Crossing-Preserving Coherence-Enhancing Diffusion on Invertible Orientation Scores

Erik Franken, Remco Duits

March 13, 2008

Abstract

Many medical image processing problems require enhancement of crossing elongated structures. This problem can not easily be solved by commonly used coherence-enhancing diffusion methods. Therefore we propose a method for coherence-enhancing diffusion in the invertible orientation score of an image. In an orientation score, the local orientation is made an explicit dimension, ensuring that crossing curves are separated from each other in this extra dimension. We consider the group structure of the orientation score domain, which is the Euclidean motion group $SE(2)$, and impose left-invariant evolution equations yielding the appropriate scale space representations of the orientation score. We describe how we can calculate regularized left-invariant derivatives, and use the Hessian to estimate three descriptive local features: curvature, deviation from horizontality, and orientedness. These local features steer a non-linear coherence-enhancing, crossing-preserving, diffusion equation on the orientation score. We propose two explicit finite difference schemes to apply the non-linear diffusion in the orientation score and provide a stability analysis. The experiments show that preservation of crossing curves is the main advantage compared to standard coherence enhancing diffusion. The use of curvature leads to improved enhancement of curves with high curvature. Furthermore, the use of deviation from horizontality makes it feasible to largely reduce the number of sampled orientations while still preserving crossings.

1 Introduction

Medical image processing problems often demand enhancement of elongated structures, such as ridges, edges, and oriented texture patterns in noisy images. Consider for example blood vessels, catheters, neural fibers, and contours of organs in noisy medical image data.

Many methods for enhancing elongated structures are based on non-linear anisotropic diffusion equations on the image, i.e. non-linear anisotropic scale spaces[1]. This idea was pioneered by Nitzberg et al. [28] and Cottet et al. [7]. Later on, Weickert proposed edge- and coherence-enhancing diffusion filtering [38, 39], which uses the structure tensor to steer the diffusion. Afterwards, various publications appeared inspired by these methods, for example Manniesing et al. [26] who propose to steer the diffusion using the vessel resemblance function.

A lot of (medical) image processing problems require the algorithm to appropriately handle crossing and bifurcating line structures. This is for example important in High Angular Resolution Diffusion Imaging (HARDI) [32], which is an extension to Diffusion Tensor Imaging (DTI) where a richer representation for angular data is used, such that crossing fibers can be distinguished. Other examples include microscopy images of for instance collagen structures, X-ray fluoroscopy images with catheters [20], and bifurcating blood vessels. At a position with a crossing, the anisotropic diffusion equation is not appropriate since one would like to diffuse in the directions of the different orientations, independent on the angles and number of crossing elongated structures. However, this is impossible to model with non-linear anisotropic diffusion equations on the image space. The straightforward, though non-optimal, solution is to inhibit the entire diffusion at crossings.

[1] Note that we do not consider non-linear diffusion of the Perona and Malick type [29] to be anisotropic diffusion, since it uses a scalar diffusivity.

Other authors try to resolve the problem of crossing oriented structures by introducing an object that is usually called *orientation space*, in this paper further on referred to as *orientation score*[2], where the local orientation is made an explicit dimension (so creating a 3D volume out of 2D image). The main advantage is that crossing curves are torn apart, meaning that it is not needed anymore to take special care for crossing e.g. by explicitly detecting them. This concept first occurs in the field of perceptual grouping / biomimicking of the biological visual system [36, 23, 27, 41, 42]. The concept is also applied for segmenting crossing structures [5, 2, 35] and estimating local orientation [16].

Kalitzin [25] was the first to propose *invertible orientation scores*, which means that it is possible to reconstruct the original image from the orientation score in a well-posed way. Duits et al. [11, 8, 9] developed a theory on the robustness of this transformation. Both authors emphasize that this invertibility is essential in order to use orientation scores (OS) for image processing/enhancement, in the following way

$$\text{Transform image to OS} \longrightarrow \text{Process OS} \longrightarrow \text{Transform OS to image.}$$

Kalitzin et al. pioneered with processing the orientation score by taking a certain power or taking non-linear combinations of Gaussian derivatives in the orientation score of an image before transforming the orientation score back to image, leading to some enhancement of lines.

Later, the authors of this paper established a neat mathematical framework for processing orientation scores [9], by considering the orientation score domain as the *Euclidean motion group*, making it possible to use results from the field of harmonic analysis on Lie groups. We studied sophisticated processing operations in the orientation score domain, namely linear convection-diffusion equations on the orientation score domain, corresponding to stochastic processes for contour enhancement [6] and completion [27]. Similar to stochastic completion fields [42] we take a product of a forward and a backward direction process to fill gaps in line structures [9, 34]. The major differences to earlier work in this field is that we use the invertibility of the orientation score and we solve these linear evolution equations by means of a special convolution (the $SE(2)$-*convolution*) on the orientation score with the corresponding Green's function, where we were the first to find the exact solutions to these equations [15, 13, 14].

This paper goes one step further as it describes a *non-linear* diffusion equation on the two-dimensional Euclidean motion group to enhance crossing elongated structures. Rather than a product of a linear forward process and a backward process, we describe how to make the diffusion tensor adaptive to three features that describe the local structure in the orientation score: orientedness, which determines whether we locally diffuse isotropic or anisotropic, and curvature and deviation from horizontality, which establish a gauge frame for the diffusion. The features are estimated using a non-symmetric Hessian matrix. The method is comparable to edge- and coherence-enhancing diffusion, with the difference that the orientation is explicitly encoded in the domain on which diffusion is applied. This paper is a companion paper of [13], written by the same authors. Here we focus in full details on the applied image analysis and algorithmic part of our work, whereas in [13] we focus on the mathematical theory and underlying differential geometry of both linear (with exact solutions) and non-linear left-invariant diffusion equations on orientation scores.

There exist several other methods that are designed for preserving or enhancing oriented structures while reducing noise. A well-known alternative approach emerges from the wavelet literature: Candès et al. proposed the curvelet transform [4, 3, 31]. This is a multi-scale and multi-angle generalization of the wavelet transform, where thresholding of coefficients in the curvelet domain is performed in order to achieve noise reduction while preserving (crossing) oriented structures. Our work has two major differences: rather than applying soft thresholds on the curvelet domain we employ the group structure in the wavelet domain and impose left-invariant evolution equations in the wavelet domain, yielding the appropriate *scale space* representations in the wavelet domain. Furthermore, our orientation score domain is a wavelet domain without scaling. Besides

[2]In this paper we would like to make a distinction between the function and the domain of a function. So the domain of an orientation score could be called an orientation space.

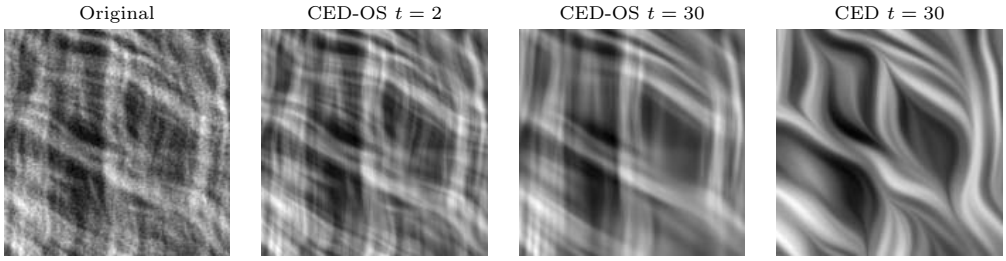


Figure 1: Example of coherence enhancing diffusion on orientation scores. The image is constructed from two rotated 2-photon images of collagen tissue in the heart. At $t = 2$ our method achieves a good enhancement of the image. Comparing the result at $t = 30$ with standard coherence enhancing diffusion, we can clearly see the superior performance for crossing structures.

a reduction of storage requirements, an important reason to leave out scaling in our orientation score transform is that admissibility conditions for wavelet transforms including scaling require the oriented wavelet to oscillate along its radial direction, which is not desirable for our diffusion equations later on. Note that we can still obtain a stable image reconstruction from a fixed scale orientation score, whereas in standard wavelet theoretical approaches it is not possible to obtain a stable image reconstruction from a fixed scale layer in the wavelet domain.

Another interesting approach for handling crossing curves is proposed Scharr [30], where instead of the gradient, the second order jet operator is used to both calculate a structure tensor and for the gradient and divergence of a diffusion-like PDE. However, the drawback of this method is that the order of the PDE high, and gets even higher if one wants to deal with crossings of more than two curves. Furthermore, it is only suitable to handle "X-junctions" (2 crossing curves with a large angle). Our approach is more general, in the sense that the number of sampled orientations n_θ determines the amount of crossing curves we can deal with (upper bounded by $n_\theta/2$) and the minimum angle between them.

1.1 Structure of the paper

This paper starts with introducing orientation scores in details, providing and the necessary theory: group structure, left invariance, the tangent space at group elements, the left-invariant diffusion equation, and the gauge frame. Then we describe how to construct invertible orientation scores in practice, and how to operationalize regularized derivatives in orientation scores. All these results are used for the next step: estimating local features in the orientation scores. These features are used next to come to a non-linear diffusion model, which corresponds to edge/coherence-enhancing diffusion, but then in the orientation score domain. Then we describe the numerical schemes that are used, including an analysis on stability. Finally we end with experimental results on the quality of curvature estimation and results of the coherence-enhancing diffusion in the orientation score.

The paper is extended work of our conference publications [18] and [19] with new results, particularly the gauge frame, stability bounds of the numerical schemes, and the concept of deviation from horizontality.

2 Theory of Orientation Scores

In this section we will describe the theory that is essential for the rest of the paper. We will start with introducing invertible orientation scores in more detail. Then we will explain the essential parts of the group theory and differential geometry that we will need to describe the algorithm. The theory is written in such a way that it should be understandable without too much prerequisites from these fields. For more theoretical underpinning and mathematical details we refer to our other publications [15, 11].

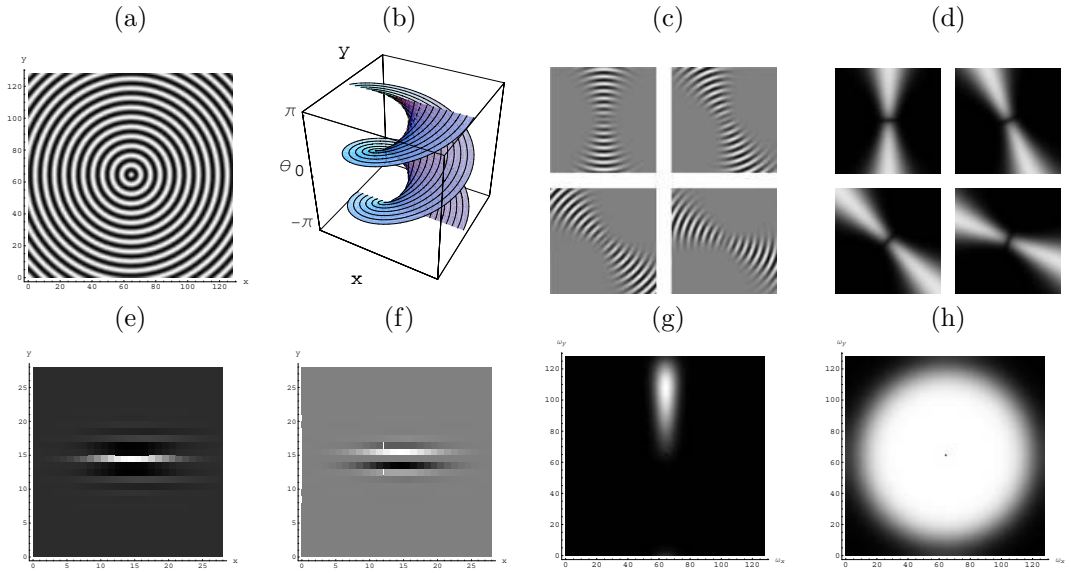


Figure 2: (a) Example of an image with concentric circles. (b) The structure of the corresponding orientation score. The circles become spirals and all spirals together span a helicoid-shaped plane. (c) Real part of the orientation score U_f displayed for 4 different orientations. (d) The absolute vale $|U_f|$ yields a phase-invariant response displayed for 4 orientations. (e) Real part of the kernel with $\theta = 0$ and parameter values $k = 2$, $q = 8$, $t = 400$, $s = 10$, $n_\theta = 64$. (f) Imaginary part. (g) Fourier transform of the kernel depicted in (e)+(f). (h) M_ψ (equation (5)), which can be seen as the Fourier transform of the net operation if no correction is applied (i.e. if reconstruction equation (10) is used).

2.1 Orientation Scores

In mathematical terms, an image f is a mapping $f : \mathbb{R}^2 \to \mathbb{R}$, which has compact support on the image domain $\Omega = [0, X] \times [0, Y]$, with image dimensions $X, Y \in \mathbb{R}^+$. If one considers oriented structures (e.g., lines, edges, oriented texture patterns) in images, the position in the domain $\mathbb{R}^2$ is not very descriptive, one only knows the position relative to the horizontal and vertical axes. The codomain is not very descriptive either, since a single grayvalue itself does not give any information on orientation. In an *orientation score* (which in most literature is called *orientation space*), we add an additional dimension to the domain, namely orientation, meaning that an orientation score U is defined as a function $\mathbb{R}^2 \times \mathbb{T} \to \mathbb{R}$ or $\mathbb{C}$, where $\mathbb{R}^2$ corresponds to the spatial (image) domain and $\mathbb{T}$ is the orientation domain. As a result, the position in the new domain contains the three essential features to locally describe an oriented structure, namely orientation and horizontal and vertical position, see Figure 2a-b.

Instead of extending the domain one could think of extending the codomain to describe oriented features, e.g. $U : \mathbb{R}^2 \to \mathbb{T} \times \mathbb{R}$. The latter approach is substantially different, since then each spatial position only maps to a single orientation, while in an orientation score each combination of spatial position and orientation maps to a number. The practical advantage is manifest: one can transparently handle crossings and bifurcations.

On the image domain, it is straightforward to develop image processing operations that are *translation invariant* [3], i.e. operations which commute with a translation of the image. One can also create *rotation invariant* operations, meaning that the operator commutes with rotation of the image. However the number of possible rotation invariant operations on images directly is limited to, for example, isotropic filtering or gauge coordinates [17]. With invertible orientation

[3]The word "covariant" would actually be better than "invariant", however we "invariant" since it is more common in literature

scores, however, it is possible to develop operations that are sensitive to oriented structures and at the same time *both* translation *and* rotation invariant, i.e. *Euclidean* invariant.

The observations on invariances lead to an important consideration in our framework: the domains of both images and orientation spaces are *Lie group manifolds*. An image f is a mapping from the group elements of the translation group $\mathbb{R}^2$ to the real numbers. Analogously, an orientation score U is a mapping from the group elements of the *Euclidean motion group* $SE(2) = \mathbb{R}^2 \rtimes \mathbb{T}$. The properties of this group will be treated in the next subsection.

2.2 The Euclidean Motion Group

The Euclidean motion group $SE(2) = \mathbb{R}^2 \rtimes \mathbb{T}$ is parameterized by the group elements $g = (\mathbf{x}, \theta)$ where $\mathbf{x} = (x, y) \in \mathbb{R}^2$ are the two spatial variables that label the domain of the image f, and $\theta \bmod 2\pi$ is the orientation angle that captures the orientation of structures in image f. We will use both short notation g and explicit notation $(\mathbf{x}, \theta)$ for group elements. The group product and group inverse of elements in $SE(2)$ are given by

$$\begin{aligned} &g\,g' = (\mathbf{x}, \theta)\,(\mathbf{x}', \theta') = (\mathbf{x} + \mathbf{R}_\theta \mathbf{x}', \theta + \theta' \bmod 2\,\pi), \\ &g^{-1} = (-\mathbf{R}_\theta^{-1}\mathbf{x}, -\theta). \end{aligned} \tag{1}$$

The Euclidean motion group is not commutative, i.e. in general $g\,g' \neq g'\,g$. Note that the translation and rotation part are not independent of each other as a rotation matrix $\mathbf{R}_\theta$ appears in the translation part. In the notation $\mathbb{R}^2 \rtimes \mathbb{T}$ this is reflected by the symbol "$\rtimes$" for the *semi*-direct product, rather than the symbol $\times$ that denotes the direct product.

To map the structure of the group to orientation scores and images, we need a so-called representation. A *representation* is a homomorphism of the form $\mathcal{R} : G \to \mathcal{B}(H)$ where H is a Hilbert space and $\mathcal{B}(H)$ is the space of bounded linear operators $\mathcal{A} : H \to H$. A representation maps a group element to an operator, i.e. $\mathcal{R} = (g \mapsto \mathcal{R}_g)$, such that $e \mapsto \mathcal{I}$ (identity element maps to identity operator), $gh \mapsto \mathcal{R}_g \mathcal{R}_h$ (group product is preserved), and consequently $g^{-1} \mapsto (\mathcal{R}_g)^{-1}$ (inverse is preserved).

Two group representations of $SE(2)$ are important in this work: on images $\mathbb{L}_2(\mathbb{R}^2)$ and orientation spaces $\mathbb{L}_2(SE(2))$. They are defined by

$$\begin{aligned} &(\mathcal{U}_g f)(\mathbf{y}) = f(\mathbf{R}_\theta^{-1}(\mathbf{y} - \mathbf{x})), \quad f \in \mathbb{L}_2(\mathbb{R}^2), \quad g = (\mathbf{x}, \theta) \in SE(2), \quad \mathbf{y} \in \mathbb{R}^2, \\ &(\mathcal{L}_g U)(h) = U(g^{-1}h), \quad U \in \mathbb{L}_2(SE(2)), \quad g, h \in SE(2). \end{aligned} \tag{2}$$

The representation $\mathcal{L}_g$ is the *left-regular* representation, since the multiplication takes place on the left side.

2.3 Invertible Orientation Scores

An invertible orientation score is obtained by correlating the image with an anisotropic kernel[4]

$$U_f(\mathbf{x}, \theta) = (\mathcal{W}_\psi f)(\mathbf{x}, \theta) = (\overline{\psi^\theta} \star f)(\mathbf{x}) = \int_{\mathbb{R}^2} \overline{\psi(\mathbf{R}_\theta^{-1}(\mathbf{x}' - \mathbf{x}))} f(\mathbf{x}') d\mathbf{x}', \tag{3}$$

where $\psi \in \mathbb{L}_2(\mathbb{R}^2)$ is the correlation kernel with orientation $\theta = 0$ (i.e. aligned with the horizontal axis in our convention), which is related to the convolution kernel by $\breve{\psi}(\mathbf{x}) = \psi(-\mathbf{x})$. The overline denotes complex conjugate, and $\psi^\theta(\mathbf{x}) = \psi(\mathbf{R}_\theta^{-1}\mathbf{x})$ where $\mathbf{R}_\theta$ is the rotation matrix $\mathbf{R}_\theta = \left(\begin{smallmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{smallmatrix}\right)$. The accompanying exact reconstruction formula is given by

$$\hat{f}\,(\mathbf{x}) = \mathcal{F}^{-1}\left[M_\psi^{-1}\mathcal{F}\left[\mathbf{x} \mapsto \int_0^{2\pi} \left(\breve{\psi}^\theta \star U_f(\cdot, \theta)\right)(\mathbf{x})\, d\theta\right]\right](\mathbf{x}), \tag{4}$$

[4]We use correlations instead of convolutions since this is consistent with earlier work by Duits [9] on invertible orientation scores. Furthermore this is common in wavelet literature.

where $M_\psi : \mathbb{R}^2 \to \mathbb{R}^+$ is calculated by

$$M_\psi(\boldsymbol{\omega}) = \int_0^{2\pi} \mathcal{F}\left[\overline{\psi^\theta}\right] \mathcal{F}\left[\psi^\theta\right] d\theta = \int_0^{2\pi} \left|\mathcal{F}\left[\psi^\theta\right]\right|^2 d\theta \,. \tag{5}$$

This function can be seen as a measure for stability of the inverse transformation: the number $M_\psi(\boldsymbol{\omega})$ specifies how well frequency component $\boldsymbol{\omega}$ is preserved by the cascade of construction and reconstruction, if the "compensation term" M_ψ^{-1} would not be included in the reconstruction equation (4). It can be verified that the construction/reconstruction equations (3) and (4) fulfill the following Plancherel's formula

$$\|f\|^2_{\mathbb{L}_2(\mathbb{R}^2)} = \|\mathcal{W}_\psi f\|^2_{M_\psi}, \tag{6}$$

where the norm $\|\cdot\|_{M_\psi}$ on the orientation score domain is defined as

$$\|\mathcal{W}_\psi f\|^2_{M_\psi} = \int_{\mathbb{R}^2} \int_0^{2\pi} |(\mathcal{F}\mathcal{W}_\psi f)(\boldsymbol{\omega}, \theta)|^2 d\theta \frac{1}{M_\psi(\boldsymbol{\omega})} d\boldsymbol{\omega}, \tag{7}$$

where $\mathcal{F}$ denotes the Fourier transform on the spatial coordinates only. Note that we have $\mathbb{L}_2$-norm preservation, i.e. $\|f\|^2_{\mathbb{L}_2(\mathbb{R}^2)} = \|\mathcal{W}_\psi f\|^2_{M_\psi} = \|\mathcal{W}_\psi f\|^2_{\mathbb{L}_2(SE(2))}$, if and only if $M_\Psi = 1$.

Theoretically, reconstruction is well-posed as long as $0 < \delta < M_\psi(\boldsymbol{\omega}) < \infty$ where δ is arbitrarily small. In practice, to prevent numerical problems, it is better to aim at $M_\psi(\boldsymbol{\omega}) \approx 1$ for $\|\boldsymbol{\omega}\| < \varrho$, meaning that all frequency components within a ball of radius ϱ are preserved. The latter is a natural choice for bandlimited images: because of finite sampling it is reasonable to assume images to be bandlimited anyway, where the bandwidth coincides with the well-known Nyquist frequency. If $M_\psi(\boldsymbol{\omega}) \approx 1$ we can approximate the reconstruction by

$$\hat{f}(\mathbf{x}) \approx \int_0^{2\pi} \left(\breve{\psi}^\theta \star U_f(\cdot, \theta)\right)(\mathbf{x})\, d\theta. \tag{8}$$

We can even further simplify the reconstruction for a special class of filters ψ that satisfy

$$M_\psi(\boldsymbol{\omega}) = \int_0^{2\pi} \left|\mathcal{F}\left[\psi^\theta\right](\boldsymbol{\omega})\right|^2 d\theta \approx \int_0^{2\pi} \left|\mathcal{F}\left[\psi^\theta\right](\boldsymbol{\omega})\right| d\theta \,. \tag{9}$$

where the reconstruction formula simplifies to integration over the orientation dimension

$$\hat{f}(\mathbf{x}) \approx \int_0^{2\pi} U_f(\mathbf{x}, \theta)\, d\theta. \tag{10}$$

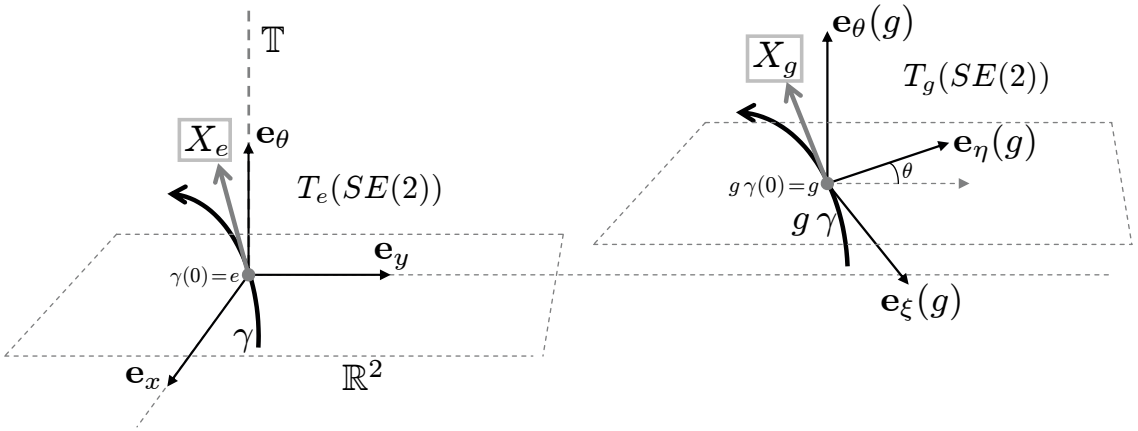
Notice that a kernel ψ with (nearly) perfect reconstruction using equation (8) can be transformed into a kernel $\tilde{\psi}$ with a perfect reconstruction using (10) as follows: $\tilde{\psi} = \left(\overline{\psi} * \psi\right)(\mathbf{x})$. However, this does not always lead to a useful kernel $\tilde{\psi}$.

Invertible orientation scores were first proposed by Kalitzin et al. [24], where he proposes a specific choice for an oriented wavelet which falls in class of proper wavelets [11]. This kernel, however, has practical disadvantages, which are explained in detail in [9, p141, subsection 4.6.2]. The oriented wavelets we use in this paper are also proper wavelets, but are of an essentially different type [9, p141, subsection 4.6.1]. A more mathematical treatment of invertible orientation scores can be found in the work by Duits et al. [11] who developed a generalization of the wavelet theory. He also first proposed the so-called "cake kernels", a more practical class of kernels with the advantage that reconstruction equation (10) can be used. We introduce this type of kernel in more detail in Section 3.

2.4 Left-invariant Operations in Orientation Scores

We want to perform image processing operations on orientation scores. Analogously to the fact that the Gabor transform of a signal makes it easier to perform operations that "manipulate"

(a) Left-invariance of tangent vectors to curves:



(b) Left-invariance of tangent vectors considered as differential operators:

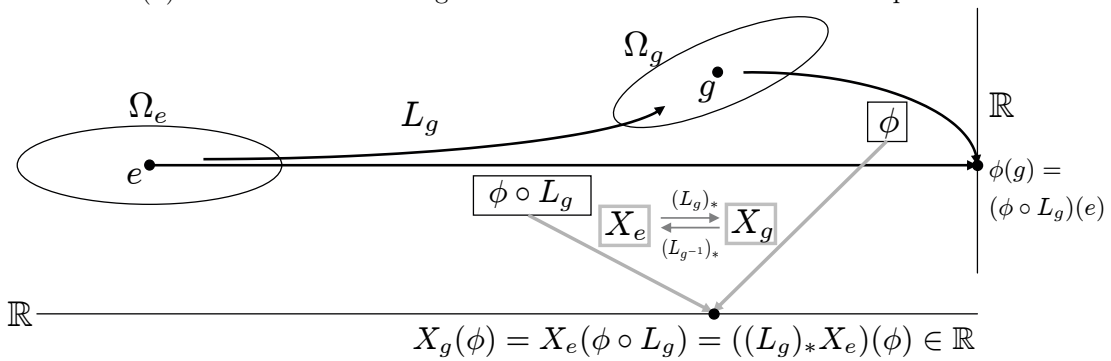


Figure 3: Illustrations of the concept of left-invariance, from two different perspectives: (a) considered as tangent vectors tangent to curves, i.e. $X_g = c^\theta \mathbf{e}_\theta(g) + c^\xi \mathbf{e}_\xi(g) + c^\eta \mathbf{e}_\eta(g)$ for all $g \in SE(2)$, and (b) considered as differential operators on locally defined smooth functions, i.e. $X_g = c^\theta \partial_\theta\big|_g + c^\xi \partial_\xi\big|_g + c^\eta \partial_\eta\big|_g$ for all $g \in SE(2)$. The push forward $(L_g)_* : T_e(SE(2)) \to T_g(SE(2))$ connects the tangent space at the unity element $T_e(SE(2))$ to all tangent spaces $T_g(SE(2))$. See text (Section 2.4) for details.

local frequencies, the orientation score transform makes it easier to apply anisotropic operations on locally oriented structures, in such a way that each orientation on each position can be "manipulated" separately.

A lot of choices for operations on orientation score exist, but not all of them are sensible. As already mentioned in Section 2.1, we want to ensure that the net operation on the image is Euclidean invariant. It can be shown [9, Theorem 2.1, p.153] that this is only the case if the operator operator $\Upsilon : \mathbb{L}_2(SE(2)) \to \mathbb{L}_2(SE(2))$ that is applied to the orientation score is *left-invariant*. Operator Υ is left-invariant iff $\mathcal{L}_g \Upsilon U = \Upsilon \mathcal{L}_g U$, for all $g \in G$ and for all $U \in \mathbb{L}_2(SE(2))$, where $\mathcal{L}_g$ is defined in (2). In other words, left-invariance means that the operator commutes with the left-regular representation $\mathcal{L}$ of the group.

Figure 3 illustrates left-invariance in two different ways: for tangent vectors and for differential operators. In Figure 3a, it is shown how a tangent vector $X_e = c^\theta \mathbf{e}_\theta + c^\xi \mathbf{e}_x + c^\eta \mathbf{e}_y \in T_e(SE(2))$ which is tangent to a curve $\gamma : \mathbb{R} \to SE(2)$ at unity element e can be "transported in a left-invariant way" to a tangent vector $X_g \in T_g(SE(2))$ which is tangent to the curve $g\gamma$ (i.e., the original curve γ is left-multiplied with $g = (\mathbf{x}, \theta) \in SE(2)$ so that the curve is translated over $\mathbf{x}$ and rotated over θ) at position g. The tangent space at the unity element is spanned by $T_e(G) = \text{span}\{\mathbf{e}_\theta, \mathbf{e}_x, \mathbf{e}_y\}$. By transporting this basis vectors in a left-invariant way (i.e., $X_g = (L_g)_* X_e$ is the *push-forward* of left-multiplication, as illustrated in the figure) we get the following basis for the left-invariant vector fields at group element g

$$\text{span}\{\mathbf{e}_\xi(g), \mathbf{e}_\eta(g), \mathbf{e}_\theta(g)\} = \text{span}\{\cos\theta\, \mathbf{e}_x + \sin\theta\, \mathbf{e}_y, -\sin\theta\, \mathbf{e}_x + \cos\theta\, \mathbf{e}_y, \mathbf{e}_\theta\} = T_g(SE(2)). \quad (11)$$

This basis for tangent vectors in g has the property that $X_g = c^\theta \mathbf{e}_\theta + c^\xi \mathbf{e}_\xi + c^\eta \mathbf{e}_\eta$ for all g, so the vector components $(c^\theta, c^\xi, c^\eta)$ of X_e and X_g for all g are the same. Furthermore, the basis vectors have a clear interpretation: $\mathbf{e}_\xi$ is always tangent to the orientation θ and $\mathbf{e}_\eta$ is always orthogonal to this orientation. For notational simplicity the dependency on g is usually omitted further on, but it is important to realize that $\mathbf{e}_\xi$ and $\mathbf{e}_\eta$ do depend on θ of the group element $g = (\mathbf{x}, \theta)$.

In Figure 3b it is shown how X_e and X_g can also be viewed as differential operators, acting on a function $\phi : SE(2) \to \mathbb{R}$ (e.g. an orientation score). In the figure, the codomain of ϕ is the vertical $\mathbb{R}$-axis. X_g can be viewed as an operator that calculates the derivative of ϕ at g, i.e.

$$X_g(\phi) = (c^\theta \partial_\theta\big|_g + c^\xi \partial_\xi\big|_g + c^\eta \partial_\eta\big|_g)\phi = \Big(c^\theta \partial_\theta + c^\xi(\cos\theta \partial_x + \sin\theta \partial_y) + c^\eta(-\sin\theta \partial_x + \cos\theta \partial_y)\Big)\,\phi \quad (12)$$

gives a scalar on the horizontal $\mathbb{R}$-axis on the bottom of the figure. The same result can also be obtained by first translating and rotating ϕ over g, i.e. $\phi \circ L_g$, such that the original neighborhood Ω_g is shifted to neighborhood Ω_e. So

$$X_g(\phi) = X_e(\phi \circ L_g) = (c^\theta \partial_\theta + c^\xi \partial_x + c^\eta \partial_y)(\phi \circ L_g). \quad (13)$$

So the direct relation between X_e and X_g is established by the push-forward operator: $X_g = (L_g)_* X_e$. Intuitively, the push-forward operator allows to move tangent vectors to tangent spaces at different group elements.

An important set of left-invariant operations are the *linear* left-invariant operations. All linear left-invariant operations that are bounded from $\mathbb{L}_2(SE(2))$ to $\mathbb{L}_\infty(SE(2))$ can be expressed as a $SE(2)$-*convolution* [9, p.113–114] of the orientation score U with a kernel $\Psi \in \mathbb{L}_1(SE(2))$

$$(\Psi \ *_{SE(2)} \ U)(\mathbf{x}, \theta) = \int_{\mathbb{R}^2} \int_0^{2\pi} \Psi(\mathbf{R}_{\theta'}^{-1}(\mathbf{x} - \mathbf{x}'), \theta - \theta') U(\mathbf{x}', \theta')\, d\theta'\, d\mathbf{x}', \quad (14)$$

Note that the $SE(2)$-convolution is a straightforward generalization to $SE(2)$ of the usual convolution on $\mathbb{R}^n$.

2.5 Tangent Spaces and Dual Tangent Spaces

For the subsequent theory in the paper we need to introduce the tangent spaces $T_g(SE(2))$ and dual tangent space $T_g^*(SE(2))$ in more detail. We will define an inner product and norm on

these spaces that we need later on. A vector in the tangent space $T_g(SE(2))$ is denoted in a basis-independent way by $c^\theta \partial_\theta\big|_g + c^\xi \partial_\xi\big|_g + c^\eta \partial_\eta\big|_g \in T_g(SE(2))$, with $c^\theta, c^\xi, c^\eta \in \mathbb{R}$. We will always use this basis further on, therefore we will work with the vector components and use the notation $\mathbf{c} = \left(c^\theta, c^\xi, c^\eta\right)^{\mathrm{T}}$. The physical dimensions of the vector components are $(1, \text{length}, \text{length})$ respectively.

Similarly a *covector* is denoted by $c_\theta d\theta\big|_g + c_\xi d\xi\big|_g + c_\eta d\eta\big|_g \in T_g^*(SE(2))$, where $d\theta$, $d\xi$, and $d\eta$ span the basis of the dual tangent space $T_g^*(SE(2))$. The relation between the tangent space and the dual tangent space is established by the Kronecker product

$$\langle dp\big|_g, \partial_q\big|_g \rangle = \delta_{pq} \quad \text{with } p, q \in \{\theta, \xi, \eta\}, \tag{15}$$

so for example $\langle d\theta, \partial_\xi \rangle = 0$ and $\langle d\theta, \partial_\theta \rangle = 1$. For the basis-dependent covector components we use the notation $\hat{\mathbf{b}} = (b_\theta, b_\xi, b_\eta)$ where the "hat" allows to distinguish between vectors and covectors. The physical dimensions of the vector components are $(1, 1/\text{length}, 1/\text{length})$ respectively.

The Kronecker product on covector components $\hat{\mathbf{c}}$ and vector components $\mathbf{b}$ is defined by

$$\langle \hat{\mathbf{b}}, \mathbf{c} \rangle = b_\theta c^\theta + b_\xi c^\xi + b_\eta c^\eta, \tag{16}$$

where the resulting number is dimensionless.

In $\mathbb{R}^n$, vectors and covectors coincide, since an inner product on vectors is defined as $(\mathbf{c}, \mathbf{b}) = \sum_{i=1}^n \sum_{j=1}^n \delta_{ij} c^i b^j$ where δ_{ij} is the Kronecker delta function ($\delta_{ij} = 1$ if $i = j$ and 0 otherwise). In the case of $T_g(SE(2))$ it would be wrong to use the same inner product, since the components c^θ, c^ξ, and c^η are dimensionally not the same. To correct for this we introduce a parameter β with physical dimension $1/\text{length}$ and define as inner product

$$(\mathbf{c}, \mathbf{b})_\beta = c^\theta b^\theta + \beta^2 c^\xi b^\xi + \beta^2 c^\eta b^\eta, \tag{17}$$

The parameter β now ensures the result is dimensionless. From this inner product we can calculate the Grammian matrix

$$\mathbf{G}_\beta = \begin{pmatrix} (\partial_\theta, \partial_\theta)_\beta & (\partial_\theta, \partial_\xi)_\beta & (\partial_\theta, \partial_\eta)_\beta \\ (\partial_\xi, \partial_\theta)_\beta & (\partial_\xi, \partial_\xi)_\beta & (\partial_\xi, \partial_\eta)_\beta \\ (\partial_\eta, \partial_\theta)_\beta & (\partial_\eta, \partial_\xi)_\beta & (\partial_\eta, \partial_\eta)_\beta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \beta^2 & 0 \\ 0 & 0 & \beta^2 \end{pmatrix}. \tag{18}$$

The Grammian matrix establishes the relation between the components of vectors and covectors by $\hat{\mathbf{c}} = \mathbf{G}_\beta \mathbf{c}$ and thus also between the inner product and Kronecker product

$$(\mathbf{c}, \mathbf{b})_\beta = \langle \mathbf{G}_\beta \mathbf{c}, \mathbf{b} \rangle = \langle \hat{\mathbf{c}}, \mathbf{b} \rangle. \tag{19}$$

Consequently, the inner product between two covectors is given by

$$(\hat{\mathbf{c}}, \hat{\mathbf{b}})_\beta = \langle \hat{\mathbf{c}}, \mathbf{G}_\beta^{-1} \hat{\mathbf{b}} \rangle = c_\theta b_\theta + \beta^{-2} c_\xi b_\xi + \beta^{-2} c_\eta b_\eta. \tag{20}$$

From the inner product on $T_g(SE(2))$ we can now induce a norm on vectors and covectors in the regular way, i.e. by

$$\|\mathbf{c}\|_\beta = \sqrt{(\mathbf{c}, \mathbf{c})_\beta} \quad \text{and} \quad \|\hat{\mathbf{c}}\|_\beta = \sqrt{(\hat{\mathbf{c}}, \hat{\mathbf{c}})_\beta}. \tag{21}$$

For a theoretical motivation of this left-invariant (and not right-invariant) inner product on each tangent space $T_g(SE(2))$ see [13].

2.6 Left-invariant Derivatives

As mentioned in Section 2.4, $\{\partial_\theta, \partial_\xi, \partial_\eta\}$ are left-invariant differential operators, and are therefore appropriate to use instead of the set $\{\partial_\theta, \partial_x, \partial_y\}$. They also have a clear interpretation, since ∂_ξ is always the spatial derivative tangent to the orientation θ and ∂_η is always orthogonal.

When constructing higher order left-invariant derivatives, it is important to note that the order of applying the derivatives matters, i.e. not all the left-invariant derivatives $\{\partial_\xi, \partial_\eta, \partial_\theta\}$ commute. The nonzero commutators (definition $[A, B] = AB - BA$) are given by

$$[\partial_\theta, \partial_\xi] = \partial_\eta, \qquad [\partial_\theta, \partial_\eta] = -\partial_\xi. \tag{22}$$

An important elementary left-invariant derivative operations is the gradient of an orientation score, which is given by

$$dU = \frac{\partial U}{\partial \theta} d\theta + \frac{\partial U}{\partial \xi} d\xi + \frac{\partial U}{\partial \eta} d\eta. \tag{23}$$

Note that this is a covector field, where the components are obtained by the nabla operator

$$\nabla U = \left(\frac{\partial U}{\partial \theta}, \frac{\partial U}{\partial \xi}, \frac{\partial U}{\partial \eta} \right)^{\mathrm{T}}. \tag{24}$$

2.7 Horizontality

A curve $q : \mathbb{R} \to SE(2)$ in the orientation score, denoted by its components as $q(t) = (x(t), y(t), \theta(t))$ is *horizontal* iff

$$\theta(t) = \angle(\frac{d}{dt}x(t), \frac{d}{dt}y(t)), \quad \forall\, t \in \mathbb{R}. \tag{25}$$

where $\angle(x, y) = \arg(x + i\, y)$. In words, horizontal curves have the property that the direction of the curve $\mathbb{P}_{\mathbb{R}^2} q$ (i.e. the curve projected to the spatial plane $\mathbb{R}^2$) coincides with the orientation θ of the curve in $SE(2)$. Therefore all tangent vectors over the curve do not contain an $\mathbf{e}_\eta$ component, i.e. an equivalent formulation for horizontality of q is

$$\left(\frac{d}{dt} q(t), \mathbf{e}_\eta(q(t)) \right)_\beta = \beta^2 \left(-\sin\theta \frac{d}{dt} x(t) + \cos\theta \frac{d}{dt} y(t) \right) = 0, \quad \forall\, t \in \mathbb{R}. \tag{26}$$

On a horizontal curve, ∂_ξ is always the spatial derivative tangent to the curve and $\mathbf{e}_\eta$ is always orthogonal to the curve.

By construction, curves in an image give *approximate* horizontal curve responses in the corresponding orientation score. It is only approximate since the oriented filters create some uncertainty in the orientation response, i.e. a curve in the image with orientation α will not render a perfect δ-spike response $\delta(\alpha - \theta)$ in the orientation score.

2.8 Exponential Curves

An *exponential curve* is a curve $\gamma : \mathbb{R} \to SE(2)$ for which the components of the tangent vector are constant over the entire parametrization, i.e.

$$\frac{d}{ds}\gamma(t) = c^\theta \mathbf{e}_\theta(\gamma(t)) + c^\xi \mathbf{e}_\xi(\gamma(t)) + c^\eta \mathbf{e}_\eta(\gamma(t)), \quad \text{for all } t \in \mathbb{R}. \tag{27}$$

Note that these curves are analogous to *straight* lines in $\mathbb{R}^n$, which also have a constant tangent vector. An exponential curve passing through point $g_0 \in SE(2)$ at $t = 0$ can be written as

$$\gamma_{\mathbf{c},g_0}(t) = g_0 \exp\left(t(c^\theta \partial_\theta\big|_{g=e} + c^\xi \partial_\xi\big|_{g=e} + c^\eta \partial_\eta\big|_{g=e}) \right). = g_0 \exp\left(t(c^\theta \partial_\theta + c^\xi \partial_x + c^\eta \partial_y) \right). \tag{28}$$

The expression for exponential curves in global $\{\mathbf{e}_\theta, \mathbf{e}_x, \mathbf{e}_y\}$ coordinates is given by (if $c^\theta \neq 0$)

$$\begin{aligned} \gamma_{\mathbf{c},g_0}(t) &= \begin{pmatrix} x_0 + \frac{c^\xi}{c^\theta}\, \mu(tc^\theta, \theta_0) + \frac{c^\eta}{c^\theta}\, \nu(tc^\theta, \theta_0) \\ y_0 + \frac{c^\xi}{c^\theta}\, \nu(tc^\theta, \theta_0) + \frac{c^\eta}{c^\theta}\, \mu(tc^\theta, \theta_0) \\ tc^\theta + \theta_0 \end{pmatrix}, \\ \text{with} \quad \mu(tc^\theta, \theta_0) &= \sin(tc^\theta + \theta_0) - \sin(\theta_0), \\ \nu(tc^\theta, \theta_0) &= \cos(tc^\theta + \theta_0) - \cos(\theta_0), \end{aligned} \tag{29}$$

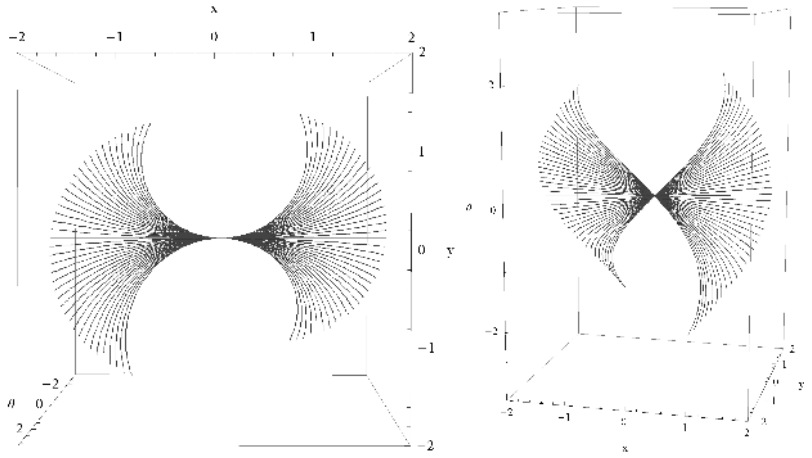


Figure 4: Horizontal exponential curves in $SE(2)$ for a range of different curvature values, shown from two different perspectives. The left-sided image shows that these curves are circular arc when projected onto the spatial plane.

which are *spirals* in $SE(2)$. The exponential curves for the special case $c^\theta = 0$ are

$$\gamma_{\mathbf{c},g_0}(t) = \begin{pmatrix} x_0 + tc^\xi \cos(\theta_0) - t\, c^\eta \sin(\theta_0) \\ y_0 + tc^\xi \sin(\theta_0) - t\, c^\eta \cos(\theta_0) \\ \theta_0 \end{pmatrix}, \tag{30}$$

which are straight lines in $SE(2)$ with constant orientation θ.

Horizontal exponential curves form a subset of all exponential curves with $c^\eta = 0$, see Figure 4.

2.9 Curvature and Deviation from Horizontality

From the tangent vector $\mathbf{c} = (c^\theta, c^\xi, c^\eta)$ of an exponential curve we define two features with a clear geometrical interpretation: *curvature* and *deviation from horizontality*, see Figure 5.

The *curvature* of an exponential curve in $SE(2)$ that is projected onto $\mathbb{R}^2$ is given by

$$\boldsymbol{\kappa}(s) = \frac{d^2}{ds^2}(\mathbb{P}_{\mathbb{R}^2}\gamma(s)) = \frac{c^\theta}{(c^\eta)^2 + (c^\xi)^2} \begin{pmatrix} -c^\eta \cos(s\, c^\theta + \theta_0) - c^\xi \sin(s\, c^\theta + \theta_0) \\ c^\xi \cos(s\, c^\theta + \theta_0) - c^\eta \sin(s\, c^\theta + \theta_0) \end{pmatrix} \tag{31}$$

where s is the arc length parametrization in the spatial plane, that is $||\frac{d}{ds}(\mathbb{P}_{\mathbb{R}^2}\gamma(s))|| = 1$. The signed norm of the curvature vector is

$$\kappa = ||\boldsymbol{\kappa}|| \operatorname{sign}(\boldsymbol{\kappa} \cdot \mathbf{e}_\eta) = \frac{c^\theta \operatorname{sign}(c^\xi)}{\sqrt{(c^\eta)^2 + (c^\xi)^2}}. \tag{32}$$

This scalar value can be intuitively interpreted: the curvature is equal to the slope at which the curve in the orientation score meets the spatial plane, see Figure 5. For a *horizontal* exponential curve we know that $c^\eta = 0$ and the curvature expression simplifies to

$$\kappa = \frac{c^\theta}{c^\xi}. \tag{33}$$

Together with $g_0 \in G$, the curvature κ fully describes a horizontal exponential curve $\gamma_{\mathbf{c},g_0}(t)\big|_{c^\eta=0}$. For a non-horizontal exponential curve, we also need the *deviation from horizontality* d_{H} which is

$$d_{\mathrm{H}} = \arctan(\frac{c^\eta}{c^\xi}), \tag{34}$$

i.e. d_{H} is the angle that the exponential curve, which is projected onto $\mathbb{R}^2$, makes with the horizontal direction $\mathbf{e}_\xi$, see Figure 5.

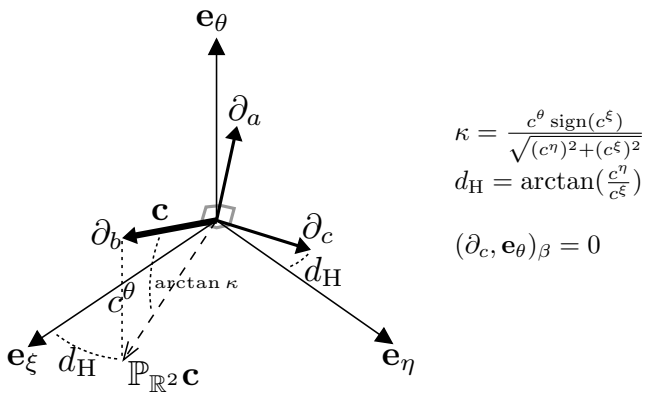


Figure 5: Illustration of curvature and deviation from horizontality (Section 2.9) and of the gauge frame (Section 2.11). Note that for visualization reasons, the lengths of the vectors are arbitrary. The true lengths are given by $\|\mathbf{c}\|_\beta = \|\mathbf{e}_\theta\|_\beta = \|\mathbf{e}_\xi\|_\beta = \|\mathbf{e}_\eta\|_\beta = 1$ and $\|\partial_a\|_\beta = \|\partial_b\|_\beta = \|\partial_c\|_\beta = \beta$.

2.10 Left-invariant Diffusion Equation on $SE(2)$

In scale space theory, one usually considers the diffusion equation on $\mathbb{R}^n$ as generating equation for a scale space of $f : \mathbb{R}^n \to \mathbb{R}$, yielding for the linear case

$$\begin{cases} \partial_t U(\mathbf{x},t) = \nabla \cdot \mathbf{D}\,\nabla U(\mathbf{x},t), \\ U(\mathbf{x},t=0) = f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n, t \geq 0, \end{cases} \tag{35}$$

where the ∇ operator is with respect to the spatial coordinates and the diffusion tensor $\mathbf{D}$ is a positive (semi)definite matrix of size $n \times n$. This diffusion equation is left-invariant with respect to the translation group.

In the same way, we construct the left-invariant diffusion equation for $SE(2)$.

$$\begin{cases} \partial_t U(g,t) = \nabla \cdot \mathbf{D}\,\nabla U(g,t) = \begin{pmatrix} \partial_\theta & \partial_\xi & \partial_\eta \end{pmatrix} \begin{pmatrix} D_{\theta\theta} & D_{\theta\xi} & D_{\theta\eta} \\ D_{\xi\theta} & D_{\xi\xi} & D_{\xi\eta} \\ D_{\eta\theta} & D_{\eta\xi} & D_{\eta\eta} \end{pmatrix} \begin{pmatrix} \partial_\theta \\ \partial_\xi \\ \partial_\eta \end{pmatrix} U(g,t), \\ U(g,t=0) = U_f(g), \quad g \in SE(2), t \geq 0 \end{cases} \tag{36}$$

where ∇ is defined in (24) and the diffusion tensor $\mathbf{D}$ a positive (semi)definite 3×3 matrix. In the linear case $\mathbf{D}$ is a constant matrix independent on g and t. The solution can be written as $U(\cdot,t) = e^{t(\nabla \cdot \mathbf{D}\,\nabla)} U_f$.

This equation generates a scale space on $SE(2)$, so it satisfies all scale space axioms as described in [12], except for the requirement of isotropy, which does not make sense in our inherently anisotropic setting. Furthermore, we have left-invariance instead of translation invariance.

Although there is no inherent notion of isotropy in $SE(2)$, we can define an artificial (but practically useful) notion of *β-isoptropic diffusion* , which is defined as

$$\partial_t U(g,t) = \left(\beta^2 \partial_\theta \partial_\theta + \partial_\xi \partial_\xi + \partial_\eta \partial_\eta\right) U(g,t). \tag{37}$$

The equation is "β-isoptropic" since $(\beta\partial_\theta, \beta\partial_\theta)_\beta = (\partial_\xi, \partial_\xi)_\beta = (\partial_\eta, \partial_\eta)_\beta = \beta^2$.

In the next section we will discuss useful choices for the diffusion tensor $\mathbf{D}$ for diffusion on $SE(2)$.

2.11 Gauge Frame for Anisotropic Diffusion in $SE(2)$

The idea of gauge coordinates in image processing [17] is to define a data-dependent orthogonal coordinate frame (the gauge frame) for the tangent space $T_{\mathbf{x}}(\mathbb{R}^N)$ at each position $\mathbf{x}$, such that

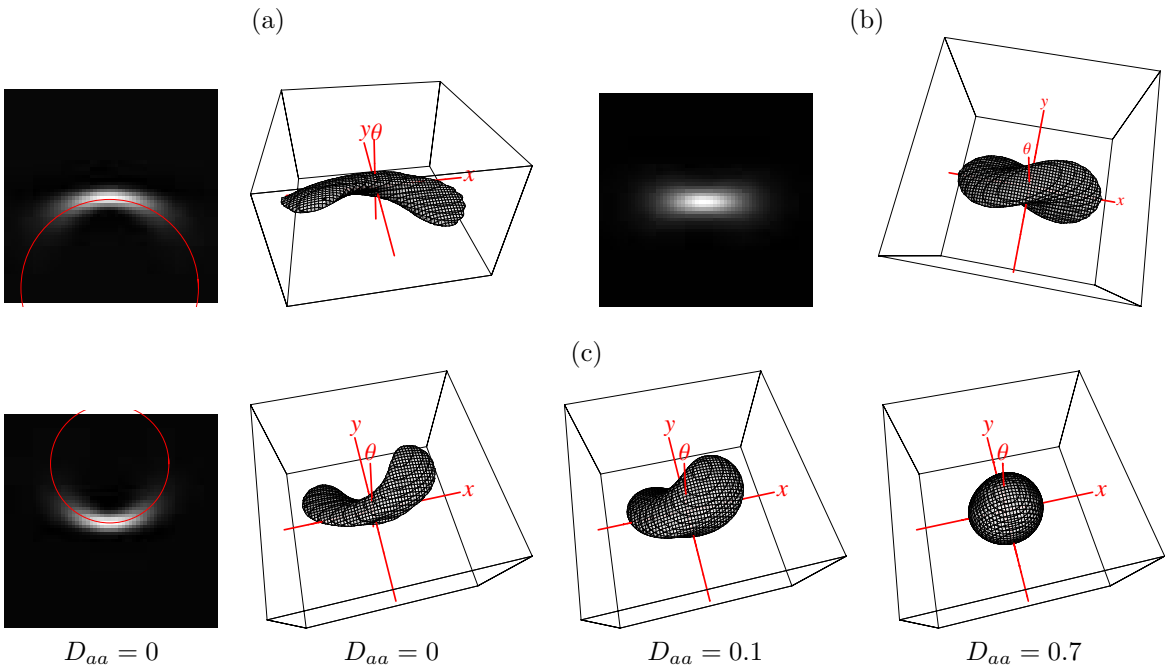


Figure 6: Illustrations of Green's functions for different parameter values, obtained using an explicit iterative numerical scheme (Section 7.2) with end time $t = 70$. (a) Shows the effect of nonzero κ. Parameters $\kappa = -0.04$, $D_{aa} = D_{cc} = 0$, $D_{bb} = 1$, and $\beta = 1$. Left: Greens functions in the spatial plane (i.e. all orientations are summed) where the superimposed circular arc shows the expected curvature, Right: isosurface in 3D. (b) Shows the effect of a nonzero D_{aa}. Parameters $D_{aa} = D_{cc} = 0.003$, $\kappa = 0$, $D_{bb} = 1$, and $\beta = 1$. (c) Shows the effect of varying $D_{aa} = D_{cc}$. Parameters $\kappa = 0.06$, $D_{bb} = 1$, and $\beta = 0.1$. As D_{aa} increases from 0 to 1, the resulting Green's function becomes more and more isotropic.

the basis vectors are in alignment with some local feature of interest in the image. The advantage of gauge coordinates is that operations described in gauge coordinates are automatically rotation invariant. The most commonly used gauge coordinates are determined by the gradient of a 2D image, where one basis vector is fixed tangent and one is fixed orthogonal to the direction of the gradient.

We apply the same idea in orientation scores. We want to establish a *β-orthogonal* gauge frame $\{\partial_a, \partial_b, \partial_c\}$, where β-orthogonality is defined as

$$(\partial_p, \partial_q)_\beta = \beta^2 \delta_{pq}, \quad p, q \in \{a, b, c\}. \tag{38}$$

Since exponential curves provide a local description of the structures we are interested in (i.e. curved elongated structures), one of the components of the gauge frame should be established by the tangent vector $\mathbf{c}(g)$ of exponential curve $\gamma_{\mathbf{c}(g),g}$ at each position $g \in SE(2)$ with $\|\mathbf{c}(g)\|_\beta = 1$, i.e. (omitting dependence on g)

$$\partial_b = \beta(c^\theta \partial_\theta + c^\xi \partial_\xi + c^\eta \partial_\eta). \tag{39}$$

The other two components ∂_a and ∂_c should span the plane orthogonal to ∂_b, and therefore can not be uniquely defined. We make an arbitrary by unique choice by ensuring that $\{\partial_a, \partial_b, \partial_c\}$ coincide with the left-invariant coordinate frame $\{\partial_\theta, \partial_\xi, \partial_\eta\}$ for the case of straight horizontal lines ($\kappa = 0$ and $d_{\mathrm{H}} = 0$), i.e.

$$\partial_a = \beta\partial_\theta, \quad \partial_b = \partial_\xi, \quad \partial_c = \partial_\eta \quad \text{iff } \mathbf{c} = (0, 1, 0). \tag{40}$$

In terms of κ and d_{H} (as defined in Section 2.9) this renders the following Gauge frame

$$\begin{pmatrix} \partial_a \\ \partial_b \\ \partial_c \end{pmatrix} = \mathbf{Q}_{\kappa,\beta}^{\mathrm{T}} \tilde{\mathbf{R}}_{d_{\mathrm{H}}}^{\mathrm{T}} \begin{pmatrix} \beta\partial_\theta \\ \partial_\xi \\ \partial_\eta \end{pmatrix}, \tag{41}$$

where

$$\tilde{\mathbf{R}}_{d_{\mathrm{H}}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos d_{\mathrm{H}} & -\sin d_{\mathrm{H}} \\ 0 & \sin d_{\mathrm{H}} & \cos d_{\mathrm{H}} \end{pmatrix}, \quad \mathbf{Q}_{\kappa,\beta} = \begin{pmatrix} \frac{\beta}{\sqrt{\beta^2+\kappa^2}} & \frac{-\kappa}{\sqrt{\beta^2+\kappa^2}} & 0 \\ \frac{\kappa}{\sqrt{\beta^2+\kappa^2}} & \frac{\beta}{\sqrt{\beta^2+\kappa^2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{42}$$

This gauge frame is illustrated in Figure 5. Note that it actually involves two rotations, since $\mathbf{Q}_{\kappa,\beta}$ is also a rotation matrix.

The class of $SE(2)$diffusions that are of our interest can now be expressed in the gauge coordinates as a diagonal diffusion equation (that is, no mixed terms)

$$\begin{cases} \partial_t U(g,t) = \Big(\partial_a D_{aa} \partial_a + \partial_b D_{bb} \partial_b + \partial_c D_{cc} \partial_c\Big) U(g,t), \\ U(g,t=0) = U_f(g). \end{cases} \tag{43}$$

Note that it is only correct to choose $D_{cc} = D_{aa}$ since ∂_a and ∂_c are arbitrarily chosen β-orthogonal to the tangent ∂_b. In left-invariant derivatives this equation can now be written as

$$\partial_t U(g,t) = \begin{pmatrix} \beta\partial_\theta & \partial_\xi & \partial_\eta \end{pmatrix} \left\{ \tilde{\mathbf{R}}_{d_{\mathrm{H}}} \mathbf{Q}_{\kappa,\beta} \begin{pmatrix} D_{aa} & 0 & 0 \\ 0 & D_{bb} & 0 \\ 0 & 0 & D_{aa} \end{pmatrix} \mathbf{Q}_{\kappa,\beta}^{\mathrm{T}} \tilde{\mathbf{R}}_{d_{\mathrm{H}}}^{\mathrm{T}} \right\} \begin{pmatrix} \beta\partial_\theta \\ \partial_\xi \\ \partial_\eta \end{pmatrix} U(g,t). \tag{44}$$

This diffusion equation will be used in the rest of the paper, where we will make D_{aa} dependent on the local differential structure of U. Figure 6 shows examples of Green's functions of linear evolutions of this type, for different values of D_{aa}, D_{bb} and κ.

3 Design of an Invertible Orientation Score Transformation

The previous section introduced the essential theory of orientation scores. The forthcoming sections will be more practical, and describe how the basic concepts are used in the algorithm. In this section we describe the practical design of an invertible orientation score transformation.

The invertibility conditions described in Section 2.3 still allow for a lot of freedom in choosing kernels Ψ. To restrict the possible choices, we first formulate some practical requirements that our transform needs to fulfill:

1. A finite number (n_θ) of orientations. This requirement is obvious from an implementational point of view.

2. Reconstruction by summing all orientations, i.e. $f(\mathbf{x}) = \sum_{i=0}^{n_\theta - 1} U_f(\mathbf{x}, i s_\theta)$, where s_θ is the orientation sample distance in radians, i.e. $s_\theta = \frac{2\pi}{n_\theta}$ if the periodicity of the orientation score 2π.

3. Polar separability in the Fourier domain, in order to design the radial and angular part separately: $\psi(\mathbf{x}) = f(\rho)h(\varphi)$ where f is the radial function and h the angular function, and $\boldsymbol{\omega} = (\omega_x, \omega_y) = (\rho\cos\varphi, \rho\sin\varphi)$.

4. The kernel should be strongly directional, i.e. the kernel should be a convex cone in the Fourier domain [1].

5. The kernel should be localized in the spatial domain, since we want to pick up local oriented structures.

6. The kernel should have the quadrature property [22]. More details can be found in Section 3.1

These requirements are similar to the requirements in [35], except for the reconstruction.

Based on these requirements we propose the following kernel

$$\psi(\mathbf{x}) = \frac{1}{N}\mathcal{F}^{-1}\left[\boldsymbol{\omega} \mapsto B^k\left(\frac{(\varphi \bmod 2\pi) - \pi/2}{s_\theta}\right) f(\rho)\right](\mathbf{x})\, G_s(\mathbf{x}) \tag{45}$$

where N is the normalization constant, $\boldsymbol{\omega} = (\rho\cos\varphi, \rho\sin\varphi)$, B^k denotes the kth order B-spline given by

$$B^k(x) = (B^{k-1} * B^0)(x), \qquad B^0(x) = \begin{cases} 1 & \text{if } -1/2 < x < +1/2 \\ 0 & \text{otherwise} \end{cases}. \tag{46}$$

Function $f(\rho)$ specifies the radial function in the Fourier domain, chosen as the Gaussian divided by its Taylor series up to order q to ensure a slower decay, i.e.

$$f(\rho) = G_t(\rho)\left(\sum_{i=0}^{q}\left(\frac{d}{d\rho'}G_t(\rho')\Big|_{\rho'=0}\right)\frac{\rho^i}{i!}\right)^{-1}, \quad G_t(\rho) = \frac{1}{2\sqrt{\pi t}}e^{-\frac{\rho^2}{4t}}. \tag{47}$$

Function G_s in (45) is a Gaussian kernel with scale s, which ensures spatial locality. Figure 2 shows an example of this orientation score transformation.

3.1 Quadrature Property and Hilbert Transform

The kernel ψ defined in equation (45) is a quadrature filter, meaning that the real part contains information about the locally even (symmetric) structures (e.g. rigdes) and the imaginary part contains information about the locally odd (anti-symmetric) structures (e.g. edges),

$$\psi(\mathbf{x}) = \psi_{\text{even}}(\mathbf{x}) - i\,\psi_{\text{odd}}(\mathbf{x}) \tag{48}$$

where ψ_{even} and ψ_{odd} are related to each other by the Hilbert transform

$$\psi_{\text{odd}} = \mathcal{H}^{\mathbf{e}_y}_{\mathbb{R}^2}\left[\psi_{\text{even}}\right], \tag{49}$$

where the Hilbert transform $\mathcal{H}^{\mathbf{v}}_{\mathbb{R}^2}$ is defined as

$$\mathcal{H}^{\mathbf{v}}_{\mathbb{R}^2}[U](\mathbf{x},\theta) = \mathcal{F}^{-1}[\boldsymbol{\omega} \mapsto i\,\text{sign}(\boldsymbol{\omega}\cdot\mathbf{v})\mathcal{F}[U(\cdot,\theta)](\boldsymbol{\omega})](\mathbf{x}), \tag{50}$$

where $\mathbf{v}$ specifies the direction in which the (in principle 1D) Hilbert transform is performed. In an orientation score, this direction is uniquely determined by $\mathbf{v} = \mathbf{e}_\eta$, leading to the following definition for the Hilbert transform on $SE(2)$:

$$\mathcal{H}_{SE(2)}[U](\mathbf{x},\theta) = \mathcal{H}^{\mathbf{e}_\eta(\theta)}_{\mathbb{R}^2}[U(\cdot,\theta)](\mathbf{x}) \tag{51}$$

Because of the quadrature property of the filter, the orientation score has the same properties and, consequently, the imaginary part does not supply any additional information that is not contained in the real-valued part. Therefore, to save memory we only need to store the real-valued part from 0 to π, i.e.

$$\text{Re}\{U_f\}(\mathbf{x},\theta) = (f * \psi^\theta_{\text{even}})(\mathbf{x}). \tag{52}$$

The complex-valued orientation score is simply found by $U_f = (\psi^\theta_{\text{even}} \star f) - i(\psi^\theta_{\text{odd}} \star f)) = \text{Re}\{U_f\}(\mathbf{x},\theta) - i\mathcal{H}_{SE(2)}[\text{Re}\{U_f\}]$. Moreover since $\psi^\theta_{\text{even}} = \psi^{\theta+\pi}_{\text{even}}$ and $\psi^\theta_{\text{odd}} = -\psi^{\theta+\pi}_{\text{odd}}$, we have the relation $U_f(\mathbf{x},\theta+\pi) = U^*_f(\mathbf{x},\theta)$.

4 Regularized Derivatives

In Section 2.6 we described left-invariant derivatives. It is well known in image processing that taking derivatives is an ill-posed problem, which is made well-posed by adding regularization. Gaussian derivatives are the most commonly used regularized derivative operators. In our orientation score framework we also need well-posed derivative operations, where the left-invariant diffusion described in Section 2.10 acts as regularizer. Left-invariant regularized derivatives on the orientation score are operationalized by $\mathcal{D}e^{t\mathcal{A}}W$ where $\mathcal{D}$ is a derivative of any order constructed from $\{\partial_\xi, \partial_\eta, \partial_\theta\}$ and $e^{t\mathcal{A}}$ accounts for diffusion. The order of the regularization operator and differential operators matters in this case, i.e. the diffusion should come first.

The exact and approximate analytic solutions for all heat kernels given by $e^{t\mathcal{A}}$ are described in [13]. In this paper, for simplicity and computation efficiency we restrict ourselves to the β-isotropic of equation (37), which can be written as

$$\partial_t W = \left(\beta^2\partial_\theta^2 + \left(\partial_\xi^2 + \partial_\eta^2\right)\right) W = \left(\beta^2\partial_\theta^2 + \left(\partial_x^2 + \partial_y^2\right)\right) W. \tag{53}$$

Since the operators ∂_θ, ∂_x, and ∂_y commute, this equation is the same as the diffusion equation in $\mathbb{R}^3$ except for the 2π-periodicity of the θ dimension. Therefore the Green's function is a Gaussian

$$G_{t_s,t_o}(x,y,\theta) = \frac{1}{8\sqrt{\pi^3 t_s^2 t_o}} e^{-\frac{x^2+y^2}{4t_s} - \frac{\theta^2}{4t_o}}, \quad \text{with } t_s = t \text{ and } t_o = \beta^2 t \tag{54}$$

In this special case we can use standard (separable) implementations of Gaussian derivatives, but we have to be careful because of the non-commuting operators. A normal (i,j,k)th order Gaussian derivative implementation for a 3D image f adheres to the right side of the following equation

$$\partial_x^i\partial_y^j\partial_z^k e^{t(\partial_x^2+\partial_y^2+\partial_z^2)} f = \partial_x^i e^{t\,\partial_x^2}\partial_y^j e^{t\,\partial_y^2}\partial_z^k e^{t\,\partial_z^2} f. \tag{55}$$

This equation is essential for the separability along the three dimensions. We want to use the same implementations to construct Gaussian derivatives in the orientation scores, meaning that we have to ensure that the same permutation of differential operators and regularization operators is allowed. By noting that

$$\begin{aligned} &\partial_\xi^i\,\partial_\eta^j\,\partial_\theta^k\, e^{t_o\partial_\theta^2 + t_s(\partial_\xi^2+\partial_\eta^2)} = \partial_\xi^i\partial_\eta^j\, e^{t_s(\partial_x^2+\partial_y^2)}\,\partial_\theta^k\, e^{t_o\partial_\theta^2}, \\ &\partial_\theta^k\,\partial_\xi^i\,\partial_\eta^j\, e^{t_o\partial_\theta^2 + t_s(\partial_\xi^2+\partial_\eta^2)} \neq \partial_\theta^k\, e^{t_o\partial_\theta^2}\,\partial_\xi^i\,\partial_\eta^j\, e^{t_s(\partial_x^2+\partial_y^2)}, \end{aligned} \tag{56}$$

we conclude that we always should ensure a certain *ordering* of the derivative operators, i.e. one should first calculate the orientational derivative ∂_θ and then the commuting spatial derivatives $\{\partial_\xi, \partial_\eta\}$, which are calculated from the Cartesian derivatives $\{\partial_x, \partial_y\}$ using (12). The commutator relations of (22) allow to rewrite the derivatives in this canonical order. For instance, the derivative $\partial_\xi\partial_\theta$ can be calculated directly with Gaussian derivatives, while $\partial_\theta\partial_\xi$ must be operationalized with Gaussian derivatives as $\partial_\xi\partial_\theta + \partial_\eta$.

5 Local Features in Orientation Scores

In order to make the diffusion in the orientation score adaptive to local line structures in the orientation score, we need to measure the local properties at each location $(\mathbf{x}, \theta)$. In our method we distinguish three local features (i.e., scalar values) at each position $g \in SE(2)$ to which the diffusion is adapted. The two features *curvature* $\kappa(g)$ and *deviation from horizontality* $d_{\mathrm{H}}(g)$ were already introduced in Section 2.9. The third important feature is *orientedness* $s(g)$: a scalar number indicating how oriented the local structure is. This means that a low value indicates that the local neighborhood is isotropic, and a high value indicates the neighborhood is anisotropic. In Figure 7 we show an example of these three local orientation score features.

For curvature and orientedness we need an estimate for the tangent vector $\mathbf{c}(g)$ at each position g. Therefore, after discussing what orientation score to use for feature estimation, in the next

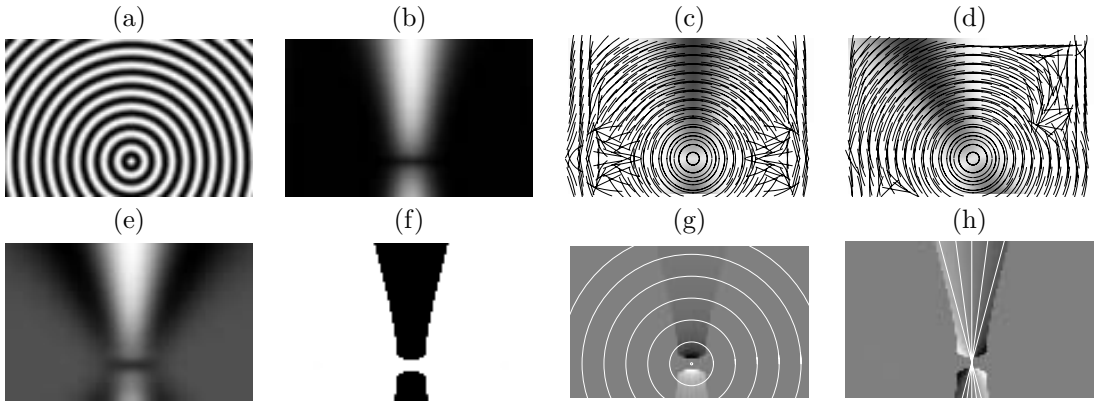


Figure 7: Example of feature estimation in an orientation score. (a) Input image f. (b) $|U_f|$ at $\theta = 0$, used for feature estimation. (c)+(d) estimated tangent vectors at orientations $\theta = 0$ and $\theta = \frac{5\pi}{32}$. The tangent vectors are displayed as circular arcs to show the estimated curvature as well as the deviation from horizontality. Note that the orientation and curvature estimation is isotropic regions in the orientation score, since the features are not well-defined there. (e) Orientedness s cf. (67) at $\theta = 0$. (f) D_{aa} calculated using (71). In this artificial image it leads to very sharp boundaries between isotropic diffusion (white area) and strongly anisotropic diffusion (black areas). (g) illustration of κ, where the curvature values are only indicated by the grayvalue in the region where $D_{aa} \approx 0$, since outside of this region the values are irrelevant. Clearly, the displayed iso-κ-contours, are situated on circular arcs. (h) d_{H}, where the deviation from horizontality values are again only indicated in the region where $D_{aa} \approx 0$. Clearly, the displayed iso-d_{H}-contours, are orthogonal to the concentric circles in the image, and the vertical line is the $d_{\mathrm{H}} = 0$ line.

subsection we will propose a method to estimate the tangent vector using first and second order left-invariant regularized derivatives. This estimate can be used to calculate curvature and deviation from horizontality with equation (32) and (34) respectively. Then, we will introduce a measure for orientedness.

5.1 Obtaining a Phase-Invariant Feature Orientation Score

For the feature estimation, we do not directly use the complex-valued orientation score U_f nor the real-valued orientation score $\mathrm{Re}\{U_f\}$. We rather aim to treat edges and ridges in a uniform manner, i.e. curvature, deviation from horizontality, and orientedness should be estimated in the same way and with the same quality independent on the local phase of the orientation score. Therefore, we use the real-valued orientation score $W = |U_f|$, yielding a *phase invariant* orientation score responding to both edges and ridges, see Figure 2d.

There is one drawback of this approach: a very regular goniometric pattern in an image, e.g. simply $\sin(x)$ (such as the image in Figure 2a), results in a flat plane response in W (see Figure 2d). This means that the tangent vector ∂_b tangent to the curves is locally not well-defined. For this kind of images, the problem is solved by forcing the deviation from horizontality to zero, as will be described in Section 5.2.1, or by using a different orientation score for feature estimation.

5.2 Tangent Vector Estimation

To estimate the tangent vector $\mathbf{c}(g)$ for all $g \in SE(2)$, we find the (horizontal) exponential curve that locally at point g fits best to the data. This local fit should not only be feasible at the centerlines of curves, but also if we shift a bit away from the centerline. At positions g in the orientation score U_f where there is no orientation, however, the tangent vector $\mathbf{c}(g)$ is not well defined, and we should design the non-linear diffusion process such that it does not take unreliable estimates of $\mathbf{c}(g)$ into account.

If we follow an oriented structure in the orientation score, the left-invariant gradient $\nabla W = (\partial_\theta, \partial_\xi, \partial_\eta)^{\mathrm{T}} W$ at all positions should remain constant. For example on the centerline of a curve the gradient remains zero, while the gradient will have a small constant η-component if we are a little bit off from the centerline. In other words, we formulate a minimization problem that minimizes over the "iso-contours" of the left-invariant gradient vector, leading to

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \left\{ \left\| \frac{d}{ds} (\nabla W(\gamma_{\mathbf{c},g_0}(s))) \big|_{s=0} \right\|_\beta^2 \,\middle|\, \|\mathbf{c}\|_\beta = 1 \right\}, \tag{57}$$

where $\mathbf{c} = \left(c^\theta, c^\xi, c^\eta\right)$, the norm $\|\cdot\|_\beta$ is defined in Section 2.5, and $\gamma_{\mathbf{c},g_0} = g_0 \exp\left(t(c^\theta \partial_\theta + c^\xi \partial_x + c^\eta \partial_y)\right)$, recall (27).

The minimizing equation in (57) is a norm of a covector and can be rewritten as

$$\begin{aligned} &\left\| \frac{d}{ds} (\nabla W(\gamma_{\mathbf{c},(s)})) \right\|_\beta^2 = \|\nabla(\nabla(\gamma_{\mathbf{c},(s)}))\, \dot{\gamma}_{\mathbf{c}}(s)\|_\beta^2 \\ &= \|\mathcal{H}W\mathbf{c}\|_\beta^2 = (\mathcal{H}W\mathbf{c}, \mathcal{H}W\mathbf{c})_\beta = (\mathbf{M}_\beta \mathcal{H}W\mathbf{c}, \mathbf{M}_\beta \mathcal{H}W\mathbf{c})_1 = (\mathbf{c}, (\mathcal{H}W)^{\mathrm{T}} \mathbf{M}_\beta^2 (\mathcal{H}W)\mathbf{c})_1 \end{aligned} \tag{58}$$

where $\mathbf{M}_\beta = \mathrm{diag}\{1, 1/\beta, 1/\beta\}$ and $(\cdot,\cdot)_1$ denotes the normal $\mathbb{R}^3$ inner product (i.e. with $\beta = 1$). The Hessian $\mathcal{H}$ on W is defined by

$$\mathcal{H}W = \nabla(\nabla W) = \begin{pmatrix} \partial_\theta^2 W & \partial_\xi \partial_\theta W & \partial_\eta \partial_\theta W \\ \partial_\theta \partial_\xi W & \partial_\xi^2 W & \partial_\eta \partial_\xi W \\ \partial_\theta \partial_\eta W & \partial_\xi \partial_\eta W & \partial_\eta^2 W \end{pmatrix}. \tag{59}$$

The side condition $\|\mathbf{c}\|_\beta = 1$ can be rewritten as

$$\|\mathbf{c}\|_\beta = (\mathbf{M}_\beta^{-1}\mathbf{c}, \mathbf{M}_\beta^{-1}\mathbf{c})_1 = (\mathbf{c}, \mathbf{M}_\beta^{-2}\mathbf{c})_1 \tag{60}$$

By Euler-Lagrange minimization $(\nabla_{\mathbf{c}} \|(\mathcal{H}W)\mathbf{c}\|_\beta^2 - \lambda(1 - \nabla_{\mathbf{c}}\|\mathbf{c}\|_\beta) = 0$ we get for the optimum $\mathbf{c}^*$:

$$(\mathcal{H}W)^{\mathrm{T}} \mathbf{M}_\beta^2 (\mathcal{H}W)\mathbf{c}^* = \lambda \mathbf{M}_\beta^{-2} \mathbf{c}^* \tag{61}$$

This can be rewritten as

$$(\mathbf{M}_\beta \mathcal{H}W \mathbf{M}_\beta)^{\mathrm{T}} (\mathbf{M}_\beta \mathcal{H}W \mathbf{M}_\beta)\, \tilde{\mathbf{c}}^* = \lambda\, \tilde{\mathbf{c}}^* \tag{62}$$

where $\tilde{\mathbf{c}}^* = \mathbf{M}_\beta^{-1}\mathbf{c}^*$ which amounts to eigensystem analysis of the symmetric 3×3 matrix $(\mathbf{M}_\beta \mathcal{H}W \mathbf{M}_\beta)^{\mathrm{T}} (\mathbf{M}_\beta \mathcal{H}W \mathbf{M}_\beta)$, where one of the three eigenvectors gives $\tilde{\mathbf{c}}^*$. The eigenvector with the smallest corresponding eigenvalue is selected as tangent vector $\tilde{\mathbf{c}}^*$, and the desired tangent vector $\mathbf{c}^*$ is then given by $\mathbf{c}^* = \mathbf{M}_\beta \tilde{\mathbf{c}}^*$.

Optionally, an increased noise-robustness can be achieved by component-wise blurring of the matrix $(\mathbf{M}_\beta \mathcal{H}W \mathbf{M}_\beta)^{\mathrm{T}} (\mathbf{M}_\beta \mathcal{H}W \mathbf{M}_\beta)$ before performing eigensystem analysis, i.e. (63) is replaced by

$$\left(G_{\rho_s,\rho_o} * (\mathbf{M}_\beta \mathcal{H}W \mathbf{M}_\beta)^{\mathrm{T}} (\mathbf{M}_\beta \mathcal{H}W \mathbf{M}_\beta)\right) \tilde{\mathbf{c}}^* = \lambda\, \tilde{\mathbf{c}}^* \tag{63}$$

where ρ_s and ρ_o are the spatial and orientational scales respectively. The post-blurring ensures that matrices that describe the local structure inaccurately, become more consistent with the surrounding. This approach is similar to the structure tensor where one applies post-blurring on the matrices formed by the dyadic product of the gradient with itself.

5.2.1 Enforcing Horizontality

An alternative is to force the curves to horizontality, which is more robust in case of regular oriented texture patterns. On non-horizontal curves, however, the expected results will be worse. Horizontality is imposed by forcing c^η to zero in (57). In the minimization term, $(\mathcal{H}W\mathbf{c})$ can now be rewritten as

$$\mathcal{H}W\mathbf{c}\big|_{c^\eta=0} = \mathcal{H}_{\mathrm{hor}} W \mathbf{c}_{\mathrm{hor}} = \begin{pmatrix} \partial_\theta^2 W & \partial_\xi \partial_\theta W \\ \partial_\theta \partial_\xi W & \partial_\xi^2 W \\ \partial_\theta \partial_\eta W & \partial_\xi \partial_\eta W \end{pmatrix} \begin{pmatrix} c^\theta \\ c^\xi \end{pmatrix} \tag{64}$$

Now the Euler Lagrange equation gives

$$(\mathbf{M}_\beta \, \mathcal{H}_{\mathrm{hor}} W \, \mathbf{M}_{\beta,\mathrm{hor}})^{\mathrm{T}} (\mathbf{M}_\beta \, \mathcal{H}_{\mathrm{hor}} W \, \mathbf{M}_{\beta,\mathrm{hor}}) \, \tilde{\mathbf{c}}^*_{\mathrm{hor}} = \lambda \, \tilde{\mathbf{c}}^*_{\mathrm{hor}} \tag{65}$$

where $\tilde{\mathbf{c}}^*_{\mathrm{hor}} = \mathbf{M}_{\beta,\mathrm{hor}} \mathbf{c}^*_{\mathrm{hor}}$ and $\mathbf{M}_{\beta,\mathrm{hor}} = \mathrm{diag}\{1, 1/\beta\}$. This amounts to eigensystem analysis of a symmetric 2×2 matrix. The eigenvector corresponding to the smallest eigenvalue should be selected and the curvature is given by (33). The deviation from horizontality is inherently zero in this case. The fact that we have 2×2 matrices instead of 3×3 is a practical advantage of this approach.

5.2.2 Structure Tensor Approach

An alternative approach to the tangent vector estimation described above, is to use the structure tensor instead of the Hessian, as was proposed by Van Ginkel [35] for the purpose of curvature estimation. In this approach one simply replaces the Hessian by the structure tensor, defined by

$$\mathcal{S}W = \tilde{\mathbf{R}}_\theta \left\{ G_{\rho_s,\rho_o} * \left(\begin{pmatrix} \partial_\theta W \\ \partial_x W \\ \partial_y W \end{pmatrix} \cdot \begin{pmatrix} \partial_\theta W \\ \partial_x W \\ \partial_y W \end{pmatrix}^{\mathrm{T}} \right) \right\} \tilde{\mathbf{R}}_\theta^{\mathrm{T}}, \tag{66}$$

where the derivatives (note: not the left-invariant ones) are implemented by Gaussian derivatives, and G_{ρ_s,ρ_o} denotes the Gaussian smoothing kernel that is applied componentwise to the structure tensor. $\tilde{\mathbf{R}}_\theta$ denotes the rotation matrix of equation (42) that makes the structure tensor left-invariant. On the resulting structure tensors we apply eigensystem analysis in exactly the same manner as described above for $(\tilde{\mathcal{H}}W)^{\mathrm{T}}(\tilde{\mathcal{H}}W)$.

5.3 Orientedness

In a 3D image, an often used measure for orientedness is the sum of the two second order derivatives orthogonal to the orientation of the elongated structure, which amounts to the sum of the two largest absolute eigenvalues of the Hessian matrix. In fact, this corresponds to the Laplacian in the 2D subspace formed by the plane orthogonal to the line structure.

In the orientation score we adopt the same approach. As measure for orientedness s we take the Laplacian in the plane orthogonal to the line, which is calculated by

$$s = -\Delta_{\mathrm{orth}} W = -(\mathbf{e}_{o1}^{\mathrm{T}} \mathcal{H} W \mathbf{e}_{o1} + \mathbf{e}_{o2}^{\mathrm{T}} \mathcal{H} W \mathbf{e}_{o2}) \tag{67}$$

where $\mathbf{e}_{o1}$ and $\mathbf{e}_{o2}$ are the two vectors that are orthonormal to the tangent vector $\mathbf{c}$ with respect to the inner product defined in equation (17), i.e.

$$(\mathbf{c}, \mathbf{e}_{o1})_\beta = 0, \quad (\mathbf{c}, \mathbf{e}_{o2})_\beta = 0, \quad (\mathbf{e}_{o1}, \mathbf{e}_{o2})_\beta = 0, \quad (\mathbf{e}_{o1}, \mathbf{e}_{o1})_\beta = 1, \quad (\mathbf{e}_{o2}, \mathbf{e}_{o2})_\beta = 1. \tag{68}$$

The minus sign in (67) is included in order to get a positive response for oriented structures: an oriented structure in $W = |U_f|$ always renders a convex hill in the intensity landscape of the orientation score, yielding a negative second order derivative.

Note that one can also use the two other eigenvectors of the matrix $(\mathbf{M}_\beta \mathcal{H} W \mathbf{M}_\beta)^{\mathrm{T}} (\mathbf{M}_\beta \mathcal{H} W \mathbf{M}_\beta)$. If $\tilde{\mathbf{e}}_{o1}$ and $\tilde{\mathbf{e}}_{o2}$ are the eigenvectors orthonormal to the selected eigenvector $\tilde{\mathbf{c}}^*$ (with respect to the $(\cdot,\cdot)_1$ inner product) then the orientedness is given by

$$s = -\left(\tilde{\mathbf{e}}_{o1}^{\mathrm{T}} \mathbf{M}_\beta \mathcal{H} W \mathbf{M}_\beta \tilde{\mathbf{e}}_{o1} + \tilde{\mathbf{e}}_{o2}^{\mathrm{T}} \mathbf{M}_\beta \mathcal{H} W \mathbf{M}_\beta \tilde{\mathbf{e}}_{o2} \right). \tag{69}$$

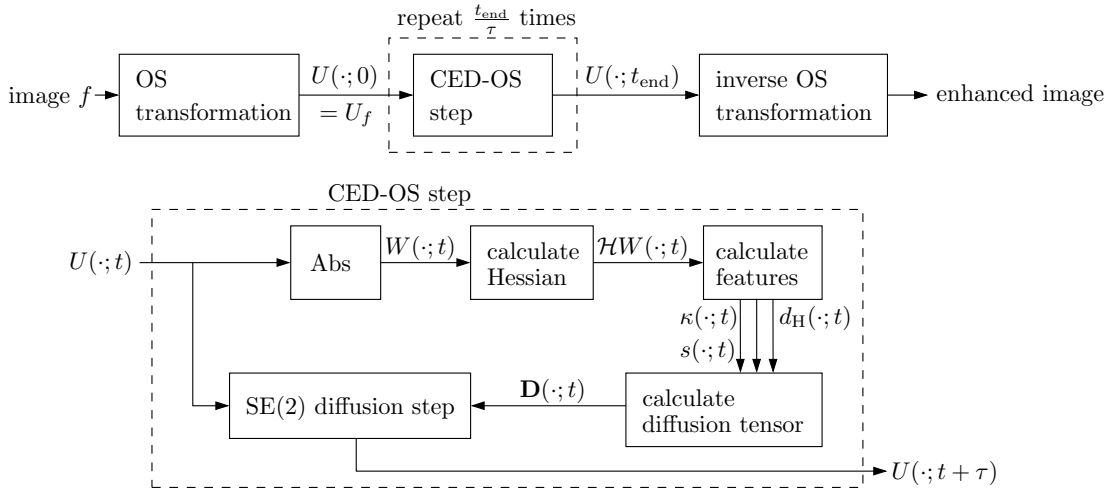


Figure 8: Flow chart of the CED-OS (Coherence Enhancing Diffusion in Orientation Score) method.

6 Non-Linear Diffusion in the Orientation Score

This section describes how to apply non-linear coherence-enhancing diffusion in the orientation score. The evolution equation we consider is the $SE(2)$ diffusion equation expressed in Gauge coordinates, equation (43), where the coefficients $D_{bb} = 1$ and $D_{cc} = D_{aa}$ are non-linearly dependent on the features,

$$\begin{cases} \partial_t U(g,t) = \Big(\partial_a D_{aa}(U)(g,t)\partial_a + \partial_b\partial_b + \partial_c D_{aa}(U)(g,t)\partial_c \Big) U(g,t), \\ U(g,t=0) = U_f(g). \end{cases} \tag{70}$$

where it should be emphasized that the derivatives ∂_a, ∂_b, and ∂_c are dependent on $U(g,t)$, although this is not explicitly indicated in the equation.

At positions in the orientation score with a strongly oriented structure, i.e. higher orientedness s, we only want to diffuse tangent to this structure, so D_{bb} should be large and $D_{aa} = D_{cc}$ should be small. If there is no strong orientation, the diffusion should be β-isotropic, so $D_{aa} = D_{bb} = D_{cc}$ should be large. Notice that in the resulting β-isotropic diffusion tensor for the latter case, the variables κ and d_{H} drop out, which is desirable since on non-oriented positions these local features are not defined.

For the conductivity function $D_{aa}(U)$ we have different choices. We propose to use

$$D_{aa}(U)(g,t) = \begin{cases} \exp\left(-\frac{s(U)(g,t)}{c}\right) & s(U)(g,t) \geq 0; \\ 1 & \text{otherwise} \end{cases} \tag{71}$$

where the non-linear function is always between zero and one, such that low values of s give $D_{aa} \approx 1$ and large values give $D_{aa} \approx 0$. The parameter c controls the behavior of the non-linear function.

Figure 8 shows how all different parts are connected together. The details of all building blocks are explained in the preceding sections, except for the "$SE(2)$ diffusion step". The next section will describe how we will numerically solve this step.

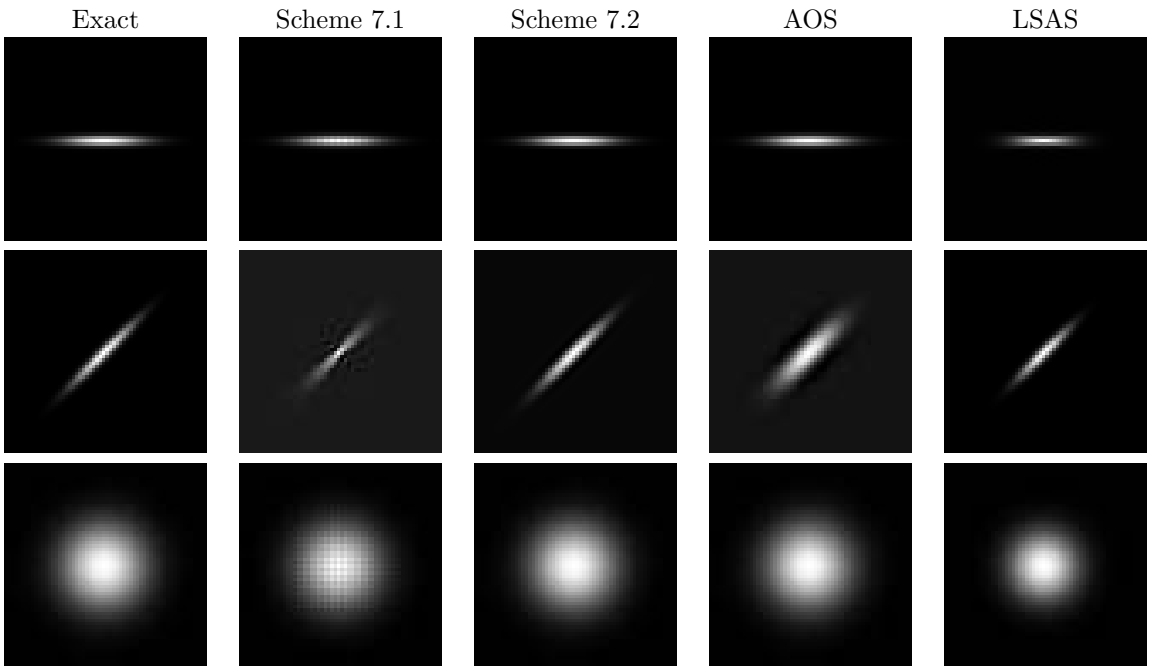


Figure 9: Comparison of rotation invariance of different numerical anisotropic diffusion schemes. The input to all algorithms is an image with dimensions 60×60 with a single Gaussian blob with scale 0.9 pixels, linear diffusion is applied with diffusion tensor $\mathbf{D} = \mathbf{R}_\theta \operatorname{diag}(D_{xx}, D_{yy}) \mathbf{R}_\theta^{-1}$, $\tau = 0.25$, and end time $t = 35$. The exact solution is found simply by convolving the input image with the anisotropic Gaussian kernel. Top row: $\theta = 0$, $D_{xx} = 1$, $D_{yy} = 0.0025$. Middle row: $\theta = \pi/4$, $D_{xx} = 1$, $D_{yy} = 0$. Bottom row: $D_{xx} = D_{yy} = 1$ (isotropic diffusion).

7 Numerical Scheme for Non-Linear Diffusion

In this section we will propose two explicit finite difference schemes to solve diffusion equation (70). We restrict to explicit schemes since implicit schemes are generally very expensive for our 3-dimensional anisotropic case. Furthermore, our anisotropic PDE requires good rotational invariance. Many efficient (semi)-implicit schemes with operator splitting, e.g. the AOS (additive operator splitting) scheme [39], are therefore discarded due to their poor rotation invariance (see Figure 9). The LSAS scheme [40] has good rotational invariance and can be performed in 3D, however it is inherently designed for isotropic grids, which is problematic in our case since we do not have a natural notion of isotropy. One could use the artificial notion of β-isotropy and use LSAS but then one would be restricted to the case $s_\theta = \beta$, and there is no good reason for this restriction.

7.1 Simple Explicit Finite Difference Scheme

A simple scheme for solving the non-linear diffusion PDE (70) is obtained by rewriting the PDE to ∂_x, ∂_y, and ∂_θ derivatives, and using centered finite differences to calculate the first order derivatives, e.g. for second order accurate finite differences this yields

$$\begin{aligned}
\partial_x U(\mathbf{x}, l) &\approx \frac{1}{2} \left(U(\mathbf{x} + \mathbf{e}_x, l) - U(\mathbf{x} - \mathbf{e}_x, l) \right), \\
\partial_y U(\mathbf{x}, l) &\approx \frac{1}{2} \left(U(\mathbf{x} + \mathbf{e}_y, l) - U(\mathbf{x} - \mathbf{e}_y, l) \right), \\
\partial_\theta U(\mathbf{x}, l) &\approx \frac{1}{2 s_\theta} \left(U(\mathbf{x}, l+1) - U(\mathbf{x} - \mathbf{e}_x, l-1) \right).
\end{aligned} \tag{72}$$

where $l \in [0, n_\theta - 1]$ denotes the sampled orientation axis where $\theta = l\, s_\theta$ with orientation sample distance s_θ. In time direction we use the first order forward finite difference $(U^{k+1} - U^k)/\tau$ where $k \in \mathbb{N}$ is the discrete time and τ the time step. The advantage of this scheme is the efficiency and good stability since the effective stencil size is 5 in each dimension rather than 3 in most other simple explicit schemes. The drawback, however, is that oscillations at the Nyquist frequency can occur, caused by the fact that a concatentation of two first order centered differences gives $\partial_x^2 U = \frac{1}{4}\left(U(\mathbf{x} + 2\mathbf{e}_x, l) - 2U(\mathbf{x}, l) + U(\mathbf{x} - 2\mathbf{e}_x, l)\right)$, i.e. the closest neighboring pixels are not taken into account. The latter problem can be resolved by adding some additional coupling between neighboring pixels, for instance by using a 3-pixel scheme to perform the isotropic part of the diffusion [37].

7.1.1 Stability Analysis

For the stability analysis, we consider the linear diffusion equation (70) assuming constant D_{aa} and find an upper bound for time step τ such that the equation remains stable for all applicable cases. We restrict to the cases that can occur in the non-linear diffusion equation of Section 6, i.e. $D_{aa} = D_{cc} \leq D_{bb} = 1$. In $\{\partial_\theta, \partial_x, \partial_y\}$ coordinates, the diffusion tensor components are given by

$$\begin{aligned}
D_{\theta\theta} &= \frac{1}{q^2}\left(D_{aa}\cos^2(\alpha) + \sin^2(\alpha)\right) \\
D_{xx} &= \cos^2(\alpha)\cos^2(\theta) + D_{aa}\left(\cos^2(\theta)\sin^2(\alpha) + \sin^2(\theta)\right) \\
D_{yy} &= D_{aa}\cos^2(\theta) + \left(\cos^2(\alpha) + D_{aa}\sin^2(\alpha)\right)\sin^2(\theta) \\
D_{\theta x} &= \frac{1}{q}(D_{aa} - 1)\cos(\alpha)\cos(\theta)\sin(\alpha) \\
D_{\theta y} &= \frac{1}{q}(D_{aa} - 1)\cos(\alpha)\sin(\alpha)\sin(\theta) \\
D_{xy} &= (1 - D_{aa})\cos^2(\alpha)\cos(\theta)\sin(\theta)
\end{aligned} \tag{73}$$

where $q = \frac{s_\theta}{\beta}$, $\sin\alpha = \frac{\kappa}{\sqrt{\beta^2+\kappa^2}}$, and $\cos\alpha = \frac{\beta}{\sqrt{\beta^2+\kappa^2}}$. The first order finite differences are defined in (72), rendering the following stencils for the second order finite differences that are applied in all three dimensions

$$\mathbf{S}_{ii} = \frac{1}{4}\begin{pmatrix} 1 & 0 & -2 & 0 & 1 \end{pmatrix}, \qquad \mathbf{S}_{ij|i\neq j} = \frac{1}{4}\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix}. \tag{74}$$

One numerical iteration can be written as

$$U^{k+1} = U^k + \tau\mathbf{A}(U^k)U^k = \underbrace{\left(\mathbf{I} + \tau\mathbf{A}(U^k)\right)}_{\mathbf{M}(U^k)}U^k \tag{75}$$

where $\mathbf{M}(U^k)$ is a square matrix with size equal to the total number of "voxels" in the orientation score. The numerical method is stable as long as the absolute values of all eigenvalues of $\mathbf{M}$ are ≤ 1. Using the Gershgorin circle theorem [21] we find that all eigenvalues are situated in a circle with center C and radius R

$$\begin{aligned}
C &= 1 - \frac{\tau}{2}(D_{\theta\theta} + D_{xx} + D_{yy}) \\
R &= \frac{\tau}{2}(|D_{\theta\theta}| + |D_{xx}| + |D_{yy}| + 4|D_{\theta x}| + 4|D_{\theta y}| + 4|D_{xy}|).
\end{aligned} \tag{76}$$

Stability requires $-1 \leq C - R$ and $C + R \leq 1$. The first inequality gives as bound for τ

$$\begin{aligned}\tau \leq 4q^2 \Big\{ & (1+q^2) + D_{aa}(1+3q^2) + \cos(2\alpha)(D_{aa}-1)(1-q^2) \\ & + 2q\,|(D_{aa}-1)\sin(2\alpha)|\;(|\cos\theta| + |\sin\theta|) \\ & + q^2\,|(1-D_{aa})\sin(2\theta)(\cos(2\alpha)+1)| \Big\}^{-1}\end{aligned} \tag{77}$$

Note that the second inequality $C + R \leq 1$ never holds as (76) shows that $C + R \geq 1$. However it can safely be discarded since $\mathbf{A}(U^k)$ is negative definite i.e. $\sigma(\mathbf{A}(U^k)) < 0$, where σ denotes the spectrum of the matrix, implying $\sigma(\mathbf{M}(U^k)) = \sigma(1 + \tau\mathbf{A}(U^k)) = 1 + \tau\sigma(\mathbf{A}(U^k)) < 1$ for all $\tau > 0$.

We want to find the case for which we get the lowest upper bound for τ, to guarantee stability in all circumstances. For the sine and cosine terms in the denominator, these worst case values are

$$\begin{aligned}\cos(2\alpha) = 1 \text{ for } \alpha = \frac{\pi}{4}, \\ \sin(2\alpha) = 1 \text{ for } \alpha = 0, \\ (|\cos\theta| + |\sin\theta|) = \sqrt{2} \text{ for } \theta = \frac{\pi}{4}.\end{aligned} \tag{78}$$

Furthermore, for D_{aa} we have to set $D_{aa} = 0$ to get the "worst case" for τ, since degenerate anisotropic diffusion is most critical concerning stability. So, we find the following upper bound for τ, which guarantees stability for all θ, α, and $0 \leq D_{aa} \leq 1$

$$\tau \leq \frac{4q^2}{1 + 2\sqrt{2}q + 3q^2 - |1 - q^2|}. \tag{79}$$

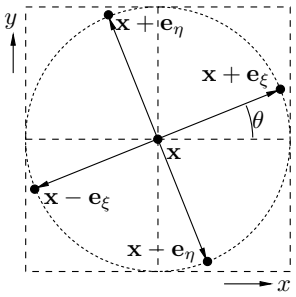
For a typical value of $q = \frac{s_\theta}{\beta} = \frac{\pi/32}{0.1}$ this yields $\tau \leq 0.60$, which is a fairly sharp upper bound if we consider our practical observations.

7.2 Left-invariant Explicit Scheme with Spline Interpolation

The important property of the differential operators ∂_ξ, ∂_η, and ∂_θ is their left-invariance. The performance of a numerical scheme should therefore be more optimal if this left-invariance is carried over to the finite differences that are used. To achieve this we should define the spatial finite differences in the directions defined by the left-invariant $\mathbf{e}_\xi$, $\mathbf{e}_\eta$ tangent basis vectors, instead of the sampled $\mathbf{e}_x, \mathbf{e}_y$ grid. In effect, the principal axes of diffusion in the spatial plane are always aligned with the finite differences as long as we do not include $d_{\mathrm{H}} \neq 0$. The reason that we do not consider deviation from horizontality is that this scheme becomes very expensive and complicated in that case.

For the numerical scheme we apply the chain rule on the right-hand side of the PDE (70) expressed in left-invariant derivatives cf. (44) with $d_{\mathrm{H}} = 0$ (i.e., analogue to 1D: $\partial_x(D\,\partial_x U) = D\,\partial_x^2 U + (\partial_x D)(\partial_x U)$). The left-invariant derivatives are replaced by the finite differences defined in Figure 10. In time direction we use the first order forward finite difference. Interpolation is required at spatial positions $\mathbf{x} \pm \mathbf{e}_\xi$ and $\mathbf{x} \pm \mathbf{e}_\eta$. For this purpose we use the algorithms for B-spline interpolation proposed by Unser et al. [33] with B-spline order 2. This interpolation algorithm consists of a prefiltering step with a separable infinite impulse response filter to determine the B-spline coefficients. The interpolation images such as $U^k(\mathbf{x} \pm \mathbf{e}_\xi)$ can then be calculated by a separable convolution with a shifted B-spline. The examples in Figure 6 and all experiments in the next section are obtained with this numerical scheme.

The main advantage of this scheme is the improved rotation invariance Figure 9. The drawback, however, is the computational speed and the some additional blurring caused by the interpolation scheme.



$$\partial_\theta U \approx \frac{1}{2s_\theta}\left(U(\mathbf{x}, l+1) - U(\mathbf{x}, l-1)\right)$$
$$\partial^2_\theta U \approx \frac{1}{s^2_\theta}\left(U(\mathbf{x}, l+1) - 2U(\mathbf{x}, l) + U(\mathbf{x}, l-1)\right)$$
$$\partial_\xi U \approx \frac{1}{2}\left(U(\mathbf{x}+\mathbf{e}^l_\xi, l) - U(\mathbf{x}-\mathbf{e}^l_\xi, l)\right)$$
$$\partial^2_\xi U \approx U(\mathbf{x}+\mathbf{e}^l_\xi, l) - 2U(\mathbf{x}, l) + U(\mathbf{x}-\mathbf{e}^l_\xi, l)$$
$$\partial_\eta U \approx \frac{1}{2}\left(U(\mathbf{x}+\mathbf{e}^l_\eta, l) - U(\mathbf{x}-\mathbf{e}^l_\eta, l)\right)$$
$$\partial^2_\eta U \approx U(\mathbf{x}+\mathbf{e}^l_\eta, l) - 2U(\mathbf{x}, l) + U(\mathbf{x}-\mathbf{e}^l_\eta, l)$$

$$\partial_\xi \partial_\theta U \approx \frac{1}{4s_\theta}\left(U(\mathbf{x}+\mathbf{e}^l_\xi, l+1) - U(\mathbf{x}+\mathbf{e}^l_\xi, l-1) - U(\mathbf{x}-\mathbf{e}^l_\xi, l+1) + U(\mathbf{x}-\mathbf{e}^l_\xi, l-1)\right)$$
$$\partial_\theta \partial_\xi U \approx \frac{1}{4s_\theta}\left(U(\mathbf{x}+\mathbf{e}^{l+1}_\xi, l+1) - U(\mathbf{x}+\mathbf{e}^{l+1}_\xi, l-1) - U(\mathbf{x}-\mathbf{e}^{l-1}_\xi, l+1) + U(\mathbf{x}-\mathbf{e}^{l-1}_\xi, l-1)\right)$$

Figure 10: Illustration of the spatial part of the stencil of the numerical scheme. The horizontal and vertical dashed lines indicate the sampling grid, which is aligned with $\{\mathbf{e}_x, \mathbf{e}_y\}$. The stencil points, indicated by the black dots, are aligned with the rotated coordinate system cf. (11) with $\theta = l\, s_\theta$.

7.2.1 Stability Bound

We derive a stability bound for this numerical scheme in the same fashion as for the previous numerical scheme. Again we restrict to the cases $D_{aa} = D_{cc} \leq D_{bb} = 1$. In $\{\partial_\theta, \partial_\xi, \partial_\eta\}$ coordinates, the diffusion tensor is given by

$$\begin{aligned} D_{\theta\theta} &= \frac{1}{q^2} D_{aa} \cos^2(\alpha) + \sin^2(\alpha) \\ D_{\xi\xi} &= \cos^2(\alpha) + D_{aa} \sin^2(\alpha) \\ D_{\eta\eta} &= D_{aa} \\ D_{\theta\xi} &= \frac{1}{q}(D_{aa} - 1)\cos(\alpha)\sin(\alpha) \\ D_{\theta\eta} &= D_{\xi\eta} = 0 \end{aligned} \tag{80}$$

where $q = \frac{s_\theta}{\beta}$. The used stencils for the second order finite differences are

$$\mathbf{S}_{ii} = \begin{pmatrix} 1 & -2 & 1 \end{pmatrix} \qquad \mathbf{S}_{ij|i\neq j} = \frac{1}{4}\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \tag{81}$$

In this scheme, all off-center stencil positions are obtained by second order spline interpolation, which has to be taken into account in the stability analysis. In the matrix $\mathbf{M}$ (with $U^{k+1} = \mathbf{M}(U^k)U^k$) this interpolation leads to more nonzero off-diagonal terms since interpolation amounts to a linear combination of a number of voxel values. Second order B-spline interpolation does not preserve the global maximum $m = \max_{g\in SE(2)}(U(g))$, but we can find a factor $1 < \chi < \infty$ which gives an upper bound such that $U_{\text{interpolated}}(g) \leq \chi \cdot m$ for all g i.e. for an arbitrary interpolation. This factor χ is found as follows

$$\chi = \max_{\Delta\in\{-1/2,+1/2\}} \sum_{x=-\infty}^{+\infty} |C_2(x+\Delta)| = \sqrt{2} \tag{82}$$

where C_2 is the cardinal spline [33] corresponding to the second order B-spline, which is in fact the net convolution kernel that is used for spline interpolation.

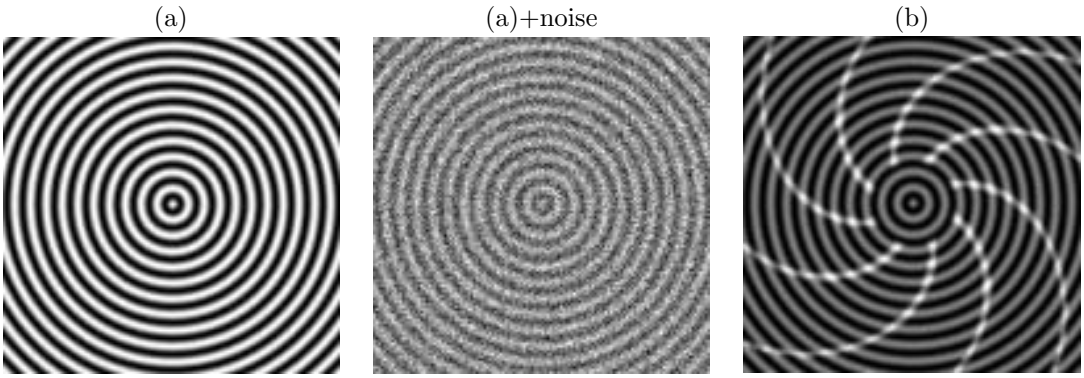


Figure 11: Curvature estimation test images.

Using Gershgorin circle theorem and taking into account $\chi = \sqrt{2}$ we find for the circle center C and circle radius R

$$\begin{aligned} C &= 1 - 2\tau \left(\left(\frac{D_{aa}}{q^2} + 1 \right) \cos^2(\alpha) + \left(D_{aa} + \frac{1}{q^2} \right) \sin^2(\alpha) + D_{aa} \right), \\ R &= 2\tau \left(\sqrt{2} D_{aa} + \sqrt{2} \left| \frac{(D_{aa} - 1)\cos(\alpha)\sin(\alpha)}{q} \right| + \right. \\ &\qquad \left. \left(D_{aa} q^{-2} + \sqrt{2} \right) \cos^2(\alpha) + \left(q^{-2} + \sqrt{2} D_{aa} \right) \sin^2(\alpha) \right) \end{aligned} \tag{83}$$

stability requires $-1 \leq C - R$ and $C + R \leq 1$. The first equation renders as bound for τ

$$\begin{aligned} \tau \leq q^2 \Bigg\{ &\left(\frac{1}{\sqrt{2}} \left| \frac{(1 - D_{aa})\sin(2\alpha)}{q} \right| + \frac{1}{2} \left(1 + \sqrt{2} \right) \left(3 D_{aa} + (1 - D_{aa})\cos(2\alpha) + 1 \right) \right) q^2 \\ &+ D_{aa} + (D_{aa} - 1)\cos(2\alpha) + 1 \Bigg\}^{-1} \end{aligned} \tag{84}$$

By inserting the "worst case values" for all sine and cosine terms, see (78), and by setting isoptopic diffusion $D_{aa} = 1$, which in this case is the worst-case for stability since it gives most off-diagonal components in matrix $\mathbf{M}$, we find the following a upper bound for τ, which guarantees stability for all θ and α and $0 \leq D_{aa} \leq 1$

$$\tau \leq \frac{2q^2}{4 + 4(1 + \sqrt{2})q^2}. \tag{85}$$

For a typical value of $q = \frac{s_\theta}{\beta} = \frac{\pi/32}{0.1}$ this yields $\tau \leq 0.15$, which coincides with practical observations.

8 Results

This section is divided in two parts. First we will quantitatively evaluate the quality of the curvature estimation, using a set of artificial test images. Then we will show results of the coherence-enhancing diffusion in orientation score (CED-OS) algorithm. We will qualitatively show the differences between including curvature and/or deviation from horizontality, and compare CED-OS with standard coherence-enhancing diffusion on the image.

8.1 Curvature Estimation Experiments

For our experiments, we apply curvature estimation on an image containing concentric circles (Figure 12). This is a useful test image since it contains a wide range of circle radii (in this case

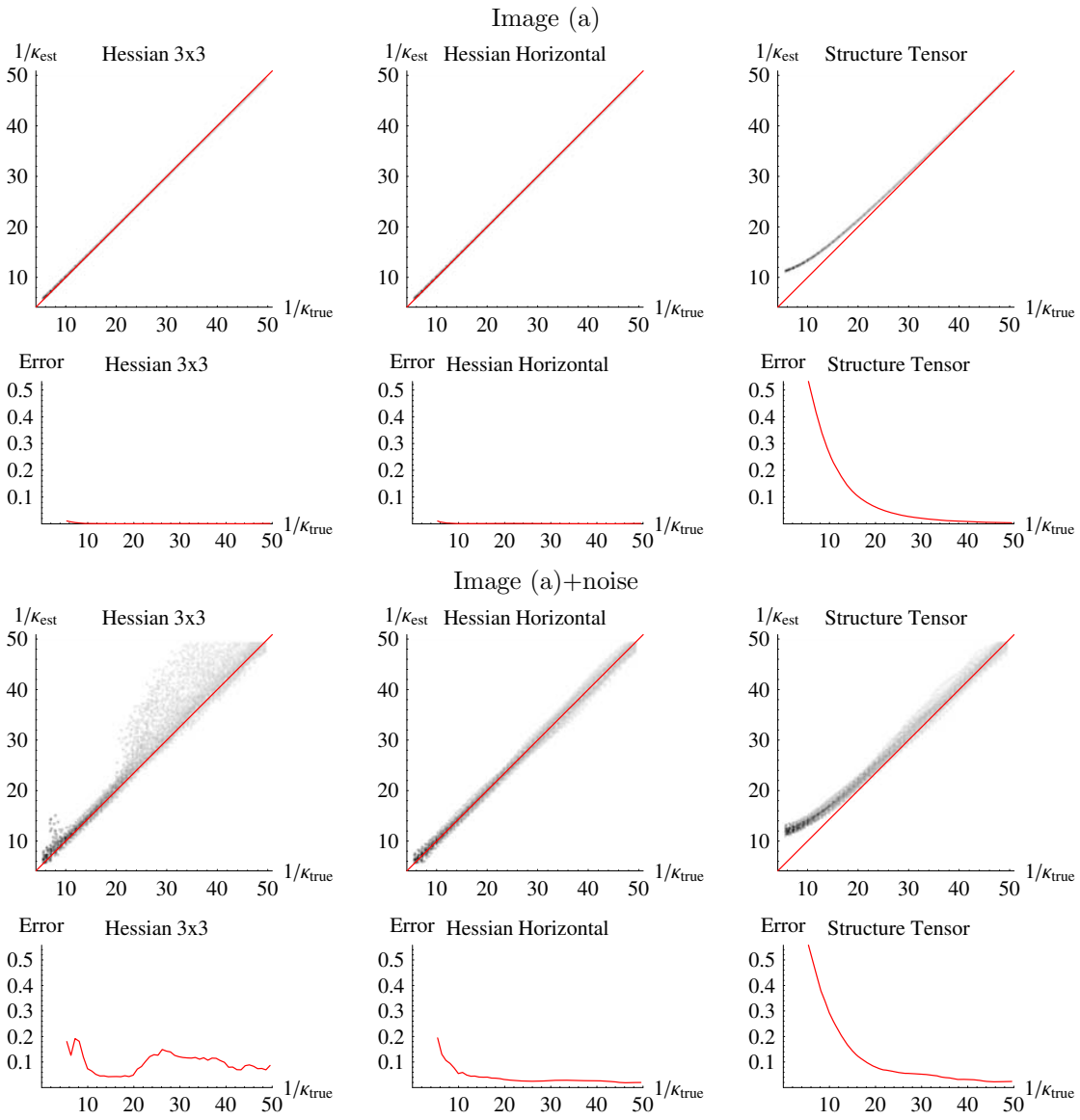


Figure 12: Curvature estimation results on image (a) and image (a)+noise for the three methods. For both images, the first row shows the density plot of true curvature κ_{true} against estimated curvature κ_{est}, and the second row shows the relative ℓ_2-error cf. (86) as function of the different curvatures.

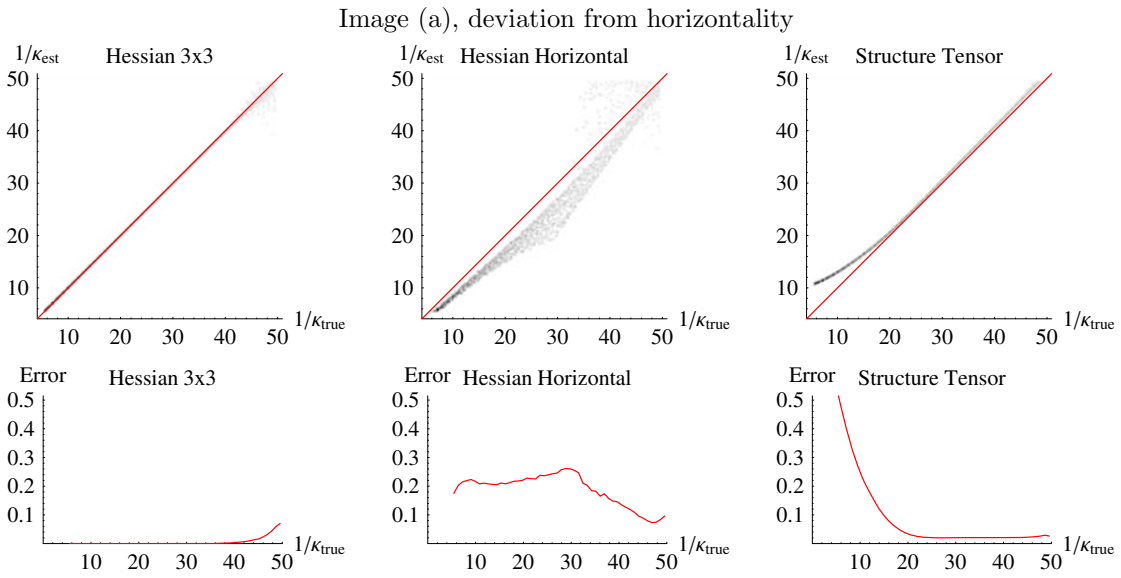


Figure 13: The effect of deviating from horizontality on image (a). For these graphs, the curvature estimation results are obtained at orientation $\theta_{\text{true}} + \pi/8$.

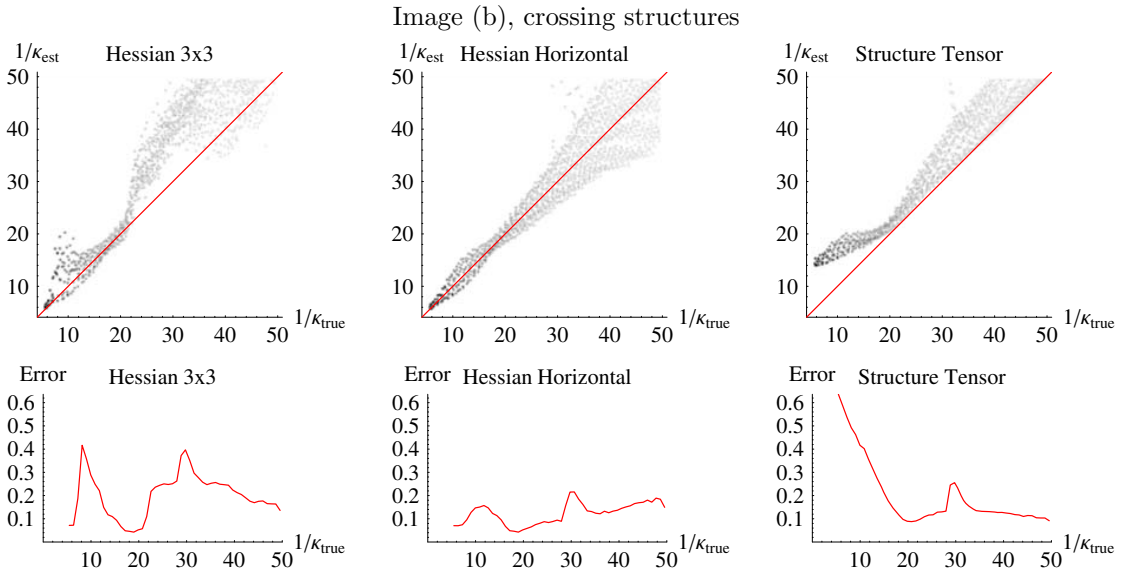


Figure 14: Shows curvature estimation results on image (b) (see Figure 11) with crossing elongated structures.

from 0 to 50). The image is converted to an orientation score by (45) using parameter values $s_\theta = \pi/32$, $k = 2$, $q = 8$, $t = 300$, and $s = 50$. For the Hessian we use $t_s = 9$, $\rho_s = 0.04$, and $\beta = 0.08$. We compare our method (with and without enforcing horizontality) with the structure tensor approach (see Section 5.2.2 and [35]), with parameters $t_s = 2.5$, $\rho_s = 7.5$, and $\beta = 0.08$. The parameters are chosen such that the total amount of Gaussian blurring is the same in both methods, i.e. the same neighborhood is taken into account.

For the test images (Figure 11) we do know the ground truth orientation θ_{true} and curvature κ_{true}. To evaluate the results, we take for each spatial position the curvature estimate κ_{est} for the known orientation and compare it to the ground truth curvature. In all experiments we display a density plot showing $1/\kappa_{\text{est}}$ (vertical) against the true curvature $1/\kappa_{\text{true}}$ (horizontal), and we show the error as function of $1/\kappa_{\text{true}}$ where the relative ℓ_2-error is defined as

$$\text{Error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\kappa_{\text{est},i} - \kappa_{\text{true}}}{\kappa_{\text{true}}}\right)^2} \tag{86}$$

where i enumerates over all N estimates $\kappa_{\text{est},i}$, $i \in \{1, 2, \ldots, N\}$, for which κ_{true} is the real curvature.

Figure 12 shows the results on an image with concentric circles. Clearly, the estimates of the structure tensor approach are too small over a wide range of the highest curvature values. This causes a quite large error for higher curvatures. Both of our approaches do not have this problem. Comparing the two Hessian-based approaches, we observe that the Hessian with enforced horizontality performs better. This is caused by the fact that on this test image the gauge frame $\{\partial_a, \partial_b, \partial_c\}$ is not always well defined on $W = |U_f|$. In images where lines do not occur in such regular sine-like patterns, it is expected that both methods perform comparable.

Figure 13 shows the effect of measuring the curvature at positions where the deviation from horizontality is nonzero. Instead of taking the curvatures at the true orientation θ_{true} we take the curvatures at $\theta_{\text{true}} + \pi/8$ to study the quality of the curvature estimation if we are not exactly at the right orientation. Clearly, since in this case the curves are not exactly horizontal, the Hessian approach that does not enforce horizontality works best.

Figure 14 shows the curvature estimation in an image with crossing curves. A slight decrease in performance can be seen since the orientation score transform can not entirely separate the responses of the different curves.

8.2 Coherence-enhancing Diffusion in Orientation Scores

In this section we compare the results of coherence enhancing diffusion in the orientation score (CED-OS) cf. (70) with results obtained by the normal coherence enhancing diffusion (CED) approach [39] where we use the LSAS numerical scheme [40].

The following parameters are used for the orientation score transformation (Section 3): $k = 2$, $q = 8$, $t = 1.6\left(\frac{X}{2\pi}\right)^2$ where X is the number of pixels of the image in x-direction, and $s = 200$. These parameters are chosen such that the reconstruction is visually indistinguishable from the original. All orientation scores have a periodicity of π. The original images all have a range of pixel values from 0 to 255. To ensure numerical stability, in the experiments where we use numerical scheme of Section 7.1 we use $\tau = 0.25$ and for the numerical scheme of Section 7.2 we use $\tau = 0.1$. Note that the resulting images we will show of CED-OS do not represent the evolving orientation score, but only the reconstructed image (i.e. after summation over all orientations).

Figure 15 shows the effect of CED-OS compared to CED on artificial images with crossing line structures. The upper image shows an additive superimposition of two images with concentric circles. Our method is able to preserve this structure, while CED can not. The same holds for the lower image with crossing straight lines, where it should be noted that our method leads to amplification of the crossings, which is because the lines in the original image are not superimposed linearly. In this experiment, no deviation from horizontality was taken into account, and the numerical scheme of Section 7.2 is used. The non-linear diffusion parameters for CED-OS are:

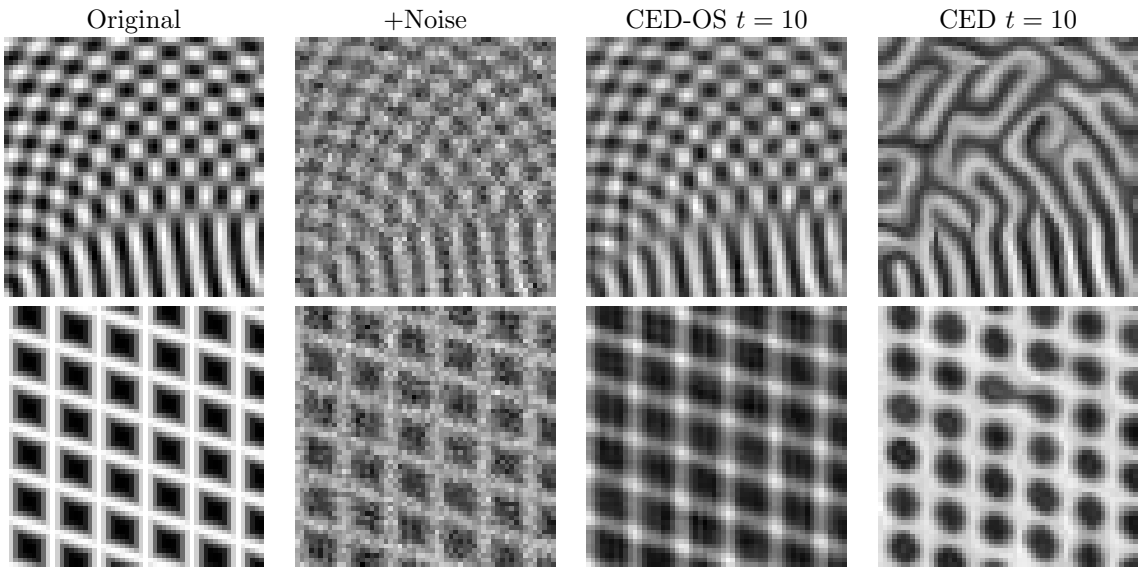


Figure 15: Shows the typical different behavior of CED-OS compared to CED. In CED-OS crossing structures are better preserved.

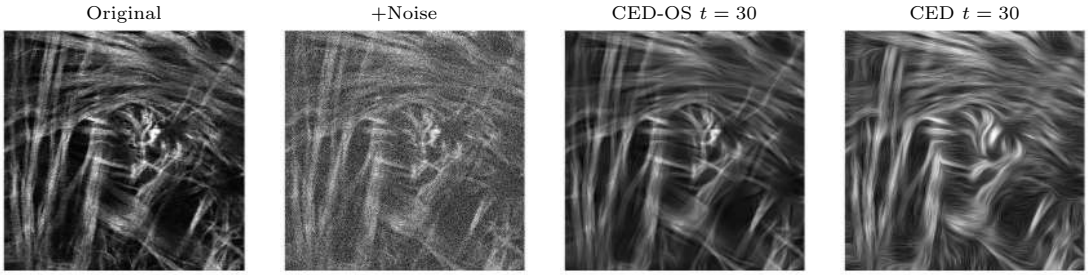


Figure 16: Result of CED-OS and CED on microscopy images of bone tissue. Additional Gaussian noise is added to verify the behaviour on noisy images.

$n_\theta = 32$, $t_s = 12$, $\rho_s = 0$, $\beta = 0.058$, and $c = 0.08$. The parameters that we used for CED are (see [39]): $\sigma = 1$, $\rho = 1$, $C = 1$, and $\alpha = 0.001$. The images have a size of 56×56 pixels.

Figure 1 at the beginning of the paper shows the results on an image of collagen fibres obtained using 2-photon microscopy. These kind of images are acquired in tissue engineering research, where the goal is to create artificial heart valves. All parameters during these experiments were set the same as the artificial images mentioned above except for CED parameter $\rho = 6$. The image size is 160×160 pixels.

Figures 16 and 17 show examples of the method on other microscopy data. The same parameters are used as above except for $t_s = 25$ in Figure 17. Clearly, the curve enhancement and noise suppression of the crossing curves is good in our method, while standard coherence enhancing diffusion tends to destruct crossings and create artificial oriented structures.

Figure 18 demonstrates the advantage of including curvature. Again, the same parameters and numerical scheme were used. Clearly, at $t = 30$ the circle with highest curvature is blurred if no curvature is taken into account. If curvature is taken into account, the diffusion process adheres much better to the spiral-shape of the circle in the orientation score. For the circles with smaller curvature, however, there is no noticeable difference.

The effect of including deviation from horizontality is especially visible if we significantly lower the number of orientation n_θ, because with a low number of orientations and without d_{H}, the elongated structures in the resulting images show strong biases towards the angles $l\frac{\pi}{n_\theta}$ with $l \in \mathbb{Z}$,

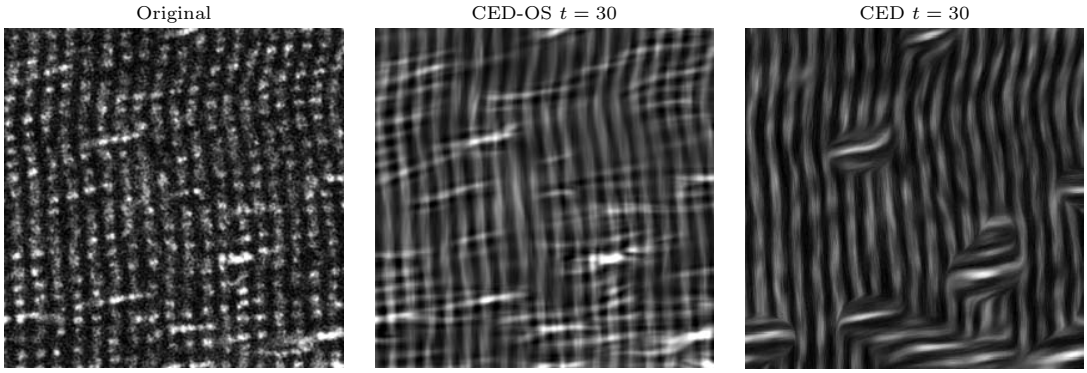


Figure 17: Result of CED-OS and CED on a microscopy image of a muscle cell.

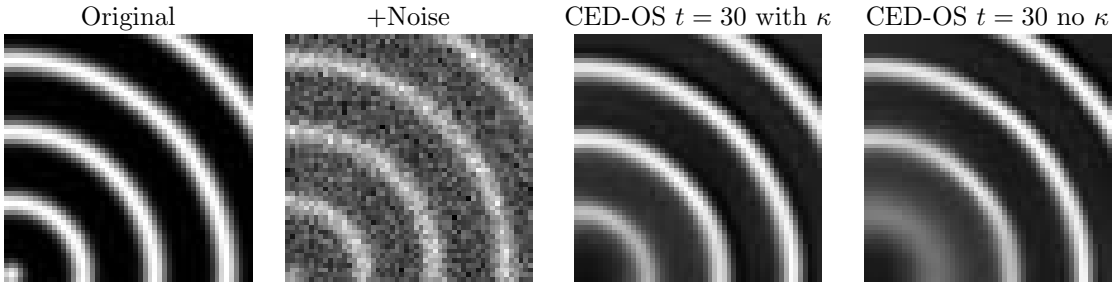


Figure 18: Shows the effect of including curvature on a noisy test image in CED-OS. At $t = 30$ the effect is visible: the circle with highest curvature is blurred if no curvature is taken into account.

while with d_{H} this problem does not occur. In Figure 19 this is illustrated. We set $n_\theta = 4$, $t_s = 5$, $\rho_s = 0$, $\beta = 0.058$, and $c = 0.1$ and use the numerical scheme of Section 7.1. Clearly we observe that 4 orientation is not enough without using d_{H}, since the orientations of the curves strongly bias towards the sampled orientations. This problem is solved if we include d_{H}, showing that even with only 4 orientations, we can appropriately handle crossing of two lines. Figure 20 shows the same effect on a microscopy image of bone tissue, where the parameters were set to $n_\theta = 4$, $t_s = 2$, $\rho_s = .5$, $\beta = 0.11$, and $c = 0.01$.

This means that including deviation from horizontality can make the algorithm much more efficient, since we can get good results with a very low number of orientations. Note, however that if one wants to handle crossings of more than 2 lines, or if the angle between the crossing lines is small, it is still necessary to increase the number of orientations.

9 Conclusions

In this paper we introduced non-linear diffusion on invertible orientation scores. Starting from a 2D image, we constructed a three-dimensional orientation score using rotated versions of a directional quadrature filter. Since an orientation score is a function on the Euclidean motion group $SE(2)$, we considered left-invariant diffusions adhering to the structure of this group. Then, we introduced a Gauge coordinate frame that is used to formulate an anisotropic diffusion that is aligned with exponential curves in $SE(2)$.

We showed how one can use normal Gaussian derivatives to calculate regularized derivatives in the orientation score. These Gaussian derivatives are used to estimate a tangent vector, which is tangent to the locally best fitting exponential curve, on each position in the orientation score. Using this tangent vector we calculate three features describing the local structure in the orientation score: curvature κ, deviation from horizontality d_{H}, and orientedness s. The features κ and d_{H}

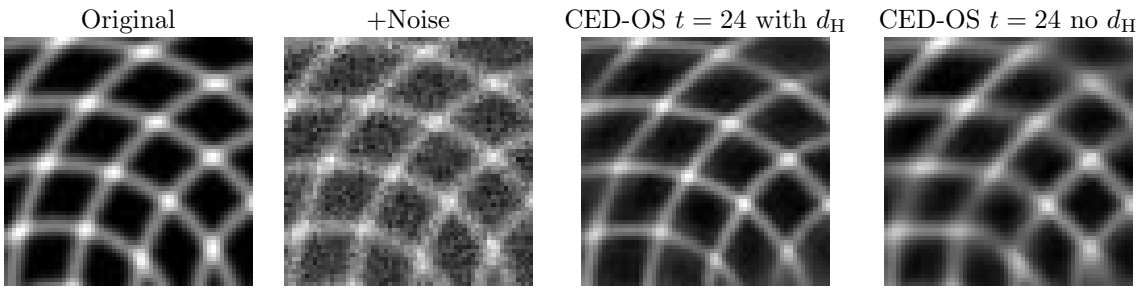


Figure 19: Shows the effect of including deviation from horizontality on a noisy test image in CED-OS. At $t = 24$ the result without deviation from horizontality clearly shows that the lines bias towards the sampled angles 0,$\pi/4$, $\pi/2$ and $3\pi/4$. If we include deviation from horizontality this problem does not occur, and even with $n_\theta = 4$ we are able to handle crossing correctly.

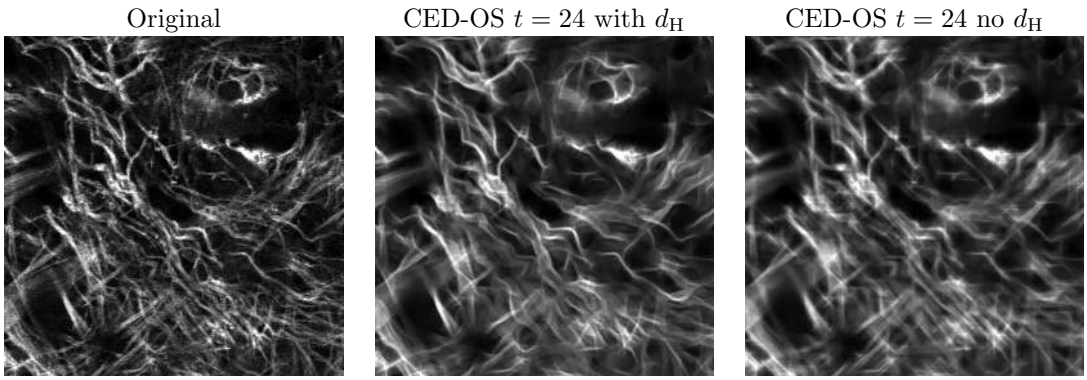


Figure 20: Shows the effect of including deviation from horizontality on a microscopy image of bone tissue. Clearly, with d_{H} we are able to handle crossings correctly even with $n_\theta = 4$.

uniquely determine the optimal Gauge frame. The non-linear diffusion is then aligned with this optimal Gauge frame and the orientedness s controls whether locally the diffusion is isotropic or anisotropic.

We proposed two explicit numerical schemes to apply the non-linear diffusion on the orientation score, and derived sharp stability bounds for both of these schemes. The simple explicit finite difference scheme is efficient but is not optimal concerning rotational invariance and oscillations at the Nyquist frequency. The left-invariant explicit finite difference scheme with spline interpolation, on the other hand, improves the left-invariance, but becomes very inefficient if deviation from horizontality is included.

The experimental results show that the curvature estimates are reliable and that we are indeed able to enhance elongated structures in images and that including curvature helps to enhance lines with large curvature. Especially at crossings our method renders a more natural result than coherence enhancing diffusion. Furthermore, deviation from horizontality d_{H} helps to get sharper results without orientational biases towards the sampled orientations, especially if we lower the number of orientations n_θ. Therefore, including d_{H} enables a large reduction in computations and storage. The diffusion shows the typical non-linear scale-space behavior when increasing time: blurring occurs, but the important features of images are preserved over a longer range of time.

Some issues could still be addressed in future work. The numerical schemes that are proposed should be improved concerning computational speed and quality. Furthermore, it would be interesting to use the same approach to other groups [10] such as the Heisenberg group (for the purpose of image enhancement via the Gabor domain), or the similitude group to use multi-scale *and* multi-orientation simultaneously to resolve the problem of selecting the appropriate scale. Finally, it is interesting to apply the same techniques in the 3D Euclidean motion group, to enhance

elongated structures in three-dimensional images. This would be especially useful for enhancing and segmenting fibers in High Angular Resolution Diffusion Imaging (HARDI) data.

References

[1] J.-P. Antoine and R. Murenzi. Two-dimensional directional wavelets and the scale-angle representation. *Signal Processing*, 52(3):241–272, 1996.

[2] J. August. *The Curve Indicator Random Field*. PhD thesis, Yale University, Dec. 2001.

[3] E. Candès and D. Donoho. Curvelets a surprisingly effective nonadaptive representation for objects with edges. In A. Cohen, C. Rabut, and L. Schumaker, editors, *Curve and Surface Fitting: Saint-Malo 1999, Nashville, TN*. Vanderbilt University Press, 1999.

[4] E. Candès and D. Donoho. Ridgelets: the key to high dimensional intermittency? *Philosophical Transactions of the Royal Society of London A*, 357:2495–2509, 1999.

[5] J. Chen, Y. Sato, and S. Tamura. Orientation space filtering for multiple orientation line segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):417–429, May 2000.

[6] G. Citti and A. Sarti. A cortical based model of perceptional completion in the roto-translation space. *Journal of Mathematical Imaging and Vision*, 24(3):307–326, 2006.

[7] G.-H. Cottet and L. Germain. Image processing through reaction combined with nonlinear diffusion. *Mathematics of Computation*, 61:659–667, 1993.

[8] M. Duits and R. Duits. A functional Hilbert space approach to the theory of wavelets. Technical Report RANA/CASA Report RANA-7-2004, Eindhoven University of Technology, Department of Mathematics, February 29 2004. URL: http://www.win.tue.nl/casa/research/casareports/2004.html.

[9] R. Duits. *Perceptual Organization in Image Analysis*. PhD thesis, Technische Universiteit Eindhoven, 2005. URL: http://www.bmi2.bmt.tue.nl/Image-Analysis/People/RDuits/THESISRDUITS.pdf.

[10] R. Duits and B. Burgeth. Scale spaces on Lie groups. pages 300–312.

[11] R. Duits, M. Duits, M. van Almsick, and B. ter Haar Romeny. Invertible orientation scores as an application of generalized wavelet theory. *Pattern Recognition and Image Analysis (PRIA)*, 17(1):42–75, 2007.

[12] R. Duits, L. Florack, J. de Graaf, and B. ter Haar Romeny. On the axioms of scale space theory. *Journal of Mathematical Imaging and Vision*, 20(3):267–298, 2004.

[13] R. Duits and E. Franken. Contour enhancement via linear and non-linear evolution equations on the Euclidean motion group. In preparation for submission to the Quarterly on Applied mathematics, AMS, 2008.

[14] R. Duits and E. Franken. Left-invariant stochastic evolution equations on $SE(2)$ and its applications to contour enhancement and contour completion via invertible orientation scores. *arXiv: 0711.0951v4*, 2007. URL: http://arxiv.org/abs/0711.0951. Also available as CASA report nr.35, 2007, Eindhoven University of Technology.

[15] R. Duits and M. van Almsick. The explicit solutions of linear left-invariant second order stochastic evolution equations on the 2D-Euclidean motion group. *AMS Quarterly of Applied Mathematics*, 2008. To appear in April.

[16] M. Felsberg, P.-E. Forssén, and H. Scharr. Channel smoothing: Efficient robust smoothing of low-level signal features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):209–222, 2006.

[17] L. M. J. Florack, B. M. t. Haar Romeny, J. J. Koenderink, and M. A. Viergever. Cartesian differential invariants in scale-space. *Journal of Mathematical Imaging and Vision*, 3(4):327–348, November 1993.

[18] E. Franken, R. Duits, and B. ter Haar Romeny. Nonlinear diffusion on the 2D Euclidean motion group. pages 461–472.

[19] E. M. Franken, R. Duits, and B. M. ter Haar Romeny. Curvature estimation for enhancement of crossing curves. In W. Niessen, C.-F. Westin, and M. Nielsen, editors, *Proceedings of the 8th IEEE Computer Society Workshop on Mathematical Methods in Biomedical Image Analysis, held in conjuction with the IEEE International Conference on Computer Vision (Rio de Janeiro, Brazil, October 14–20, 2007)*. Omnipress, 2007. Digital proceedings.

[20] E. M. Franken, P. Rongen, M. van Almsick, and B. M. ter Haar Romeny. Detection of electrophysiology catheters in noisy fluoroscopy images. pages 25–32.

[21] S. Gerschgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk 7*, pages 749–754, 1931.

[22] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision*. Kluwer Academic Publishers, 1995.

[23] F. Heitger and R. von der Heydt. A computational model of neural contour processing. In *Proceedings of the 4th International Conference on Computer Vision (Berlin, Germany, June 20–23, 1993)*, pages 32–40. Washington D.C.: IEEE Computer Society Press, 1993.

[24] S. N. Kalitzin, B. M. t. Haar Romeny, and M. A. Viergever. Invertible apertured orientation filters in image analysis. *International Journal of Computer Vision*, 31(2/3):145–158, April 1999.

[25] S. N. Kalitzin, B. M. ter Haar Romeny, and M. A. Viergever. Invertible orientation bundles on 2D scalar images. In B. M. ter Haar Romeny, L. Florack, J. Koenderink, and M. Viergever, editors, *Scale Space Theory in Computer Vision*, pages 77–88, 1997.

[26] R. Manniesing, M. A. Viergever, and W. Niessen. Vessel enhancing diffusion: A scale space representation of vessel structures. *Medical Image Analysis*, 10(6):815–825, 2006.

[27] D. Mumford. Elastica and computer vision. In C. L. Bajaj, editor, *Algebraic Geometry and its Applications*, pages 491–506. Springer-Verlag, New York, 1994.

[28] M. Nitzberg and T. Shiota. Nonlinear image filtering with edge and corner enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:826–833, 1992.

[29] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.

[30] H. Scharr. Diffusion-like reconstruction schemes from linear data models. In *Pattern Recognition : 28th DAGM Symposium, Berlin, Germany, September 12-14*, volume 4174 of *Lecture Notes in Computer Science*, pages 51–60. Springer-Verlag, 2006.

[31] J.-L. Starck, E. J. Candès, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, June 2002.

[32] D. Tuch, R. Weisskoff, J. Belliveau, and V. Wedeen. High angular resolution diffusion imaging of the human brain. In *Proc. of the 7th Annual Meeting of ISMRM, Philadelphia*, page 321, 1999.

[33] M. Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, Nov. 1999.

[34] M. A. van Almsick. *Context Models of Lines and Contours*. PhD thesis, Eindhoven University of Technology, Department of Biomedical Engineering, Eindhoven, The Netherlands, 2007.

[35] M. van Ginkel. *Image Analysis using Orientation Space based on Steerable Filters*. PhD thesis, Technische Universiteit Delft, The Netherlands, 2002.

[36] D. Walters. Selection of image primitives for general-purpose visual processing. *Computer Vision, Graphics, and Image Processing*, 37(2):261–298, 1987.

[37] J. Weickert and H. Scharr. A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. *Journal of Visual Communication and Image Representation*, pages 103–118, 2002.

[38] J. A. Weickert. *Anisotropic Diffusion in Image Processing*. ECMI Series. Teubner, Stuttgart, January 1998.

[39] J. A. Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2/3):111–127, April 1999.

[40] M. Welk, J. Weickert, and G. Steidl. From tensor-driven diffusion to anisotropic wavelet shrinkage. pages 391–403.

[41] L. R. Williams and D. W. Jacobs. Stochastic completion fields: a neural model of illusory contour shape and salience. *Neural Comput.*, 9(4):837–858, 1997.

[42] J. Zweck and L. R. Williams. Euclidean group invariant computation of stochastic completion fields using shiftable-twistable functions. *Journal of Mathematical Imaging and Vision*, 21(2):135–154, Sept. 2004.