otimes \chi^2 + \chi^3 \otimes \chi^3) \right). \quad (46)$$

In this expression, the cost function C and parameters β and α are defined in the same way as they were in the anisotropic gauge frame metric tensor field (45). The new parameter $\varepsilon \in [0, 1]$ decides how strongly the model has to adhere to the forward gear. Note that the case $\varepsilon = 1$ does allow backwards motion. On the other hand, when $\varepsilon \rightarrow 0$, only in point rotations and forward movement are allowed.

Note that the way we defined forward movement (in the direction of \mathcal{B}_1) could mean that backwards motion in the left invariant frame compared to the previous position is allowed, when the gauge frames are not nicely aligned. However, in practice, we have not encountered any cases where this led to problematic situations. This could be solved by defining forward movement in the direction of \mathcal{A}_1 , but since this means that we will have to be working with two different frames, we chose to not implement this.

We already explained that the gauge frame is able to adapt for curvature, whereas the left invariant frame is not. We illustrate this in Figure 17. We see that the control sets corresponding to the gauge frame are always aligned with the underlying vessel structure, whereas the control sets belonging to the left invariant frame, rotate and support movement in all directions, depending on the orientation of the point \mathbf{p} .

3 Numerical Approach

In this section, we will provide more details of the numerical approach towards identifying where the vessels are in the retina images. First, we give a general overview of the steps that need to be taken in order to find the shortest path (geodesic), as you can find in Section 3.1. In Section 3.2 we focus on how to solve the eikonal equation using both finite differences and fast marching. Then, we explain how we track the vessels (from their end point to their start point) in Section 3.3. Lastly, we compare the results of the two numerical methods that we discussed to the exact solution as one can read in Section 3.4.

3.1 General Approach

There are 5 steps that need to be followed when calculating the optimal geodesics in a manifold \mathbb{M} . Let us consider the procedure in a metric space $(\mathbb{M}, d_{\mathcal{F}})$ where $d_{\mathcal{F}}$ denotes the distance measure and where \mathcal{F} represents a Finsler metric.

[Step 1] Compute the dual Finsler function \mathcal{F}^* . Note that the Finsler function itself, \mathcal{F} , is known. Then, \mathcal{F}^* satisfies

$$\mathcal{F}^*(\mathbf{p}, \hat{\mathbf{p}}) = \sup_{\dot{\mathbf{p}} \in T_{\mathbf{p}}(\mathbb{M})} \frac{\langle \dot{\mathbf{p}}, \hat{\mathbf{p}} \rangle}{\mathcal{F}(\mathbf{p}, \dot{\mathbf{p}})}, \text{ where } \hat{\mathbf{p}} \in T_{\mathbf{p}}^*(\mathbb{M}).$$

[Step 2] Derive the eikonal system. That means that one needs to write out

$$\begin{cases} \mathcal{F}^*(\mathbf{p}, dU(\mathbf{p})) = 1 & \text{for all } \mathbf{p} \in \mathbb{M} \setminus \{\mathbf{p}_S\} \\ U(\mathbf{p}_S) = 0. \end{cases} \quad (47)$$

where $U = d_{\mathcal{F}}(\mathbf{p}_S, \cdot)$ denotes the distance map. In other words, we need to write Eq. (47) in function of \mathbf{p} and $dU(\mathbf{p})$.

[Step 3] Obtain the distance map as the viscosity solution of the eikonal system.

[Step 4] Compute the derivative $d\mathcal{F}^*$ of the dual Finsler function (w.r.t. 2nd entry),

$$d_{\hat{\mathbf{p}}}\mathcal{F}^*(\mathbf{p}, \hat{\mathbf{p}}) := \frac{\partial}{\partial \hat{\mathbf{p}}}\mathcal{F}^*(\mathbf{p}, \hat{\mathbf{p}}).$$

[Step 5] Apply backtracking via an intrinsic gradient descent on the distance map. That means that we have to calculate

$$\begin{cases} \dot{\gamma}(t) = -L d_{\hat{\mathbf{p}}}\mathcal{F}^*(\gamma(t), dU(\gamma(t))), & L := d_{\mathcal{F}}(\mathbf{p}, \mathbf{p}_S) \\ \gamma(0) = \mathbf{p}, & \gamma(1) = \mathbf{p}_S. \end{cases}$$

where γ is a minimal geodesic from \mathbf{p}_S to some $\mathbf{p} \in \mathbb{M}$.

In the setting that we will consider, we have a Riemannian metric. This means that the Finsler metric \mathcal{F} is induced by a metric tensor field \mathcal{G} on Riemannian manifold $(\mathbb{M}, \mathcal{G})$, where the following relation holds

$$\mathcal{F}(\mathbf{p}, \dot{\mathbf{p}}) = \sqrt{\mathcal{G}|_{\mathbf{p}}(\dot{\mathbf{p}}, \dot{\mathbf{p}})}.$$

3.2 Solving the Eikonal Equation in Riemannian Setting

In order to find the shortest curves between any end point (or tip) \mathbf{q} and a starting point (or seed) \mathbf{p}_S , we need to solve the eikonal equation. Let us introduce a distance function $U = d_{\mathcal{G}}(\mathbf{p}_S, \cdot)$. This function is found by solving the viscosity solution of the eikonal partial differential equation (PDE)

$$\begin{cases} \|\text{grad } U\|_{\mathcal{G}} = 1 \\ U(\mathbf{p}_S) = 0, \end{cases} \quad (48)$$

where $U : \mathbb{M}_2 \rightarrow \mathbb{R}$ is the continuous viscosity function of Eq. (47). Here, the gradient $\text{grad } U$, which is defined almost everywhere, is the vector reciprocal to the differential dU :

$$\mathcal{G}(\text{grad } U, \cdot) = dU. \quad (49)$$

The norm induced by the tensor metric field \mathcal{G} is calculated as explained in Section 1.1.6.

3.2.1 Finite Differences

First, we consider the finite difference approach in detail. We discuss the discretization schemes of all metric tensor fields that were implemented as part of this project. For all metric tensors, both the “symmetric” and “forward” gear versions are discussed.

Earlier, we explained that we consider the Eikonal equation (with $U : \mathbb{M}_2 \rightarrow \mathbb{R}$) given by

$$\begin{cases} \|\text{grad } U(\mathbf{p})\|_{\mathcal{G}} = 1 & \forall \mathbf{p} \in \mathbb{M}_2, \\ U(\mathbf{p}_S) = 0. \end{cases}$$

We are looking for the viscosity solution of this equation, as explained in [Step 3]. One approach to find this viscosity solution, is by considering the Hamilton-Jacobi PDE (with $W : \mathbb{M}_2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$). Let $\epsilon > 0$ fixed, $r \in [r_n, r_{n+1}]$, $r_n := n\epsilon$ with $n \in \mathbb{N}$. Then, the Hamilton-Jacobi PDE is given by

$$\begin{cases} \partial_r W_{n+1}^\epsilon(\mathbf{p}, r) = 1 - \|\text{grad } W_{n+1}^\epsilon(\mathbf{p}, r)\|_{\mathcal{G}} & \forall \mathbf{p} \in \mathbb{M}_2, r \geq 0, & (50a) \\ W_{n+1}^\epsilon(\mathbf{p}, r_n) = W_n^\epsilon(\mathbf{p}, r_n) & \forall \mathbf{p} \neq \mathbf{p}_S, & (50b) \\ W_{n+1}^\epsilon(\mathbf{p}_S, r_n) = 0, & & (50c) \end{cases}$$

for $n = 1, 2, \dots$, and for $n = 0$, we have

$$\begin{cases} \partial_r W_1^\epsilon(\mathbf{p}, r) = 1 - \|\text{grad } W_1^\epsilon(\mathbf{p}, r)\|_{\mathcal{G}} & \forall \mathbf{p} \in \mathbb{M}_2 \\ W_1^\epsilon(\mathbf{p}, 0) = \delta_{\mathbf{p}_S}^M(\mathbf{p}), \end{cases} \quad (51a)$$

$$(51b)$$

where $\delta_{\mathbf{p}_S}^M$ is the morphological delta given by

$$\delta_{\mathbf{p}_S}^M(\mathbf{p}) = \begin{cases} 0 & \text{if } \mathbf{p} = \mathbf{p}_S, \\ +\infty & \text{else.} \end{cases} \quad (52)$$

We use the end condition of $W_n^\epsilon(\mathbf{p}, r_n)$ of the n th evolution as an initial condition $W_{n+1}^\epsilon(\mathbf{p}, r_n)$ of the $(n+1)$ th evolution. Only for $\mathbf{p} = \mathbf{p}_S$, we have a different initial condition, being $W_{n+1}^\epsilon(\mathbf{p}_S, r_n) = 0$. Then, the pointwise limit $W^\infty(\mathbf{p})$ is defined as

$$W^\infty(\mathbf{p}) := \lim_{\epsilon \rightarrow 0} \left(\lim_{n \rightarrow \infty} W_{n+1}^\epsilon(\mathbf{p}, (n+1)\epsilon) \right). \quad (53)$$

It is important that each initial value problem solution $W_{n+1}^\epsilon(\mathbf{p}, r)$ is the unique viscosity solution of Eq. (50). In that case, the viscosity property for the viscosity solutions of Eq. (53) carry over the viscosity solutions of Eq. (47). So, we obtain $W = W^\infty$ as the unique viscosity solution of Eq. (48). More information on the limit in Eq. (53) can be found in Appendix E of [6]. Here, it is also explained in detail why it is necessary to have two limits. For an intuitive explanation, we turn to Figure 18. We see that, for $\epsilon > 0$, we obtain staircasing (due to a discrete rounding of the distance function). Taking the limit $\epsilon \downarrow 0$, the solution $W^\infty(\mathbf{p}) = W(\mathbf{p})$ is obtained.

Remark 3.1. The staircasing limit depicted in Figure 18 is similar to the basic eikonal boundary value problem on \mathbb{R} with solution $d(x, 0) = |x|$. On \mathbb{R} the approach, described by Eq. (50), (51) and (53), provides the pointwise limit

$$|x| = \lim_{\epsilon \rightarrow 0} \sum_{m=0}^{\infty} r_{m+1} \mathbb{1}_{[r_m, r_{m+1}]}(|x|) = \lim_{\epsilon \rightarrow 0} \sum_{m=0}^{\lceil \frac{|x|}{\epsilon} \rceil} r_{m+1} \mathbb{1}_{[r_m, r_{m+1}]}(|x|), \quad (54)$$

with $r_m = m\epsilon$ [6].

Note that Eq. (53) is a limit that is independent of the PDE discretization. When taking $\epsilon > 0$, we do discretize the morphological convolution solving the PDE which explains the stairs in Figure 18.

Definition 3.1 (Morphological convolution). The morphological convolution \square of two functions f_1 and f_2 is defined by

$$(f_1 \square f_2)(\mathbf{x}) = \inf_{\mathbf{y} \in \Omega} \{f_1(\mathbf{x} - \mathbf{y}) + f_2(\mathbf{y})\}. \quad (55)$$

Remark 3.2. Note that the morphological convolution where one of the functions is the morphological delta, that we introduced in Eq. (52), for some function f and some point \mathbf{x} in domain Ω , results in

$$\begin{aligned} (f \square \delta_0^M)(\mathbf{x}) &= \inf_{\mathbf{y} \in \Omega} \{f(\mathbf{x} - \mathbf{y}) + \delta_0^M(\mathbf{y})\} = f(\mathbf{x}); \\ (\delta_0^M \square f)(\mathbf{x}) &= \inf_{\mathbf{y} \in \Omega} \{\delta_0^M(\mathbf{x} - \mathbf{y}) + f(\mathbf{y})\} = f(\mathbf{x}). \end{aligned}$$

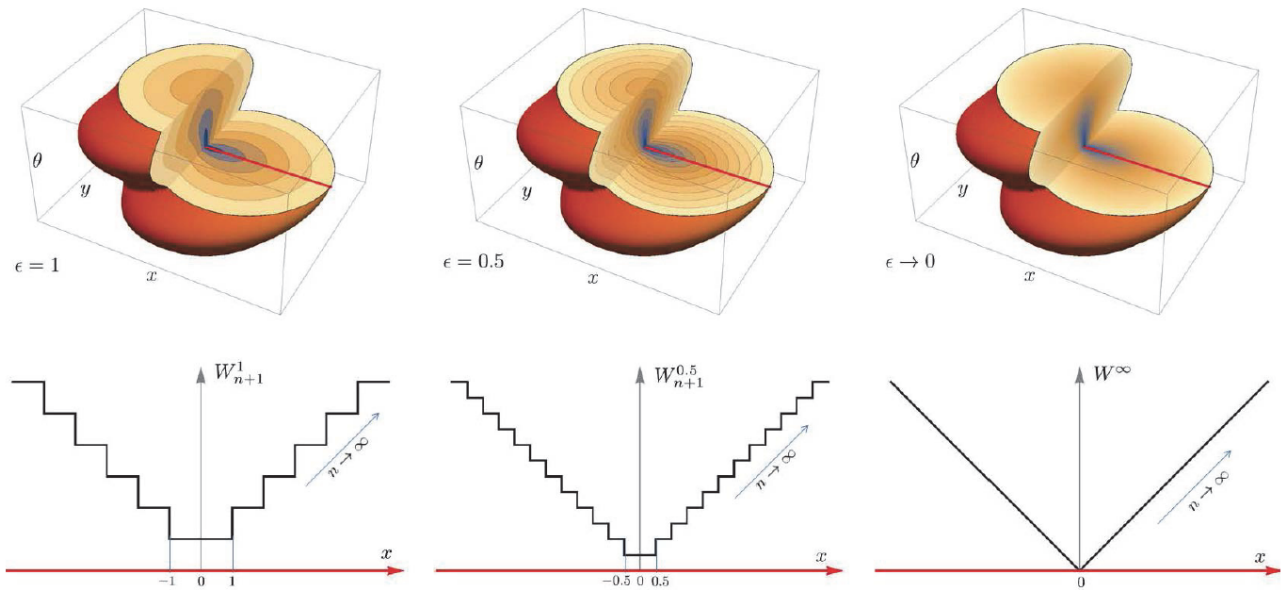


Figure 18: Illustration of the pointwise limits in Eq. (53). Top: plot of $\mathbf{p} \mapsto W_{n+1}^\epsilon(\mathbf{p}, r_{n+1})$ (from left to right, for $\epsilon = 1$, $\epsilon = 0.5$ and $\epsilon \downarrow 0$). Along the red axis $\{(x, 0, 0) | x \in \mathbb{R}\}$ we have $x = d(\mathbf{p}, \mathbf{e})$. Bottom: the corresponding graph of $x \mapsto W_{n+1}^\epsilon((x, 0, 0), r_{n+1})$. As n grows the staircase grows; as $\epsilon \rightarrow 0$ the size of the steps in the staircase vanishes, and we see $W^\infty = d(\mathbf{p}, \mathbf{e})$ in the rightmost column. Taken from [6].

Before, we move on to the actual discretization of the equation, we give an interpretation of the solution. When we solve the Eikonal equation, we calculate the distances from the seed \mathbf{p}_S to every other point $\mathbf{p} \in SE(2)$. It is useful to interpret these distances, and their corresponding distance function U , in terms of wavefront propagation. The wave of a viscous fluid originates from the starting point \mathbf{p} and propagates throughout the domain. This explains the viscosity solution W , which needs to be calculated in [Step 3] of the introduced approach. In the Eikonal equation, function values represent the distance from the seed to reach the specific point. In the viscosity solution, these function value represents the time it takes for the wave to travel towards the specific end value. The shortest path is found by tracking back along the path which was followed by the wavefronts. [6]

Left Invariant Frame We start with the left invariant metric tensor field defined by Eq. (43). Let us recall

$$\mathcal{G}_{\text{LIF}} = \tilde{C} (\xi^2 \omega^1 \otimes \omega^1 + \xi^2 \zeta^{-2} \omega^2 \otimes \omega^2 + \omega^3 \otimes \omega^3),$$

where $\tilde{C} = \frac{1}{(1+\beta/C)^2}$ is the mobility of the metric tensor. Let us denote

$$dW(\mathbf{p}) = (\mathcal{A}_1|_{\mathbf{p}} W) \omega^1|_{\mathbf{p}} + (\mathcal{A}_2|_{\mathbf{p}} W) \omega^2|_{\mathbf{p}} + (\mathcal{A}_3|_{\mathbf{p}} W) \omega^3|_{\mathbf{p}}, \quad (56)$$

where ω^i ($i = 1, 2, 3$) form the covector basis and $\mathcal{A}_i|_{\mathbf{p}} W$ ($i = 1, 2, 3$) are the coefficients in \mathbb{R} . Then, the norm becomes

$$\begin{aligned} \mathcal{G}_{\mathbf{p}} \left(\mathcal{G}_{\mathbf{p}}^{-1} \left((\mathcal{A}_1|_{\mathbf{p}} W) \omega^1|_{\mathbf{p}} + (\mathcal{A}_2|_{\mathbf{p}} W) \omega^2|_{\mathbf{p}} + (\mathcal{A}_3|_{\mathbf{p}} W) \omega^3|_{\mathbf{p}} \right), \mathcal{G}_{\mathbf{p}}^{-1} \left((\mathcal{A}_1|_{\mathbf{p}} W) \omega^1|_{\mathbf{p}} + (\mathcal{A}_2|_{\mathbf{p}} W) \omega^2|_{\mathbf{p}} + (\mathcal{A}_3|_{\mathbf{p}} W) \omega^3|_{\mathbf{p}} \right) \right) \\ = \frac{1}{\tilde{C}} \left(\frac{1}{\xi^2} (\mathcal{A}_1|_{\mathbf{p}} W)^2 + \frac{\zeta^2}{\xi^2} (\mathcal{A}_2|_{\mathbf{p}} W)^2 + (\mathcal{A}_3|_{\mathbf{p}} W)^2 \right), \end{aligned} \quad (57)$$

using Eq. (21) with $a_{11} = \sqrt{\tilde{C}}\xi$, $a_{22} = \sqrt{\tilde{C}}\xi/\zeta$ and $a_{33} = \sqrt{\tilde{C}}$.

Since we want to calculate the eikonal PDE using finite differences, we need to find the correct discretization of Eq. (57). Here, it is important to realize that $\mathcal{A}_i|_{\mathbf{p}} W$ also contains derivatives (for $i = 1, 2, 3$) as we have seen in Section 1.3, Eq. (25). We recall that these derivatives are given by

$$\mathcal{A}_1 W(\mathbf{x}, \theta) = \cos \theta \partial_x W(\mathbf{x}, \theta) + \sin \theta \partial_y W(\mathbf{x}, \theta) = \lim_{h \downarrow 0} \frac{W(\mathbf{x} + h \mathbf{e}_\phi^\theta, \theta) - W(\mathbf{x}, \theta)}{h}; \quad (58a)$$

$$\mathcal{A}_2 W(\mathbf{x}, \theta) = -\sin \theta \partial_x W(\mathbf{x}, \theta) + \cos \theta \partial_y W(\mathbf{x}, \theta) = \lim_{h \downarrow 0} \frac{W(\mathbf{x} + h \mathbf{e}_\rho^\theta, \theta) - W(\mathbf{x}, \theta)}{h}; \quad (58b)$$

$$\mathcal{A}_3 W(\mathbf{x}, \theta) = \partial_\theta W(\mathbf{x}, \theta) = \lim_{h \downarrow 0} \frac{W(\mathbf{x}, \theta + h) - W(\mathbf{x}, \theta)}{h}, \quad (58c)$$

where $\mathbf{e}_\phi^\theta = (\cos \theta, \sin \theta)$ and $\mathbf{e}_\rho^\theta = (-\sin \theta, \cos \theta)$. Let there be a distance $\Delta r > 0$ such that

$$\mathcal{A}_1 W(\mathbf{x}, \theta) \approx \frac{W(\mathbf{x} + \Delta r \mathbf{e}_\phi^\theta, \theta) - W(\mathbf{x}, \theta)}{\Delta r}; \quad (59a)$$

$$\mathcal{A}_2 W(\mathbf{x}, \theta) \approx \frac{W(\mathbf{x} + \Delta r \mathbf{e}_\rho^\theta, \theta) - W(\mathbf{x}, \theta)}{\Delta r}; \quad (59b)$$

$$\mathcal{A}_3 W(\mathbf{x}, \theta) \approx \frac{W(\mathbf{x}, \theta + \Delta \theta) - W(\mathbf{x}, \theta)}{\Delta \theta}. \quad (59c)$$

The discretization of the forward difference variant of these derivatives yields

$$(\mathcal{A}_1 w_{i,j,k}^n)^f = (\mathcal{A}_1 W(i\Delta x, j\Delta y, k\Delta \theta))^f \approx \frac{W(\mathbf{x} + \Delta r \mathbf{e}_\phi^\theta, \theta) - W(\mathbf{x}, \theta)}{\Delta r} = \frac{w_{i+\frac{\Delta r}{\Delta x} \cos(k\Delta \theta), j+\frac{\Delta r}{\Delta y} \sin(k\Delta \theta), k}^n - w_{i,j,k}^n}{\Delta r}; \quad (60a)$$

$$(\mathcal{A}_2 w_{i,j,k}^n)^f = (\mathcal{A}_2 W(i\Delta x, j\Delta y, k\Delta \theta))^f \approx \frac{W(\mathbf{x} + \Delta r \mathbf{e}_\rho^\theta, \theta) - W(\mathbf{x}, \theta)}{\Delta r} = \frac{w_{i-\frac{\Delta r}{\Delta x} \sin(k\Delta \theta), j+\frac{\Delta r}{\Delta y} \cos(k\Delta \theta), k}^n - w_{i,j,k}^n}{\Delta r}; \quad (60b)$$

$$(\mathcal{A}_3 w_{i,j,k}^n)^f = (\mathcal{A}_3 W(i\Delta x, j\Delta y, k\Delta \theta))^f \approx \frac{W(\mathbf{x}, \theta + \Delta \theta) - W(\mathbf{x}, \theta)}{\Delta \theta} = \frac{w_{i,j,k+1}^n - w_{i,j,k}^n}{\Delta \theta}. \quad (60c)$$

where $\mathcal{A}_l w_{i,j,k}^n$ represents the discretization of $\mathcal{A}_l W(i\Delta x, j\Delta y, k\Delta \theta)$ for $l = 1, 2, 3$, and where the superscript f indicates that we are considering the forward finite differences. Note that for example $w_{i+\frac{\Delta r}{\Delta x} \cos(k\Delta \theta), j+\frac{\Delta r}{\Delta y} \sin(k\Delta \theta), k}^n$ does not necessarily lie on the grid that we use for the calculations. Hence, we use interpolation to estimate those values that are not on the grid. The Lie-package that was developed for Mathematica assumes that $\Delta x = \Delta y = \Delta r = 1$. That means that we consider the discretization scheme:

$$(\tilde{\mathcal{A}}_1 w_{i,j,k}^n)^f = (\tilde{\mathcal{A}}_1 W(\mathbf{x}, \theta))^f \approx (W(\mathbf{x} + \mathbf{e}_\phi^\theta, \theta) - W(\mathbf{x}, \theta)) = w_{i+\cos(k\Delta \theta), j+\sin(k\Delta \theta), k}^n - w_{i,j,k}^n; \quad (61a)$$

$$(\tilde{\mathcal{A}}_2 w_{i,j,k}^n)^f = (\tilde{\mathcal{A}}_2 W(\mathbf{x}, \theta))^f \approx (W(\mathbf{x} + \mathbf{e}_\rho^\theta, \theta) - W(\mathbf{x}, \theta)) = w_{i-\sin(k\Delta \theta), j+\cos(k\Delta \theta), k}^n - w_{i,j,k}^n; \quad (61b)$$

$$(\tilde{\mathcal{A}}_3 w_{i,j,k}^n)^f = (\tilde{\mathcal{A}}_3 W(\mathbf{x}, \theta))^f \approx \frac{1}{\Delta \theta} (W(\mathbf{x}, \theta + \Delta \theta) - W(\mathbf{x}, \theta)) = \frac{w_{i,j,k+1}^n - w_{i,j,k}^n}{\Delta \theta}, \quad (61c)$$

where $\tilde{\mathcal{A}}_l w_{i,j,k}^n$ represents the discrete version of $\mathcal{A}_l W(i\Delta x, j\Delta y, k\Delta \theta)$ which was implemented in the Lie-package for $l = 1, 2, 3$.

In the left invariant frame, the discretization of Eq. (50a), using forward differences, results in

$$\begin{cases} w_{i,j,k}^{n+1} = w_{i,j,k}^n - \Delta t \left(\tilde{C}^{-1/2} \sqrt{\xi^{-2} (\tilde{\mathcal{A}}_1 w_{i,j,k}^n)^2 + \xi^{-2} \zeta^2 (\tilde{\mathcal{A}}_2 w_{i,j,k}^n)^2 + (\tilde{\mathcal{A}}_3 w_{i,j,k}^n)^2 - 1} \right) \\ w_{\mathbf{p}_S} = 0, \end{cases}$$

where $\tilde{\mathcal{A}}_l w_{i,j,k}^n = \max \left\{ -(\tilde{\mathcal{A}}_l w_{i,j,k}^n)^f, (\tilde{\mathcal{A}}_l w_{i,j,k}^n)^b, 0 \right\}$ for $l = 1, 2, 3$, and where $w_{\mathbf{p}_S}$ is the discretized representation of $W(\mathbf{p}_S)$. Forward and backward finite differences are respectively denoted by superscripts f and b . This system of equations can easily be solved using software like MATLAB or Mathematica.

Forward Gear Left Invariant Frame When we consider the forward gear version of the left invariant frame, we need to remark that the constraint only affects $\tilde{\mathcal{A}}_1|_{\mathbf{p}} W$, which should always have a positive value for $\epsilon = 1$. Therefore, the discretization of Eq. (50a) for the forward gear left invariant frame leaves us with

$$\begin{cases} w_{i,j,k}^{n+1} = w_{i,j,k}^n - \Delta t \left(\tilde{C}^{-1/2} \sqrt{\xi^{-2} \left((\tilde{\mathcal{A}}_1 w_{i,j,k}^n)_+ \right)^2 + \xi^{-2} \epsilon^2 \left((\tilde{\mathcal{A}}_1 w_{i,j,k}^n)_- \right)^2 + \xi^{-2} \zeta^2 (\tilde{\mathcal{A}}_2 w_{i,j,k}^n)^2 + (\tilde{\mathcal{A}}_3 w_{i,j,k}^n)^2 - 1} \right) \\ w_{\mathbf{p}_S} = 0, \end{cases}$$

where $\tilde{\mathcal{A}}_l w_{i,j,k}^n$ is the discretized form of the derivatives as defined in (61) for $l = 1, 2, 3$.

Gauge Frame Now, we move on to the gauge frame metric tensor field with was given by Eq. (45). We recall

$$\mathcal{G}_{\text{GF}} := \tilde{C} \left(\chi^1 \otimes \chi^1 + \frac{1}{(1-\alpha)^2} (\chi^2 \otimes \chi^2 + \chi^3 \otimes \chi^3) \right),$$

where $\tilde{C} = \frac{1}{(1+\beta/C)^2}$ is the mobility of the metric tensor. Our goal is to express Eq. (50a) with respect to the gauge frame.

Let us denote

$$dW(p) = (\mathcal{B}_1|_{\mathbf{p}} W) \chi^1 + (\mathcal{B}_2|_{\mathbf{p}} W) \chi^2 + (\mathcal{B}_3|_{\mathbf{p}} W) \chi^3, \quad (62)$$

where χ^i ($i = 1, 2, 3$) form the covector basis of the gauge frame and $(\mathcal{B}_i|_{\mathbf{p}} W)$ are coefficients in \mathbb{R} for $i = 1, 2, 3$. Then, the norm becomes

$$\begin{aligned} & \mathcal{G}_{\mathbf{p}}(\mathcal{G}_{\mathbf{p}}^{-1}((\mathcal{B}_1|_{\mathbf{p}} W) \chi^1 + (\mathcal{B}_2|_{\mathbf{p}} W) \chi^2 + (\mathcal{B}_3|_{\mathbf{p}} W) \chi^3), \mathcal{G}_{\mathbf{p}}^{-1}((\mathcal{B}_1|_{\mathbf{p}} W) \chi^1 + (\mathcal{B}_2|_{\mathbf{p}} W) \chi^2 + (\mathcal{B}_3|_{\mathbf{p}} W) \chi^3)) \\ &= \tilde{C}^{-1} \left((\mathcal{B}_1|_{\mathbf{p}} W)^2 + (1-\alpha)^2 (\mathcal{B}_2|_{\mathbf{p}} W)^2 + (1-\alpha)^2 (\mathcal{B}_3|_{\mathbf{p}} W)^2 \right), \end{aligned} \quad (63)$$

using Eq. (21) with $a_{11} = \sqrt{\tilde{C}}$ and $a_{22} = a_{33} = \sqrt{\tilde{C}}/(1-\alpha)$.

In order to determine the discretization of Eq. (50a) in the gauge frame, we need to figure out how we determine the correct expression for $(B_l w_{i,j,k}^n)^f$ and $(B_l w_{i,j,k}^n)^b$ for $l = 1, 2, 3$ and f and b indicating forward and backward finite differences respectively and where $B_l w_{i,j,k}^n$ denotes the discretization of $\mathcal{B}_l W(i\Delta x, j\Delta y, k\Delta\theta)$ for $l = 1, 2, 3$. We know that \mathcal{B}_j is constructed as a weighted sum of \mathcal{A}_i where $i, j = 1, 2, 3$ as introduced in Section 1.4, Eq. (36) and (38). We recall that

$$\mathcal{B}_l|_{\mathbf{p}} W = \sum_{m=1}^3 c_l^m \mathcal{A}_m|_{\mathbf{p}} W,$$

where $c_l^m \in \mathbb{R}$. These constants also influence the forward and backward finite differences for the gauge frame. We recall the definition of the first order upwind scheme.

Definition 3.2 (First order upwind scheme in \mathbb{R}). We consider the ordinary differential equation

$$\frac{\partial y}{\partial t} = a \frac{\partial y}{\partial x}, \quad a \in \mathbb{R}.$$

Defining

$$a_+ = \max\{a, 0\}, \quad a_- = \min\{a, 0\}$$

and

$$y_x^- = \frac{y_i^n - y_{i-1}^n}{\Delta x}, \quad y_x^+ = \frac{y_{i+1}^n - y_i^n}{\Delta x}$$

the first order upwind scheme is given by

$$y_i^{n+1} = y_i^n + \Delta t (a_+ y_x^- + a_- y_x^+).$$

The theory in Definition 3.2 translates for the left invariant derivatives and gauge derivatives to

$$\begin{aligned} (B_l w_{i,j,k}^n)^f &= \sum_{m=1}^3 \left((c_l^m)_+ (\tilde{A}_m w_{i,j,k}^n)^f + (c_l^m)_- (\tilde{A}_m w_{i,j,k}^n)^b \right) \\ (B_l w_{i,j,k}^n)^b &= \sum_{m=1}^3 \left((c_l^m)_- (\tilde{A}_m w_{i,j,k}^n)^f + (c_l^m)_+ (\tilde{A}_m w_{i,j,k}^n)^b \right). \end{aligned}$$

Then, the discretization of Eq. (50a) becomes

$$\begin{cases} w_{i,j,k}^{n+1} = w_{i,j,k}^n - \Delta t \left(\tilde{C}^{-1/2} \sqrt{(B_1 w_{i,j,k}^n)^2 + (1-\alpha)^2 (B_2 w_{i,j,k}^n)^2 + (1-\alpha)^2 (B_3 w_{i,j,k}^n)^2} - 1 \right) \\ w_{\mathbf{p}_S} = 0, \end{cases}$$

where $B_l w_{i,j,k}^n = \max\left\{- (B_l w_{i,j,k}^n)^f, (B_l w_{i,j,k}^n)^b, 0\right\}$ for $l = 1, 2, 3$.

Forward Gear Gauge Frame Lastly, we need to consider the forward gear gauge frame. In Eq. (46), we introduced the metric tensor field for the forward gear variant of the gauge frame. We see that the only difference compared to the symmetric gauge frame is that we are only considering the positive part and a rescaled negative part of $\mathcal{B}_1|_{\mathbf{p}} W$. Therefore, the discretization of Eq. (50a) becomes

$$\begin{cases} w_{i,j,k}^{n+1} = w_{i,j,k}^n - \Delta t \left(\tilde{C}^{-1/2} \sqrt{\left((B_1 w_{i,j,k}^n)_+ \right)^2 + \varepsilon^2 \left((B_1 w_{i,j,k}^n)_- \right)^2 + (1-\alpha)^2 (B_2 w_{i,j,k}^n)^2 + (1-\alpha)^2 (B_3 w_{i,j,k}^n)^2} - 1 \right) \\ w_{\mathbf{p}_S} = 0, \end{cases}$$

where $B_l w_{i,j,k}^n = \max\left\{- (B_l w_{i,j,k}^n)^f, (B_l w_{i,j,k}^n)^b, 0\right\}$ for $l = 1, 2, 3$.

3.2.2 (Hamiltonian) Fast Marching

An alternative method to calculate the distances from an arbitrary point \mathbf{p} to a seed \mathbf{p}_S is called fast marching. This is an efficient calculation method in which the shortest path can be found in one pass when the distance map satisfies two conditions (monotony, causality) which will be discussed later on in this chapter. Therefore, this method should be a lot faster than the earlier discussed finite differences, since it is not an iterative method; not all values are updated every time step, but just a targeted group. We use the implementation written by Jean-Marie Mirebeau [26]. However, we also need to know how this method works [27; 28]. First, we will give an overview of the different steps, the code goes through. Thereafter, we will explain the different mechanisms that are used.

The fast marching code follows a number of steps. Different theoretical aspects are used in order to calculate a geodesic. They are presented in chronological order below.

1. Call the primal metric;
2. Calculate the dual metric using Lemma 3.1;
3. Use Selling's algorithm to calculate Voronoi's first reduction of quadratic forms of the dual metric. This algorithm determines which points can be reached from each seed;
4. Discretize the eikonal PDE;
5. Calculate Fast Marching Algorithm 2.

We start off with introducing some notation. Just like before, we work on a discretized space. We will be using the Cartesian grid space, denoted by $\mathbb{L} := \mathbb{Z}^d$, where d is the dimension of the position space. The Euclidean space is denoted by $\mathbb{E} = \mathbb{R}^d$. In this report, we have $d = 2$. Then, the domain $\Omega \subseteq \mathbb{R}^d$ needs to be discretized and is denoted by

$$\Omega_h := \Omega \cap h\mathbb{L}, \quad \partial\Omega_h = (\mathbb{E} \setminus \Omega) \cap h\mathbb{L},$$

where $h > 0$. We denote the dual vector space of \mathbb{E} and \mathbb{L} by \mathbb{E}^* and $\mathbb{L}^* := \{\hat{\mathbf{p}} \in \mathbb{E}^* : \forall \mathbf{p} \in \mathbb{L}, \langle \hat{\mathbf{p}}, \mathbf{p} \rangle \in \mathbb{Z}\}$ respectively, where $\langle \cdot, \cdot \rangle$ denotes the duality bracket. Note that we have $\mathbb{E} \cong \mathbb{E}^*$ and $\mathbb{L} \cong \mathbb{L}^*$ since $d < \infty$.

Let us denote $S(\mathbb{E}) \subseteq L(\mathbb{E}, \mathbb{E}^*)$, $S^+(\mathbb{E})$ and $S^{++}(\mathbb{E})$ as the space of symmetric linear maps, semi-definite symmetric linear maps and positive definite symmetric linear maps respectively. Additionally, we introduce $\text{GL}(\mathbb{E}) \subseteq L(\mathbb{E}, \mathbb{E})$ which denotes the group of invertible linear transformations, and $\text{GL}(\mathbb{L}) \subseteq \text{GL}(\mathbb{E})$ the subgroup of those which leave the Cartesian grid \mathbb{L} invariant (i.e. their matrix has integer coefficients and determinant ± 1). In the fast marching code, we need to supply the primal metric with respect to the standard frame. The function that we call transforms the primal metric to a dual metric. The python-script does this transformation using Lemma 3.1 as introduced in [17].

Lemma 3.1. Let $\mathbf{w} \in \mathbb{R}^n$ and let $M \in \mathbb{R}^{n \times n}$ be symmetric, positive definite, where $n = d(d+1)/2$. Define the norm $\mathcal{F}_{M, \mathbf{w}} : \mathbb{R}^n \rightarrow \mathbb{R}^+$ by

$$\mathcal{F}_{M, \mathbf{w}}(\mathbf{v}) = \sqrt{(M\mathbf{v}, \mathbf{v}) + (\mathbf{w}, \mathbf{v})_-^2}.$$

Then its dual norm

$$\mathcal{F}_{M, \mathbf{w}}^*(\hat{\mathbf{v}}) = \sqrt{(\hat{\mathbf{v}}, \hat{M}\hat{\mathbf{v}}) + (\hat{\mathbf{v}}, \hat{\mathbf{w}})_+^2},$$

with $\hat{M} = (M + \mathbf{w} \otimes \mathbf{w})^{-1}$ and $\hat{\mathbf{w}} = \frac{M^{-1}\mathbf{w}}{\sqrt{1 + (\mathbf{w}, M^{-1}\mathbf{w})}}$.

More information on how the primal metric with respect to the standard, static frame is calculated, can be found in Appendix A.

When one want to decompose an arbitrary tensor $D \in S^{++}(\mathbb{E})$ in the form

$$D = \sum_{\hat{\mathbf{e}} \in \mathbb{L} \setminus \{0\}} \rho^{\hat{\mathbf{e}}}(D) \hat{\mathbf{e}} \otimes \hat{\mathbf{e}}, \quad \text{where } \forall \hat{\mathbf{e}} \in \mathbb{L} \setminus \{0\}, \rho^{\hat{\mathbf{e}}}(D) \geq 0, \quad (64)$$

with finitely many nonzero coefficients $\rho^{\hat{\mathbf{e}}}(D)$, while maximizing

$$\sum_{\hat{\mathbf{e}} \in \mathbb{L} \setminus \{0\}} \rho^{\hat{\mathbf{e}}}(D),$$

one can use Voronoi's first reduction of quadratic forms [27].

Theorem 1 (Voronoi). *Let us define*

$$\mathcal{P} := \{M \in S^{++}(\mathbb{R}); \forall \dot{\mathbf{e}} \in \mathbb{L} \setminus \{0\}, \|\dot{\mathbf{e}}\|_M \geq 1\}, \quad \mathcal{L}(D) := \inf_{M \in \mathcal{P}} \text{Tr}(MD).$$

The linear problem $\mathcal{L}(D)$ is feasible, for any $D \in S^{++}(\mathbb{E}^*)$, in the sense that the set of minimizers is non-empty and compact. In addition, the convex polytope $\mathcal{P} \subseteq S(\mathbb{E})$ has a finite number of equivalence classes of vertices under the action of $GL(\mathbb{L}) \subseteq GL(\mathbb{E})$.

In other words, the minimization problem is given by

$$\text{minimize } \langle\langle M, D \rangle\rangle \text{ subject to } \langle\langle M, \dot{\mathbf{e}} \otimes \dot{\mathbf{e}} \rangle\rangle \geq 1 \text{ for all } \dot{\mathbf{e}} \in \mathbb{L} \setminus \{0\}, \quad (65)$$

where $\langle\langle M, D \rangle\rangle := \text{Tr}(MD)$.

Since we are considering dimensions $d = 2$, we can use Selling's algorithm to solve Eq. (65). This algorithm returns a superbase, of which we introduce the definition below.

Definition 3.3. [Superbase] A superbase of \mathbb{Z}^d is a $(d+1)$ -tuple $\mathbf{b} = (\mathbf{v}_0, \dots, \mathbf{v}_d) \in (\mathbb{Z}^d)^{d+1}$ such that $|\det(\mathbf{v}_1, \dots, \mathbf{v}_d)| = 1$ and $\mathbf{v}_0 + \dots + \mathbf{v}_d = \mathbf{0}$. One also defines $M_{\mathbf{b}} := \frac{1}{2} \sum_{i=1}^d \mathbf{v}_i \mathbf{v}_i^T \in S^{++}(\mathbb{E})$. A superbase is said M -obtuse, where $M \in S^{++}(\mathbb{E})$, iff $\langle \dot{\mathbf{e}}_i, \dot{\mathbf{e}}_j \rangle_M \leq 0$ for all $0 \leq i < j \leq d$.

In Algorithm 1, we state Selling's algorithm. Since Voronoi's first reduction was feasible (Theorem 1), we know that the algorithm will end. Selling's algorithm returns a D -obtuse superbase (as defined in Definition 3.3). The vertex $M_{\mathbf{b}}$ which is associated with this superbase \mathbf{b} is optimal, and the corresponding Karush-Kuhn-Tucker relations are given by

$$D = - \sum_{0 \leq i < j \leq d} \langle \mathbf{v}_i, D\mathbf{v}_j \rangle \mathbf{e}_{ij} \otimes \mathbf{e}_{ij},$$

where $\mathbf{e}_{ij} := \mathbf{v}_k^\perp$ with $\{i, j, k\} = \{0, 1, 2\}$.

Algorithm 1: Selling's algorithm [27].

Input: matrix D and superbase $\mathbf{b} = (\mathbf{v}_0, \dots, \mathbf{v}_d)$

Define $I \subseteq \{1, \dots, d\}$, such that $\langle \mathbf{v}_i, D\mathbf{v}_j \rangle > 0$ for all $i < j, i, j \in I$.

while $I \neq \emptyset$ **do**

 Replace the superbase \mathbf{b} by $\mathbf{b}' = (-\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_i - \mathbf{v}_j)$

 Redefine $I \subseteq \{1, \dots, d\}$, such that $\langle \mathbf{v}_i, D\mathbf{v}_j \rangle > 0$ for all $i < j, i, j \in I$.

Output: D -obtuse superbase, i.e. $\langle \mathbf{v}_i, D\mathbf{v}_j \rangle \leq 0$ for all $i \neq j$.

In Figure 19, one can see a visualization of the dual unit ball and different discretizations. The discretizations are the offsets associated by Voronoi's first reduction to a tensor D involved in the decomposition (Eq. (64)). The points where the arrows end are considered to be having distance 1 from the center and are therefore within reach. During the Fast Marching updates, the algorithm can reach these points from its center.

We create acute stencils when we use Selling's algorithm and Voronoi's theorem.

Definition 3.4 (Acuteness). Let \mathcal{F} be an asymmetric norm on \mathbb{E} . We say that two vectors $u, v \in \mathbb{E} \setminus \{0\}$ form an \mathcal{F} -acute angle if

$$\mathcal{F}(u + \delta v) \geq \mathcal{F}(u) \quad \text{and} \quad \mathcal{F}(v + \delta u) \geq \mathcal{F}(v) \quad \text{for all } \delta \geq 0.$$

We say that a finite conforming triangulation \mathcal{T} is \mathcal{F} -acute if

1. The union of triangles $T \in \mathcal{T}$ is a neighborhood of the origin.
2. The vertices of each $T \in \mathcal{T}$ lie on \mathbb{L} , one of them is the origin 0 and T has area 1/2.
3. The non-zero vertices of each triangle $T \in \mathcal{T}$ form an \mathcal{F} -acute angle.

When a stencil is acute, it is also causal.

Proposition 3.1 (Causality Property). Let \mathcal{F} be an asymmetric norm on \mathbb{E} , let $u, v \in \mathbb{E}$ be linearly independent, and let $d_u, d_v \in \mathbb{R}$. Assume that u and v form an \mathcal{F} -acute angle. Define

$$d_w := \min_{t \in [0, 1]} t d_u + (1-t) d_v + \mathcal{F}(t u + (1-t) v),$$

and assume that this minimum is not attained for $t \in \{0, 1\}$. Then $d_u < d_w$ and $d_v < d_w$.

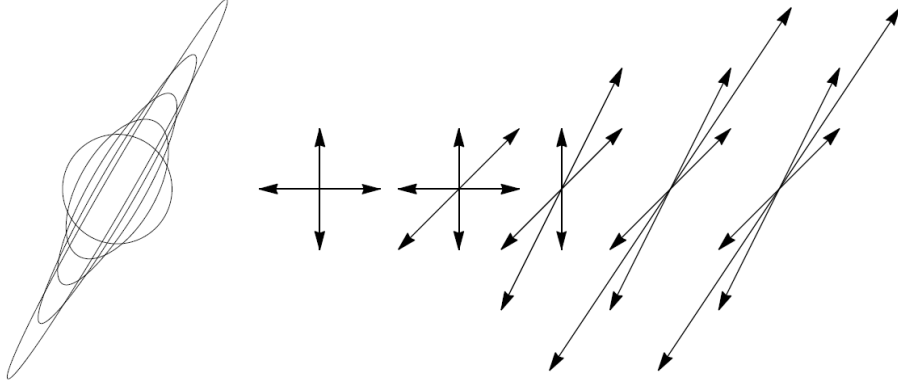


Figure 19: Dual unit ball $\{\mathbf{p}; \|\mathbf{p}\|_{D^{-1}} \leq 1\}$, and offsets associated by Voronoi's first reduction to a tensor D , involved in the decomposition in $d = 2$. Taken from [28].

This proposition is proved in [29]. When this causality property is satisfied and the resulting map is monotone, the tracking can be solved in one pass.

An alternative definition of a causal map is given in Definition 3.5. We will use this definition to illustrate why the our stencil is causal and provide more information on the updating process of fast marching.

Definition 3.5 (Monotony, causality). A numerical scheme on a finite set Z is a map $H : Z \times \mathbb{R} \times \mathbb{R}^Z \rightarrow \mathbb{R}$. It is said

- Monotone iff H is non-decreasing w.r.t. the second and (each of the) third variables.
- Causal iff H only depends on the positive part of the third variable(s).

One denotes for any $U : Z \rightarrow \mathbb{R}$ and any $\mathbf{p} \in Z$

$$\tilde{H}U(\mathbf{p}) := H(\mathbf{p}, U(\mathbf{p}), (U(\mathbf{p}) - U(\mathbf{q}))_{\mathbf{q} \in Z})$$

In our case, we have $\tilde{H}U(\mathbf{p}) := U(\mathbf{p})$ and $Z = \mathbb{M}_2$. Both properties automatically follow from inserting the upwind finite differences in the dual which is of the form

$$\mathcal{F}^*(\mathbf{p}, \hat{\mathbf{p}})^2 \approx \max_{1 \leq k \leq K} \left(\sum_{1 \leq i \leq I} \alpha_{ik} \langle \hat{\mathbf{p}}, \dot{\mathbf{e}}_{ik} \rangle_+^2 + \sum_{1 \leq j \leq J} \beta_{jk} \langle \hat{\mathbf{p}}, \dot{\mathbf{f}}_{jk} \rangle^2 \right),$$

where $\mathbf{p} \in \Omega$ and co-vector $\hat{\mathbf{p}} \in \mathbb{E}^*$. The integers I, J, K are meta-parameters which are fixed for each class of metric, in contrast to $\alpha_{ik}, \beta_{jk} \geq 0$ and the offsets $\dot{\mathbf{e}}_{ik}, \dot{\mathbf{f}}_{jk} \in \mathbb{Z}^d$ implicitly depend on the current point $\mathbf{p} \in X$, and where $1 \leq i \leq I$, $1 \leq j \leq J$ and $1 \leq k \leq K$ [28].

For simplification, we assume $K = 1$. The sets I and J remain arbitrary. Then, we have

$$\mathcal{F}^*(\mathbf{p}, \hat{\mathbf{p}})^2 \approx \sum_{1 \leq i \leq I} \alpha_i(\mathbf{p}) \langle \hat{\mathbf{p}}, \dot{\mathbf{e}}_i \rangle_+^2 + \sum_{1 \leq j \leq J} \beta_j(\mathbf{p}) \langle \hat{\mathbf{p}}, \dot{\mathbf{e}}_j \rangle^2.$$

The weights $\alpha_i = \alpha_i(\mathbf{p}) \geq 0$, $\beta_j = \beta_j(\mathbf{p}) \geq 0$ and offsets $\dot{\mathbf{e}}_i = \dot{\mathbf{e}}_i(\mathbf{p}) \in \mathbb{Z}^d$ both depend on the point $\mathbf{p} \in X$ in which they are evaluated. The discretization scheme is given by

$$(\mathcal{F}_h U(\mathbf{p}))^2 := h^{-2} \sum_{1 \leq i \leq I} \alpha_i(\mathbf{p}) \max\{0, U(\mathbf{p}) - U(\mathbf{p} - h\dot{\mathbf{e}}_i)\}^2 + \sum_{1 \leq j \leq J} \beta_j(\mathbf{p}) \max\{0, U(\mathbf{p}) - U(\mathbf{p} - h\dot{\mathbf{e}}_j), U(\mathbf{p}) - U(\mathbf{p} + h\dot{\mathbf{e}}_j)\}^2,$$

where $\mathcal{F}_h U - 1 = 0$ should be satisfied. Without loss of generality, we can reduce this system to

$$(\mathcal{F}_h U(\mathbf{p}))^2 := h^{-2} \sum_{1 \leq i \leq I} \tilde{\alpha}_i(\mathbf{p}) \max\{0, U(\mathbf{p}) - U(\mathbf{q}_i)\}^2,$$

where $\mathbf{q}_i := \mathbf{p} - h\dot{\mathbf{e}}_i(\mathbf{p})$ are the points which can be reached from point \mathbf{p} and which are accepted. In addition to that, $\tilde{\alpha}_i = \tilde{\alpha}_i(\mathbf{p})$ is a linear combination of α_i and β_j , depending on the values of $U(\mathbf{p})$ and $U(\mathbf{q}_i)$. When calculating the fast marching algorithm (Algorithm 2), it is important to know how the value of $U(\mathbf{p})$ is updated.

In the fast marching algorithm, we divide the points in three categories: Far, Trial and Accepted. During an update, we only consider the set of “trial” points. After one of these points moves to the “accepted” set, one redefines the set of “trial” points with points from the far set, which have become within reach of the accepted points. Once a point \mathbf{p} is tagged as “accepted”, its function value $U(\mathbf{p})$ becomes frozen as well. When all the points have moved to the “accepted” category, a geodesic can be easily calculated. The exact algorithm can be found in Algorithm 2.

Algorithm 2: Fast Marching [28].

Pre-computed stencils $(V[\mathbf{p}])_{\mathbf{p} \in X \cup \partial X}$, where $V[\mathbf{q}] := \{\mathbf{p} \in X; \mathbf{q} \in \mathcal{V}(\mathbf{p})\}$, with

$$\mathcal{V}(\mathbf{p}) := \{\mathbf{q} = \mathbf{p} - h\mathbf{e}_i(\mathbf{p}); 1 \leq i \leq I, \mathbf{q} \text{ is visible from } \mathbf{p}\} \cap (X \cup \partial X).$$

Tag points of ∂X as Trial, and other points as Far.

Initialize $U : X \cup \partial X \rightarrow (-\infty, \infty]$ to the value $+\infty$ on X and the boundary condition on ∂X .

while there remain Trial points **do**

 Find a Trial point \mathbf{q} minimizing U .

 Tag \mathbf{q} as Accepted, and call Post-process(\mathbf{q}).

for each neighbor $\mathbf{p} \in V[\mathbf{q}]$, either Far or Trial **do**

if \mathbf{p} is Far **then**

 tag it as Trial, and call Pre-process (\mathbf{p}).

 Update $U(\mathbf{p})$, taking into account the value $U(\mathbf{q})$ of the last Accepted point.

Now, we take a closer look at the updating process of the values $U(\mathbf{p})$. Let us consider a point $\mathbf{p} \in X$ that is tagged as Trial, and which has value $U(\mathbf{p})$. This point \mathbf{p} needs to be updated w.r.t. the surrounding Accepted points. Then, the value of $U(\mathbf{p})$ needs to be replaced with the largest solution $\lambda \in \mathbb{R}$ to the following quadratic equation:

$$h^{-2} \sum_{1 \leq i \leq I} \tilde{\alpha}_i(\mathbf{p}) \max\{0, (\lambda - U(\mathbf{q}_i))\}^2 = 1. \quad (66)$$

First, we note that we only consider the function values at locations \mathbf{q}_i , where $\mathbf{q}_i \in \mathcal{V}(\mathbf{p})$, so within reach of the point \mathbf{p} . In addition to that, we use that we have $\lambda \geq U(\mathbf{q}_i)$ for \mathbf{q}_i Accepted (due to the monotonicity property of U). Therefore, we can rewrite Eq. (66) to

$$h^{-2} \sum_{i \in \Upsilon} \tilde{\alpha}_i(\mathbf{p}) (\lambda - U(\mathbf{q}_i))^2 = 1, \quad \text{where } \Upsilon := \{1 \leq i \leq I; \mathbf{q}_i \in \mathcal{V}(\mathbf{p}), \mathbf{q}_i \text{ is Accepted}\}. \quad (67)$$

The neighbor points \mathbf{q}_i where $i \in \Upsilon$ are the points that are tagged as Accepted and which can be reached from point \mathbf{p} and are within the domain $X \cup \partial X$. Eq. (67) always has two real roots. This holds because of the ordering, causality and monotony of the stencils and is a consequence of the construction of the fast marching algorithm [28].

When calculating the fast marching algorithm, the smallest root of Eq. (67) is a numerical artifact and therefore not important when determining $U(\mathbf{p})$. However, the largest root denoted by λ^* satisfies by construction $\lambda^* \geq U(\mathbf{q}_i)$ for all $i \in \Upsilon$, and defines the updated value of $U(\mathbf{p})$. Note that

$$\lambda^* = \frac{\sum_{i \in \Upsilon} \tilde{\alpha}_i(\mathbf{p}) U(\mathbf{q}_i) + \sqrt{(\sum_{k \in \Upsilon} \tilde{\alpha}_k(\mathbf{p}) U(\mathbf{q}_k))^2 - (\sum_{l \in \Upsilon} \tilde{\alpha}_l(\mathbf{p})) (\sum_{m \in \Upsilon} \tilde{\alpha}_m(\mathbf{p}) U(\mathbf{q}_m)^2 - 1)}}{\sum_{j \in \Upsilon} \tilde{\alpha}_j(\mathbf{p})}. \quad (68)$$

The updating process of value $U(\mathbf{p})$ is also described in Algorithm 3.

Algorithm 3: Update of $U(\mathbf{p})$ in CPU algorithm [28].

Input: U, \mathbf{p}

Define:

Set $\Upsilon := \{1 \leq i \leq I; \mathbf{q}_i \in \mathcal{V}(\mathbf{p}), \mathbf{q}_i \text{ Accepted}\}$.

Variables $a = \sum_{i \in \Upsilon} \tilde{\alpha}_i(\mathbf{p})$, $b = \sum_{i \in \Upsilon} \tilde{\alpha}_i(\mathbf{p}) U(\mathbf{q}_i)$, $c = \sum_{i \in \Upsilon} \tilde{\alpha}_i(\mathbf{p}) U(\mathbf{q}_i)^2 - 1$.

Calculate $\lambda^* = \frac{b + \sqrt{b^2 - ac}}{a}$.

Output: $U^{\text{new}}(\mathbf{p}) = \lambda^*$

Above, we described the CPU-implementation of Fast Marching. This is used when the discretization problem is monotone and causal. However, when this is not the case, the GPU-implementation can be used. In that case, Eq. (66) cannot be simplified to (67) since we do not know on beforehand which points are in the set of

accepted points and consequently, we update the values simultaneously. Therefore, the updating works a little bit differently. We consider all values in the surroundings of a point \mathbf{p} . First, we sort all surrounding values $U(\mathbf{q}_i)$, and denote the ordered neighbor points by u_{i_1}, \dots, u_{i_I} . Then, we calculate the value of the point \mathbf{p} , represented by $\lambda := U(\mathbf{p})$ for the scenarios in which points u_{i_1} up to u_{i_r} are included (where $1 \leq r \leq I$), denoted by λ_r . We accept the value λ_{r-1} for $U(\mathbf{p})$ when $\lambda_{r-1} \leq u_{i_r}$. This is also described in Algorithm 4.

Algorithm 4: Update of $U(\mathbf{p})$ in GPU algorithm [30].

Variables $a_0 = 0, b_0 = 0, c_0 = -h^2, \lambda = \infty$.

Sort the indices, so that $u_{i_1} \leq \dots \leq u_{i_I}$.

for r from 1 to I : **do**

if $\lambda \leq u_{i_r}$ **then**

\perp break.

$a_r = a_{r-1} + \alpha_{i_r}; b_r = b_{r-1} + \alpha_{i_r} u_{i_r}; c_r = c_{r-1} + \alpha_{i_r} u_{i_r}^2$

$\lambda = \frac{b_r + \sqrt{b_r^2 - a_r c_r}}{a_r}$

Return λ

3.3 Backtracking from Tip to Seed

Once we have found the distances from (all possible) tips to the seed, we need to determine the shortest curve (geodesic). We know that the distances are defined (and calculated) in such a way that the distance from the seed to the seed itself is 0. The further you move away from the location and orientation of the seed, the larger the distance to the seed gets. Do note that this distance is not (necessarily) the Euclidean distance. We use steepest descent to determine the shortest path from a given tip to the seed. First, we recall how the steepest descent algorithm is defined.

Definition 3.6 (Directional Derivative). Let $\mathbf{p} \in \mathbb{M}_2$ and $\dot{\mathbf{p}} \in T_{\mathbf{p}}\mathbb{M}_2$. If $(\dot{\mathbf{p}}, \dot{\mathbf{p}})_{\mathcal{G}} = 1$, then the directional derivative $D_{\dot{\mathbf{p}}}f(\mathbf{p})$ is given by

$$D_{\dot{\mathbf{p}}}f(\mathbf{p}) = (\nabla f(\mathbf{p}), \dot{\mathbf{p}})_{\mathcal{G}}.$$

Lemma 3.2 (Direction of Steepest Descent). The directional derivative $D_{\dot{\mathbf{p}}}f(\mathbf{p}) = (\nabla f(\mathbf{p}), \dot{\mathbf{p}})_{\mathcal{G}}$ where $\|\dot{\mathbf{p}}\|_{\mathcal{G}} = 1$ is minimal for $\mathbf{X}_{\mathbf{p}}[f] = -\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|_{\mathcal{G}}}$. Therefore,

$$\mathbf{X}_{\mathbf{p}}[f] = -\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|_{\mathcal{G}}} \in T_{\mathbf{p}}(M)$$

is also known as the direction of steepest descent.

Proof. By definition 3.6, we can write the directional derivative as

$$D_{\dot{\mathbf{p}}}f(\mathbf{p}) = (\nabla f(\mathbf{p}), \dot{\mathbf{p}})_{\mathcal{G}}.$$

One of the properties of the inner product is that it is maximal when both vectors point into the same direction ($\dot{\mathbf{p}} = \nabla f(\mathbf{p}) / \|\nabla f(\mathbf{p})\|$). The inner product is minimal when both vectors point in opposite directions. Since we have $\|\dot{\mathbf{p}}\|_{\mathcal{G}} = 1$, we know that the inner product is minimal for

$$\mathbf{X}_{\mathbf{p}}[f] = -\frac{\nabla f(\mathbf{p})}{\|\nabla f(\mathbf{p})\|_{\mathcal{G}}}.$$

□

Theorem 2 (Steepest Descent Algorithm). Let a point $\mathbf{p}^{(0)}$ be given. The steepest descent algorithm is an iterative method in which all points $\mathbf{p}^{(k)}$ are defined by

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - t_k \frac{\nabla f(\mathbf{p}^{(k)})}{\|\nabla f(\mathbf{p}^{(k)})\|_{\mathcal{G}}}. \quad (69)$$

where k denotes the iteration, and t_k denotes the step size.

Next, we have to take a look at how we implement the steepest descent algorithm correctly for the different frames that we use to calculate the shortest curves. First, we take a look at a general example to get a feeling of how the the implementation of steepest descent should be implemented on different frames.

Example 3.1. Let us consider the function

$$f(x, y, z) = \frac{1}{2} (ax^2 + by^2 + cz^2).$$

Calculating the direction of steepest descent w.r.t. the canonical frame, we find

$$\mathbf{X}_{\mathbf{p}}[f] = -\frac{\nabla f(x, y, z)}{\|\nabla f(x, y, z)\|} = -\frac{ax \nabla x + by \nabla y + cz \nabla z}{\|\nabla f(x, y, z)\|}, \text{ where } \nabla x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \nabla y = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \text{ and } \nabla z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

When we want to calculate the steepest descent direction with respect to a different frame than the frame in which f is expressed, let's say $(\tilde{x}, \tilde{y}, \tilde{z})$. This influences the direction of steepest descent, and it is given by

$$\mathbf{X}_{\mathbf{p}}[f] = -\frac{ax \tilde{\nabla} x + by \tilde{\nabla} y + cz \tilde{\nabla} z}{\|\nabla f(x, y, z)\|}, \text{ where } \tilde{\nabla} x = \begin{pmatrix} \partial_{\tilde{x}} x \\ \partial_{\tilde{y}} x \\ \partial_{\tilde{z}} x \end{pmatrix}.$$

3.3.1 Left Invariant Frame

First, we take a closer look at the steepest descent for the left invariant frame. Our goal is to determine an expression which represents the direction of steepest descent such that we can use the steepest descent method to estimate the geodesic (or shortest curve) from a given tip to the seed. We have the left invariant frame, given by $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3\}$ and the standard canonical frame defined by $\{\partial_x, \partial_y, \partial_\theta\}$. As discussed in Section 1.3, the left invariant frame can be described by

$$\begin{pmatrix} \mathcal{A}_1 W(\mathbf{p}) \\ \mathcal{A}_2 W(\mathbf{p}) \\ \mathcal{A}_3 W(\mathbf{p}) \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \partial_x W(\mathbf{p}) \\ \partial_y W(\mathbf{p}) \\ \partial_\theta W(\mathbf{p}) \end{pmatrix}.$$

Considering the left invariant tensor field given in Eq. (43), we find the direction of steepest descent to be

$$\mathbf{X}_{\mathbf{p}}[W] = -(1 + \beta / C|_{\mathbf{p}})^2 \left(\xi^{-2} \mathcal{A}_1|_{(x,y,\theta)} W \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} + \zeta^2 \xi^{-2} \mathcal{A}_2|_{(x,y,\theta)} W \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{pmatrix} + \mathcal{A}_3|_{(x,y,\theta)} W \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right). \quad (70)$$

Taking Eq. (70) and implementing it in the steepest descent algorithm given in Eq. (69), we find

$$\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \\ \theta^{(k+1)} \end{pmatrix} = \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ \theta^{(k)} \end{pmatrix} + \tilde{t}_k (1 + \beta / C|_{\mathbf{p}^{(k)}})^2 \begin{pmatrix} \xi^{-2} \cos \theta^{(k)} \mathcal{A}_1|_{\mathbf{p}^{(k)}} W - \zeta^2 \xi^{-2} \sin \theta^{(k)} \mathcal{A}_2|_{\mathbf{p}^{(k)}} W \\ \xi^{-2} \sin \theta^{(k)} \mathcal{A}_1|_{\mathbf{p}^{(k)}} W + \zeta^2 \xi^{-2} \cos \theta^{(k)} \mathcal{A}_2|_{\mathbf{p}^{(k)}} W \\ \mathcal{A}_3|_{\mathbf{p}^{(k)}} W \end{pmatrix},$$

where $\tilde{t}_k = \frac{t_k}{\|\mathbf{X}_{\mathbf{p}^{(k)}}[W]\|_{\mathcal{G}}}$.

Forward Gear We also need to consider the effect of the forward gear to the backtracking such as in the metric tensor field introduced in Eq. (44). In our models, that means that we only consider the positive part of the dual of $\omega^1 \otimes \omega^1$. We denote this by $(\mathcal{A}_1|_{\mathbf{p}} W)_+ = \max\{0, \mathcal{A}_1|_{\mathbf{p}} W\}$. We know that taking the derivative of $(\frac{1}{2}x)_+^2$ leaves us with $(x)_+$. Therefore, we find the steepest descent algorithm to be

$$\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \\ \theta^{(k+1)} \end{pmatrix} = \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ \theta^{(k)} \end{pmatrix} + \tilde{t}_k (1 + \beta / C|_{\mathbf{p}^{(k)}})^2 \begin{pmatrix} \xi^{-2} \cos \theta^{(k)} (\mathcal{A}_1|_{\mathbf{p}^{(k)}} W)_+ + \varepsilon^2 \xi^{-2} \cos \theta^{(k)} (\mathcal{A}_1|_{\mathbf{p}^{(k)}} W)_- - \zeta^2 \xi^{-2} \sin \theta^{(k)} \mathcal{A}_2|_{\mathbf{p}^{(k)}} W \\ \xi^{-2} \sin \theta^{(k)} (\mathcal{A}_1|_{\mathbf{p}^{(k)}} W)_+ + \varepsilon^2 \xi^{-2} \sin \theta^{(k)} (\mathcal{A}_1|_{\mathbf{p}^{(k)}} W)_- + \zeta^2 \xi^{-2} \cos \theta^{(k)} \mathcal{A}_2|_{\mathbf{p}^{(k)}} W \\ \mathcal{A}_3|_{\mathbf{p}^{(k)}} W \end{pmatrix},$$

where $\tilde{t}_k = \frac{t_k}{\|\mathbf{X}_{\mathbf{p}^{(k)}}^+[W]\|_{\mathcal{G}}}$ and

$$\begin{aligned} \mathbf{X}_{\mathbf{p}}^+[W] = & -(1 + \beta / C|_{\mathbf{p}})^2 \left(\xi^{-2} (\mathcal{A}_1|_{(x,y,\theta)} W)_+ \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} + \varepsilon^2 \xi^{-2} (\mathcal{A}_1|_{(x,y,\theta)} W)_- \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} \right. \\ & \left. + \zeta^2 \xi^{-2} \mathcal{A}_2|_{(x,y,\theta)} W \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{pmatrix} + \mathcal{A}_3|_{(x,y,\theta)} W \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right). \end{aligned}$$

3.3.2 Gauge Frame

Next, we investigate the backtracking in the gauge frame. Here, we have the gauge frame given by $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ as introduced in Section 1.4. We showed that we can denote the Gauge frame using the left invariant frame;

$$\begin{pmatrix} \mathcal{B}_1|_{\mathbf{p}} W \\ \mathcal{B}_2|_{\mathbf{p}} W \\ \mathcal{B}_3|_{\mathbf{p}} W \end{pmatrix} = \begin{pmatrix} c_1^1 & c_1^2 & c_1^3 \\ c_2^1 & c_2^2 & c_2^3 \\ c_3^1 & c_3^2 & c_3^3 \end{pmatrix} \begin{pmatrix} \mathcal{A}_1|_{\mathbf{p}} W \\ \mathcal{A}_2|_{\mathbf{p}} W \\ \mathcal{A}_3|_{\mathbf{p}} W \end{pmatrix}.$$

When we calculate the steepest descent direction w.r.t. the canonical frame for Eq. (45), we find (using the chain rule)

$$\begin{aligned} \mathbf{X}_{\mathbf{p}}[W] = & -(1 + \beta / C|_{\mathbf{p}})^2 \left(\begin{pmatrix} c_1^1 \mathcal{B}_1|_{\mathbf{p}} W + (1 - \alpha)^2 (c_2^1 \mathcal{B}_2|_{\mathbf{p}} W + c_3^1 \mathcal{B}_3|_{\mathbf{p}} W) \\ c_1^2 \mathcal{B}_1|_{\mathbf{p}} W + (1 - \alpha)^2 (c_2^2 \mathcal{B}_2|_{\mathbf{p}} W + c_3^2 \mathcal{B}_3|_{\mathbf{p}} W) \\ c_1^3 \mathcal{B}_1|_{\mathbf{p}} W + (1 - \alpha)^2 (c_2^3 \mathcal{B}_2|_{\mathbf{p}} W + c_3^3 \mathcal{B}_3|_{\mathbf{p}} W) \end{pmatrix} \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} \right. \\ & + \left. \begin{pmatrix} c_1^2 \mathcal{B}_1|_{\mathbf{p}} W + (1 - \alpha)^2 (c_2^2 \mathcal{B}_2|_{\mathbf{p}} W + c_3^2 \mathcal{B}_3|_{\mathbf{p}} W) \\ c_1^3 \mathcal{B}_1|_{\mathbf{p}} W + (1 - \alpha)^2 (c_2^3 \mathcal{B}_2|_{\mathbf{p}} W + c_3^3 \mathcal{B}_3|_{\mathbf{p}} W) \end{pmatrix} \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{pmatrix} \right. \\ & \left. + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right). \end{aligned} \quad (71)$$

Implementing Eq. (71) into the steepest descent algorithm given in Eq. (69) gives

$$\begin{aligned} \begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \\ \theta^{(k+1)} \end{pmatrix} = & \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ \theta^{(k)} \end{pmatrix} + \tilde{t}_k (1 + \beta / C|_{\mathbf{p}^{(k)}})^2 \left(\begin{pmatrix} c_1^1 \mathcal{B}_1|_{\mathbf{p}^{(k)}} W + (1 - \alpha)^2 (c_2^1 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^1 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \\ c_1^2 \mathcal{B}_1|_{\mathbf{p}^{(k)}} W + (1 - \alpha)^2 (c_2^2 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^2 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \\ c_1^3 \mathcal{B}_1|_{\mathbf{p}^{(k)}} W + (1 - \alpha)^2 (c_2^3 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^3 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \end{pmatrix} \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} \right. \\ & + \tilde{t}_k (1 + \beta / C|_{\mathbf{p}^{(k)}})^2 \left(\begin{pmatrix} c_1^2 \mathcal{B}_1|_{\mathbf{p}^{(k)}} W + (1 - \alpha)^2 (c_2^2 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^2 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \\ c_1^3 \mathcal{B}_1|_{\mathbf{p}^{(k)}} W + (1 - \alpha)^2 (c_2^3 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^3 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \end{pmatrix} \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{pmatrix} \right. \\ & \left. + \tilde{t}_k (1 + \beta / C|_{\mathbf{p}^{(k)}})^2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right), \end{aligned}$$

where $\tilde{t}_k = \frac{t_k}{\|\mathbf{X}_{\mathbf{p}^{(k)}}[W]\|_{\mathcal{G}}}$.

Forward Gear We also need to check what effect the ‘‘forward gear’’ has on the gauge frame. We will use the forward gear gauge frame metric tensor field given in Eq. (46). Here, there is a constraint on $\mathcal{B}_1|_{\mathbf{p}^{(k)}} W$, of which we consider a rescaled negative part. The effects on the steepest descent algorithm are similar to the effect of forward gear on the left invariant frame. Hence, we find the algorithm to be

$$\begin{aligned} \begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \\ \theta^{(k+1)} \end{pmatrix} = & \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ \theta^{(k)} \end{pmatrix} + \tilde{t}_k (1 + \beta / C|_{\mathbf{p}^{(k)}})^2 \left(\begin{pmatrix} c_1^1 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_+ + c_1^1 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^1 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^1 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \\ c_1^2 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_+ + c_1^2 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^2 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^2 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \\ c_1^3 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_+ + c_1^3 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^3 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^3 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \end{pmatrix} \begin{pmatrix} \cos \theta^{(k)} \\ \sin \theta^{(k)} \\ 0 \end{pmatrix} \right. \\ & + \tilde{t}_k (1 + \beta / C|_{\mathbf{p}^{(k)}})^2 \left(\begin{pmatrix} c_1^2 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_+ + c_1^2 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^2 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^2 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \\ c_1^3 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_+ + c_1^3 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}^{(k)}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^3 \mathcal{B}_2|_{\mathbf{p}^{(k)}} W + c_3^3 \mathcal{B}_3|_{\mathbf{p}^{(k)}} W) \end{pmatrix} \begin{pmatrix} -\sin \theta^{(k)} \\ \cos \theta^{(k)} \\ 0 \end{pmatrix} \right. \\ & \left. + \tilde{t}_k (1 + \beta / C|_{\mathbf{p}^{(k)}})^2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right), \end{aligned}$$

where $\tilde{t}_k = \frac{t_k}{\|\mathbf{X}_{\mathbf{p}^{(k)}}^+[W]\|_{\mathcal{G}}}$ and

$$\begin{aligned} \mathbf{X}_{\mathbf{p}}^+[W] = & -(1 + \beta / C|_{\mathbf{p}})^2 \left(\begin{pmatrix} c_1^1 (\mathcal{B}_1|_{\mathbf{p}} W)_+ + c_1^1 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^1 \mathcal{B}_2|_{\mathbf{p}} W + c_3^1 \mathcal{B}_3|_{\mathbf{p}} W) \\ c_1^2 (\mathcal{B}_1|_{\mathbf{p}} W)_+ + c_1^2 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^2 \mathcal{B}_2|_{\mathbf{p}} W + c_3^2 \mathcal{B}_3|_{\mathbf{p}} W) \\ c_1^3 (\mathcal{B}_1|_{\mathbf{p}} W)_+ + c_1^3 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^3 \mathcal{B}_2|_{\mathbf{p}} W + c_3^3 \mathcal{B}_3|_{\mathbf{p}} W) \end{pmatrix} \begin{pmatrix} \cos \theta \\ \sin \theta \\ 0 \end{pmatrix} \right. \\ & + \left(\begin{pmatrix} c_1^2 (\mathcal{B}_1|_{\mathbf{p}} W)_+ + c_1^2 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^2 \mathcal{B}_2|_{\mathbf{p}} W + c_3^2 \mathcal{B}_3|_{\mathbf{p}} W) \\ c_1^3 (\mathcal{B}_1|_{\mathbf{p}} W)_+ + c_1^3 \varepsilon^2 (\mathcal{B}_1|_{\mathbf{p}} W)_- + (1 - \alpha)^2 (\varepsilon^2 c_2^3 \mathcal{B}_2|_{\mathbf{p}} W + c_3^3 \mathcal{B}_3|_{\mathbf{p}} W) \end{pmatrix} \begin{pmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{pmatrix} \right. \\ & \left. + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right). \end{aligned}$$

3.4 Comparison of Results to Exact Solution

In order to make sure that our calculated results correspond with the exact solution, we have compared them for both the finite difference method and fast marching. If we know the exact solution to the differential problem that we are considering, we can compare for which scaling of variables both solutions are the most alike in shape, and where the deviation between them is as small as possible.

We compare the results for the both methods to the exact solutions. These exact solutions have been calculated before, and an explanation on how they are obtained can be found in [31]. We note that the results of the exact solutions are on a volume around zero. To be precise, they are given on $[-6, 6] \times [-6, 6] \times [-\pi, \pi]$. The results for the PDE-solver are given on the interval $[1, N_x] \times [1, N_y] \times [0, 2\pi]$. For the Fast Marching method, we can choose on which interval we want the results to be calculated. In order to be able to compare all the answers, we need to make sure that they are on the same interval. Here, we need to use the scaling homothety principle. We will consider the metric tensor field given by

$$\mathcal{F}_0(\mathbf{p}, \dot{\mathbf{p}})^2 := \begin{cases} \mathcal{C}_1^2(\mathbf{p}) |\dot{\mathbf{x}} \cdot \mathbf{n}|^2 + \mathcal{C}_2^2(\mathbf{p}) \|\dot{\mathbf{n}}\|^2 & \text{if } \dot{\mathbf{x}} \propto \mathbf{n} \\ +\infty & \text{otherwise.} \end{cases} \quad (72)$$

which includes relative cost-functions \mathcal{C}_1 and \mathcal{C}_2 , and where $\dot{\mathbf{x}} \propto \mathbf{n} \Leftrightarrow \exists \lambda \in \mathbb{R}$ s.t. $\dot{\mathbf{x}} = \lambda \mathbf{n}$, $\mathbf{p} := (\mathbf{x}, \mathbf{n})$, $\dot{\mathbf{p}} = (\dot{\mathbf{x}}, \dot{\mathbf{n}})$.

In addition to that, we also introduce the spheres

$$\mathcal{S}_R^{\mathcal{F}}(\mathbf{p}_S) := \{\mathbf{p} \in \mathbb{M} \mid d_{\mathcal{F}}(\mathbf{p}_S, \mathbf{p}) = R\} \text{ with } \mathcal{F} = \mathcal{F}_0. \quad (73)$$

Lemma 3.3 (Scaling homothety). Consider $\mathbf{p}_S := (0, 0, 0)$ and $d = 2$. Let $\xi > 0$ denote the stiffness parameter $\xi^{-1} > 0$ with physical dimension [Length].

Consider the metric tensor field given in Eq. (72). Let $\mathcal{F}_{0,\xi}$ denote the special case where the Finsler function is induced by constant relative costs: $\mathcal{C}_1 = \xi > 0$ and $\mathcal{C}_2 = 1$. Let $d_{\mathcal{F}_{0,\xi}}$ be the metric induced by $\mathcal{F} = \mathcal{F}_{0,\xi}$ given in Eq. (9). Consider the corresponding spheres $\mathcal{S}_R^{\mathcal{F}_{0,\xi}}(\mathbf{p}_S)$ given in Eq. (73). Define a scaling operator on \mathbb{M} and the scaled output curve by

$$\Phi_{\xi}(x, y, \theta) = (\xi x, \xi y, \theta), \quad \Gamma(\cdot) = (X(\cdot), Y(\cdot), \Theta(\cdot)) = \Phi_{\xi}(\gamma(\cdot)), \quad (74)$$

respectively.

Then, we have for all $t \in (0, 1)$:

$$\mathcal{F}_{0,\xi}(\gamma(t), \dot{\gamma}(t)) = \mathcal{F}_{0,1}(\Gamma(t), \dot{\Gamma}(t)). \quad (75)$$

The physical dimensions are consistent in this identity. In addition to that, we have for all $R > 0$,

$$\mathcal{S}_R^{\mathcal{F}_{0,\xi}}(\mathbf{p}_S) = \Phi_{\xi}^{-1}\left(\mathcal{S}_R^{\mathcal{F}_{0,1}}(\mathbf{p}_S)\right). \quad (76)$$

Proof. We divide the proof of the lemma in three parts:

1. Show that $\mathcal{F}_{0,\xi}(\gamma(t), \dot{\gamma}(t)) = \mathcal{F}_{0,1}(\Gamma(t), \dot{\Gamma}(t))$ for all $t \in (0, 1)$.
2. Show that $\mathcal{S}_R^{\mathcal{F}_{0,\xi}}(\mathbf{p}_S) = \Phi_{\xi}^{-1}\left(\mathcal{S}_R^{\mathcal{F}_{0,1}}(\mathbf{p}_S)\right)$ for all $R > 0$.
3. Show that the physical dimensions match of $\mathcal{F}_{0,\xi}(\gamma(\cdot), \dot{\gamma}(\cdot))$ and $\mathcal{F}_{0,1}(\Gamma(\cdot), \dot{\Gamma}(\cdot))$.

First, we show that Part 1 is true. We start with explicitly writing down both the left and right hand side of Eq. (75). Let $\gamma(t) := (\mathbf{x}(t), \mathbf{n}(t))$ and $\Gamma(t) := (\mathbf{X}(t), \mathbf{N}(t))$. Then,

$$\mathcal{F}_{0,\xi}(\gamma(t), \dot{\gamma}(t))^2 = \begin{cases} \xi^2 |\dot{\mathbf{x}}(t) \cdot \mathbf{n}(t)|^2 + \|\dot{\mathbf{n}}(t)\|^2 & \text{if } \dot{\mathbf{x}}(t) \propto \mathbf{n}(t) \\ +\infty & \text{else.} \end{cases} \quad (77)$$

$$= \begin{cases} |\xi \dot{\mathbf{x}}(t) \cdot \mathbf{n}(t)|^2 + \|\dot{\mathbf{n}}(t)\|^2 & \text{if } \dot{\mathbf{x}}(t) \propto \mathbf{n}(t) \\ +\infty & \text{else.} \end{cases} \quad (78)$$

$$\mathcal{F}_{0,1}(\Gamma(t), \dot{\Gamma}(t))^2 = \begin{cases} |\dot{\mathbf{X}}(t) \cdot \mathbf{N}(t)|^2 + \|\dot{\mathbf{N}}(t)\|^2 & \text{if } \dot{\mathbf{X}}(t) \propto \mathbf{N}(t) \\ +\infty & \text{else.} \end{cases} \quad (79)$$

We see that Eq. (78) and (79) are equal iff $\|\dot{\mathbf{n}}(t)\|^2 = \|\dot{\mathbf{N}}(t)\|^2$ and $\dot{X}(t) = \xi \dot{\mathbf{x}}(t)$. The last part automatically follows from (74). The first part is easily proven by looking at the following:

$$\begin{aligned} \mathbf{n}(t) &= \lambda \begin{pmatrix} \cos \theta(t) \\ \sin \theta(t) \end{pmatrix}, \text{ so } \dot{\mathbf{n}} = \lambda \begin{pmatrix} -\sin \theta \\ \cos \theta \end{pmatrix}, \text{ and hence } \|\dot{\mathbf{n}}(t)\| = \lambda; \\ \mathbf{N}(t) &= \lambda \begin{pmatrix} \cos \Theta(t) \\ \sin \Theta(t) \end{pmatrix}, \text{ so } \dot{\mathbf{N}} = \lambda \begin{pmatrix} -\sin \Theta \\ \cos \Theta \end{pmatrix}, \text{ and hence } \|\dot{\mathbf{N}}(t)\| = \lambda. \end{aligned}$$

So we find $\|\dot{\mathbf{n}}(t)\| = \|\dot{\mathbf{N}}(t)\|$. Therefore, we can conclude that Eq. (77) and (79) are equal. In other words, for every $t \in [0, 1]$ and $\dot{\mathbf{x}}(t) \propto \dot{\mathbf{n}}(t)$, we have

$$\begin{aligned} \mathcal{F}_{0,1} \left(\Gamma(t), \frac{d\Gamma}{dt}(t) \right) &= 1 |\dot{\mathbf{X}}(t) \cdot \mathbf{N}(t)|^2 + |\dot{\Theta}(t)|^2 \\ &= \xi^2 |\dot{\mathbf{x}}(t) \cdot \mathbf{n}(t)| + |\dot{\theta}(t)|^2 \\ &= \mathcal{F}_{0,\xi}(\gamma(t), \dot{\gamma}(t)). \end{aligned}$$

We move on to Part 2 of the proof. We start proving that the statement holds from left to right (\Rightarrow). Let

$$\mathbf{p} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \in S_R^{\mathcal{F}_{0,\xi}}(\mathbf{p}_S).$$

Then, we have

$$d_{\mathcal{F}_{0,\xi}}(\mathbf{p}_S, \mathbf{p}) = R = \int_0^1 \mathcal{F}_{0,\xi}(\gamma_m(t), \dot{\gamma}_m(t)) dt,$$

where γ_m is a minimizing curve connecting \mathbf{p}_S and \mathbf{p} . Now,

$$\Gamma_m(t) = \Phi_\xi(\gamma_m(t)) = \begin{pmatrix} \xi x_m(t) \\ \xi y_m(t) \\ \theta_m(t) \end{pmatrix}$$

satisfies

$$\mathcal{F}_{0,1}(\Gamma_m(t), \dot{\Gamma}_m(t)) = \mathcal{F}_{0,\xi}(\gamma_m(t), \dot{\gamma}_m(t)).$$

This holds because of what we have proven in Part 1 since it holds for all pairs $(\gamma(t), \Gamma(t))$. Then, Γ_m is also the minimizing curve for

$$d_{\mathcal{F}_{0,1}}(\Phi_\xi(\mathbf{p}_S), \Phi_\xi(\mathbf{p})) = \min_{\substack{\Gamma(0)=\mathbf{p}_S, \\ \Gamma(1)=\Phi_\xi(\mathbf{p})}} \int_0^1 \mathcal{F}_{0,1}(\Gamma(t), \dot{\Gamma}(t)) dt = R, \quad \mathbf{p}_S = \Phi(\mathbf{p}_S) \text{ since } \mathbf{p}_S = (0, 0, 0).$$

Hence, we can conclude that $\mathbf{p} \in S_R^{\mathcal{F}_{0,\xi}}(\mathbf{p}_S)$ implies that

$$\Phi_\xi(\mathbf{p}) = \begin{pmatrix} \xi x \\ \xi y \\ \theta \end{pmatrix} = \begin{pmatrix} X \\ Y \\ \Theta \end{pmatrix} \in S + R^{\mathcal{F}_{0,1}}(\mathbf{p}_S).$$

Rewriting this yields $\mathbf{p} \in \Phi_\xi^{-1}(S^{\mathcal{F}_{0,1}}(\mathbf{p}_S))$.

Next, we need to show that the reverse also holds (\Leftarrow). Let $\mathbf{p} \in \Phi_\xi^{-1}(S^{\mathcal{F}_{0,1}}(\mathbf{p}_S))$ be given. Then we have $\Phi_\xi(\mathbf{p}) \in S^{\mathcal{F}_{0,1}}(\mathbf{p}_S)$. That means that we find

$$d_{\mathcal{F}_{0,1}}(\mathbf{p}_S, \Phi(\mathbf{p})) = R = d_{\mathcal{F}_{0,1}}(\Phi(\mathbf{p}_S), \Phi(\mathbf{p})) = \min_{\substack{\Gamma(0)=\mathbf{p}_S, \\ \Gamma(1)=\mathbf{p}}} \int_0^1 \mathcal{F}_{0,1}(\Gamma(t), \dot{\Gamma}(t)) dt.$$

by the way $S_R^{\mathcal{F}_{0,1}}$ is defined in Eq. (73). Using analogous argumentation as before, we find $\mathbf{p} \in S_R^{\mathcal{F}_{0,1}}(\mathbf{p}_S)$.

Last, we show that the physical dimensions match, as Part 3 of this proof. We note that we have dimensions

$$[x] = \text{Length}; \quad [y] = \text{Length}; \quad [\theta] = 1; \quad [\xi] = \text{Length}^{-1}.$$

Therefore, we have

$$[X] = [\xi x] = 1; \quad [Y] = [\xi x] = 1; \quad [\Theta] = 1.$$

This results in

$$\begin{aligned} [\mathcal{F}_{0,\xi}(\gamma(\cdot), \dot{\gamma}(\cdot))] &= [|\xi \dot{\mathbf{x}}(t) \cdot \mathbf{n}(t)|^2 + \|\dot{\mathbf{n}}(t)\|^2] = 1 \\ [\mathcal{F}_{0,1}(\Gamma(\cdot), \dot{\Gamma}(\cdot))] &= [|\dot{\mathbf{X}}(t) \cdot \mathbf{N}(t)|^2 + \|\dot{\mathbf{N}}(t)\|^2] = 1, \end{aligned}$$

which finalizes our proof. \square

This means that we can compare the results of the different solution strategies (exact, PDE-solver and Fast Marching). Now, we have given an elaborate explanation on how and why we are able to compare the different results.

3.4.1 Comparison of Exact Solution and PDE-method

Before we are able to compare the solutions of the PDE-method to the exact solutions, we need to find a relation between ξ_{Exact} and ξ_{PDE} being the stiffness parameters of the exact solution and the PDE-method respectively. We claim that the correct ratio is given by

$$\xi_{\text{PDE}} = \xi_{\text{Exact}} \Delta r, \quad (80)$$

where Δr is the distance between two points in the discretization. In practice, we chose $\Delta r = \Delta x = \Delta y$, so therefore we have in the exact interval $\Delta r = \frac{x_{\text{max}} - x_{\text{min}}}{N_x - 1}$, with N_x the number of pixels in the x -direction. This claim is supported when we take a look at the discretization of the PDE. We recall the discretizations introduced in Eq. (60) and Eq. (61). For the PDE-method, we implicitly assume that the distance between two pixels equals 1. Then the corresponding finite difference formula is given by

$$w_{i,j,k}^{n+1} = w_{i,j,k}^n - \Delta t \cdot \left(\sqrt{\left(\frac{1}{\xi_{\text{PDE}}} (\tilde{A}_1 w_{i,j,k}^n) \right)^2 + \left(\frac{\zeta}{\xi_{\text{PDE}}} (\tilde{A}_2 w_{i,j,k}^n) \right)^2 + \left((\tilde{A}_3 w_{i,j,k}^n) \right)^2 - 1} \right), \quad (81)$$

as introduced in (61). However, we want our method to be able to handle situations in which the distance between two grid points is not equal to one, but still have the same results. Therefore, we rewrite Eq. (81) to the following system

$$w_{i,j,k}^{n+1} = w_{i,j,k}^n - \Delta t \cdot \left(\sqrt{\left(\frac{1}{\xi_{\text{Exact}}} (A_1 w_{i,j,k}^n) \right)^2 + \left(\frac{\zeta}{\xi_{\text{Exact}}} (A_2 w_{i,j,k}^n) \right)^2 + \left((A_3 w_{i,j,k}^n) \right)^2 - 1} \right) \quad (82a)$$

$$= w_{i,j,k}^n - \Delta t \cdot \left(\sqrt{\left(\frac{1}{\xi_{\text{PDE}}} (\tilde{A}_1 w_{i,j,k}^n) \right)^2 + \left(\frac{\zeta}{\xi_{\text{PDE}}} (\tilde{A}_2 w_{i,j,k}^n) \right)^2 + \left((\tilde{A}_3 w_{i,j,k}^n) \right)^2 - 1} \right) \quad (82b)$$

$$\Leftrightarrow \xi_{\text{PDE}} = \xi_{\text{Exact}} \Delta r. \quad (82c)$$

It is still important that we have $\Delta x = \Delta y$.

We support this claim by analysing the results for different parameter choices. First, we check whether the results are the same for different $\Delta x = \Delta y = 0.12$ and $\Delta x = \Delta y = 0.24$ on the domain $[-6, 6] \times [-6, 6] \times [0, 2\pi)$ with $N_x = N_y = 101$ and $N_x = N_y = 51$ respectively. We use $N_\theta = 32$ for both simulations. The results are visible in Figure 20. We see that the calculated solutions roughly follow the exact solution, which is what we hoped to find. In addition to that, we see that the calculated solutions are the same for both choices of $N_x = N_y$. Due to numerical approximation errors, some of the solutions differ from the exact solution. However, the calculated curves are a decent match to the exact solutions.

Next, we take a look at the importance of the pixels being the same size, so $\Delta x = \Delta y$ where Δx is as defined earlier, and $\Delta y = \frac{y_{\text{max}} - y_{\text{min}}}{N_y - 1}$. Implicitly, we have assumed in our calculation that this is the case as we illustrated in the calculation in Eq. (82), where it is assumed that the grid size equals 1. However, this grid size differs when we try to approximate the solution on the domain $[-6, 6] \times [-6, 6] \times [0, 2\pi)$ with the same amount of grid points. Rescaling our image, which used to be on the domain $[1, N_x] \times [1, N_y] \times [1, N_\theta]$, also means that we have to rescale the length of the pixels in x -, y - and θ -direction. In Figure 21, we show that it is not possible to rescale the solution to the PDE-method such that it fits the exact solution better for all solutions, either in the x - or the y -direction.

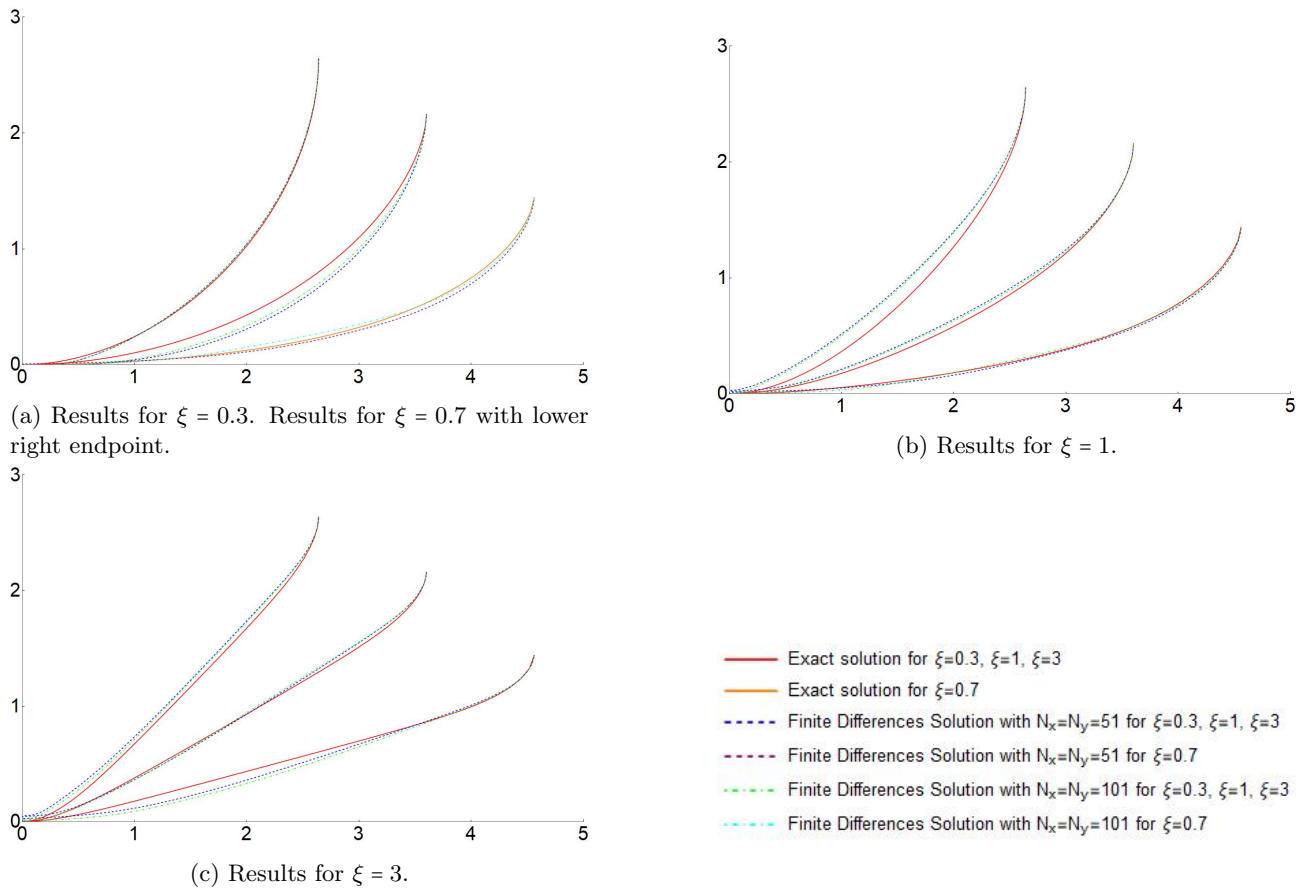


Figure 20: Results of ξ_{PDE} for different numbers of pixels $N_x = N_y$ and different values for ξ_{Exact} .

Lastly, we illustrate that the importance to ensure that the ratio between the number of angles that are considered and the number of pixels is large enough. In Figure 22, we show what happens for different numbers of orientations. We see that for a low number of orientations, the tracking will move into one direction, do an in place rotation and move towards the end point. For higher numbers of orientations, the calculation time gets a lot higher. In addition to that, when the ratio N_x/N_θ is too small, we find that the simulation becomes unstable. Therefore, we need to use a good ratio between $N_x = N_y$ and N_θ .

Remark 3.3. We note that the computation time to build a volume and calculate the shortest path in $(N_x, N_y, N_\theta) = (51, 51, 32)$, $(N_x, N_y, N_\theta) = (51, 51, 64)$ and $(N_x, N_y, N_\theta) = (51, 51, 128)$ are respectively 19.8s, 37.0s and 150.7s. These calculations have been performed in the left invariant frame. Calculation times for the gauge frame are even higher.

3.4.2 Comparison of Exact Solution and Fast Marching

We end the comparison of calculated results and the exact solution with the fast marching method. The input we need to give, is the metric itself, instead of the dual metric, which we used in the PDE-solver. Therefore, we find that the relation between ξ_{Exact} and ξ_{FM} is given by

$$\xi_{\text{FM}} = \xi_{\text{Exact}}^{-1}. \quad (83)$$

We check whether this is correct by comparing the results we find to the exact solution. In Figure 23, we see that both solutions are of similar form. However, there are still some deviations, where the biggest deviations appear in the tracking to the most right end point. We notice that with more orientations, these deviations become smaller. That suggests that they are (partly) due to discretization errors. Therefore, we conclude that Eq. (83) expresses the correct relation between ξ_{Exact} and ξ_{FM} .

Remark 3.4. The computation times for the fast marching algorithm executed on $(N_x, N_y, N_\theta) = (51, 51, 32)$, $(N_x, N_y, N_\theta) = (51, 51, 64)$ and $(N_x, N_y, N_\theta) = (51, 51, 128)$ are respectively approximately 3s, 5.7s and 11.1s. The fast marching algorithm is approximately 6 to 12 times faster than the finite difference model. Similar relations hold for the calculation times for the finite differences and fast marching in the gauge frame.

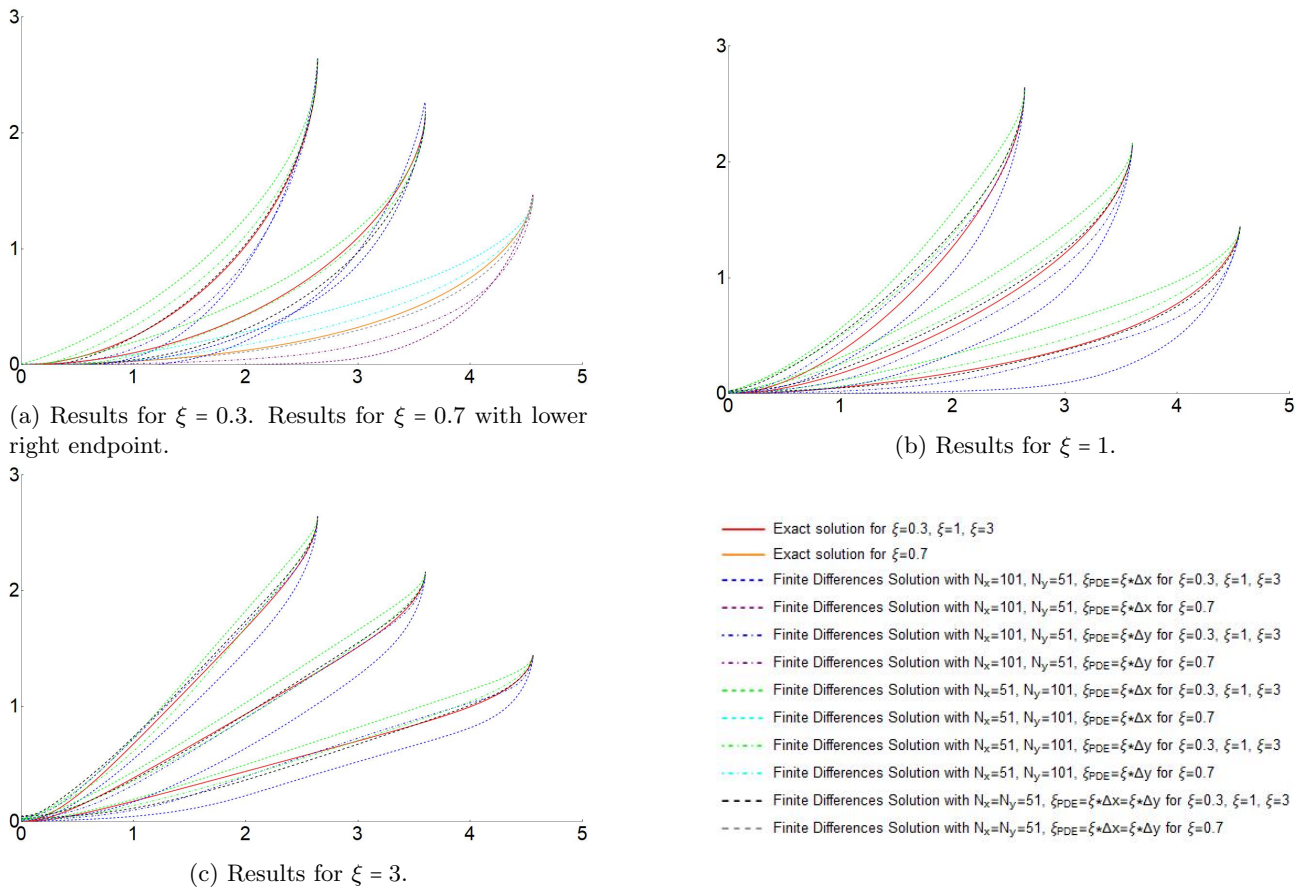


Figure 21: Results illustrating the importance that $\Delta x = \Delta y$ in the calculations.

3.4.3 Conclusion

We have compared the finite difference and fast marching solutions to the exact solution. We concluded for the finite differences and fast marching methods that the underlying relation to ξ_{Exact} was given by

$$\begin{aligned}\xi_{\text{FM}} &= \xi_{\text{Exact}}^{-1} \\ \xi_{\text{PDE}} &= \xi_{\text{Exact}} \Delta r\end{aligned}$$

respectively. That means that we find that the relation between ξ_{FM} and ξ_{PDE} is given by

$$\frac{\xi_{\text{PDE}}}{\Delta r} = \xi_{\text{FM}}^{-1}.$$

In addition to this relation between both weighting parameters, we found the following conclusions:

1. When using the finite difference approach:
 - (a) It is important that we have $\Delta r = \Delta x = \Delta y$, otherwise the aforementioned relation does not hold because of underlying assumptions in the Lie package for Mathematica. This is also illustrated in Figures 20 and 21.
 - (b) It is essential that we consider sufficient angular directions, as illustrated by Figure 22. When we consider a limited number of orientations, the result is less precise.
2. When using the fast marching algorithm:
 - (a) It is crucial that we use sufficient orientations. In Figure 23, we saw that using more orientations results in more accurate tracking results.
3. The calculation times of the fast marching algorithm are much lower compared to the finite difference approach. On the other hand, the finite difference approach is more accurate than the fast marching algorithm. Therefore, we need to decide per experiment what is more important to us while calculating the results; accuracy or calculation times.

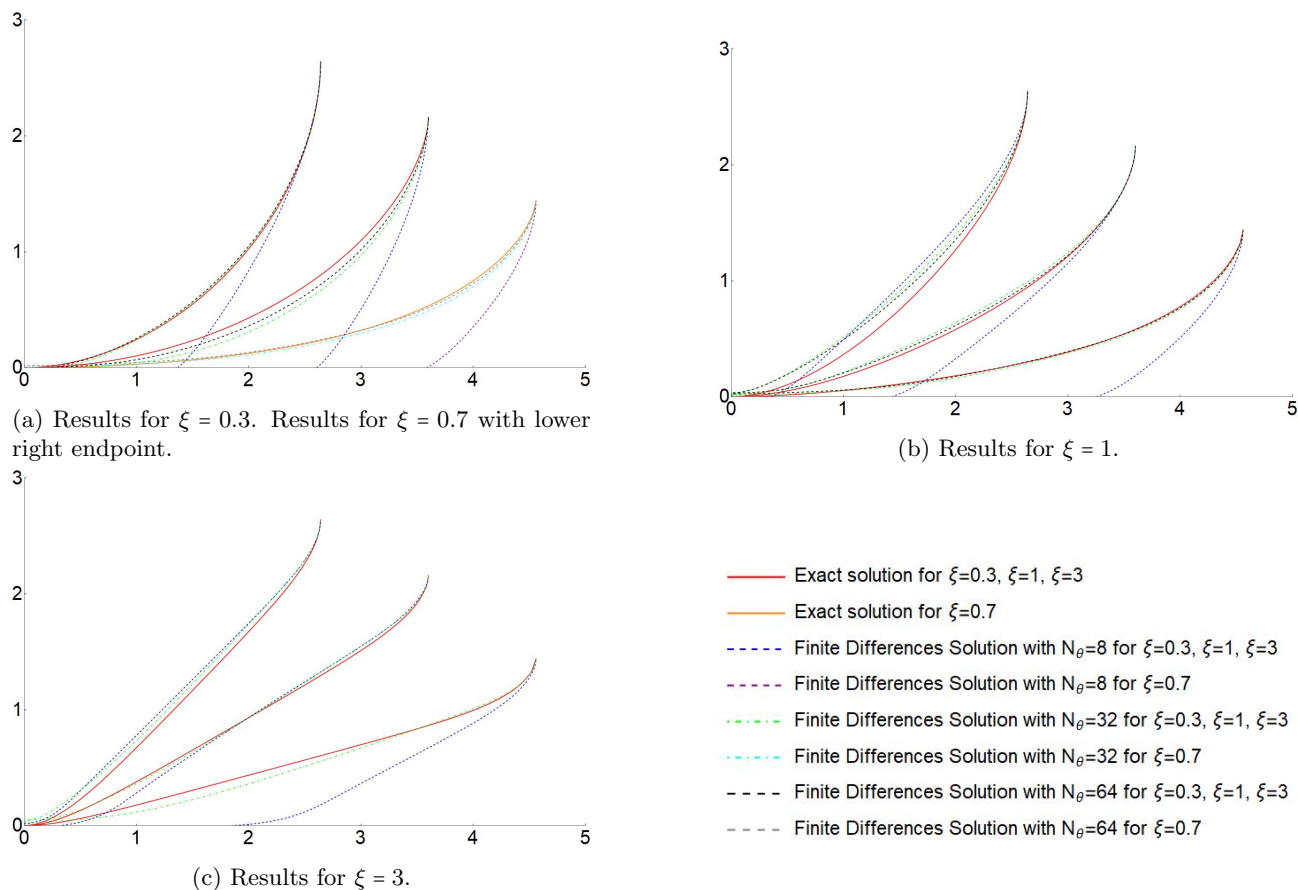


Figure 22: Results for different ratios of the number of orientations N_θ and the number of pixels $N_x = N_y = 51$, illustrating the importance of using sufficient angular directions.

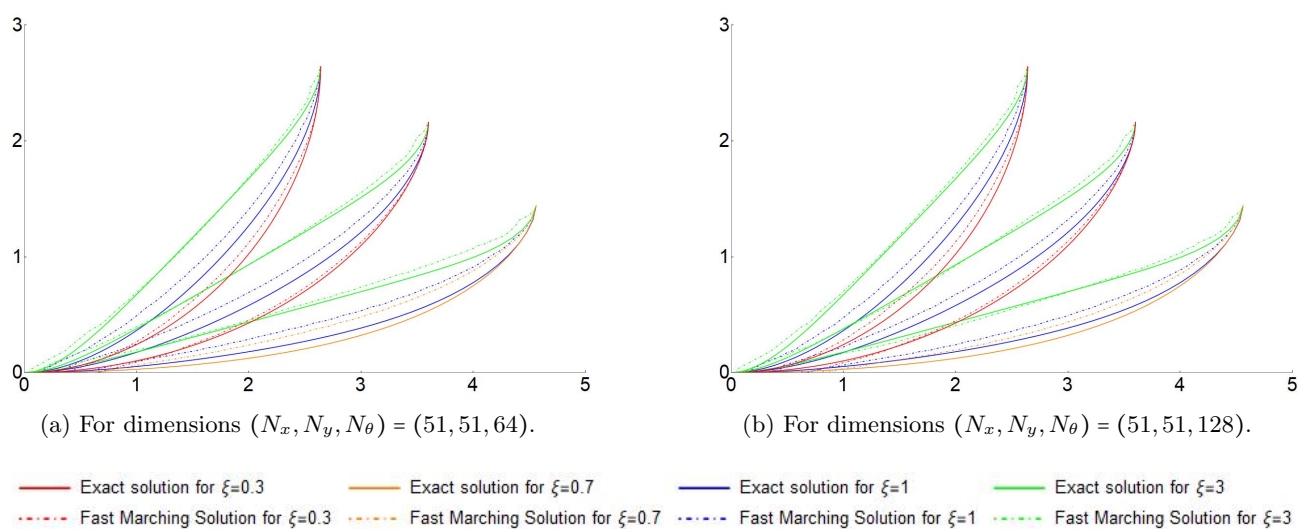


Figure 23: Comparison of fast marching results (dash-dotted lines) with the exact solution (solid lines). The red, orange, blue and green lines represent the results for the values $\xi = 0.3, \xi = 0.7, \xi = 1$ and $\xi = 3$ respectively.

4 Vessel Tracking

In this section, we will discuss the results of the vessel tracking. First, we will give an overview of the different parameters that play a role in the metric tensor fields as defined in Section 2 and how they are defined. Then, we discuss the results for different experiments that we conducted.

4.1 Experimental Settings

We start with introducing different ways to construct a cost function. Thereafter, we define the values of the different parameters that are used in the calculations for the different experiments.

4.1.1 Cost Function

There are different ways in which the cost function can be defined. Here, we will mention the different versions that we implemented.

1. The first cost function that we implemented is the same as introduced in [6]. We define

$$\mathcal{C}(x, y, \theta) = \frac{1}{1 + \lambda \left| \frac{(A_2^2 U_f)_+(x, y, \theta)}{\|(A_2^2 U_f)_+\|_\infty} \right|^p}, \quad (84)$$

where $\lambda \geq 0$, $p \in \mathbb{N}$, $\|\cdot\|_\infty$ the supremum-norm and $U_f : SE(2) \rightarrow \mathbb{R}$ is a lift of the image. We note that \mathcal{C} is bounded from below and above by $\delta = \frac{1}{1+\lambda} \leq \mathcal{C} \leq 1$. The parameter λ increases the contrast in the cost function, and the parameter p controls its steepness. Recall that the lift is done using real-valued anisotropic wavelets ψ :

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

Note that the cost function in Eq. (84) relies on the left invariant frame.

2. Since we are not always just considering the left invariant frame, we also implemented a cost function that relies on the gauge frame. It is introduced by Zhang [32]. First, we introduce the second order derivatives w.r.t. the Gauge Frame as

$$\begin{aligned} \Phi_2^{\sigma_s, \sigma_o}(U_f) &:= \mathcal{B}_{22}(G_{\sigma_s, \sigma_o} * U_f), \\ \Phi_3^{\sigma_s, \sigma_o}(U_f) &:= \mathcal{B}_{33}(G_{\sigma_s, \sigma_o} * U_f), \end{aligned}$$

where $\mathcal{B}_{22} := \mathcal{B}_2 \mathcal{B}_2$ and $\mathcal{B}_{33} := \mathcal{B}_3 \mathcal{B}_3$ denote the second derivatives with respect to the second and third Gauge frame vector respectively and where $G_{\sigma_s, \sigma_o}(\mathbf{x}, \theta) := G_{\sigma_s}(\mathbf{x}) \cdot G_{\sigma_o}(\theta)$ blurs the image resulting in more smooth derivatives. Then, the cost function is calculated by

$$\mathcal{C}^{\sigma_s, \sigma_o}(\mathbf{x}, \theta) = \frac{1}{1 + \lambda \left| \frac{(-\Phi_2^{\sigma_s, \sigma_o}(U_f)(\mathbf{x}, \theta))_+}{\|(-\Phi_2^{\sigma_s, \sigma_o} U_f)_+\|_\infty} \right|^p}. \quad (85)$$

In addition to that, we also implemented the cost function

$$\mathcal{C}^{\sigma_s, \sigma_o}(\mathbf{x}, \theta) = \frac{1}{1 + \lambda \left| \frac{(-\Phi_2^{\sigma_s, \sigma_o}(U_f)(\mathbf{x}, \theta) - \Phi_3^{\sigma_s, \sigma_o}(U_f)(\mathbf{x}, \theta))_+}{\|(-\Phi_2^{\sigma_s, \sigma_o} U_f - \Phi_3^{\sigma_s, \sigma_o} U_f)_+\|_\infty} \right|^p}. \quad (86)$$

3. The last cost function that we introduce is based on the orientation scores of the Gauge Frame, and hence are given by

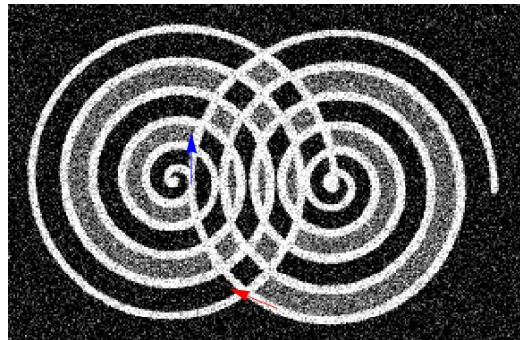
$$\mathcal{C}(\mathbf{x}, \theta) = \frac{1}{1 + \lambda |S^k(U_f)(\mathbf{x}, \theta)|^p}, \quad (87)$$

where $\lambda \geq 0$, $p \in \mathbb{N}$, $\|\cdot\|_\infty$ the sup-norm, and where $S^k(U_f)$ is as defined in Eq. (41). The main difference with Eq. (86) is that we omitted the blurring of the lifted orientation scores.

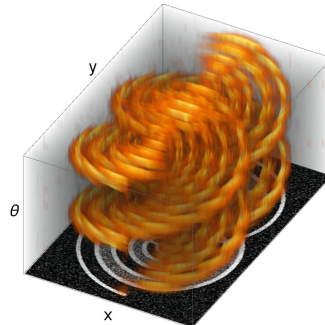
The values of p and λ can be chosen freely, but in this section, we chose to use $p = 2$ and $\lambda = 200$ in order to get the best results.

	ξ	N_θ	σ_s	σ_o	ε	β	k
Experiment 1 & 2	0.08	16	2.8	$0.87s_\theta$	0	5	1
Experiment 3	0.08	16	1.5	$0.87s_\theta$	0	5	1
Experiment 4	0.05	16	7	$1.2s_\theta$	0	5	1

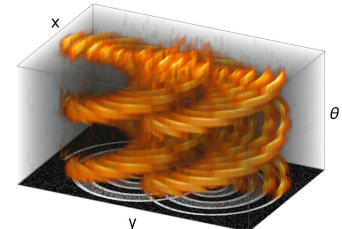
Table 1: Values used for the different parameters in the experiments where $s_\theta = \frac{2\pi}{N_\theta}$.



(a) The task is to trace the line starting in the red arrow to the blue arrow with corresponding orientations.



(b) Orientation Scores of the image in Fig. 24a.



(c) Orientation Scores of the image in Fig. 24a.

Figure 24: Experiment 1: setting and visualization of orientation scores.

4.1.2 Parameter settings

In the experiments, the parameters take the values as mentioned in Table 1, unless mentioned otherwise. In addition to that, we speak of an isotropic metric tensor field when $\zeta = 1$ and of an anisotropic metric tensor field when $\zeta = 1/8$. The values of α and β are mentioned per experiment.

4.2 Visualization Results

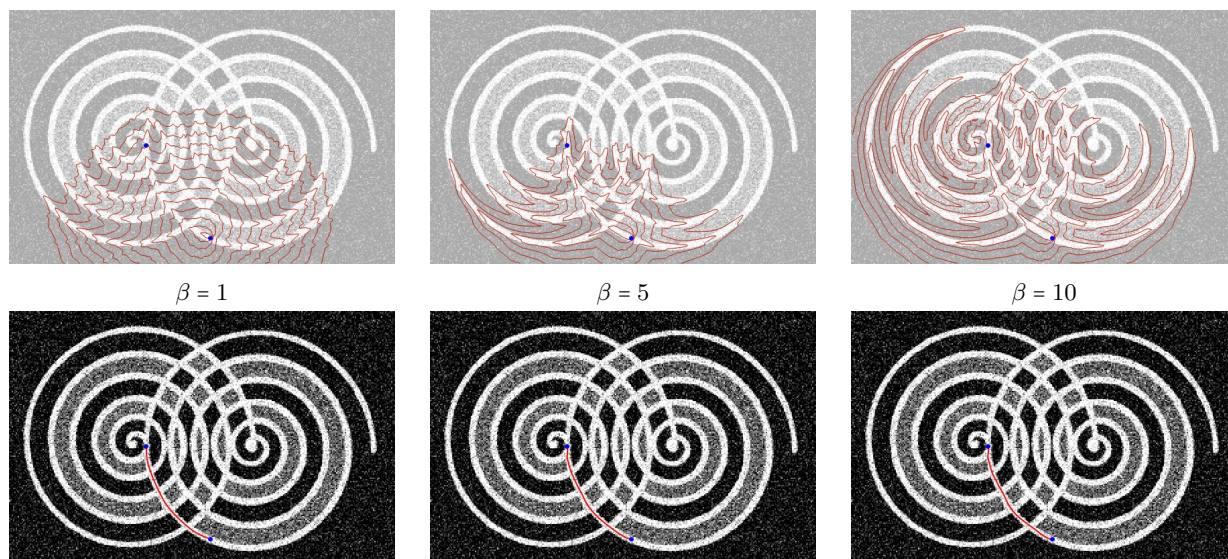
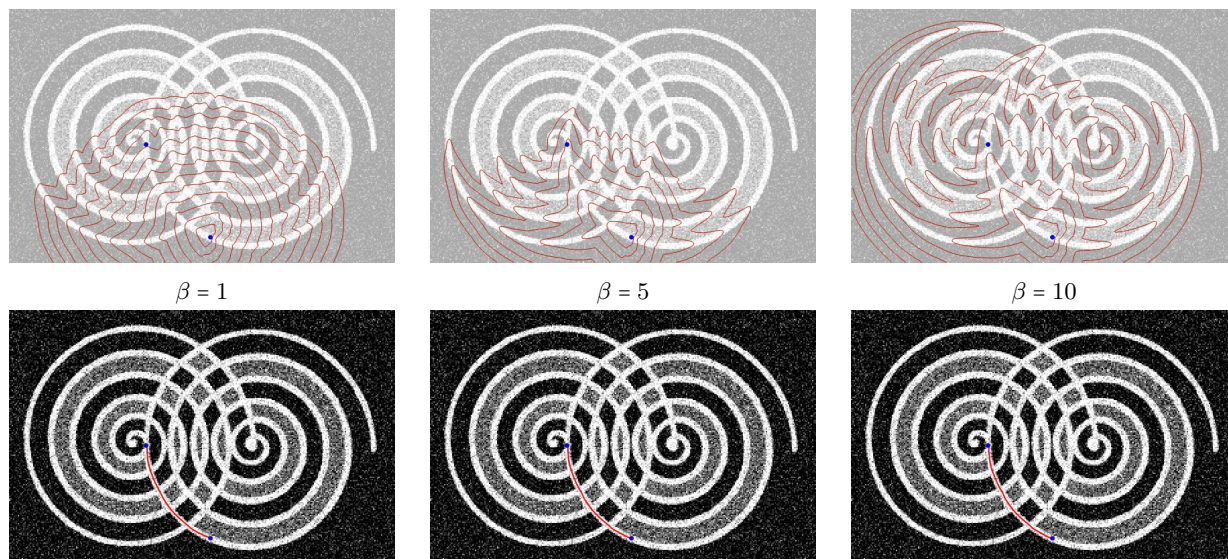
Before we show the results, we first explain how they are visualized. For each tracking, we show both the distances to the seed, projected onto the image. We show them as level curves on with the original image as background. In addition to that, we also show the tracking. Since our calculation are done in $SE(2)$, and we are considering an image in \mathbb{R}^2 , the geodesics are also projected onto the image.

4.3 Experiments

4.3.1 Experiment 1

The goal of the first experiment is to get acquainted with the way different metrics work and the influence of the different parameters. In Figure 24a, the setting of the experiment is clarified. For these experiments, we have used the orientation scores of the Gauge Frame to construct the cost function as written down in Eq. (87). We take a closer look at the two different frames that one can use and the influence of the parameter β . The orientation scores of the image are visualized in Figure 24b and 24c. These orientation scores are used while calculating the cost function used for Experiment 1 and 2 and the visualizations show how the different orientations are separated in the position and orientation space.

Left Invariant Frame For the (forward gear) left invariant frame, we considered both an anisotropic ($\zeta = 0.125$) and an isotropic ($\zeta = 1$) frame. In Figure 25 and Figure 26, we see the results for both models (anisotropic and isotropic respectively). Comparing the results for the same value of β , we note that the anisotropic left invariant frame gives fronts that are more narrow and sharper at the end than the isotropic left invariant frame. This is in line with our expectations when looking at the control sets, which are visualized in Figure 47. Here, we see that for the isotropic left invariant frame, the range of points with distance 1 from the origin is comparable to (half) a flattened ball. From the seed you can move the same distance (in the Euclidean frame) to the side and forward, and a smaller distance in the angular direction. However, this is not the case for the anisotropic frame. In that frame, the range of points with distance 1 from the origin is comparable to (half a) rugby ball. In the Euclidean distance, you can walk further forward than sideways and in the angular direction. Therefore, we expect the fronts to be pointier as well, which is the case.

Figure 25: Anisotropic forward gear left invariant frame where $\zeta = 0.125$.Figure 26: Isotropic forward gear left invariant frame where $\zeta = 1$.

Furthermore, we note that the vessel does not contain sharp turns. Therefore, we do not expect any cusps or in place rotations. The tracking result is a smooth line that follows the shape of the vessel as we expected.

Gauge Frame In addition to the left invariant frame, we also consider the gauge frame. The latter one should be able to follow line structures in a better way since it is able to bend along line structures as discussed in Section 1.4 and Section 2. In Figure 27, we see the results of the same experiments as deduced in Figure 25 and Figure 26, but now, using the gauge frame as the reference system. We see that it is harder to rotate in this frame, since sideways movement costs the same amount of energy as angular movement. In the left invariant frame, it was easier to move along the crossing vessels than it is in the gauge frame. This confirms that the gauge frame adheres better to the forward direction of the vessel structure.

In addition to that, we see that the tracking does not contain any cusps or in place rotations. The results are as expected.

Scaling parameter β Finally, we take a look at the influence of the parameter β in all three frames (isotropic, anisotropic left invariant, anisotropic gauge frame). The parameter β determines how much the frame needs to adhere to the cost function. In other words, this indicates how much it is punished for moving outside of the vessel structure. We see that this coincides with the distances to reach specific points calculated by the simulation. We see that the distance from the seed to a point between two vessels is higher for $\beta = 5$ than for $\beta = 1$. In addition to that, we also see that for $\beta = 10$, the simulation has trouble building the volume (thus

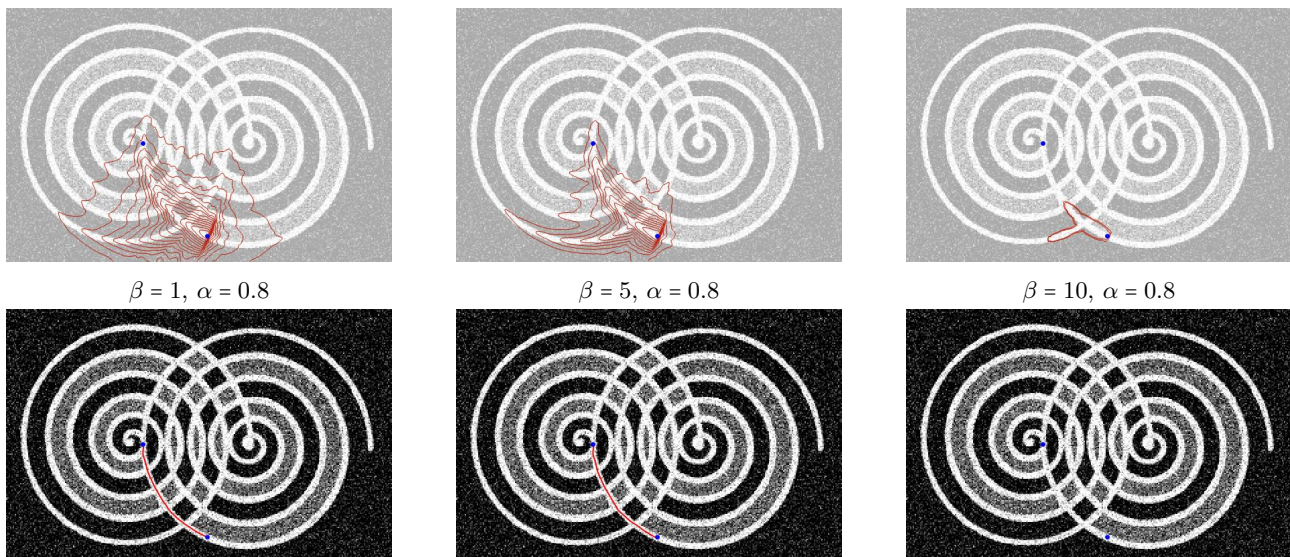


Figure 27: Anisotropic forward gear gauge frame.

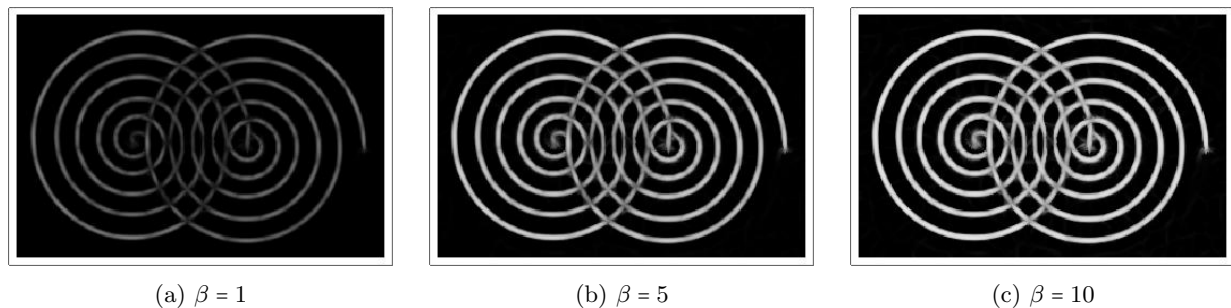


Figure 28: Visualization of the cost function \tilde{C} for different values of β , where the cost function is defined by $\tilde{C}(\mathbf{x}, \theta) = 1 + \beta \frac{\mathcal{C}(\mathbf{x}, \theta) - \min\{\mathcal{C}(\mathbf{x}, \theta)\}}{\max\{\mathcal{C}(\mathbf{x}, \theta)\} - \min\{\mathcal{C}(\mathbf{x}, \theta)\}}$ with \mathcal{C} as defined in Eq. (87). In these figures, black and white indicate high and low values respectively.

calculating the distances) in the gauge frame metric since the simulation becomes unstable. This is most likely due to using a too big step size in the integration step. In Figure 28, we see a visualization of the different cost functions. We see that the light areas coincide with the vessel structure in the image. The locations in these white areas have lower costs than the darker areas. Looking at the different solutions in Figure 25, Figure 26 and Figure 27 combined with the cost function in Figure 28, we decide that $\beta = 5$ is the best fit for the calculations.

4.3.2 Experiment 2

The second experiment uses the same image as Experiment 1. However, the start and end position are different, as you can see in Figure 29. With this setup, it is possible to take a different route than what we would expect (by making a turn at one of the intersections).

This experimental setting is ideal to show the **influence of the forward gear**. We expect that the forward gear models are more likely to follow the line structure as one would expect (without taking unnecessary turns). In addition to that, we expect the gauge frame to adhere more easily to the line structures, and therefore to be more likely to return the correct geodesic. In Figure 30, we see a comparison between models where the forward gear is and is not activated. We see that sometimes the model opts to use a “shortcut” by using the reverse gear, such as in Figure 30c. When we deactivate this possibility and force the tracking to keep moving forward, it is only possible that in place rotations take place.

In the isotropic left invariant frame, we see that the tracking favours to take a wrong turn when the forward gear is not activated. We do not see that the tracking moves backward. However, we do see that it takes a shortcut. When the forward gear is activated, the tracking follows the line structure as expected. This can be explained by looking at the distances to reach the seed from each point. In the symmetric variant of the model, it is cheaper to opt for the left vessel, whereas in the forward gear variant, following the vessel structure

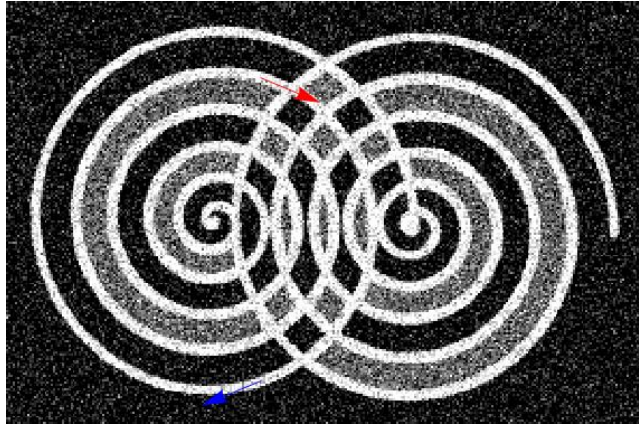


Figure 29: The task is to trace the line starting in the red arrow to the blue arrow with corresponding orientations.

is favoured (since it has lower costs).

In the symmetric anisotropic left invariant frame, we see that the tracking “switches gears” at the first crossing. Then, at the last crossing, it changes its moving direction again, after which it reaches its end point. In the forward gear variant of the model, the tracking does not switch gears anymore. Instead of taking the wrong exit at the first crossing, it follows the vessel structure. The tracking does not completely follow the correct path, since it uses the wrong vessel structure at the final crossing. However, the tracking does not need to change gears to complete the tracking.

In the symmetric anisotropic gauge frame, we see that the tracking uses the reverse gear at the first crossing, and at the last crossing. Consequently, the tracking differs from what we expected. However, when we activate the forward gear, we see that the tracking does not have an in place rotation at the place where the switching of gears used to take place. Instead, the tracking follows the vessel structure and takes a turn at the last crossing. This can be related to the functioning of the gauge frame. As we discussed before, the gauge frame adapts to the vessel structure. In this specific case, the vessels are very curvy and where the vessels touch each other, the frames adapt to the “wrong” vessel. Therefore, the tracking favours taking a slight turn at the last but one crossing.

4.3.3 Experiment 3 - Low Resolution

It is important to test our methods on retina images as well, since these are not necessarily of the same quality as the image that we used in the previous two experiments. Therefore, we first consider an image with a low spatial resolution. There are two blood vessels parallel to each other. Consequently, it is possible to take a different route to the end point, by opting for the lower vessel instead of the upper one, as you can see in Figure 31.

First, we take a closer look at the cost function that we will use. We note that, when we use the cost function as defined in Eq. (87), the tracking is tempted to take the lower vessel. We saw that the tracking is not correct in that case. This can be easily explained by looking at the cost function that we used. In Figure 32a and Figure 32b, the cost functions created by Eq. (87) are visualized. We note that the width of the upper vessel is quite small, which is a reasonable explanation for why the tracking favours the lower vessel. In addition to that, the costs of moving here are higher. In an effort to boost the upper vessels accessibility, we temper the costs at locations cheaper than the start or end point. We introduce the cost function given by

$$\mathcal{C}(\mathbf{x}, \theta) = \frac{1}{1 + \lambda |S^k(U_f)(\mathbf{x}, \theta) \cdot \exp((\min\{S^k(U_f)(\mathbf{x}_0, \theta_0), S^k(U_f)(\mathbf{x}_1, \theta_1)\} - S^k(U_f)(\mathbf{x}, \theta))_-)|^p}, \quad (88)$$

where $S^k(U_f)$ is defined in Eq. (41).

Remark 4.1. We note that, by definition, we have $S^k(V)(\mathbf{p}) \in [0, 1]$ for all $\mathbf{p} \in \mathbb{M}_2$. By taking the smallest orientation confidence at the start and end position and subtracting the orientation confidence value at each point, we determine where the orientation confidence is bigger and where it is smaller than this minimum value. By only considering the negative values, we ensure that we only consider those locations that are better aligned to the vessel structure. The exponential dampens the reaction. Finally, by multiplication to the original cost function, we find a dampened orientation confidence, which we use to calculate a cost function.

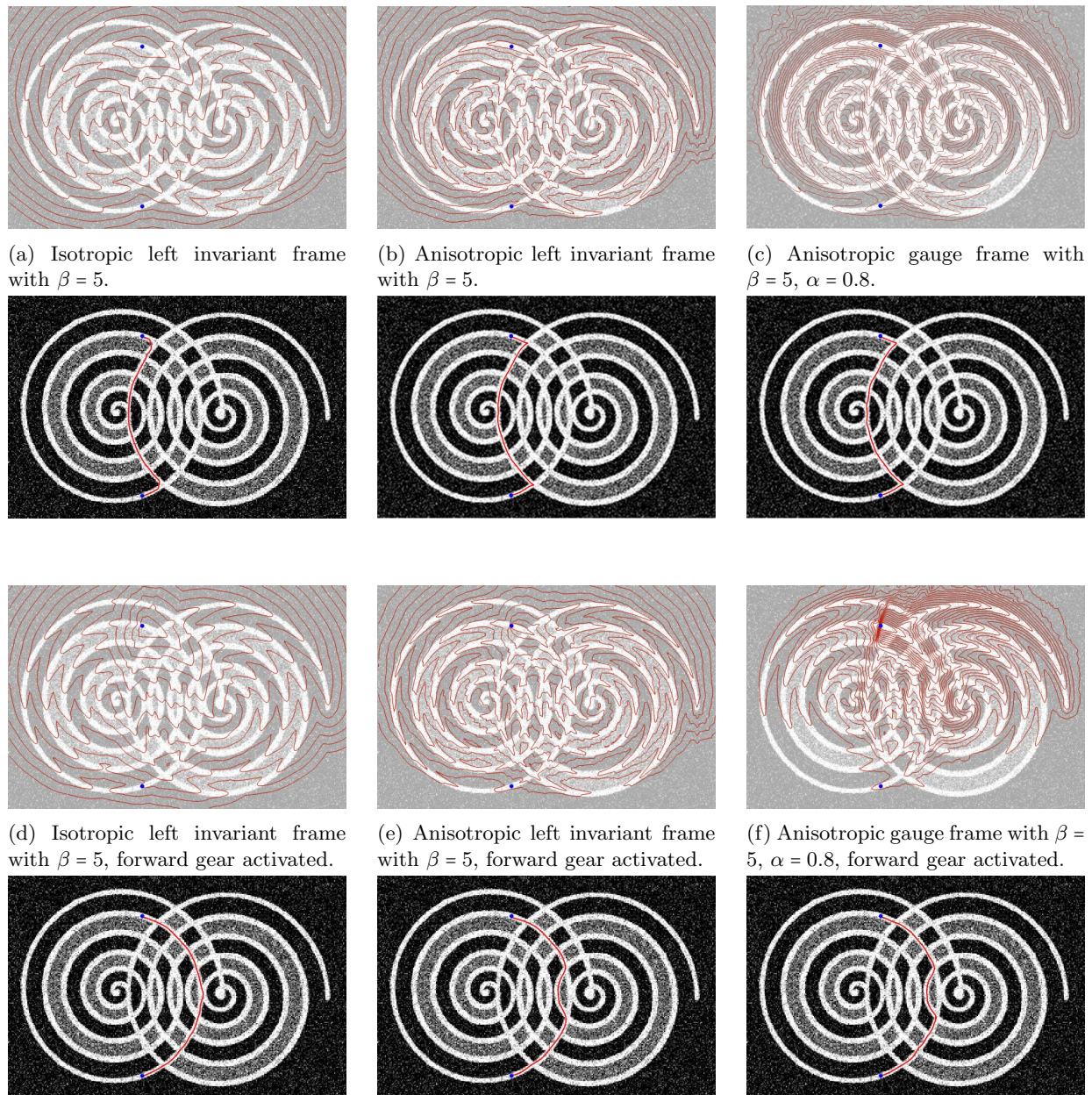


Figure 30: Comparison models with and without forward gear activated.

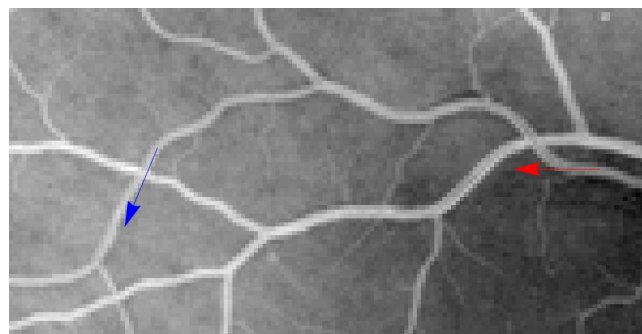


Figure 31: The task is to trace the line starting in the red arrow to the blue arrow with corresponding orientations.

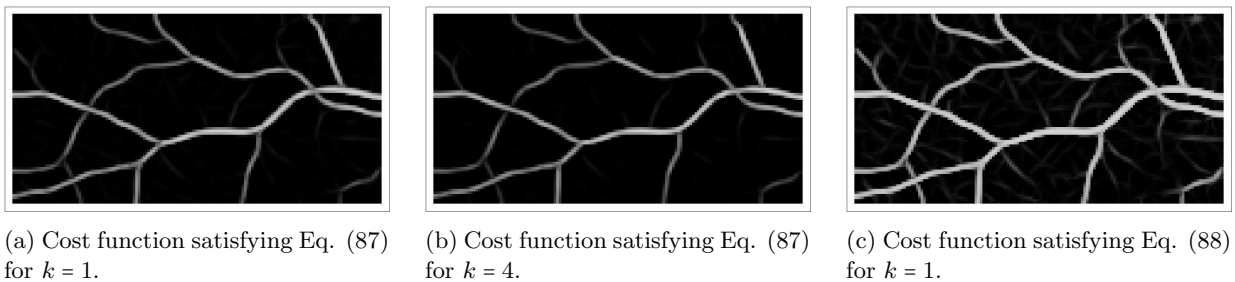


Figure 32: Visualization of different cost functions $\tilde{\mathcal{C}}$, where $\tilde{\mathcal{C}}(\mathbf{x}, \theta) = 1 + \beta \frac{\mathcal{C}(\mathbf{x}, \theta) - \min\{\mathcal{C}(\mathbf{x}, \theta)\}}{\max\{\mathcal{C}(\mathbf{x}, \theta)\} - \min\{\mathcal{C}(\mathbf{x}, \theta)\}}$, where \mathcal{C} is defined in Eq. (87) for Figure 32a-32b, and where \mathcal{C} is defined in Eq. (88) for Figure 32c. In these figures, black and white indicate high and low values respectively.

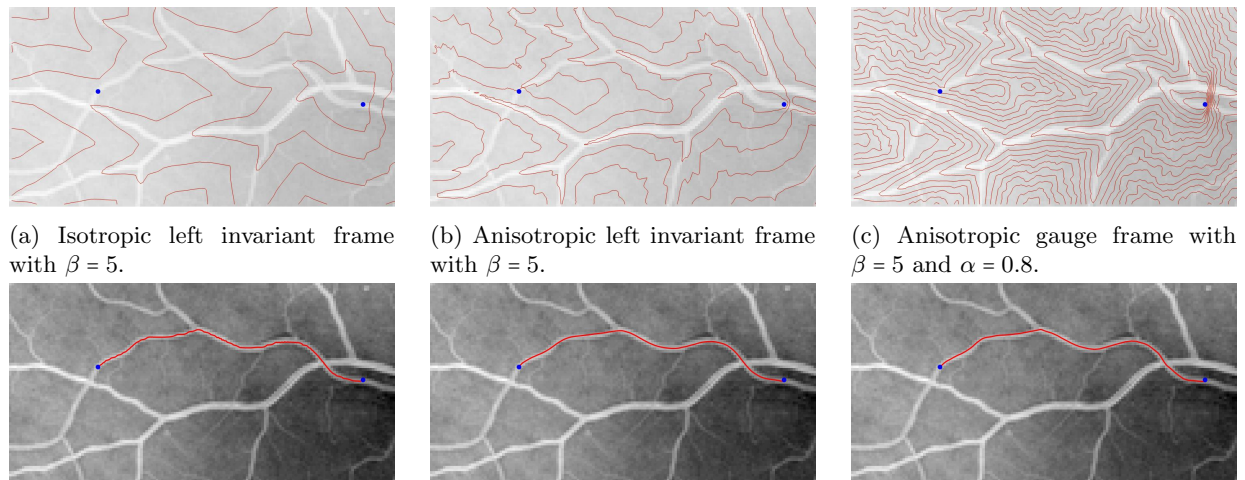


Figure 33: Results for different frames where $\beta = 5$. All simulations use $N_\theta = 16$ and $\xi_{\text{PDE}} = 0.08$.

In Figure 32c, we see that using the alternative cost function introduced in Eq. (88) ensures that the upper vessel is cheaper. Consequently, the tracking may be more likely to follow the upper vessel while calculating the shortest route. Therefore, we use the cost function introduced in Eq. (88) where $k = 1$ for this experiment.

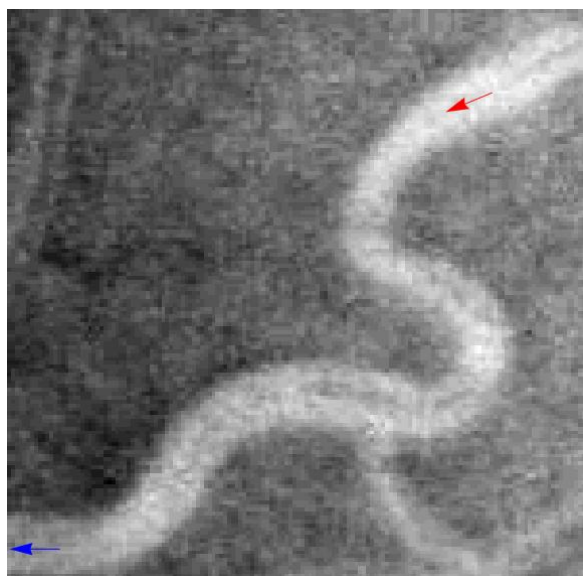
We study the tracking for the different frames. The results are visualized in Figure 33. First, we notice that the tracking chose the correct vessel for all three frames. The tracking never moves backwards. The only thing that is important to note is that the tracking in the isotropic left invariant frame is somewhat jittery (Figure 33a). This is most likely due to the fact that the tracking is allowed to make bigger steps to the sides, as we can see in the control sets (Figure 47).

4.3.4 Experiment 4 - High Tortuosity

Last, we consider an image containing a blood vessel that is highly tortuous. We are interested in knowing whether the models are still able to follow the vessel structure. The exact experimental setting is given in Figure 34 along with the used cost function. We will also look at the influence of the number of orientations and the value of the weighting parameter ξ in this experiment.

First, we take a look at the results for a reference case. In Figure 35, we have $N_\theta = 16$ and $\xi = 0.05$. We see that both the isotropic and anisotropic left invariant frame struggle following the line structure. They both take a shortcut. In other words, they do not follow the vessel as one would expect. On the other hand, we have the gauge frame, which is able to adapt to the line structure of the vessel very well. The tracking gives a good approximation of the vessel. There are no in place rotations or cusps, but the tracking gives a line that looks continuous.

Number of orientations We want to know the influence of the number of orientations. We expect that using less than 16 orientations leads to worse tracking results. We calculated the results for $N_\theta = 8$ which are visible in Figure 36. One can see immediately that the tracking results are less accurate than they were when $N_\theta = 16$ (in Figure 35). We see that the isotropic and anisotropic left invariant frame have trouble following the curvature of the vessel. Instead of bending along its structure, they take a shortcut and move outside the vessel, just like they did when we considered $N_\theta = 16$.

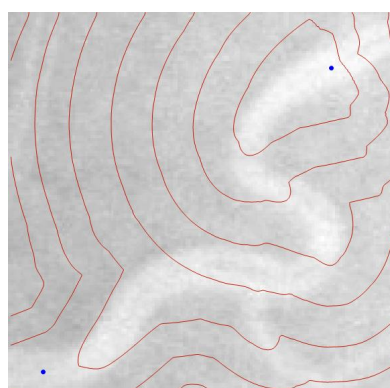


(a) The task is to trace the line starting in the red arrow to the blue arrow with corresponding orientations.

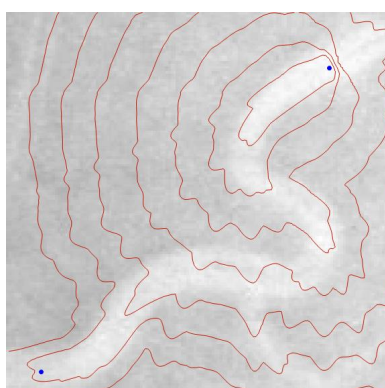


(b) Visualization of the used cost function introduced in Eq. (87). Dark areas represent high costs, where lighter areas represent lower costs.

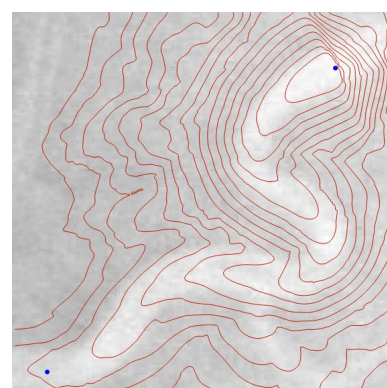
Figure 34: Visualization of both the experimental setting and the used cost function



(a) Isotropic Left Invariant Frame



(b) Anisotropic Left Invariant Frame



(c) Anisotropic Gauge Frame with $\alpha = 0.8$.

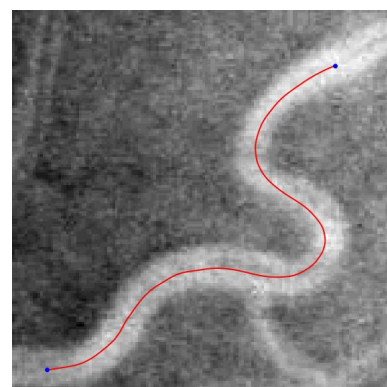
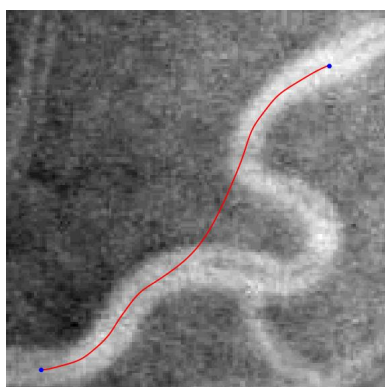
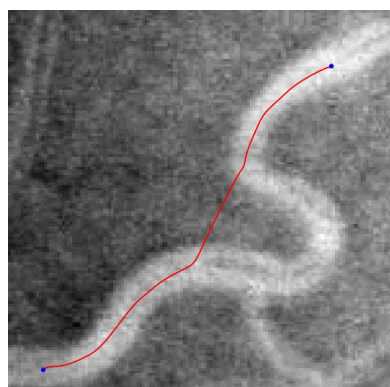


Figure 35: Results for the different frames where $\beta = 5$. All simulations use $N_\theta = 16$ and $\xi_{\text{PDE}} = 0.05$.

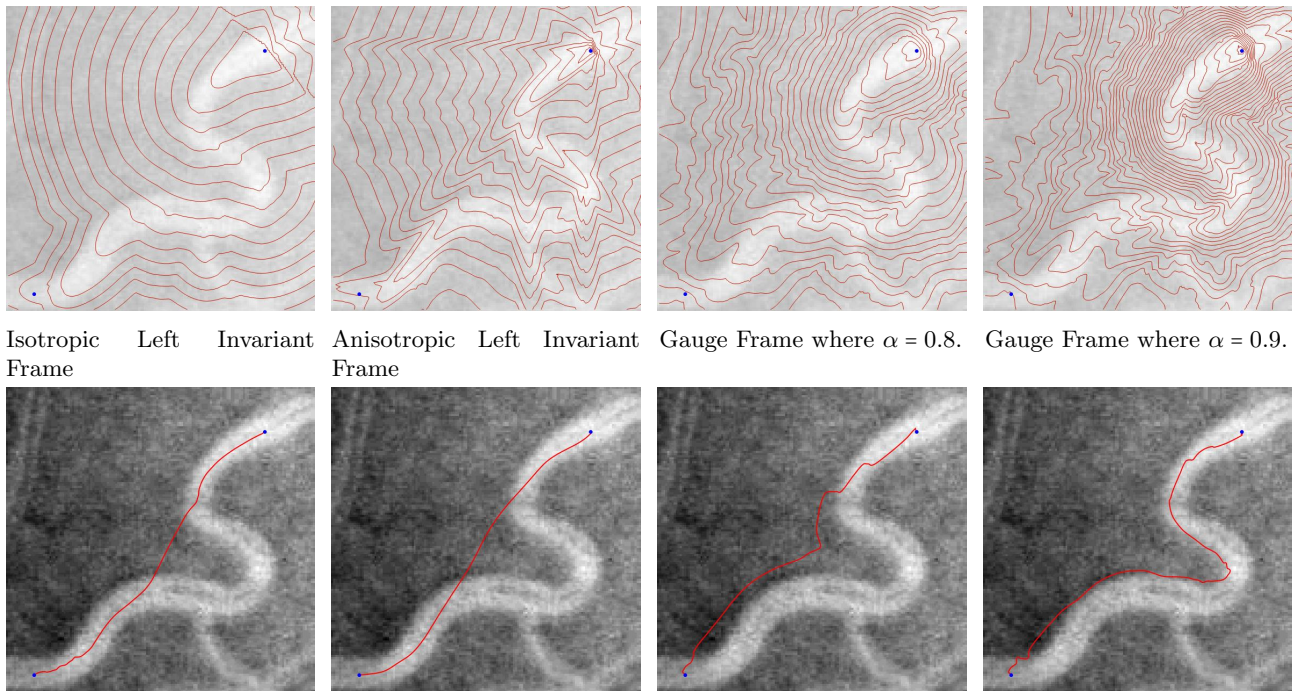


Figure 36: Results for the different frames where $\beta = 5$. All simulations use $N_\theta = 8$ and $\xi_{\text{PDE}} = 0.05$.

We consider the gauge frame using both $\alpha = 0.8$ and $\alpha = 0.9$. In the case where $\alpha = 0.8$, we see that the tracking does not give the correct result. The tracking is more dynamic, but it does not complete the tracking correctly. This could be related to a relative domination of the \mathcal{B}_2 and \mathcal{B}_3 direction over the \mathcal{B}_1 direction. Therefore, we check what happens when we consider $\alpha = 0.9$, resulting in a relatively bigger component in the \mathcal{B}_1 direction. In this case, we see that the tracking is able to follow the vessel structure better. Do note that it is not perfect; the tracking is somewhat jittery and does not always stay within the vessel. However, this tracking is much better than the one we found for $\alpha = 0.8$. This also illustrates a big advantage of the gauge frames. In this specific case, both left invariant frames were not able to calculate a correct tracking. However, using the anisotropic gauge frames, we were able to find a value of α which did result in a correct tracking.

Weighting parameter ξ We investigate the effect of the weighting parameter ξ . We will show that, for high values of ξ , the tracked vessel will be very flexible (like cooked spaghetti), where for low values of ξ , the tracked vessel will be hard to bend and thus, act like steel. In Figure 37, we see the results for $\xi_{\text{PDE}} = 0.01$, which is a low value compared to $\xi_{\text{PDE}} = 0.05$. In Figure 38, we see the results for $\xi_{\text{PDE}} = 0.25$, which is relatively high.

First, we discuss the results for a low value of ξ . In Figure 37, we see that the tracking results are different from what we would expect. For the isotropic left invariant frame, we see that the tracking moves straight ahead, rotates in one point, and then continues to move straight ahead for some distance. We see a clear difference between the symmetric model and the forward gear variant. In the first one, the tracking moves backward towards the tip. In the forward gear variant of the model, this is not allowed. Therefore, we see that the tracking changes. Instead, the tracking now moves from side to side inside the vessel structure, but does not move backwards, as was the case in the symmetric model. In the anisotropic left invariant frame, we see that the tracking results do not follow the vessel structure. However, the tracking makes fewer sharp angles, and therefore seems to be smoother. Towards the seed, the forward gear variant of the model makes a sudden movement, which appears to be backward. Most likely, an in place rotation takes place. In addition to that, we see that when the tracking approaches the seed in the forward gear variant of the anisotropic left invariant frame, it first passes the seed, after which it turns (using different smaller steps) in order to arrive at the correct location. In the symmetric version of the same metric frame, we see that the tracking follows a smooth line from tip to seed without passing the seed before arriving at the correct location. The last frame we consider, is the anisotropic gauge frame where $\alpha = 0.8$. In this model, we see that the tracking is very different for the symmetric and forward gear variants of the model. In the symmetric variant, we see that initially, the tracking follows the vessel structure better. However, before the vessel bends, the tracking finds a shorter way to reach the seed and it crosses the vessel structures moving straight towards the seed. Towards the end of the tracking, the tracking uses the reverse gear (at least) once. Therefore, this tracking cannot be found in the forward gear variant of the model. When we consider the forward gear variant, we see that the tracking does not bend along with the vessel structure, but takes a shortcut immediately. Towards, the end of the tracking, it has some

trouble moving towards the seed, and it moves around it while searching for the exact location. The general impression for the tracking results for a low value of ξ is that the tracking contains many angles and rotation points. Overall, it has trouble following the vessel structures in a way that we would do it ourselves.

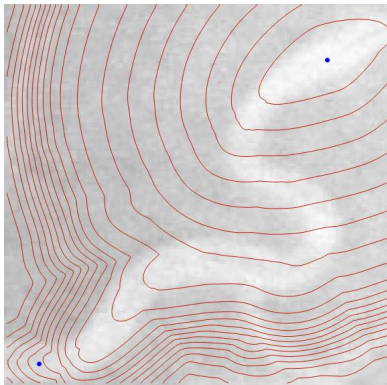
In Figure 38, we see the tracking results for a (relatively) high value of ξ . First, we take a closer look at the *isotropic left invariant frame*. Here, we see that the tracking does not follow the vessel structure as expected. However, we do see that the tracking results for the symmetric and forward gear model are very similar. Both tracking results take a shortcut and do not follow the vessel structure along its hairpin bend. In the *anisotropic left invariant frame*, we see that the tracking does not follow the vessel structure either. The tracking results for the forward gear and symmetric variant are again very similar. There are small differences with the isotropic left invariant frame. The most prominent difference is the way both trackings take their shortcut. In the isotropic model, the model makes a somewhat sharper turn in the vessel, and it makes a small bend in between both vessels. In the anisotropic model, the tracking seems to be making a more continuous turn. The last model we consider is the *anisotropic gauge frame where $\alpha = 0.8$* . We see a difference between the tracking of the symmetric and the forward gear variants of this model. First, it is important to note that both tracking results do not move outside of the vessel structure. However, we see some interesting behaviour where the vessel bends. In the symmetric model, the tracking uses the reverse gear twice where the tracking runs into the edge of the vessel. However, this is not possible in the forward gear. Here, we see that the both times the tracking meets the edge of the vessel, the tracking makes a sharp turn or in place rotation in order to prevent using the reverse gear. We can clearly see that the tracking results are more flexible than they were when $\xi = 0.08$. This confirms the analogy to cooked spaghetti we made at the beginning of this paragraph.

4.4 Conclusion

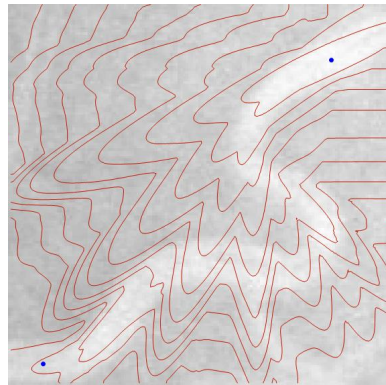
In this chapter we discussed the tracking results for different experimental settings. Here, we summarize the most important findings.

1. All experiments have illustrated the *difference between the isotropic and the anisotropic metric tensor field diagonalized in the left invariant frame*. We have seen that the anisotropic metric tensor field discourages the tracking to move sideways. Therefore, it favors angular movement and forward/backward movement.
2. The *gauge frame follows the vessel structure better* than the left invariant frame. This can be explained since the first gauge vector is aligned with the underlying vessel structure. Therefore, the tracking is stimulated (by design) to move into the direction of the vessel.
3. The *forward gear model strongly discourages backward movement*. The tracking does not switch gears anymore. Since in-place rotations are often more expensive, there is only a limited number of keypoints.
4. The *choice of the cost function* is very important. When the cost function favours some vessels over others, this will be visible in the tracking. In addition to that, the cost function has a high influence on where the tracking goes, since it tries to find the shortest route, which is the optimal path.
5. The gauge frame needs a *lower number of orientations* to find a correct tracking, as we saw in Experiment 4 (Figure 36). Where the left invariant frame has trouble following the vessel structure with 16 orientations, the gauge frame is able to find a correct tracking with only 8 orientations. This is in line with previous findings [12].
6. The gauge frame *adapts its direction to the underlying dominant vessel structure*. We saw that this might lead to problems at crossings (in the artificial image). We will investigate this in more detail in Section 5, and will give our corresponding conclusions there.
7. The *parameter ξ denotes the bending stiffness and hence influences the flexibility of the tracking*. When its value is very low, the tracking is very stiff and therefore one could say that “the curves look like bended steel”. However, when its value is relatively high, the tracking is able to rotate very easily and “the tracking behaves like cooked spaghetti”.

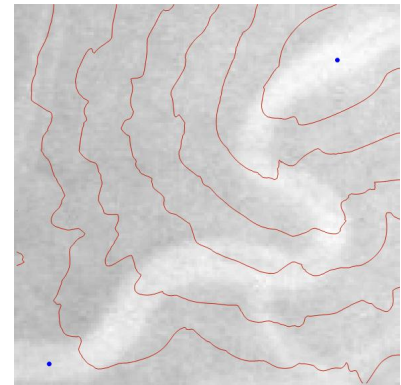
Overall, we can conclude that when we consider an image containing only one wide vessel (consisting of a high number of pixels/sample points), it is beneficial to use the gauge frame.



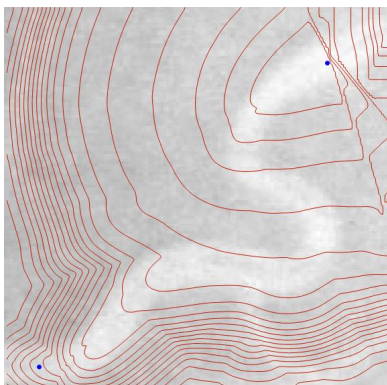
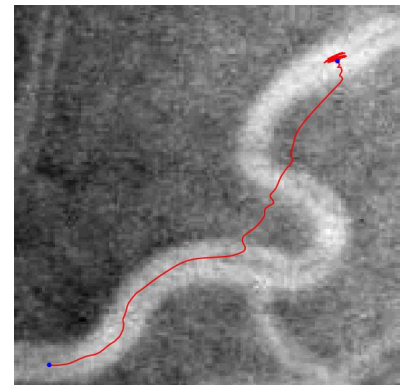
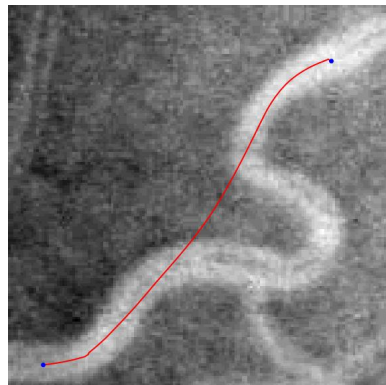
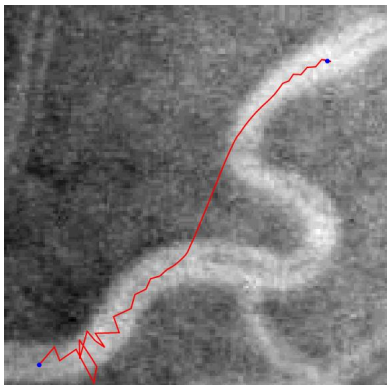
Symmetric Isotropic Left Invariant Frame.



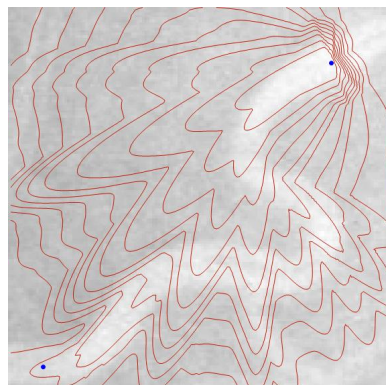
Symmetric Anisotropic Left Invariant Frame.



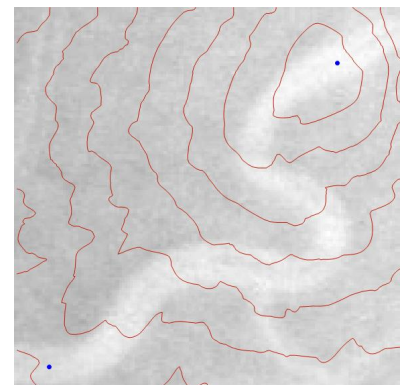
Symmetric Gauge Frame with $\alpha = 0.8$.



Isotropic Left Invariant Frame with Forward Gear.



Anisotropic Left Invariant Frame with Forward Gear.



Gauge Frame with Forward Gear where $\alpha = 0.8$.

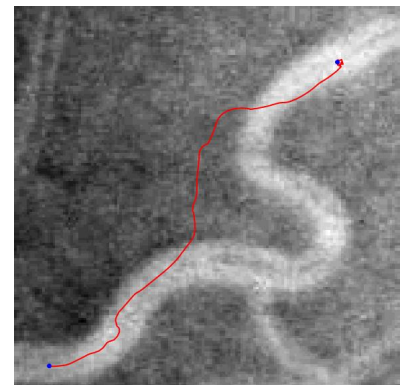
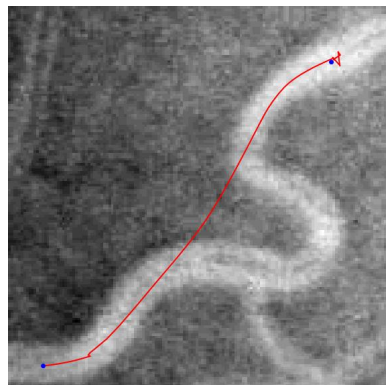
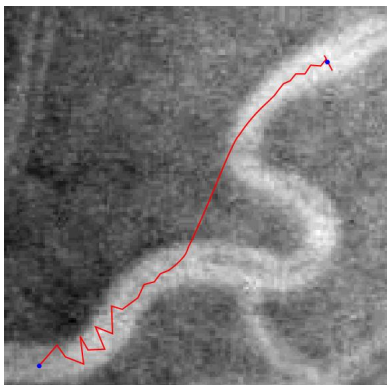
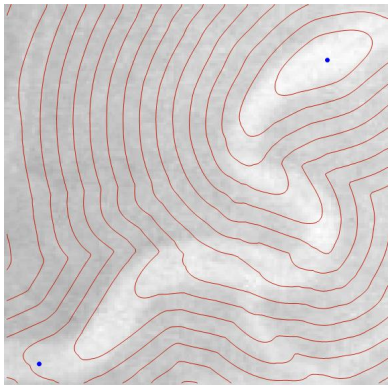
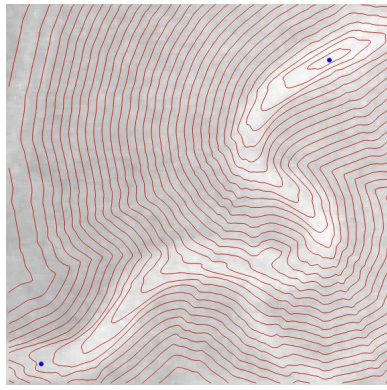


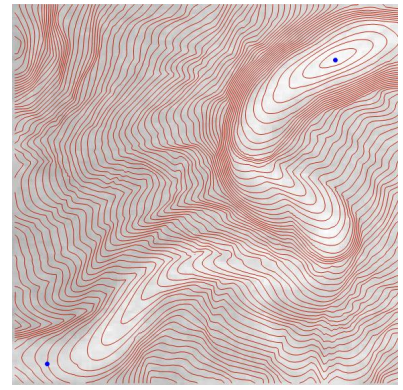
Figure 37: Results for the different frames where $\beta = 5$. All simulations use $N_\theta = 16$ and $\xi_{\text{PDE}} = 0.01$



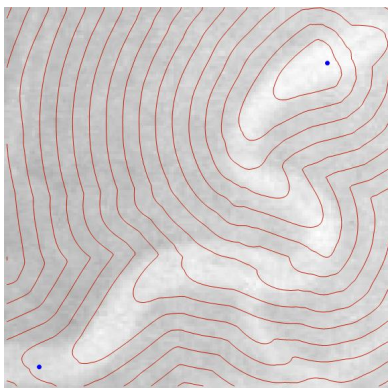
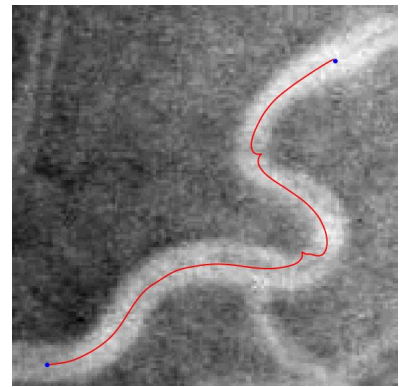
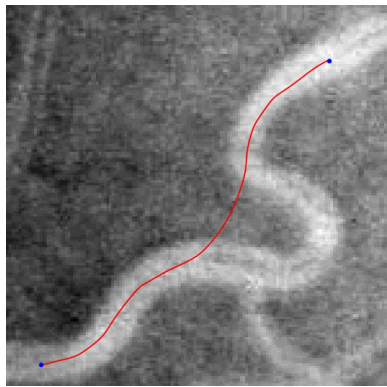
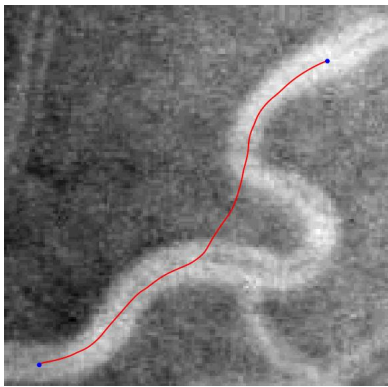
Symmetric Isotropic Left Invariant Frame.



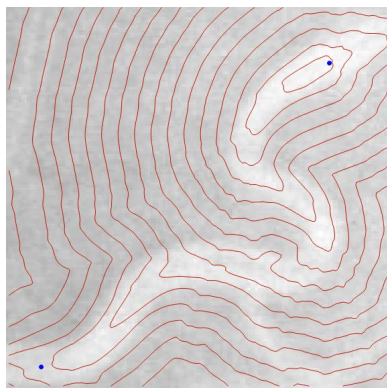
Symmetric Anisotropic Left Invariant Frame.



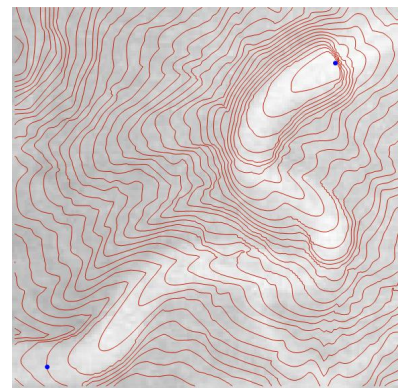
Symmetric Gauge Frame with $\alpha = 0.8$.



Isotropic Left Invariant Frame with Forward Gear.



Anisotropic Left Invariant Frame with Forward Gear.



Gauge Frame with Forward Gear where $\alpha = 0.8$.

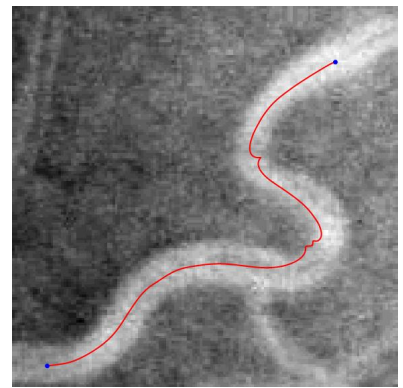
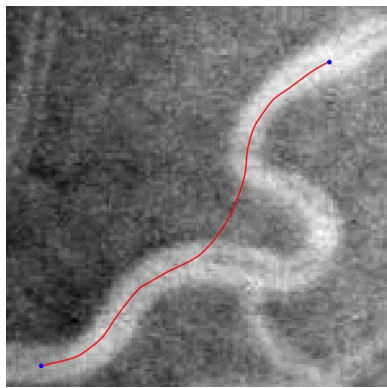
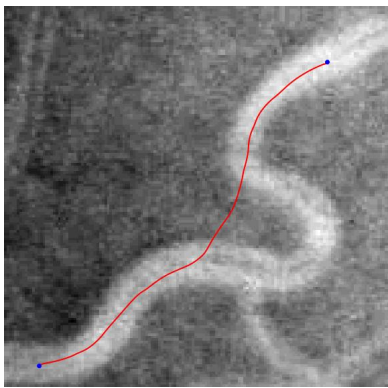


Figure 38: Results for the different frames where $\beta = 5$. All simulations use $N_\theta = 16$ and $\xi_{\text{PDE}} = 0.25$

5 Tracking Blood Vessels in Vascular Trees

In the previous section, we studied the tracking results in the situation where we consider one seed and one tip. However, often there are multiple vessels in an image. It is interesting to study how the tracking behaves when we consider multiple seeds and tips. We note that our implementation (PDE-method) has no trouble dealing with one seed and multiple tips. In that case, the backtracking algorithm needs to be executed multiple times, and no problems arise. However, when considering multiple seeds, we need to adapt our current implementation. Since the algorithm written by Jean-Marie Mirebeau using Fast Marching is able to build a volume with multiple seeds, we decide to use the Fast Marching code in this section. We will consider one image in which we track multiple vessels. Beforehand, we indicate the seeds and tips for all vessels. Then, the Fast Marching script is asked to calculate all geodesics. Once the results are known, we check whether the tracking has been done correctly. If this is not the case, we adapt our model.

The script we use goes through the following steps for each processed image:

1. Load image and annotations (seeds, tips, vessel structures and more);
2. Down sample the image and its annotations (make sure the image has fewer pixels, but is still “the same image” and the annotations are still in the same place) to improve calculation times;
3. Import the seeds and tips. This information is saved in the annotation data we imported in Step 1.
 - (a) Locations: tips are vessels that are not a parent vessel, start denotes the starting points of the vessels, where seeds contain both the starting points of the vessels, and the bifurcations of vessels;
 - (b) Orientations: for each location, we compute the location exactly in the middle between the left and right boundary of the vessel for the first n points (we use $n = 10$). Then, we compute the location that is exactly in between two subsequent locations. Afterwards, we calculate the average location of the $n - 1$ average points we just calculated. Last, we calculate the angle between the initial location (the seed or tip that we identified before) and the calculated location on the vessel. This angle yields the orientation of the vessel at the specific point. In formula’s, we did the following:

Consider the first n locations (x_i, y_i) . Then calculate the locations exactly in the middle represented by $(\tilde{x}_i, \tilde{y}_i)$ by calculating

$$(\tilde{x}_i, \tilde{y}_i) = \left(\frac{x_i + x_{i+1}}{2}, \frac{y_i + y_{i+1}}{2} \right) \quad \text{for } i = 1, \dots, n - 1.$$

Next, we calculate the mean value of $(\tilde{x}_i, \tilde{y}_i)$ for $i = 1, \dots, n - 1$, which is saved in (\bar{x}, \bar{y}) :

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n-1} \sum_{i=1}^{n-1} \tilde{x}_i, \frac{1}{n-1} \sum_{i=1}^{n-1} \tilde{y}_i \right).$$

Last, calculate the angle θ by

$$\theta = \arg(\bar{x} + i\bar{y}) = \arctan\left(\frac{\bar{y}}{\bar{x}}\right),$$

taking into account the quadrant in which (\bar{x}, \bar{y}) is located.

4. Select the part of the image you want to analyse (and the vessel structures, seeds and tips that are located there). Do note that it can be the case that only parts of vessels are in this part of the image. Therefore, it may be possible that for some vessels, only tips or seeds are included. Check this before calculating the geodesics;
5. Set parameters for the geodesics ($\mu = 0.05$, $\varepsilon = 0.1$, $N_\theta = 64$, periodicity $= 2\pi$, $\Delta\theta = 2\pi/N_\theta$, $\Delta x = 1$);
6. Calculate the orientation score transform;
7. From end points to starting points (not bifurcations) on lifted vessel tree
 - (a) Convert all coordinates to the correct notation. In the notebook, we use discrete notation for the positions and the corresponding angles. That means that when we consider the continuous angle $\theta_c = \pi$, we use the discrete angle $\theta_d = \frac{\theta_c}{\Delta\theta} + 1 = 33$ in the calculations. In addition to that, we make sure that the coordinates are in the correct domain (by returning modulo N_x , N_y and N_θ) and all are integers (rounding with 0 decimals). In addition to that, we also calculate all centerline positions (notations before was $(\tilde{x}_i, \tilde{y}_i)$). For each position, we also calculate a corresponding angle. These positions and orientations will be used to calculate the lift and cost function.

- (b) Create lift by placing delta peaks at the centerline position and orientations; we construct two different lift functions V , which are used later on when we define the cost function. Both lift functions have the same dimensions as the orientation scores.

The first lift function is based on the centerline positions. At all these positions, the lift function gets value 1. Since the lift function should not only support movement in direction θ_i at location (x_i, y_i) , also the values in the lift function that correspond to orientation $\theta_i \pm \pi$ at this location should get function value 1. All other values of the lift function are 0.

The second lift function is an expansion of the first one. Again, we have an array with the same dimension as the orientation score. The function values at the locations corresponding to the locations and orientations of the centerlines and their rotations are 1. However, in this case, we support angular movement at seeds. Therefore, the lift function is assigned value 1 for all orientations at the seeds. This way, it is encouraged to visit the seeds.

Afterwards, a Gaussian filter is applied to the lift functions such that the function values are slightly blurred and approach continuous behaviour better.

- (c) Construct cost function: The costs \mathcal{C} are defined by calculating

$$\mathcal{C}(\mathbf{x}, \theta) = \frac{1}{1 + \lambda V(\mathbf{x}, \theta)^p}, \quad (89)$$

where $\lambda = 400$ and $p = 2$.

- (d) Identify crossings: Last, we identified all crossings in the upper half of the image. For each location, we identified the orientation of the crossing vessels as well. For now, we identified these locations and orientations by hand.
8. Vessel tracking (building volume and backtracking). Depending on the frame in which our tracking takes place, we calculate different metrics. As explained earlier, we use Jean-Marie Mirebeau's Fast Marching code to find the geodesics.

- Left Invariant Frame and Gauge Frame: Unfortunately, the function we use for the forward gear gauge frame (“AsymmetricQuadratic3”) does not have the option to calculate the angular component periodically. Therefore, we duplicate the metric and stack them on top of each other. We use 3 copies. This way, the tracking is able to follow the vessel, and will not be limited by the non-periodicity of the angular component. In an effort to make the computations more efficient, we duplicated the seeds and tips to “each angular layer”. Note that using 3 copies is equivalent to using 5 copies without copying the seed. In the end, we select the shortest path from all the equivalent tips. We determine the metric, as explained in Appendix A, including the cost function, the dimensions of the domain, the area of the domain, and the seeds and tips. When we consider the forward gear variant, we also provide the forward direction, the number of copies that should be considered, and the variable ε .
- Mixed Frame: In Section 1.4, we discussed that the gauge frame is a good fit on the vessels, but not necessarily at other locations. Therefore, we consider a (convex) linear combination of both frames. When the cost function is small, it is cheap to move, so we are at a vessel structure. When we are on a vessel, we want to use the gauge frame. The other way around, when the cost function is high, it is expensive to move, so we are outside a vessel structure, and therefore, we want to use the left invariant frame. That leads us to the following frame

$$\mathcal{G}_{\text{Mixed}}^{(f)} = (1 - \mathcal{C}) \mathcal{G}_{\text{GF}}^{(f)} + \mathcal{C} \mathcal{G}_{\text{LIF}}^{(f)}, \quad (90)$$

where \mathcal{C} denotes the cost function.

The forward direction adapts according to the same scaling as the frame does.

Remark 5.1. We used the ground truth while identifying the cost function in Step 7b. The ground truth contains all information on where the vessels are located in the image. Although we do not have this information under normal circumstances, we can justify using this information. There are different neural network techniques in which an accuracy up to 96% can be achieved [33].

The values of the different parameters in the experiments are given in Table 2. Note that using more orientations would result in a better vessel tracking result. However, due to calculation capacity of the computer, we decided to use $N_\theta = 64$. Computations are a lot faster and use less memory when we choose to use a smaller N_θ . However, the fast marching algorithm needs sufficient angular directions in order to find smooth results

	ξ	ζ	N_θ	σ_s	σ_a	ε	α
Cost function in Figure 40	0.1	0.1	64	3 or 7 (see text)	$0.87 s_\theta$	0.1	$2/3$
Cost function in Figure 44	0.1	1	32	1.5	0.6	0.1	$2/3$

Table 2: Values used for the different parameters in the experiment using different cost functions for vessel tracking in a tree, with $s_\theta = \frac{2\pi}{N_\theta}$.

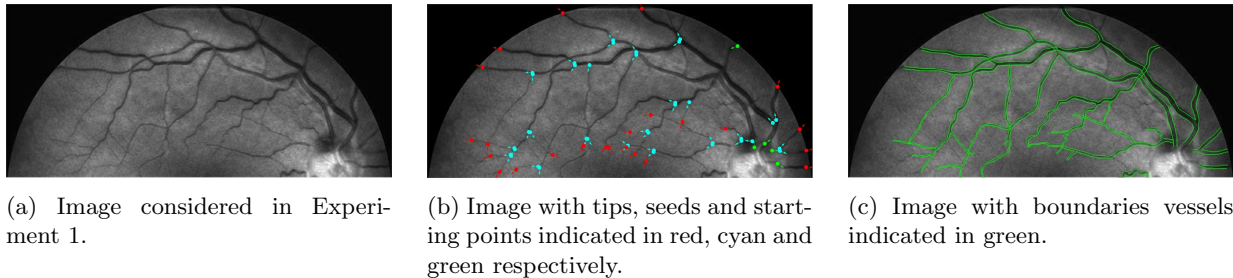


Figure 39: Setting of Experiment 1 (tracking in vessel tree).

5.1 Experiment

The retina image that is used in Experiment 1 is shown in Figure 39. As mentioned before, we will track multiple vessels at once. All geodesics need to find a way from each tip (red point) to a starting point (green point). The cost functions belonging to this retina image (explained in both item 7b and 7c), are visualized in Figure 40. We see that the cost function consists of cheap line segments that meet at bifurcation points. For the cost function with elevators at the seeds, we see that all line segments are glued together at bifurcation points. Since we want to stimulate the tracking to rotate (and connect different line segments) at these bifurcation points, we will use the cost function with elevators. Note that we know the locations of bifurcation points for this image. However, under normal circumstances, one does not have such a classification of points. Therefore, it is important to look for a way to classify both points automatically.

First, we compare the tracking results of the left invariant frame and gauge frame for both the symmetric and forward gear variants of the model. In Figure 41, we see the results of these tracking algorithms. We see that the results of the tracking for the symmetric and forward gear left invariant frame are identical. That means that the model did not have to switch gears in the symmetric model. We see that the tracking is performed correctly for all vessels. Since we argued in the Section 4 that the gauge frame was better able to follow specific structures in an image, we check how the tracking results are for the vessel tree given in this image. We see that the tracking of symmetric anisotropic gauge frame switches to a different vessel from what was expected twice. The first time, the tracking uses a crossing as a bifurcation point. This could be related to the alignment of the gauge frame. The second time, the tracking switches vessels, the two vessels are located very close to each other. When the gauge frame is not correctly aligned with the vessel, it is possible that the tracking is stimulated to move towards a different vessel and continue its way there. This seems to be confirmed when we look at the direction of the first gauge vector in Figure 42. For the forward gear anisotropic gauge frame, the results are very similar to the symmetric case. Last, we consider the mixed frame as introduced in Eq. (90). This frame uses the left invariant frame where the cost function is high, and the gauge frame where the cost function is low. We see that the results for the symmetric and forward gear variants of this model are also very similar. However, they do differentiate from what we found for the anisotropic left invariant frame and the anisotropic gauge frame. In this model, two vessels move in a direction that we did not anticipate. We note that the vessel that decided before to switch vessels at a point where the two vessels are very close together, follows its own vessel structure correctly now. However, at the first crossing that it comes across, it treats the crossing as a bifurcation, moving in a different direction from what we would expect. Another vessel first meets a bifurcation point, where it does not move in the direction towards the seed, but towards the different tip. Then, when it meets the same crossing as described before, it moves in the same direction as all other vessels. This is different from what we would expect, since the it moves over a larger distance now.

We discussed that the mistakes in the tracking could be related to the gauge frame that was used to calculate the results. Therefore, we take a closer look at the exponential curves belonging to the gauge frame. They are plotted in Figure 42. We calculated the gauge frame for different variables, and see that the alignment of the exponential curves to the vessel structure is the best in Figure 42c, where we use $\sigma_s = 3$ and wavelet size 5. Therefore, we also consider the tracking results where the gauge frame is calculated with these variables.

The tracking results where the gauge frame is constructed using $\sigma_s = 3$ and wavelet size 5 are given in Figure

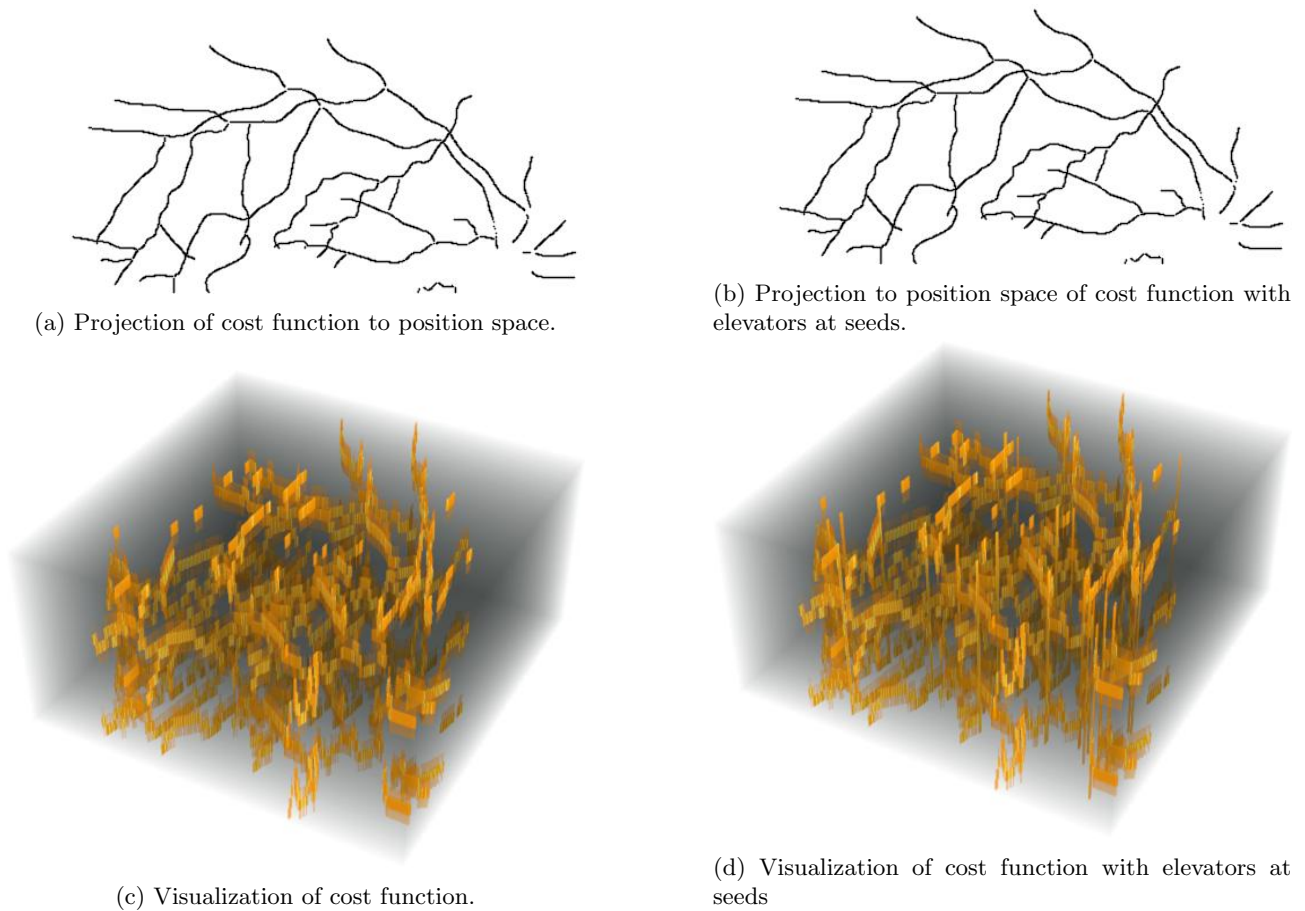


Figure 40: Cost function corresponding to Experiment 1 with the methods explained in items 7b and 7c.

43. In the symmetric anisotropic gauge frame tracking, we see that there are two locations where the tracking goes wrong. These are the same as in the tracking where we used the gauge frame constructed with $\sigma_s = 7$ and wavelet size 15. In the forward gear variant of this model, there is an extra location where the tracking does not follow the vessel correctly. There is one vessel that contains several bends, which is not tracked correctly. In the symmetric and forward gear mixed model the tracking is correct for the most part. The tracking only switches vessels where they are very close together. Again, this could be related to the fitting of the gauge frame. Looking at the image, it is understandable that it assumes that there is a crossing/bifurcation at this location, since the orientation of the vessels is also very similar.

In conclusion, we see that the gauge frame needs to be fitted very well on the image that we are considering in order to find good results. This is not trivial, since there are a lot of parameters that can be adapted in an effort to find a better gauge frame. Unfortunately, we have not been able to find the ideal settings for the image presented in Figure 39. This still requires more research.

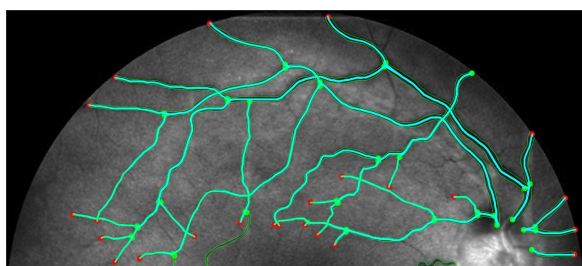
We note that, due to our very strict cost function, the tracking rarely moves outside of the vessel structure. Therefore, we consider the tracking for a different, less precise cost function as well. This cost function is inspired by the cost function introduced in [34]. We implemented a single scale vesselness filter, instead of a multi-scale version. The mobility, a.k.a. vesselness filter, uses an anisotropy measure \mathcal{R} and a structure measure \mathcal{S} . These are based on the second order derivatives in the coordinate frame that we are considering. The convexity criterion \mathcal{Q} is chosen to select transversal vessel profiles. The vesselness filter $\mathcal{V}(U_f) : SE(2) \rightarrow \mathbb{R}^+$ is given by

$$\mathcal{V}(U_f) = \begin{cases} 0 & \text{if } \mathcal{Q} \leq 0 \\ \exp\left(-\frac{\mathcal{R}^2}{2\sigma_1^2}\right) \left[1 - \exp\left(-\frac{\mathcal{S}}{2\sigma_2}\right)\right] & \text{if } \mathcal{Q} > 0. \end{cases} \quad (91)$$

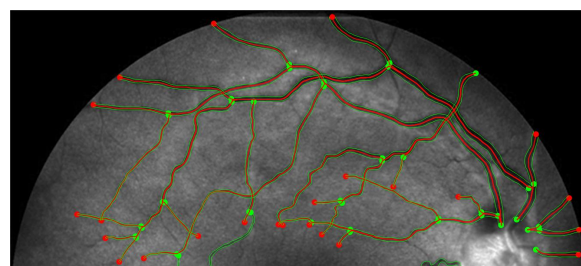
In the above expression, variables σ_1 and σ_2 are always chosen to be 0.5 and $0.2 \|\mathcal{S}\|_\infty$ respectively.

We use the left invariant frame while calculating the cost function. Then, the different measures are given by

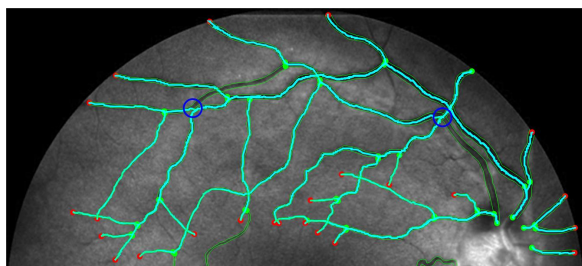
$$\mathcal{R} = \frac{\mathcal{A}_1^2 U_f^{s,\beta}}{\mathcal{A}_2^2 U_f^{s,\beta}}, \quad \mathcal{S} = \left(\mathcal{A}_1^2 U_f^{s,\beta}\right)^2 + \left(\mathcal{A}_2^2 U_f^{s,\beta}\right)^2, \quad \mathcal{Q} = \mathcal{A}_2^2 U_f^{s,\beta}. \quad (92)$$



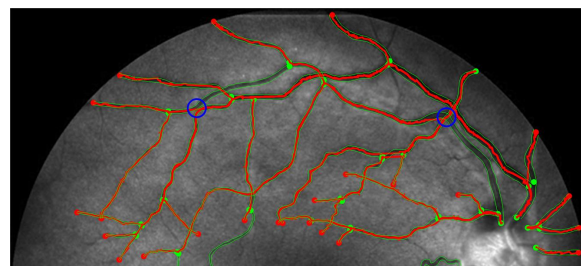
(a) Backtracking results for symmetric anisotropic left invariant frame.



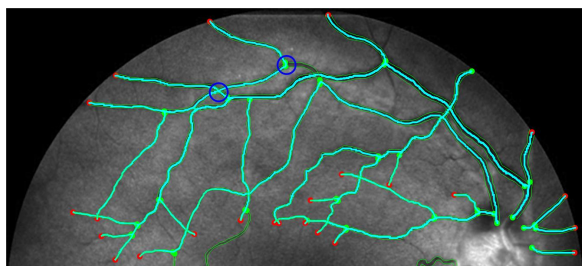
(b) Backtracking results for forward gear anisotropic left invariant frame.



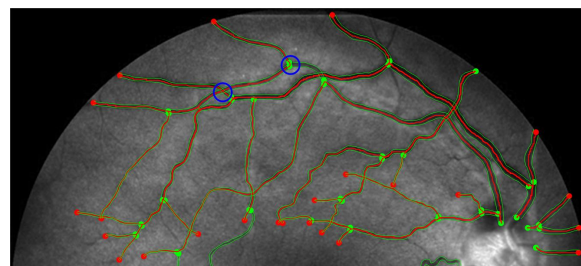
(c) Backtracking results for symmetric anisotropic gauge frame with $\sigma_s = 7$, $\sigma_a = 0.87s_\theta$ and wavelet size 15.



(d) Backtracking results for forward gear anisotropic gauge frame with $\sigma_s = 7$, $\sigma_a = 0.87s_\theta$ and wavelet size 15.

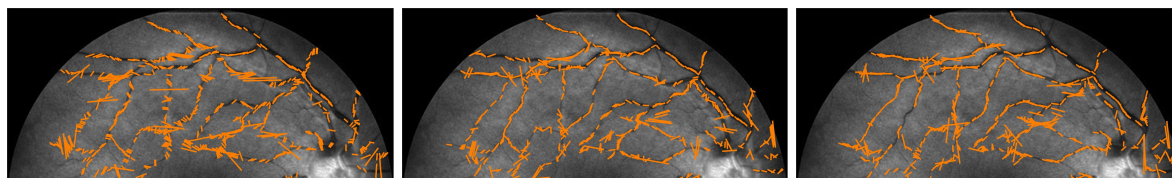


(e) Backtracking results for symmetric mixed frame with $\sigma_s = 7$, $\sigma_a = 0.87s_\theta$ and wavelet size 15.



(f) Backtracking results for forward gear mixed frame with $\sigma_s = 7$, $\sigma_a = 0.87s_\theta$ and wavelet size 15.

Figure 41: The tracking of the vessels is indicated in cyan and red (symmetric and forward gear respectively). Blue circles are placed around the locations where the tracking behaves differently from what we expect. The vessel boundaries are indicated in green. The used cost function was introduced in items 7b and 7c.

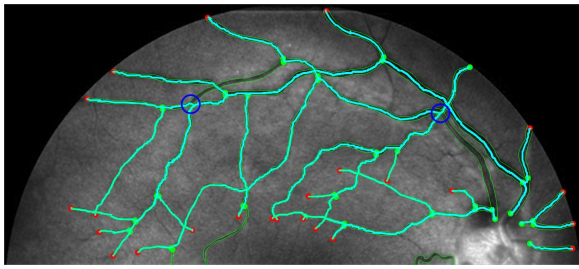


(a) $\sigma_s = 7$, wavelet size 15.

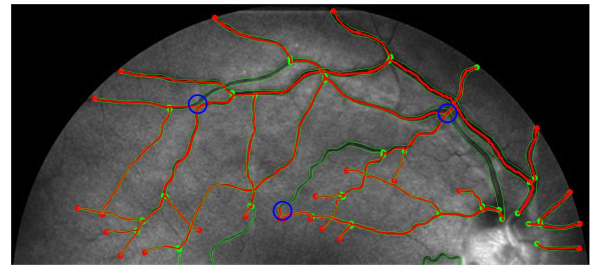
(b) $\sigma_s = 3$, wavelet size 15.

(c) $\sigma_s = 3$, wavelet size 5.

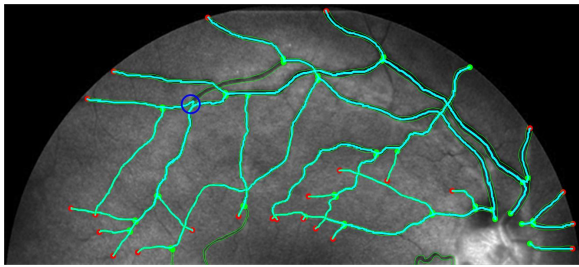
Figure 42: Exponential curves for gauge frame using different spatial blurring coefficients and wavelets sizes.



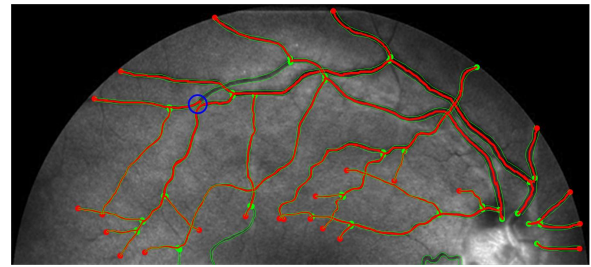
(a) Backtracking results for symmetric anisotropic gauge frame with $\sigma_s = 3$, $\sigma_a = 0.87s_\theta$ and wavelet size 5.



(b) Backtracking results for forward gear anisotropic gauge frame with $\sigma_s = 3$, $\sigma_a = 0.87s_\theta$ and wavelet size 5.



(c) Backtracking results for symmetric mixed frame with $\sigma_s = 3$, $\sigma_a = 0.87s_\theta$ and wavelet size 5.



(d) Backtracking results for forward gear mixed frame with $\sigma_s = 3$, $\sigma_a = 0.87s_\theta$ and wavelet size 5.

Figure 43: The tracking of the vessels is indicated in cyan and red (symmetric and forward gear respectively). Blue circles are placed around the locations where the tracking behaves differently from what we expect. The vessel boundaries are indicated in green. The used cost function was introduced in items 7b and 7c.

Here, the superscripts s, β indicate Gaussian derivatives at spatial scale $s = \frac{1}{2}\tilde{\sigma}_s$ and angular scale $\frac{1}{2}(\beta\tilde{\sigma}_s)^2$. In this example, we chose $\beta = 0.75/\tilde{\sigma}_s$, and $\tilde{\sigma}_s = 2$.

We note that the retinal image in Figure 39 contains a black mask in the shape of a disk. The edge that arises on the boundary between the image and the mask will be recognized as a vessel-structure. Therefore, we remove this structure from the vesselness filter that we construct. We do so by calculating the vesselness filter for only the mask (without the retinal image), and set all points where the vesselness filter of the mask goes under a certain threshold (in our case 10^{-4}) to zero. Since angular movement was better supported than we would like it to be, we used erosion in the angular direction to limit angular movement. The result of the aforementioned manipulations are saved in $\tilde{\mathcal{V}}$, which is rescaled such that all values are in the interval $[0, 1]$. Last, we suppose that we know where the bifurcations are located. Here, we support movement in all directions. These points are spatially blurred with a scale of 1.5, such that angular movement is also slightly supported close to the bifurcation point. These values are rescaled such that they are in the interval $[0, 1]$. The values are saved in the Bifurcation Point Density \mathbf{B} . Then, the mobility \mathcal{M} is given by

$$\mathcal{M}(\mathbf{x}, \theta) = \max\{\tilde{\mathcal{V}}(\mathbf{x}, \theta), 1/4 \cdot \mathbf{B}(\mathbf{x}, \theta)\}.$$

Then, this mobility is rescaled to be in the interval $[0, 1]$. The corresponding cost function is given by

$$\mathcal{C}_{\mathcal{V}}(\mathbf{x}, \theta) = \frac{1}{1 + \lambda \cdot \mathcal{M}(\mathbf{x}, \theta)^p}, \quad (93)$$

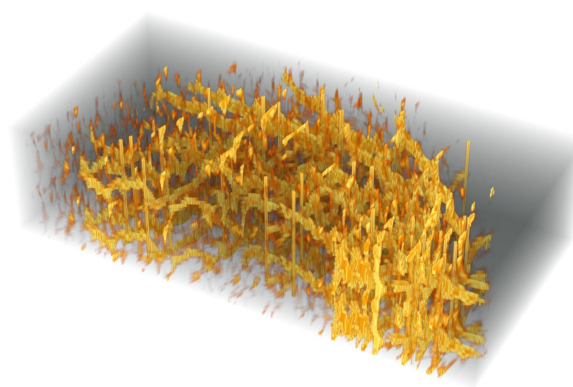
where $\lambda = 800$ and $p = 2$. The visualization of this cost function is given in Figure 44. The tracking results belonging to the cost function given in Figure 44 are visualized in Figure 45.

First, we discuss the results of the tracking in the left invariant frame. We see that the tracking is less accurate compared to when we used the “exact cost function”. However, the tracking is still very decent. The blue circles indicate where the tracking goes wrong. These are located at places where either crossings occur or where both vessels are close to each other. Last, we note that there are no significant differences between the symmetric and the forward gear left invariant trackings.

Next, we consider the gauge frame tracking. We see that this tracking is less accurate. Instead of discussing all vessels, we highlight two of them which illustrate the recurring problems accurately. In Figure 46, we see in both images that the tracking is better at adapting for curvature (in the white ovals). However, there are also places where the tracking results are not that well fitted. This occurs at crossings of blood vessels. The gauge frame was fitted such that it follows the vessel structure better. However, the gauge frame always prefers

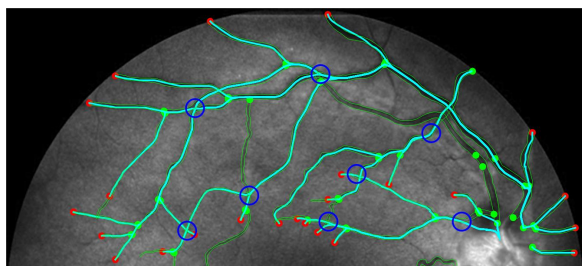


(a) Projection of alternative cost function to position space.

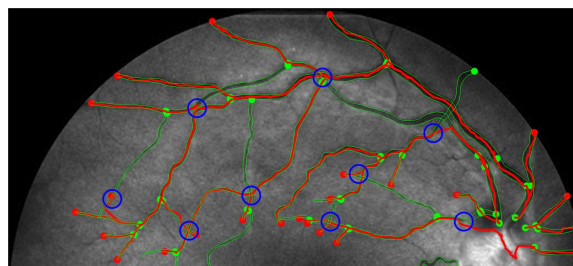


(b) Visualization of alternative cost function.

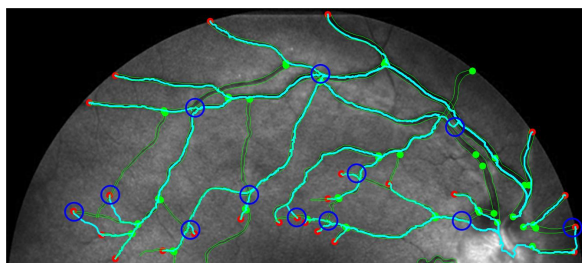
Figure 44: Visualizations of alternative cost function explained in Eq. (93) corresponding to Experiment 1.



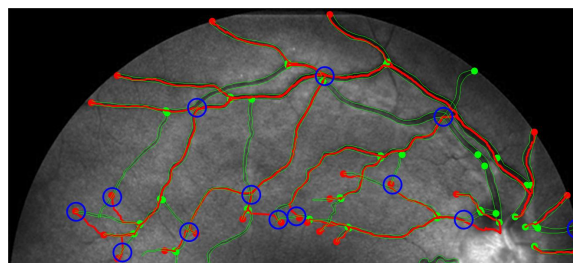
(a) Backtracking results for symmetric left invariant frame.



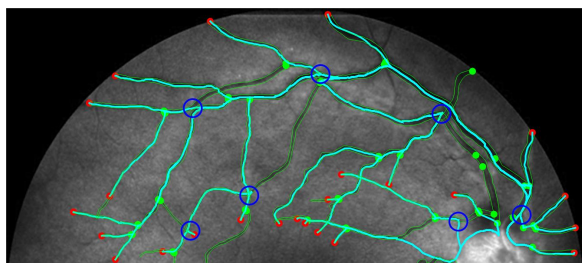
(b) Backtracking results for forward gear anisotropic left invariant frame.



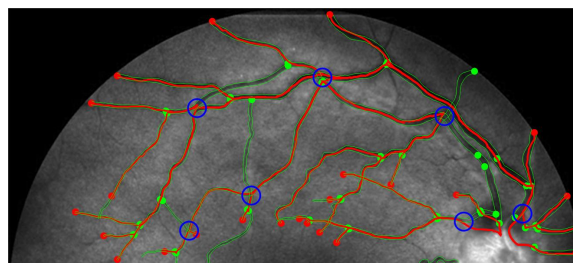
(c) Backtracking results for symmetric anisotropic gauge frame with $\sigma_s = 3$, $\sigma_a = 0.87s_\theta$ and wavelet size 75.



(d) Backtracking results for forward gear anisotropic gauge frame with $\sigma_s = 3$, $\sigma_a = 0.87s_\theta$ and wavelet size 75.

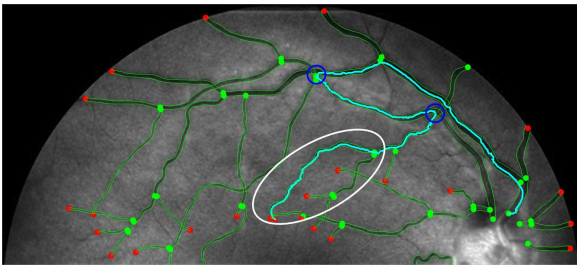


(e) Backtracking results for symmetric mixed frame with $\sigma_s = 3$, $\sigma_a = 0.87s_\theta$ and wavelet size 75.

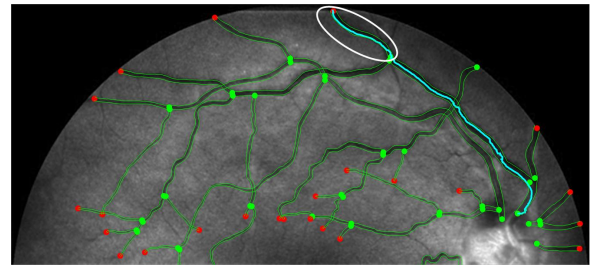


(f) Backtracking results for forward gear mixed frame with $\sigma_s = 3$, $\sigma_a = 0.87s_\theta$ and wavelet size 75.

Figure 45: The tracking of the vessels is indicated in cyan and red (symmetric and forward gear respectively). Blue circles are placed around the locations where the tracking behaves differently from what we expect. The vessel boundaries are indicated in green. The used cost function is introduced in Eq. (93).



(a) Tortuous vessel that runs into several crossings.



(b) Tortuous vessel.

Figure 46: Backtracking results for symmetric mixed frame with $\sigma_s = 1.5$, $\sigma_a = 0.6$ and wavelet size 75. The seeds and bifurcation points are indicated by the green points. The tips are indicated by the red points. The vessel structure is visualized by the green boundaries. In the white ovals, the regions where the gauge frame is better able to adapt to curvature compared to the left invariant frame are surrounded. At the blue circles, the tracking opts for the wrong/unexpected blood vessel. The used cost function is introduced in Eq. (93).

the dominant structure. That means that at crossings, the gauge frame stimulates the tracking to follow the dominant vessel. This is not always the vessel that we intend to track, as we can see in Figure 46a, where this happens twice (see blue circles).

Last, we see in Figure 45 that the tracking results are a combination of the left invariant frame and the gauge frame (as expected). We see that at the “highly tortuous vessels”, the left invariant frame tracking is used. At these places, the cost function is less precise, and therefore the left invariant frame takes over when the tracking moves outside the cheap areas of the cost function. We still see that at crossings (and places where the two vessels are very close to each other), the tracking prefers the dominant vessel. The cost function could possibly be improved by considering a gauge frame based variant of the vesselness function.

5.2 Conclusion

We discussed the results of the different tracking methods using different cost functions. We summarize our main conclusions:

1. The *choice of the cost function* is very important. Choosing a very precise cost function (as suggested in Figure 40), the tracking results will be very good. However, obtaining such a perfect lifted cost function requires too much training in a deep learning approach. Geometrical methods provide reasonable crossing preserving vesselness cost functions, but in cases where bifurcations or crossings are close to each other, tracking errors start to occur. Using deep learning segmentations with basic forward networks provide worse segmentations. In the future, we either must optimize/train the (geometric) crossing preserving vesselness parameters, or enlarge the network architecture with state of the art computer science techniques. In addition to that, prior classification (e.g. via deep learning) of crossings and bifurcations is still desirable to achieve correct vessel tracking (without artificial keypoints and skeletons [35; 36]).
2. It is very important that the *gauge frame is well-fitted* to the retina image. When the gauge frame is not fitted optimally, one can easily see this in the tracking results, which will become more jittery or leave the vessel structure. This happens when the underlying frame encourages the tracking to move outside of the vessel structure. The cost function discourages this movement, leading to a zigzag path. An alternative explanation would be that the “main direction” of the gauge frame differs a lot per location. In that case, it is also possible that the tracking is not smooth despite stabilization of the gauge frame [5].
3. The gauge frames *main direction often bends into the direction with the highest response*. However, at crossings, this direction is not always the direction of the vessel that we are considering. Therefore, all vessels arriving at a crossing are encouraged to start moving along the most dominant vessel. This is not the vessel that all trackings should be following. Therefore, it is important to construct a cost function which ensures cross-profile consistency. This possibly makes it less likely for trackings to change direction at crossings. In addition to that, one should consider to use a different frame near crossings (e.g. left invariant frame). One could implement this using the non-crossing conserving vesselness map, e.g. the vesselness function of Frangi [37].

Since the fast marching algorithm needs a higher number of orientations to find a good tracking, the advantage of using fewer orientations is not interesting anymore. The advantages of the reduced computation time are less, but they are still applicable, since the fast marching algorithm is a lot faster than the finite difference method.

6 Future Research

In this report, we discussed the implementation of the forward gear variant of the gauge frame model and compared it to other existing models. Even though we have considered and implemented a lot of theory, there are still aspects that need to be studied more closely. In this section, we will elaborate on them.

We have used two different solution strategies to solve the Eikonal PDE. The first one used a finite differences approach. We implemented this such that it can track several tips to one single seed in one go. In practice, we often need to track several tips, but it is not always clear beforehand which seeds correspond to the tips. Therefore, it is important to ensure that the volume is built such that it can consider multiple seeds as well. In practice, this means that we give all seeds \mathbf{p}_S value 0 in every iteration. Note that this has already been implemented in the fast marching method of Jean-Marie Mirebeau [26].

While studying the finite differences approach, it is important to also do a stability analysis. Erik Franken has already preformed this for the left invariant frame [11]. We used the same condition for the gauge frame, but this may not always be the best condition to use, since we are considering a slightly different model. Therefore, it would be good to have a close look at this stability condition and its validity. At the same time, it would also be interesting to check the order of accuracy of the used scheme.

In some specific situations, there were a some problems with the tracking; it would reach a local minimum before reaching the global minimum. These local minima were not located on grid points, but in between. We used a list interpolation to construct a continuous function out of a discrete list. However, we also expect there only to be minima at the seeds. When a local minimum is reached at a different point, this is not in line with our expectations. Therefore, it is important to do some more research on why these local minima occur, and how we can prevent them from occurring. One of the solution strategies would be adding Gaussian derivatives while calculating the steepest descent algorithm. At this point in time, we solved the tracking problems by slightly blurring only in the angular component.

In Section 5, we discussed vessel tracking in a vessel tree. We learned that it is very important that the constants which are used to fit the gauge frame, are chosen correctly. If we want to automatize this algorithm in an application, it is important that these constants can be chosen automatically, in an efficient manner. Therefore, it is relevant to search for a “rule of thumb” which results in a well-fitted gauge frame. It was illustrated that using a different gauge frame results in a different tracking, which might differ from the expected, correct result.

In Section 5, we discussed the best cost function. We concluded that, in order to find correct results, we need to connect the cost function at bifurcation points. In order to do this correctly, we need to identify the bifurcations. It is important to distinguish between crossings and bifurcations, since we do not want to stimulate rotation at crossings. When we use this tracking in applications, it is crucial that it is possible to classify bifurcations and crossings correctly. Therefore, it is valuable to work on an automated recognition of crossings and bifurcations and their corresponding directions.

Throughout this thesis, we noted in several scenarios that the gauge frame is well able to adapt for curvature. This is one of the main advantages of this frame. However, exactly this adaptation for curvature also leads to crossings being interpreted as bifurcations. The gauge frame will in that case adapt towards the dominant vessel, which is undesirable for the other, crossing, vessel which will most likely be tracked less well. Therefore, it is important to come up with a way to use the gauge frame only when it is not near crossings. One of the ways to do this, would be by identifying, besides the bifurcations, the crossings as well. A more automated way to “identify crossings and bifurcations”, would be by using the Frangi-vesselness function on the image [37]. This cost function is weak at crossings and bifurcations, and therefore would be ideal to use in the mixed tensor field as given in Eq. (90).

Last, the tracking results in the fast marching method are calculated with an algorithm that is very similar to Dijkstra’s algorithm. In this thesis, we used this implemented algorithm to calculate the tracking results. Therefore, the tracking is not smooth. There are different ways to resolve this. First, one can calculate the weighted average of each point. By calculating $\mathbf{x}_i^{\text{new}} = \frac{\mathbf{x}_{i-1} + 2\mathbf{x}_i + \mathbf{x}_{i+1}}{4}$, one finds a tracking result which will be less jittery. Alternatively, one could use the steepest descent tracking that was used in the finite differences method. This also leads to a smooth tracking result.

A (Gauge Frame and Left Invariant) Metric Expressed with Respect to Static Frame

In the Fast Marching code of Jean-Marie Mirebeau, we need to supply the metric with respect to the standard, static frame. This means that we need to find \mathcal{G}_ϵ , where $\epsilon = \{\epsilon^1, \epsilon^2, \epsilon^3\} := \{dx, dy, d\theta\}$ denotes the standard dual basis and $e = \{e_1, e_2, e_3\} := \{\partial_x, \partial_y, \partial_\theta\}$ denotes the standard basis. In Eq. (45), \mathcal{G}_χ is expressed relative to the gauge frame (recall that the gauge basis is given by $\chi = \{\chi^1, \chi^2, \chi^3\}$). In order to find \mathcal{G}_ϵ , we first have to express \mathcal{G}_χ relative to the left invariant frame, which we denote by \mathcal{G}_ω (recall that the left invariant basis is given by $\omega = \{\omega^1, \omega^2, \omega^3\}$). Once, we have found an expression to do this, we still need to express \mathcal{G}_ω relative to the standard basis, which we denote by \mathcal{G}_ϵ , and denotes the metric that we need to enter in the code. Writing these steps as formulas, we get

$$\mathcal{G} = \sum_{i,j=1}^3 [\mathcal{G}_\chi]_{ij} \chi^i \otimes \chi^j = \sum_{i,j=1}^3 [\mathcal{G}_\omega]_{ij} \omega^i \otimes \omega^j = \sum_{i,j=1}^3 [\mathcal{G}_\epsilon]_{ij} \epsilon^i \otimes \epsilon^j \quad (94)$$

First, we start by expressing Eq. (45) in terms of the left invariant basis. We recall that we have

$$[\mathcal{G}_\chi] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & (1-\alpha)^{-2} & 0 \\ 0 & 0 & (1-\alpha)^{-2} \end{pmatrix}. \quad (95)$$

In addition to that, we recall that the gauge frame vectors \mathcal{B}_i are constructed out of the left invariant frame vectors \mathcal{A}_j . More specifically, we have

$$\mathcal{B}_i|_{\mathbf{p}} = \sum_{j=1}^3 c_i^j(\mathbf{p}) \mathcal{A}_j|_{\mathbf{p}}, \quad \chi^j|_{\mathbf{p}} = \sum_{i=1}^3 \bar{c}_i^j(\mathbf{p}) \omega^i|_{\mathbf{p}}.$$

We can rewrite this to

$$\begin{pmatrix} \mathcal{B}_1|_{\mathbf{p}} \\ \mathcal{B}_2|_{\mathbf{p}} \\ \mathcal{B}_3|_{\mathbf{p}} \end{pmatrix} = \mathbf{C}_{\mathbf{p}} \begin{pmatrix} \mathcal{A}_1|_{\mathbf{p}} \\ \mathcal{A}_2|_{\mathbf{p}} \\ \mathcal{A}_3|_{\mathbf{p}} \end{pmatrix}, \quad \begin{pmatrix} \chi^1|_{\mathbf{p}} \\ \chi^2|_{\mathbf{p}} \\ \chi^3|_{\mathbf{p}} \end{pmatrix} = \bar{\mathbf{C}}_{\mathbf{p}}^\top \begin{pmatrix} \omega^1|_{\mathbf{p}} \\ \omega^2|_{\mathbf{p}} \\ \omega^3|_{\mathbf{p}} \end{pmatrix}.$$

where

$$\mathbf{C}_{\mathbf{p}} := \begin{pmatrix} c_1^1(\mathbf{p}) & c_1^2(\mathbf{p}) & c_1^3(\mathbf{p}) \\ c_2^1(\mathbf{p}) & c_2^2(\mathbf{p}) & c_2^3(\mathbf{p}) \\ c_3^1(\mathbf{p}) & c_3^2(\mathbf{p}) & c_3^3(\mathbf{p}) \end{pmatrix}, \quad \bar{\mathbf{C}}_{\mathbf{p}} := \begin{pmatrix} \bar{c}_1^1(\mathbf{p}) & \bar{c}_1^2(\mathbf{p}) & \bar{c}_1^3(\mathbf{p}) \\ \bar{c}_2^1(\mathbf{p}) & \bar{c}_2^2(\mathbf{p}) & \bar{c}_2^3(\mathbf{p}) \\ \bar{c}_3^1(\mathbf{p}) & \bar{c}_3^2(\mathbf{p}) & \bar{c}_3^3(\mathbf{p}) \end{pmatrix}. \quad (96)$$

First, we need to identify the variables \bar{c}_i^j . Here, we use that for dual vectors, one should have $\langle \chi^j|_{\mathbf{p}}, \mathcal{B}_i|_{\mathbf{p}} \rangle = \delta_i^j$ in the gauge frame, just like $\langle \omega^j|_{\mathbf{p}}, \mathcal{A}_i|_{\mathbf{p}} \rangle = \delta_i^j$ in the left invariant frame. That means that we have, for $i, j = 1, 2, 3$ and for unknown \bar{c}_j^i

$$\begin{aligned} \delta_i^j &= \langle \chi^j|_{\mathbf{p}}, \mathcal{B}_i|_{\mathbf{p}} \rangle = \left\langle \sum_{k=1}^3 \bar{c}_k^j(\mathbf{p}) \omega^k|_{\mathbf{p}}, \sum_{l=1}^3 c_l^i(\mathbf{p}) \mathcal{A}_l|_{\mathbf{p}} \right\rangle \\ &= \sum_{k,l=1}^3 \bar{c}_k^j(\mathbf{p}) c_l^i(\mathbf{p}) \langle \omega^k|_{\mathbf{p}}, \mathcal{A}_l|_{\mathbf{p}} \rangle = \sum_{k=1}^3 \bar{c}_k^j(\mathbf{p}) c_k^i(\mathbf{p}). \end{aligned} \quad (97)$$

We can rewrite the expression in Eq. (97) to

$$\begin{aligned} &\begin{pmatrix} c_1^1(\mathbf{p}) & c_1^2(\mathbf{p}) & c_1^3(\mathbf{p}) \\ c_2^1(\mathbf{p}) & c_2^2(\mathbf{p}) & c_2^3(\mathbf{p}) \\ c_3^1(\mathbf{p}) & c_3^2(\mathbf{p}) & c_3^3(\mathbf{p}) \end{pmatrix} \begin{pmatrix} \bar{c}_1^1(\mathbf{p}) & \bar{c}_1^2(\mathbf{p}) & \bar{c}_1^3(\mathbf{p}) \\ \bar{c}_2^1(\mathbf{p}) & \bar{c}_2^2(\mathbf{p}) & \bar{c}_2^3(\mathbf{p}) \\ \bar{c}_3^1(\mathbf{p}) & \bar{c}_3^2(\mathbf{p}) & \bar{c}_3^3(\mathbf{p}) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \Leftrightarrow &\begin{pmatrix} \sum_{k=1}^3 c_1^k(\mathbf{p}) \bar{c}_k^1(\mathbf{p}) & \sum_{k=1}^3 c_1^k(\mathbf{p}) \bar{c}_k^2(\mathbf{p}) & \sum_{k=1}^3 c_1^k(\mathbf{p}) \bar{c}_k^3(\mathbf{p}) \\ \sum_{k=1}^3 c_2^k(\mathbf{p}) \bar{c}_k^1(\mathbf{p}) & \sum_{k=1}^3 c_2^k(\mathbf{p}) \bar{c}_k^2(\mathbf{p}) & \sum_{k=1}^3 c_2^k(\mathbf{p}) \bar{c}_k^3(\mathbf{p}) \\ \sum_{k=1}^3 c_3^k(\mathbf{p}) \bar{c}_k^1(\mathbf{p}) & \sum_{k=1}^3 c_3^k(\mathbf{p}) \bar{c}_k^2(\mathbf{p}) & \sum_{k=1}^3 c_3^k(\mathbf{p}) \bar{c}_k^3(\mathbf{p}) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \Leftrightarrow &\mathbf{C}_{\mathbf{p}} \bar{\mathbf{C}}_{\mathbf{p}} = \mathbf{I}. \end{aligned}$$

That means that we can easily calculate $\bar{\mathbf{C}}_{\mathbf{p}}$ by taking the inverse of $\mathbf{C}_{\mathbf{p}}$, so

$$\bar{\mathbf{C}}_{\mathbf{p}} = \mathbf{C}_{\mathbf{p}}^{-1}.$$

Next, we need to find out how we can calculate $[\mathcal{G}_\omega]$. We do so by rewriting Eq. (94), which gives us

$$\begin{aligned}\mathcal{G} &= \sum_{i,j=1}^3 [\mathcal{G}_\chi]_{ij} \chi^i \otimes \chi^j \\ &= \sum_{i,j=1}^3 [\mathcal{G}_\chi]_{ij} \left(\sum_{k=1}^3 \bar{c}_k^i(\mathbf{p}) \omega^k \right) \otimes \left(\sum_{l=1}^3 \bar{c}_l^j(\mathbf{p}) \omega^l \right).\end{aligned}\quad (98)$$

Since we know the tensor product is a linear transformation, we can rewrite Eq. (98) to

$$\begin{aligned}\mathcal{G} &= \sum_{i,j=1}^3 [\mathcal{G}_\chi]_{ij} \sum_{k,l=1}^3 \bar{c}_k^i(\mathbf{p}) \bar{c}_l^j(\mathbf{p}) \omega^k \otimes \omega^l \\ &= \sum_{k,l=1}^3 \left(\sum_{i,j=1}^3 \bar{c}_k^i(\mathbf{p}) [\mathcal{G}_\chi]_{ij} \bar{c}_l^j(\mathbf{p}) \right) \omega^k \otimes \omega^l \\ &= \sum_{k,l=1}^3 \left(\sum_{i,j=1}^3 [\bar{\mathbf{C}}_{\mathbf{p}}]_{ki} [\mathcal{G}_\chi]_{ij} [\bar{\mathbf{C}}_{\mathbf{p}}]_{lj} \right) \omega^k \otimes \omega^l \\ &= \sum_{k,l=1}^3 \left([\bar{\mathbf{C}}_{\mathbf{p}} \mathcal{G}_\chi \bar{\mathbf{C}}_{\mathbf{p}}^\top]_{kl} \right) \omega^k \otimes \omega^l \\ &= \sum_{k,l=1}^3 [\mathcal{G}_\omega]_{kl} \omega^k \otimes \omega^l.\end{aligned}$$

We can express $[\mathcal{G}_\omega]$ as a matrix product, being

$$[\mathcal{G}_\omega] = \bar{\mathbf{C}}_{\mathbf{p}} [\mathcal{G}_\chi] \bar{\mathbf{C}}_{\mathbf{p}}^\top = \mathbf{C}_{\mathbf{p}}^{-1} [\mathcal{G}_\chi] \mathbf{C}_{\mathbf{p}}^{-\top}.$$

With similar calculations, we can express the metric in terms of the standard basis ϵ . That means that we need to find the transformation matrices $\mathbf{Q}_{\mathbf{p}}$ and $\bar{\mathbf{Q}}_{\mathbf{p}}$ such that

$$\begin{pmatrix} \mathcal{A}_1|_{\mathbf{p}} \\ \mathcal{A}_2|_{\mathbf{p}} \\ \mathcal{A}_3|_{\mathbf{p}} \end{pmatrix} = \mathbf{Q}_{\mathbf{p}} \begin{pmatrix} \mathbf{e}_1|_{\mathbf{p}} \\ \mathbf{e}_2|_{\mathbf{p}} \\ \mathbf{e}_3|_{\mathbf{p}} \end{pmatrix}, \quad \begin{pmatrix} \omega^1|_{\mathbf{p}} \\ \omega^2|_{\mathbf{p}} \\ \omega^3|_{\mathbf{p}} \end{pmatrix} = \bar{\mathbf{Q}}_{\mathbf{p}}^\top \begin{pmatrix} \epsilon^1|_{\mathbf{p}} \\ \epsilon^2|_{\mathbf{p}} \\ \epsilon^3|_{\mathbf{p}} \end{pmatrix}$$

is satisfied. That means that it is given by

$$\mathbf{Q}_{\mathbf{p}} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \bar{\mathbf{Q}}_{\mathbf{p}} = \mathbf{Q}_{\mathbf{p}}^{-1}.\quad (99)$$

Repeating the arguments above, we find

$$[\mathcal{G}_\epsilon] = \bar{\mathbf{Q}}_{\mathbf{p}} [\mathcal{G}_\omega] \bar{\mathbf{Q}}_{\mathbf{p}}^\top = \mathbf{Q}_{\mathbf{p}}^{-1} \mathbf{C}_{\mathbf{p}}^{-1} [\mathcal{G}_\chi] \mathbf{C}_{\mathbf{p}}^{-\top} \mathbf{Q}_{\mathbf{p}}^{-\top} = (\mathbf{C}_{\mathbf{p}} \mathbf{Q}_{\mathbf{p}})^{-1} [\mathcal{G}_\chi] (\mathbf{C}_{\mathbf{p}} \mathbf{Q}_{\mathbf{p}})^{-\top},\quad (100)$$

where

$$\mathcal{G} = \sum_{m,n=1}^3 [\mathcal{G}_\epsilon]_{mn} \epsilon^m \otimes \epsilon^n.$$

The more extended arguments for the statement in Eq. (100) are

$$\begin{aligned}\mathcal{G} &= \sum_{k,l=1}^3 [\mathcal{G}_\omega]_{kl} \omega^k \otimes \omega^l \\ &= \sum_{k,l=1}^3 [\mathcal{G}_\omega]_{kl} \left(\sum_{m=1}^3 [\bar{\mathbf{Q}}_{\mathbf{p}}]_{mk} \epsilon^m \right) \otimes \left(\sum_{n=1}^3 [\bar{\mathbf{Q}}_{\mathbf{p}}]_{nl} \epsilon^n \right) \\ &= \sum_{m,n=1}^3 \left(\sum_{k,l=1}^3 [\bar{\mathbf{Q}}_{\mathbf{p}}]_{mk} [\mathcal{G}_\omega]_{kl} [\bar{\mathbf{Q}}_{\mathbf{p}}]_{ln} \right) \epsilon^m \otimes \epsilon^n \\ &= \sum_{m,n=1}^3 [\bar{\mathbf{Q}}_{\mathbf{p}} \mathcal{G}_\omega \bar{\mathbf{Q}}_{\mathbf{p}}^\top]_{mn} \epsilon^m \otimes \epsilon^n.\end{aligned}$$

In addition to the metric, we also need to supply the directions in which one may only move forward or backwards. In our case, the dual vectors are denoted by \mathcal{B}_1 , so the primal vector is given by χ^1 . We need to express this vector in the standard frame as well. That means that we have

$$\chi^1|_{\mathbf{p}} = \sum_{i=1}^3 \bar{c}_i^1(\mathbf{p}) \omega^i|_{\mathbf{p}} = \begin{pmatrix} \bar{c}_1^1(\mathbf{p}) \\ \bar{c}_2^1(\mathbf{p}) \\ \bar{c}_3^1(\mathbf{p}) \end{pmatrix} \cdot \begin{pmatrix} \omega^1|_{\mathbf{p}} \\ \omega^2|_{\mathbf{p}} \\ \omega^3|_{\mathbf{p}} \end{pmatrix} = [\bar{\mathbf{C}}_{\mathbf{p}}^{\top}]_{1,\cdot} \begin{pmatrix} \omega^1|_{\mathbf{p}} \\ \omega^2|_{\mathbf{p}} \\ \omega^3|_{\mathbf{p}} \end{pmatrix}.$$

A.1 Conclusion

In conclusion, we found that we can express the gauge metric with respect to the static frame by

$$[\mathcal{G}_{\epsilon}] = (\mathbf{C}_{\mathbf{p}} \mathbf{Q}_{\mathbf{p}})^{-1} [\mathcal{G}_{\chi}] (\mathbf{C}_{\mathbf{p}} \mathbf{Q}_{\mathbf{p}})^{-\top}, \quad (101)$$

where $\mathbf{C}_{\mathbf{p}}$ denotes the relationship between \mathcal{B}_i and \mathcal{A}_j , and where $\mathbf{Q}_{\mathbf{p}}$ contains the relationship between \mathcal{A}_j and ϵ_k , which are given in Eq. (96) and Eq. (99) respectively. Furthermore, $[\mathcal{G}_{\chi}]$ is based on the considered metric, and is given in Eq. (95) for the metric tensor field introduced in Eq. (45).

The forward direction in the gauge frame is given by

$$\chi^1|_{\mathbf{p}} = [\mathbf{C}_{\mathbf{p}}^{-\top}]_{1,\cdot} \begin{pmatrix} \omega^1|_{\mathbf{p}} \\ \omega^2|_{\mathbf{p}} \\ \omega^3|_{\mathbf{p}} \end{pmatrix} = [\mathbf{C}_{\mathbf{p}}^{-\top}]_{1,\cdot} \mathbf{Q}_{\mathbf{p}}^{-\top} \begin{pmatrix} \epsilon^1|_{\mathbf{p}} \\ \epsilon^2|_{\mathbf{p}} \\ \epsilon^3|_{\mathbf{p}} \end{pmatrix}.$$

When we want to express the left invariant frame in terms of the static frame $[\mathcal{G}_{\epsilon}]$, we can still use the relations in Eq. (101). In this special case, we have $\mathbf{C}_{\mathbf{p}} = \mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^3$ denotes the identity matrix. The matrix $[\mathcal{G}_{\chi}]$ is replaced by $[\tilde{\mathcal{G}}_{\omega}]$ which is given by

$$[\tilde{\mathcal{G}}_{\omega}] = \begin{pmatrix} \xi^2 & 0 & 0 \\ 0 & \xi^2/\zeta^2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

for the metric tensor field introduced in Eq. (43). The forward direction of the left invariant frame is given by

$$\omega^1|_{\mathbf{p}} = [\mathbf{Q}_{\mathbf{p}}^{-\top}]_{1,\cdot} \begin{pmatrix} \epsilon^1|_{\mathbf{p}} \\ \epsilon^2|_{\mathbf{p}} \\ \epsilon^3|_{\mathbf{p}} \end{pmatrix}.$$

B Control Sets

We discussed in Section 1.1.6 that the control sets need to be closed and vary continuously with respect to the point $\mathbf{p} \in \mathbb{M}_2$ in the sense of the Hausdorff distance. In [17], a proof is provided that the left invariant frame meets these requirements. However, this still needs to be shown for the gauge frame. Here, we will give a short summary of the needed theory and conclusions for the left invariant frame. Proofs can be found in Appendix A of [17].

First, we introduce some general theory. Afterwards, we will elaborate on what is needed for the specific frames that we are considering.

Definition B.1. Given a metric space \mathbb{E} , let $\mathcal{K}(\mathbb{E})$ be the collection of non-empty compact subsets of \mathbb{E} .

Given a normed vector space V , let $\mathfrak{C}(V) \subset \mathcal{K}(V)$ be the collection of non-empty compact subsets of V , which are convex and contained in the unit ball.

We introduce the distance function d_A and the Hausdorff distance $\mathcal{H}(A, B)$.

Definition B.2. Given a metric space \mathbb{E} , let $A, B \in \mathcal{K}(\mathbb{E})$. Then, the distance function $d_A : \mathbb{E} \rightarrow \mathbb{R}^+$ and the Hausdorff distance $\mathcal{H}(A, B)$ are defined respectively by

$$d_A(x) := \inf_{y \in A} d(x, y), \quad \mathcal{H}(A, B) := \max \left\{ \sup_{x \in B} d_A(x), \sup_{x \in A} d_B(x) \right\}.$$

Definition B.3. A family of controls \mathcal{B} on the set \mathbb{X} is an element of the set \mathfrak{B} defined by

- If $\mathbb{X} \subset \mathbb{E}$ an Euclidean vector space, then $\mathfrak{B} := C^0(\mathbb{X}, \mathfrak{C}(\mathbb{E}))$.
- If $\mathbb{X} \subset \mathbb{M}$ a Riemannian manifold, then $\mathfrak{B} := \{\mathcal{B} \in C^0(\mathbb{X}, \mathcal{K}(T\mathbb{M})) \mid \forall \mathbf{p} \in \mathbb{X}, \mathcal{B}(\mathbf{p}) \in \mathfrak{C}(T_{\mathbf{p}}\mathbb{M})\}$.

Definition B.4. A path γ is $T\mathcal{B}$ -admissible, where $\gamma \in \Gamma$, $T \in \mathbb{R}^+$ and $\mathcal{B} \in \mathfrak{B}$, iff for almost every $t \in [0, 1]$

$$\dot{\gamma}(t) \in T\mathcal{B}(\gamma(t)).$$

Definition B.5. For any $\mathcal{B} \in \mathfrak{B}$, $\mathbf{p}, \mathbf{q} \in \mathbb{X}$, we let

$$T_{\mathcal{B}}(\mathbf{p}, \mathbf{q}) := \inf \{T \geq 0 \mid \exists \gamma \in \Gamma, \gamma(0) = \mathbf{p}, \gamma(1) = \mathbf{q} \text{ and } \gamma \text{ is } T\mathcal{B}\text{-admissible}\}.$$

Proposition B.1. With the notations of Definition 1.19, the sets $\mathbf{p} \in \mathbb{M} \mapsto \mathcal{B}_{\mathcal{G}}(\mathbf{p})$ form a family of controls on $(\mathbb{M}, \delta \|\cdot\|)$. In addition for all $\mathbf{p}, \mathbf{q} \in \mathbb{M}$

$$d_{\mathcal{G}}(\mathbf{p}, \mathbf{q}) = T_{\mathcal{B}_{\mathcal{G}}}(\mathbf{p}, \mathbf{q}).$$

B.1 Left Invariant Frame

In Section 2, we introduced the left invariant metric tensor fields \mathcal{G}_{LIF} and $\mathcal{G}_{\text{LIF}}^f$. Both tensor fields are given by

$$\begin{aligned} \mathcal{G}_{\text{LIF}} &:= \frac{1}{(1 + \beta/C)^2} \left(\xi^2 \left(\omega^1 \otimes \omega^1 + \frac{1}{\zeta^2} \omega^2 \otimes \omega^2 \right) + \omega^3 \otimes \omega^3 \right). \\ \mathcal{G}_{\text{LIF}}^f &:= \frac{1}{(1 + \beta/C)^2} \left(\xi^2 \left(\omega^1 \otimes \omega^1 + (\varepsilon^{-2} - 1) (\omega^1 \otimes \omega^1)_- + \frac{1}{\zeta^2} \omega^2 \otimes \omega^2 \right) + \omega^3 \otimes \omega^3 \right). \end{aligned}$$

Now, we will denote these metric tensor fields by $\mathcal{F}_{\varepsilon} := \sqrt{\mathcal{G}_{\text{LIF}}}$ and $\mathcal{F}_{\varepsilon}^f := \sqrt{\mathcal{G}_{\text{LIF}}^f}$ to stress the ε dependency of the metric, and we choose $\zeta = \varepsilon$. We note that we have

$$\begin{aligned} (\mathcal{F}_0)^2 &= \begin{cases} \frac{1}{(1 + \beta/C)^2} (\xi^2 \omega^1 \otimes \omega^1 + \omega^3 \otimes \omega^3) & \text{if } \dot{\mathbf{x}} \propto \mathbf{n}, \\ +\infty & \text{else.} \end{cases} \\ (\mathcal{F}_0^f)^2 &= \begin{cases} \frac{1}{(1 + \beta/C)^2} (\xi^2 \omega^1 \otimes \omega^1 + \omega^3 \otimes \omega^3) & \text{if } \dot{\mathbf{x}} \propto \mathbf{n} \text{ and } \dot{\mathbf{x}} \cdot \mathbf{n} \geq 0, \\ +\infty & \text{else.} \end{cases} \end{aligned}$$

Proposition B.2. The metrics $\mathcal{F}_{\varepsilon}$ and $\mathcal{F}_{\varepsilon}^f$ are indeed metrics in the sense of Definition 1.19 for all $0 \leq \varepsilon \leq 1$. The associated control sets $\mathcal{B}_{\varepsilon} := \mathcal{B}_{\mathcal{F}_{\varepsilon}}$ and $\mathcal{B}_{\varepsilon}^f := \mathcal{B}_{\mathcal{F}_{\varepsilon}^f}$ depend continuously on the parameter $\varepsilon \in [0, 1]$, and satisfy the inclusions $\mathcal{B}_{\varepsilon}(\mathbf{p}) \subset \mathcal{B}_{\varepsilon'}(\mathbf{p})$ and $\mathcal{B}_{\varepsilon}^f(\mathbf{p}) \subset \mathcal{B}_{\varepsilon'}^f(\mathbf{p})$ for any $\mathbf{p} \in \mathbb{M}$ and $0 \leq \varepsilon \leq \varepsilon' \leq 1$.

In order to prove Proposition B.2, we need the following two lemmas.

Lemma B.1. Let B be a compact subset of a metric space \mathbb{E} , and let $\varphi \in C^0(B, \mathbb{E})$. Then

$$\mathcal{H}(B, \varphi(B)) \leq \sup_{x \in B} d(x, \varphi(x)).$$

We identify the following similarities:

$$\begin{aligned} \|\dot{\mathbf{x}}\|^2 &= \left(\omega^1|_{\mathbf{p}} \otimes (\omega^1|_{\mathbf{p}})\right)(\dot{\mathbf{p}}, \dot{\mathbf{p}}) + \left((\omega^2|_{\mathbf{p}} \otimes (\omega^2|_{\mathbf{p}})\right)(\dot{\mathbf{p}}, \dot{\mathbf{p}}); \\ |\dot{\mathbf{x}} \cdot \mathbf{n}|^2 &= \left((\omega^1|_{\mathbf{p}} \otimes (\omega^1|_{\mathbf{p}})\right)(\dot{\mathbf{p}}, \dot{\mathbf{p}}); \\ \|\dot{\mathbf{x}} \wedge \mathbf{n}\|^2 &:= \|\dot{\mathbf{x}}\|^2 - |\dot{\mathbf{x}} \cdot \mathbf{n}|^2 = \left((\omega^2|_{\mathbf{p}} \otimes (\omega^2|_{\mathbf{p}})\right)(\dot{\mathbf{p}}, \dot{\mathbf{p}}); \\ \|\dot{\mathbf{n}}\|^2 &= \left((\omega^3|_{\mathbf{p}} \otimes (\omega^3|_{\mathbf{p}})\right)(\dot{\mathbf{p}}, \dot{\mathbf{p}}). \end{aligned}$$

Therefore, Lemma 6 from [17] states the same as Lemma B.2.

Lemma B.2. Let $\mathbf{n}_1, \mathbf{n}_2 \in S^1$, let $a_1, a_2, b_1, b_2 \geq 1$ and let $\varepsilon_1, \varepsilon_2 \in [0, 1]$. For each $i \in \{1, 2\}$, let B_i be the collection of all $(\dot{\mathbf{x}}, \dot{\mathbf{n}}) \in \mathbb{R}^d \times \mathbb{R}^d$ obeying

$$\begin{aligned} \dot{\mathbf{n}} \cdot \mathbf{n}_i &= 0, \\ a_i^2 \|\dot{\mathbf{n}}\|^2 + b_i^2 \left(|\dot{\mathbf{x}} \cdot \mathbf{n}_i|^2 + \varepsilon_i^{-2} \|\dot{\mathbf{x}} \wedge \mathbf{n}_i\|^2 \right) &\leq 1 \text{ if } \varepsilon_i > 0 \\ a_i^2 \|\dot{\mathbf{n}}\|^2 + b_i^2 |\dot{\mathbf{x}} \cdot \mathbf{n}_i|^2 &\leq 1 \text{ and } \dot{\mathbf{x}} \wedge \mathbf{n}_i = 0 \text{ if } \varepsilon_i = 0. \end{aligned}$$

Then

$$\mathcal{H}(B_1, B_2) \leq |a_1^{-1} - a_2^{-1}| + |b_1^{-1} - b_2^{-1}| + \sqrt{2(1 - \mathbf{n}_1 \cdot \mathbf{n}_2)} + |\zeta \varepsilon_1 - \zeta \varepsilon_2|.$$

The same estimate holds for the sets B_i^f , $i \in \{1, 2\}$, defined by the inequalities

$$\begin{aligned} \dot{\mathbf{n}} \cdot \mathbf{n}_i &= 0, \\ a_i^2 \|\dot{\mathbf{n}}\|^2 + b_i^2 \left((\dot{\mathbf{x}} \cdot \mathbf{n}_i)_+^2 + \varepsilon_i^{-2} \|\dot{\mathbf{x}} \wedge \mathbf{n}_i\|^2 + (\dot{\mathbf{x}} \cdot \mathbf{n}_i)_-^2 \right) &\leq 1 \text{ if } \varepsilon_i > 0 \\ a_i^2 \|\dot{\mathbf{n}}\|^2 + b_i^2 (\dot{\mathbf{x}} \cdot \mathbf{n}_i)_+^2 &\leq 1 \text{ and } \dot{\mathbf{x}} \wedge \mathbf{n}_i = 0, \dot{\mathbf{x}} \cdot \mathbf{n}_i \geq 0 \text{ if } \varepsilon_i = 0. \end{aligned}$$

B.1.1 Visualization of the Control Sets

There are multiple ways in which we can calculate them. The first method uses that we know the expression for an ellipsoid. For the (an)isotropic left invariant frame, using $\theta = 0$, this is given by

$$(\xi x)^2 + \left(\frac{\xi y}{\zeta}\right)^2 + \theta^2 = 1.$$

In the visualization tool, we need to give the distance r_i from the center to the surface for 162 different directions. Therefore, we assume that we are looking in direction (x_i, y_i, θ_i) . We assume that $\|(x_i, y_i, \theta_i)\|^2 = 1$ such that we have

$$r^2 x_i^2 + r^2 y_i^2 + r^2 \theta_i^2 = r^2.$$

Then, one should have

$$\begin{cases} \tilde{x}_i^2 + \tilde{y}_i^2 + \tilde{\theta}_i^2 = r_i^2, \\ (\xi \tilde{x}_i)^2 + \left(\frac{\xi \tilde{y}_i}{\zeta}\right)^2 + \tilde{\theta}_i^2 = 1, \end{cases} \quad (102)$$

where $(\tilde{x}_i, \tilde{y}_i, \tilde{\theta}_i) = (rx_i, ry_i, r\theta_i)$. Note that the first equation is always satisfied by definition. Rewriting the second equation given in Eq. (102) results in

$$r = \pm \sqrt{\frac{1}{(\xi x_i)^2 + \left(\frac{\xi y_i}{\zeta}\right)^2 + \theta_i^2}}. \quad (103)$$

Rotation of the axes over an angle θ gives

$$x_{\text{new}} = \cos \theta x + \sin \theta y, \quad y_{\text{new}} = -\sin \theta x + \cos \theta y, \quad \theta_{\text{new}} = \theta.$$

Resulting in radii

$$r = \pm \sqrt{\frac{1}{(\xi (\cos \theta x_1 + \sin \theta y_1))^2 + \left(\frac{\xi (-\sin \theta x_1 + \cos \theta y_1)}{\zeta}\right)^2 + \theta_1^2}}. \quad (104)$$

We note that we can express the radii using matrices

$$\begin{aligned} r &= \pm \left((x_i \ y_i \ \theta_i) \begin{pmatrix} \cos \theta & -\sin \theta & \\ \sin \theta & \cos \theta & \\ & & 1 \end{pmatrix} \begin{pmatrix} \xi^2 & & \\ & \xi^2/\zeta^2 & \\ & & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta & \\ -\sin \theta & \cos \theta & \\ & & 1 \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ \theta_i \end{pmatrix} \right)^{-1/2} \\ &= \pm \left((x_i \ y_i \ \theta_i) Q_{\mathbf{p}}^{-1} [\tilde{\mathcal{G}}_{\omega}] Q_{\mathbf{p}}^{-\top} \begin{pmatrix} x_i \\ y_i \\ \theta_i \end{pmatrix} \right)^{-1/2}, \end{aligned} \quad (105)$$

where $Q_{\mathbf{p}}$ was defined in Appendix A and with

$$[\tilde{\mathcal{G}}_{\omega}] = \begin{pmatrix} \xi^2 & & \\ & \xi^2/\zeta^2 & \\ & & 1 \end{pmatrix}. \quad (106)$$

The control sets of the (an)isotropic left invariant frame are visualized in Figure 47. We see that the Euclidean distance to \mathbf{p} is bigger when moving forward, than it is moving backward. In addition to that, moving forward is encouraged over moving sideways or changing angular direction.

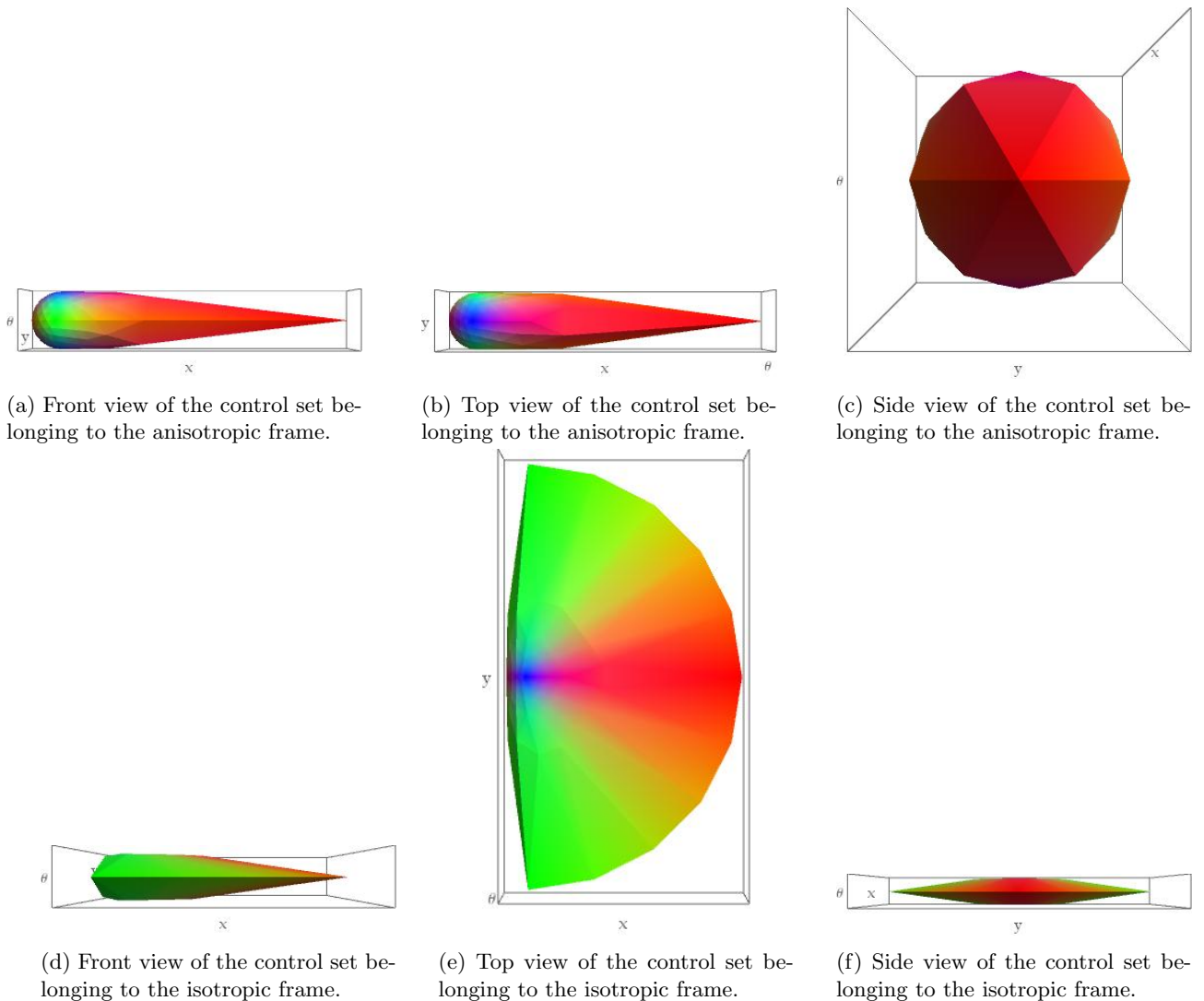


Figure 47: Control sets for the isotropic and anisotropic left invariant frame ($\xi = 0.1$ and $\zeta = 0.1$ or $\zeta = 1$).

B.2 Gauge Frame

We state that the control sets of the gauge frame are also continuous with respect to the Hausdorff distance since the control sets are continuous deformations of the left invariant control sets. However, this is not formally proven yet.

B.2.1 Visualization of the Control Sets

We can calculate the control sets of the gauge frame in a way similar to what we did in the left invariant frame. Looking at the expression in Eq. (105), we realize that it is similar to Eq. (100). Generalizing this to the gauge frame yields

$$r_i = \left((x_i \ y_i \ \theta_i) Q_{\mathbf{P}}^{-1} C_{\mathbf{P}}^{-1} [\mathcal{G}_{\chi}] C_{\mathbf{P}}^{-\top} Q_{\mathbf{P}}^{-\top} \begin{pmatrix} x_i \\ y_i \\ \theta_i \end{pmatrix} \right)^{-1/2} = \left((x_i \ y_i \ \theta_i) (C_{\mathbf{P}} Q_{\mathbf{P}})^{-1} [\mathcal{G}_{\chi}] (C_{\mathbf{P}} Q_{\mathbf{P}})^{-\top} \begin{pmatrix} x_i \\ y_i \\ \theta_i \end{pmatrix} \right)^{-1/2}. \quad (107)$$

An example of a visualization is given in Figure 48. Here, we used the gauge frame, where

$$C_{\mathbf{P}} = \begin{pmatrix} 3.38282 & -0.717793 & -0.938303 \\ 2.07567 & 9.78221 & 0. \\ 0.265325 & -0.0562988 & 0.999632 \end{pmatrix}, \quad \theta = \frac{\pi}{8}.$$

We see that the control sets are a deformation of the control sets in the left invariant frame (Figure 47). The control sets are better aligned with line structures in the image on which these control sets were based.

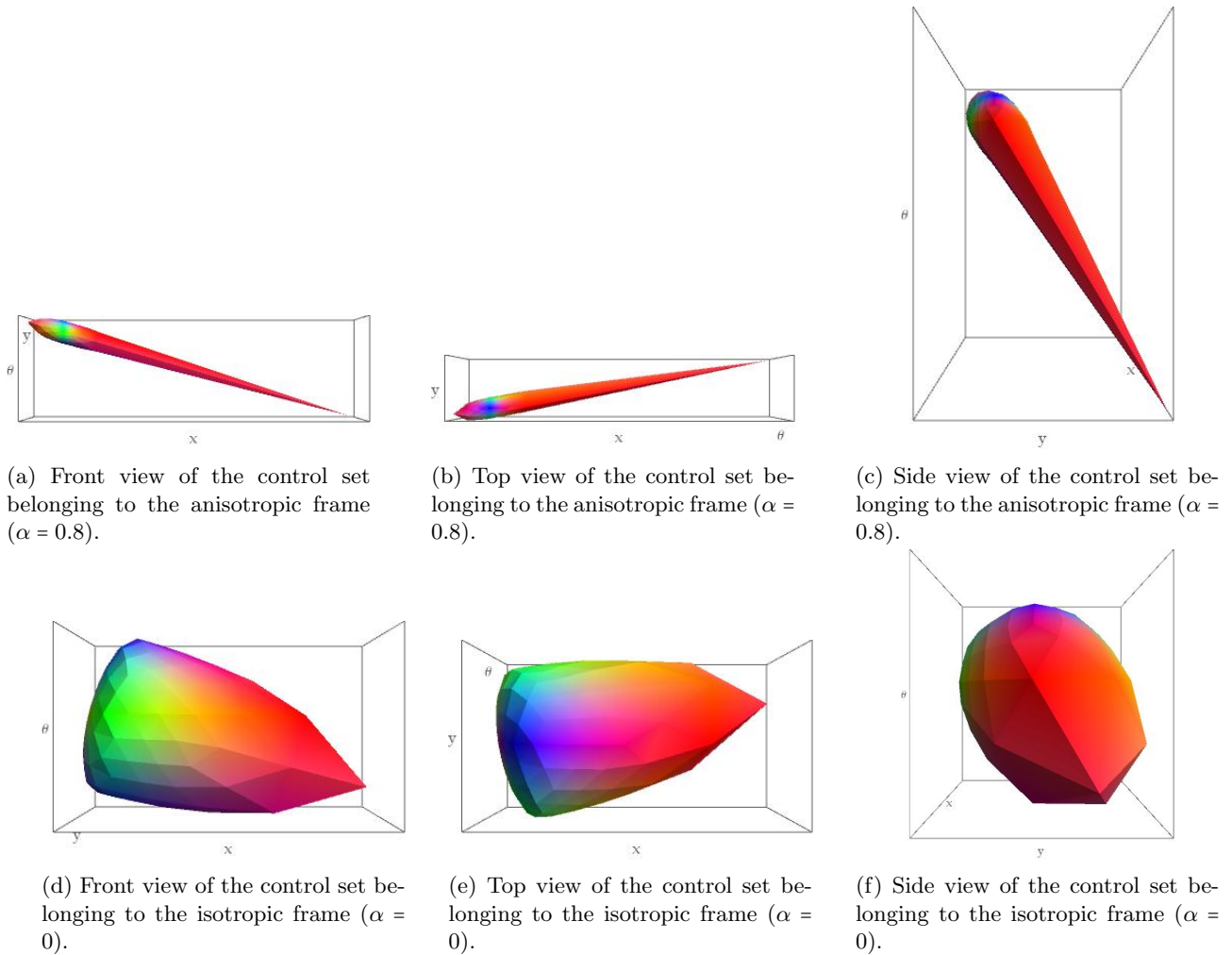


Figure 48: Control sets for the isotropic and anisotropic left invariant frame for the gauge frame given by $\begin{pmatrix} 3.38282 & -0.717793 & -0.938303 \\ 2.07567 & 9.78221 & 0. \\ 0.265325 & -0.0562988 & 0.999632 \end{pmatrix}$, using $\xi = 0.1$, $\zeta = 0.1$ and $\alpha = 0$ or $\alpha = 0.9$ for the isotropic and anisotropic frames respectively.

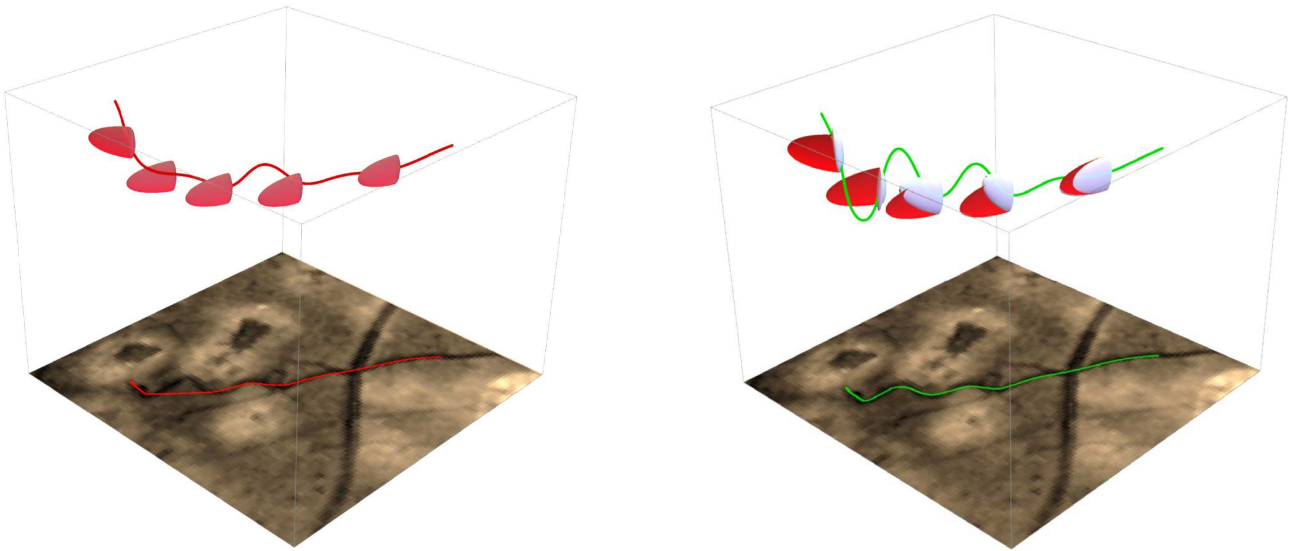
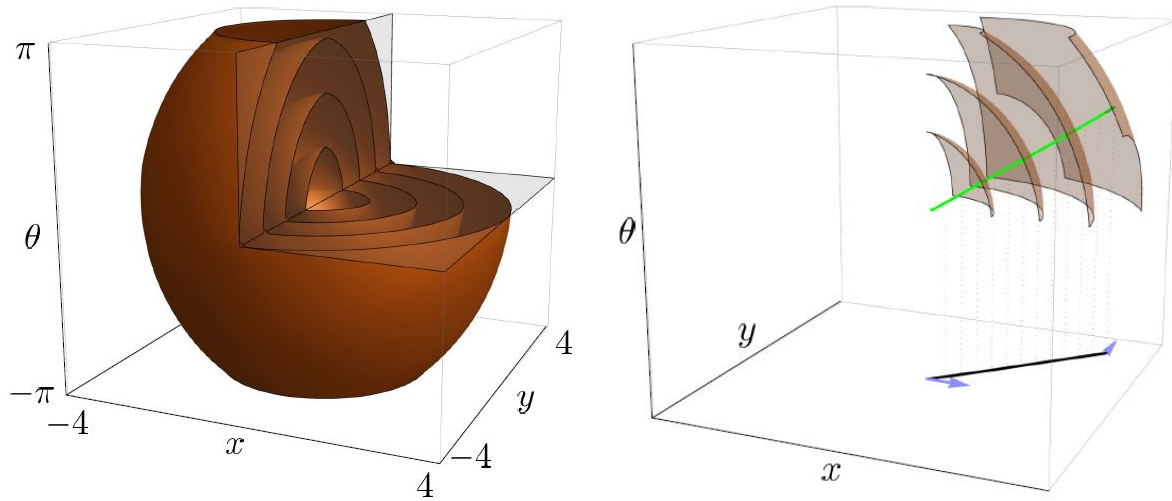


Figure 49: The red control sets correspond to the left invariant frame, whereas the white control sets correspond to the gauge frame. The control sets of the gauge frame are better aligned with the underlying vessel structure. Figures received from R. Duits.

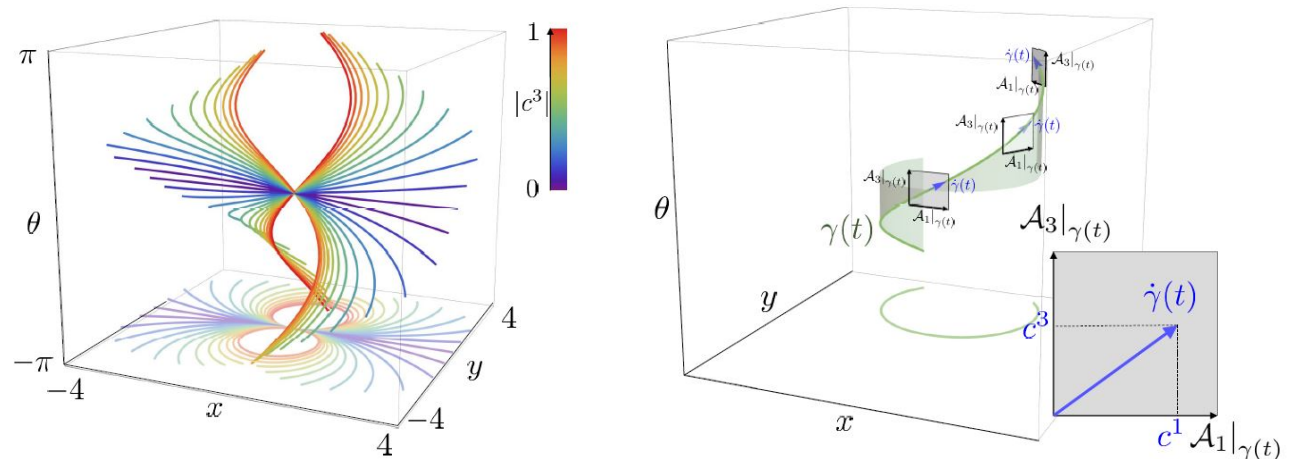
A visualization of both the left invariant and gauge frame can be found in Figure 49. We see that the control sets of the gauge frame (white) are better aligned with the underlying vessel structure than those of the left invariant frame (red). Therefore, the vessel tracking result in the gauge frame (green) is a more accurate than the tracking result in the left invariant frame (red).

C Cartan Connections

In this section, we give a short overview of Cartan connections and their influence on shortest and straight curves. In Figure 50, we illustrate that short curves are not necessarily the same as straight curves, for which we state the corresponding theorems in this section.



(a) Short curves in the Riemannian manifold. The geodesics are straight lines. Left: Geodesically equidistant surfaces \mathcal{S}_t^1 . Right: Riemannian geodesics for the isotropic Riemannian case $\epsilon = 1$, $\mathcal{C} = 1$.



(b) Straight curves. A set of horizontal exponential curves in $SE(2)$ for which $\dot{\gamma}(\tau) = c^1 \mathcal{A}_1|_{\gamma(\tau)} + c^3 \mathcal{A}_3|_{\gamma(\tau)}$, with constant tangent vector components c^1 and c^3 . These curves are auto-parallel ('straight curves' in torqued and curved geometry modeled by Lie-Cartan connection $\nabla^{[1]}$).

Figure 50: Visualization of the difference between straight and short curves in the Riemannian manifold. Taken from [38].

We will only consider the Hamiltonian on a Riemannian manifold. The theorems (and corresponding proofs) of the sub-Riemannian case with the left invariant frame as reference frame can be found in [38]. We will consider the Lie-Cartan connections in support of understanding Riemannian geometry when the Lie group G is considered as a Riemann manifold (G, \mathcal{G}) with a left-invariant Riemannian metric tensor field given by

$$\mathcal{G} = \sum_{i,j=1}^n g_{ij} \omega^i \otimes \omega^j, \quad (108)$$

with g_{ij} constant relative to the left-invariant co-frame ω^i introduced in Section 1.3 such that $[g_{ij}] \in \mathbb{R}^{n \times n}$ is symmetric positive definite. We restrict ourselves to the diagonal case in Theorem 3, Theorem 4 and Theorem 5. The metric isomorphism between TM and T^*M \mathcal{G} is written as

$$\tilde{\mathcal{G}}(X) = \mathcal{G}(X, \cdot). \quad (109)$$

Before we state the theorems in the left invariant and gauge frame, we need to introduce some theory. We start with the general definition of Cartan connections on a Lie group.

Definition C.1 (Cartan/canonical connection on Lie group). A Cartan (or canonical) connection on a Lie group is a vector bundle connection with the following additional properties:

1. left invariance:

$$X, Y \text{ are left invariant vector field} \Rightarrow \nabla_X Y \text{ is a left invariant vector field,} \quad (110)$$

2. for any $\mathbf{a} \in T_e(G)$ the exponential curve and auto-parallel curve coincide:

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = 0 \quad \text{where} \quad \gamma(t) = \gamma(0) \exp(\mathbf{a}t). \quad (111)$$

C.1 Left Invariant Frame

Then, we consider the Lie-Cartan connection on the left invariant frame specifically.

Definition C.2 (Lie-Cartan Connection). Consider a Lie group with Lie brackets $[\cdot, \cdot]$ and structure constants $c_{ij}^k \in \mathbb{R}$ s.t. $[\mathcal{A}_i, \mathcal{A}_j] = \sum_k c_{ij}^k \mathcal{A}_k$. Then the Lie-Cartan connection indexed with $\nu \in \mathbb{R}$ equals

$$\nabla^{[\nu]} := \sum_{i,k=1}^n \omega^i \otimes (\mathcal{A}_i \circ \omega^k(\cdot)) \mathcal{A}_k + \sum_{i,j,k=1}^n \omega^i \otimes \omega^j \nu c_{ij}^k \mathcal{A}_k. \quad (112)$$

Then the Lie-Cartan connection with covariant derivative of a field $Y = \sum_{k=1}^n y^k \mathcal{A}_k$, along a smooth vector field $X = \sum_{i=1}^n x^i \mathcal{A}_i$, is given by

$$\nabla_X^{[\nu]} Y = \nabla^{[\nu]}(X, Y) = \sum_{k=1}^n \left(\dot{y}^k + \sum_{i,j=1}^n \nu c_{ij}^k x^i y^j \right) \mathcal{A}_k,$$

where $\dot{y}^k(t) = \frac{d}{dt} y^k(\gamma(t))$ which equals $\dot{y}^k(t) = (X y^k)(\gamma(t))$ and $x^i = \dot{\gamma}(t)$ where $x^i|_{\gamma(t)} := \dot{\gamma}^i(t) = \langle \omega^i|_{\gamma(t)}, \dot{\gamma}(t) \rangle$ along the flow-lines γ of smooth vector field X . A ‘flowline’ is a smooth curve γ satisfying $\dot{\gamma}(t) = X_{\gamma(t)}$. With slight abuse of notation, we write

$$\nabla_{\dot{\gamma}}^{[\nu]} Y = \sum_{k=1}^n \left(\dot{y}^k + \sum_{i,j=1}^n \nu c_{ij}^k \dot{\gamma}^i \gamma^j \right) \mathcal{A}_k.$$

The corresponding dual connection on the co-tangent bundle is given by

$$\nabla_{\dot{\gamma}}^{[\nu],*} \lambda = \sum_{i=1}^n \left(\dot{\lambda}_i + \sum_{k,j=1}^n \nu c_{ij}^k \lambda_k \dot{\gamma}^j \right) \omega^i,$$

where $\lambda = \sum_{i=1}^n \lambda_i \omega^i \in T^*(G)$.

Next, we give the definition of Poisson brackets.

Definition C.3 (Poisson brackets). The Poisson bracket of two scalar functions $f_1, f_2 \in C^\infty(T^*\mathbb{M})$ is given by

$$\{f_1, f_2\} = \sum_{i=1}^n (\mathcal{A}_i f_2) \frac{\partial f_1}{\partial \lambda_i} - \frac{\partial f_2}{\partial \lambda_i} (\mathcal{A}_i f_1). \quad (113)$$

Last, we introduce g^{ij} . These contravariant components are defined as

$$\begin{pmatrix} g^{11} & \cdots & g^{1n} \\ \vdots & \ddots & \vdots \\ g^{n1} & \cdots & g^{nn} \end{pmatrix} := \begin{pmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{n1} & \cdots & g_{nn} \end{pmatrix}^{-1}.$$

Theorem 3 (Theorem 2 from [38]). *The Hamiltonian on Riemannian manifold (G, \mathcal{G}) , with left-invariant metric tensor field \mathcal{G} introduced in Eq. (108) equals*

$$\mathfrak{h} = \frac{1}{2} \sum_{i=1}^n \lambda^i \lambda_i = \frac{1}{2} \sum_{i,j=1}^n g^{ij} \lambda_i \lambda_j, \quad \lambda^i = \sum_{j=1}^n g^{ij} \lambda_j \quad (114)$$

and the corresponding Hamiltonian flow (generated by the Hamiltonian vector field $\vec{\mathfrak{h}}$) can be written as (recall the definition of linear map $\tilde{\mathcal{G}}(X) := \mathcal{G}(X, \cdot)$)

$$\begin{aligned} \dot{\mathbf{v}} = \vec{\mathfrak{h}}(v) &\Leftrightarrow \begin{cases} \tilde{\mathcal{G}}^{-1}\lambda = \dot{\gamma} & (\text{horizontal part}) \\ \nabla_{\dot{\gamma}}^{[1],*}\lambda = 0 & (\text{vertical part}) \end{cases} \\ &\Leftrightarrow \begin{cases} \dot{\gamma}^i = u^i = \lambda^i := \sum_{j=1}^n g^{ij}\lambda_j & (\text{horizontal part}) \\ \dot{\lambda}_i = \{\mathfrak{h}, \lambda_i\} = -\sum_{j,k=1}^n c_{ij}^k \lambda_k u^j & (\text{vertical part}) \end{cases} \end{aligned}$$

with velocity controls $u^i := \dot{\gamma}^i = \langle \omega^i|_{\gamma(\cdot)}, \dot{\gamma} \rangle$ and $\mathbf{v}(t) = (\gamma(t), \lambda(t))$ a curve in the co-tangent bundle $T^*(G)$ where the geodesics $\gamma(t) \in G$ and the momentum along the geodesic $\lambda(t) \in T_{\gamma(t)}^*(G)$, and with $\{\cdot, \cdot\}$ denoting Poisson bracket as defined in Eq. (113).

This theorem is proven in [38].

In the last theorem, the difference between straight curves and shortest curves is shown. In this theorem, we have an upper bound for the time in $\gamma(t)$, denoted by t_{cut} . This denotes the time one can move along a geodesics until it reaches a point \mathbf{q} which is in the cut locus of \mathbf{p} . This means that there exist two length minimizers joining \mathbf{p} and \mathbf{q} . In other words, there are multiple minimizing geodesics connecting point \mathbf{q} to the end point \mathbf{p} [39] (Definition 8.52). The conjugate time t_{conj} is the distance you can move along a geodesic until you meet a conjugate point. The conjugate points are, roughly, all points \mathbf{p} that can almost be joined by a 1-parameter family of geodesics. These are the points where local optimality is lost. The exact definition can be found in [39], Definition 8.43.

Example C.1. We consider two different manifolds where conjugate points do (not) exist.

- Consider a sphere. The north-pole and the south-pole can be connected by any median, and their distance is always the same.
- Consider the plane \mathbb{R}^2 . There are no conjugate points; there are not multiple geodesics connecting two points.

The variable t_{cut} denotes how far you can move along a geodesic until the Riemannian exponential curves loose their optimality.

Theorem 4 (Theorem 1 from [38]). *In a Riemannian manifold $(G, T(G), \mathcal{G})$, with the tangent bundle $T(G)$ and metric tensor field \mathcal{G} defined in Eq. (108) and induced metric $d_{\mathcal{G}}$ defined in Eq. (9), and the Lie-Cartan connection $\nabla^{[\nu]}$ for $\nu = 1$ defined in Eq. (112), we have the following relations for “straight” curves:*

$$\begin{aligned} \gamma \text{ is a } \nabla^{[1]} \text{ - straight curve} &\Leftrightarrow \gamma \text{ is an exponential curve} \\ \Leftrightarrow \nabla_{\dot{\gamma}}^{[1]}\dot{\gamma} = 0 &\Leftrightarrow \gamma \text{ has } \nabla^{[1]} \text{ - auto parallel velocity,} \end{aligned}$$

and the following for “shortest” curves (minimizers in Eq. (9)), recall Eq. (109):

$$\gamma \text{ is a shortest curve} \Leftrightarrow \gamma \text{ is a minimizing curve in } d_{\mathcal{G}} \quad (115)$$

$$\Rightarrow \begin{cases} \nabla_{\dot{\gamma}}^{[1],*}\lambda = 0 \\ \dot{\gamma} = \tilde{\mathcal{G}}^{-1}\lambda \end{cases} \Leftrightarrow \gamma \text{ has } \nabla^{[1],*} \text{ - parallel momentum.} \quad (116)$$

For a minimizing curve between $g_1 = \gamma(0)$ and $g_2 = \gamma(t)$ one must have $0 \leq t \leq t_{\text{cut}} = \min \{t_{\text{conj}}(\lambda(0)), t_{\text{max},1}(\lambda(0))\}$, cf. [40; 41] for details. They are found by steepest descent:

$$\gamma(t) = \gamma(0) + \int_0^t \text{grad}_{\mathcal{G}} W(\gamma(s)) ds, \quad (117)$$

on distance maps $W(g) = d_{\mathcal{G}}(g, e)$ that are viscosity solutions of the eikonal PDE

$$\begin{cases} \|\text{grad}_{\mathcal{G}} W(g)\| = \sqrt{\mathcal{G}_g(\text{grad}_{\mathcal{G}} W(g), \text{grad}_{\mathcal{G}} W(g))} = 1, \\ W(e) = 0, \end{cases} \quad (118)$$

with (metric-intrinsic) gradient $\text{grad}_{\mathcal{G}} W(g) = \tilde{\mathcal{G}} dW(g)$, as this only gives global minimizing curves.

C.2 Gauge Frame

We just discussed the difference between shortest curve and straight curve with the left invariant frame as reference framework. Jelle Wemmenhove has worked on a the same theorem, but with the gauge frame as the reference framework during his master thesis [13]. Here, we state the corresponding theorem (including the corresponding notation).

Definition C.4. Given a gauge frame $\{\mathcal{B}_i\}_{i=1}^d$ with dual frame $\{\chi^i\}_{i=1}^d$, we propose the family of connections $\widehat{\nabla}^{[\nu]}$ with $\nu \in \mathbb{R}$ on $T(M)$ given by

$$\widehat{\nabla}_X^{[\nu]} Y := \sum_{k=1}^d X(y^k) \mathcal{B}_k + \sum_{i,j=1}^d x^i y^j \nu [\mathcal{B}_i, \mathcal{B}_j], \quad (119)$$

for vector fields $X = \sum_{j=1}^d x^j \mathcal{B}_j$ and $Y = \sum_{j=1}^d y^j \mathcal{B}_j$.

Corollary C.1. The dual $\widehat{\nabla}^*$ of the connection given in Eq. (119) is given by

$$\widehat{\nabla}_X^{[\nu],*} \lambda = \sum_{i=1}^d \left(X(\lambda_i) + \sum_{j,k=1}^d x^j \lambda_k \nu C_{ij}^k \right) \chi^i, \quad (120)$$

for $X = \sum_{j=1}^d x^j \mathcal{B}_j$, $\lambda = \sum_{j=1}^d \lambda_j \chi^j$ and where C_{ij}^k is defined by $\sum_{k=1}^d C_{ij}^k \mathcal{B}_k = [\mathcal{B}_i, \mathcal{B}_j]$.

The proof of the corollary can be found in [13].

Theorem 5 (Theorem 20 from [13]). *If γ is a shortest curve, i.e. it locally minimizes $d_{\mathcal{G}}$, then γ is the horizontal part of a curve $v = (\gamma, p)$ in $TM \times T^*(M)$ which has momentum $p = \tilde{\mathcal{G}}(\dot{\gamma})$ that is parallel w.r.t. the dual connection $\widehat{\nabla}^{[1],*}$:*

$$\widehat{\nabla}_{\dot{\gamma}}^{[1],*} p = 0.$$

Furthermore, γ is a straight curve w.r.t. the connection $\widehat{\nabla}^{[1]}$ if and only if its velocity $\dot{\gamma}$ is a constant linear combination of the basis $\{\mathcal{B}_i\}_{i=1}^n$:

$$\widehat{\nabla}_{\dot{\gamma}}^{[1]} \dot{\gamma} = 0 \quad \text{iff} \quad \dot{\gamma} = \sum_{i=1}^n u^i \mathcal{B}_i, \text{ where } u^i \in \mathbb{R}.$$

The proof for this theorem can be found in [13].

References

- [1] Donnell J. Creel. Clinical Electrophysiology, July 2007. Accessed: 2021-05-03.
- [2] Henry J. Kaplan. Retina Image Bank; Severe NPDR. <https://imagebank.asrs.org/file/5343/severe-npdr>, March 2013. Accessed: 2021-05-03.
- [3] P. Vishnu Priya, A. Srinivasa Rao, and J.V.C. Sharma. Diabetic Retinopathy - Can Lead to Complete Blindness. *International Journal of Science Inventions Today*, 2:254–265, January 2013.
- [4] 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. *Diabetologia*, 54:2409–2416, September 2011.
- [5] Erik J. Bekkers. *Retinal Image Analysis using Sub-Riemannian Geometry in SE(2)*. PhD thesis, Eindhoven University of Technology, 2017.
- [6] Erik J. Bekkers, Remco Duits, Alexey Mashtakov, and Gonzalo R. Sanguinetti. A PDE Approach to Data-driven Sub-Riemannian Geodesics in SE(2). *SIAM Journal on Imaging Sciences*, 8(4):2740–2770, 2015.
- [7] Jean-Marie Mirebeau. Fast-Marching Methods for Curvature Penalized Shortest Paths. *Journal of Mathematical Imaging and Vision*, 60:784–815, July 2018.
- [8] B. Franceschiello, A. Mashtakov, G. Citti, and A. Sarti. Geometrical Optical Illusion via Sub-Riemannian Geodesics in the Roto-Translation Group. *Differential Geometry and its Applications*, 65:55–77, 08 2019.
- [9] G. Citti, B. Franceschiello, G. Sanguinetti, and A. Sarti. Sub-Riemannian Mean Curvature Flow for Image Processing. *SIAM Journal on Imaging Sciences*, 9:212–237, 2016.
- [10] Ugo Boscain, Roman Chertovskih, Jean-Paul Gauthier, Dario Prandi, and Alexey Remizov. Cortical-Inspired Image Reconstruction via Sub-Riemannian Geometry and Hypoelliptic Diffusion. *ESAIM: Proceedings and Surveys*, 64:37–53, 2018.
- [11] Erik M. Franken. *Enhancement of Crossing Elongated Structures in Images*. PhD thesis, Eindhoven University of Technology, 2008.
- [12] Erik M. Franken and Remco Duits. Crossing-Preserving Coherence-Enhancing Diffusion on Invertible Orientation Scores. *International Journal of Computer Vision*, 85:1573–1405, February 2009.
- [13] A.J. Wemmenhove. Improved Vessel Tracking and New Data-Driven Connections on SE(2). Master’s thesis, Eindhoven University of Technology, 2020.
- [14] Remco Duits and Erik M. Franken. Left-Invariant Parabolic Evolutions on SE(2) and Contour Enhancement via Invertible Orientation Scores Part II: Nonlinear Left-Invariant Diffusions on Invertible Orientation Scores. *Quarterly of Applied Mathematics*, 68:293–331, February 2010.
- [15] Remco Duits. *Perceptual Organization in Image Analysis*. PhD thesis, Eindhoven University of Technology, 2005.
- [16] R. Duits, M.H.J Janssen, J. Hannink, and G.R. Sanguinetti. Locally Adaptive Frames in the Roto-Translation Group and Their Applications in Medical Imaging. *Journal of Mathematical Imaging and Vision*, 56:367–402, November 2016.
- [17] Remco Duits, Stephan P. L. Meesters, Jean-Marie Mirebeau, and Jorg M. Portegies. Optimal Paths for Variants of the 2D and 3D Reeds-Shepp Car with Applications in Image Analysis. *Journal of Mathematical Imaging and Vision*, 2017.
- [18] Remco Duits, Maurice Duits, Markus van Almsick, and Bart ter Haar Romeny. Invertible Orientation Scores as an Application of Generalized Wavelet Theory. *Pattern Recognition and Image Analysis*, 17:42–75, March 2007.
- [19] Remco Duits and Erik 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). *Quarterly of Applied Mathematics*, 68:255–292, June 2010.
- [20] Remco Duits and Erik J. Bekkers. Lecture notes Differential Geometry for Image Processing - Part II, November 2020.

- [21] Da Chen. *New Minimal Paths Models for Tubular Structure Extraction and Image Segmentation*. PhD thesis, University Paris-Dauphine, 2016.
- [22] Remco Duits, Erik J. Bekkers, and Jorg M. Portegies. Lecture Notes Differential Geometry for Image Processing - Part III: Tracking in Orientation Scores Optimal Paths for Variants of the 2D and 3D Reeds-Shepp Car, November 2020.
- [23] Remco Duits and Erik J. Bekkers. Lecture Notes Differential Geometry for Image Processing - Part I, November 2020.
- [24] Luc Florack. Course Notes Tensor Calculus and Differential Geometry (2WAH0), March 2020.
- [25] Remco Duits and Bernhard Burgeth. Scale Spaces on Lie Groups. In Fiorella Sgallari, Almerico Murli, and Nikos Paragios, editors, *Scale Space and Variational Methods in Computer Vision*, pages 300–312, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [26] Jean-Marie 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, n.d. Accessed: 2021-03-17.
- [27] Jean-Marie Mirebeau. Riemannian fast-marching on cartesian grids using Voronoi’s first reduction of quadratic forms. hal-01507334v4, April 2017.
- [28] Jean-Marie Mirebeau and Jorg M. Portegies. Hamiltonian Fast Marching: A Numerical Solver for Anisotropic and Non-Holonomic Eikonal PDEs. *Image Processing On Line*, 9:47–93, 2019. <https://doi.org/10.5201/ipo1.2019.227>.
- [29] Jean-Marie Mirebeau. Efficient Fast Marching with Finsler Metrics, 2013.
- [30] Jean-Marie Mirebeau, Lionel Gayraud, Rémi Barrère, Da Chen, and Francois DESQUILBET. Massively Parallel Computation of Globally Optimal Shortest Paths with Curvature Penalization. Working paper or preprint, March 2021.
- [31] Remco Duits, Ugo Boscain, Francesco Rossi, and Yuri Sachkov. Association Fields via Cuspless Sub-Riemannian Geodesics in SE(2). *Journal of Mathematical Imaging and Vision*, 49:384–417, 06 2014.
- [32] Jiong Zhang, Behdad Dashtbozorg, Erik J. Bekkers, Josien P.W. Pluim, Remco Duits, and Bart M. ter Haar Romeny. Robust Retinal Vessel Segmentation via Locally Adaptive Derivative Frames in Orientation Scores. *IEEE Transactions on medical imaging*, 35(12):2631–2644, 2016.
- [33] Huazhu Fu, Yanwu Xu, Stephen Lin, Damon Wing Kee Wong, and Jiang Liu. Deepvessel: Retinal Vessel Segmentation via Deep Learning and Conditional Random Field. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 132–139. Springer, 2016.
- [34] Julius Hannink, Remco Duits, and Erik J. Bekkers. Crossing-Preserving Multi-scale Vesselness. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, pages 603–610, Cham, 2014. Springer International Publishing.
- [35] Erik J. Bekkers, Da Chen, and Jorg M. Portegies. Nilpotent Approximations of Sub-Riemannian Distances for Fast Perceptual Grouping of Blood Vessels in 2D and 3D. *Journal of Mathematical Imaging and Vision*, 60:882–899, 2018.
- [36] Laurent Cohen. Multiple Contour Finding and Perceptual Grouping using Minimal Paths. *Journal of Mathematical Imaging and Vision*, 14:225–236, 05 2001.
- [37] Ro Frangi, W.J. Niessen, Koen Vincken, and Max Viergever. Multiscale Vessel Enhancement Filtering. *Medical Image Computing and Computer-Assisted Intervention*, 1496:130–137, 02 2000.
- [38] R. Duits, B.M.N. Smets, A.J Wemmenhove, J.W. Portegies, and E.J. Bekkers. Geometric Flows in Multi-Orientation Image Processing via a Cartan Connection, 2020.
- [39] Andrei A. Agrachev, Davide Barilari, and Ugo Boscain. Introduction to Riemannian and Sub-Riemannian Geometry. <https://webusers.imj-prg.fr/~daveide.barilari/2016-11-21-ABB.pdf>, 2016.
- [40] Andrei A. Agrachev and Yuri L. Sachkov. *Control Theory from the Geometrical Viewpoint*, volume 87. Springer-Verlag, 2004.
- [41] Yuri L. Sachkov. Cut Locus and Optimal Synthesis in the Sub-Riemannian Problem on the Group of Motions of a Plane. *ESAIM: COCV*, 17(2):293–321, 2011.