Int J Comput Vis (2009) 85: 253–278
DOI 10.1007/s11263-009-0213-5

Crossing-Preserving Coherence-Enhancing Diffusion on Invertible Orientation Scores

Erik Franken · Remco Duits

Received: 31 January 2008 / Accepted: 13 January 2009 / Published online: 24 February 2009

Abstract Many image processing problems require the enhancement of *crossing* elongated structures. These problems cannot easily be solved by commonly used coherence-enhancing diffusion methods. Therefore, we propose a method for coherence-enhancing diffusion on the invertible orientation score of a 2D image. In an orientation score, the local orientation is represented by an additional third dimension, ensuring that crossing elongated structures are separated from each other. We consider orientation scores as functions on the Euclidean motion group, and use the group structure to apply left-invariant diffusion equations on orientation scores. 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 orientation confidence. These local features are used to adapt a nonlinear coherence-enhancing, crossing-preserving, diffusion equation on the orientation score. We propose two explicit finite-difference schemes to apply the nonlinear diffusion in the orientation score and provide a stability analysis. Experiments on both artificial and medical images show that preservation of crossings 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 reduce the number of sampled orientations while still preserving crossings.

Keywords Crossing elongated structures · Image enhancement · Orientation scores · Nonlinear diffusion · Coherence-enhancing diffusion · Euclidean motion group

1 Introduction

Image-processing problems often demand enhancement of elongated structures, such as lines, contours, and oriented textures, in noisy images. Many methods for enhancing elongated structures are based on *nonlinear anisotropic diffusion* equations on the image, i.e. adaptive diffusion processes where an unequal amount of diffusion is applied in different directions. This idea was pioneered by Nitzberg and Shiota (1992) and Cottet and Germain (1993). Later on, Weickert proposed edge- and coherence-enhancing diffusion filtering (Weickert 1998, 1999), which uses the structure tensor to steer the diffusion. Afterwards, various publications appeared inspired by these methods. For example, Manniesing and Niessen (2005), Manniesing et al. (2006) proposed to steer the diffusion using the vessel resemblance function, which is based on the Hessian instead of the structure tensor, and Tschumperlé (2006) proposed to include curvature into the diffusion process in order to improve enhancement of curved structures.

Many image processing problems, especially medical ones, require the algorithm to handle crossing and bifurcating line structures appropriately. This is for example important for High Angular Resolution Diffusion Imaging (HARDI) (Tuch et al. 1999), which is an extension to Diffusion Tensor Imaging (DTI), where a richer representation

E. Franken · R. Duits
Department of Biomedical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands

E. Franken
e-mail: e.m.franken@tue.nl

R. Duits (✉)
Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: r.duits@tue.nl

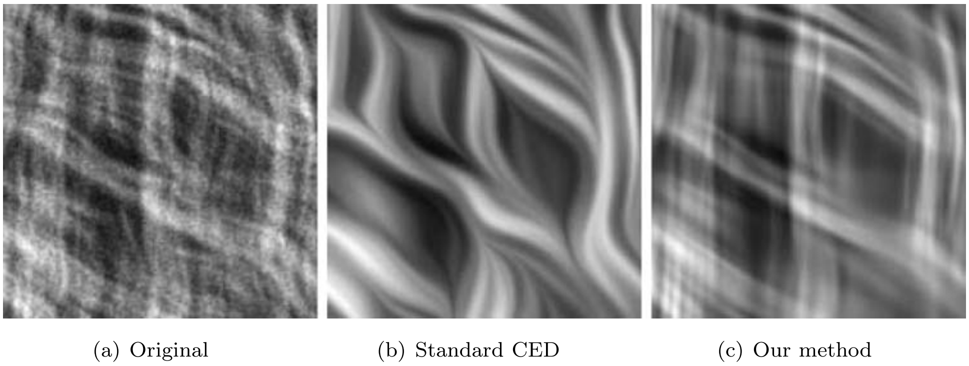


Fig. 1 Example of a microscopy image with crossing collagen fibers, showing the problems occurring at crossing. With the standard coherence-enhancing diffusion method, cf. Weickert (1998), abbreviated as CED, crossing elongated structures are not preserved. The method described in this paper is able to handle crossings appropriately. The parameters that are used to obtain these images are given in Sect. 8

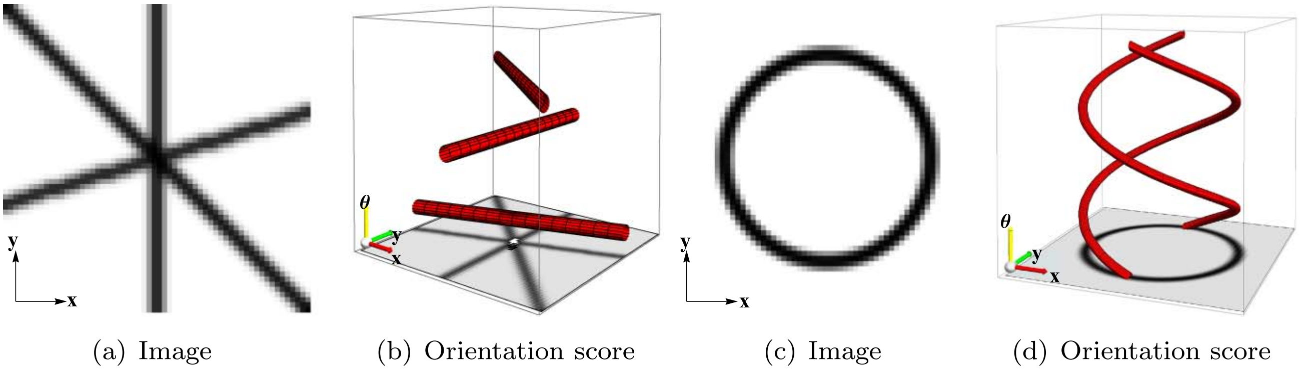


Fig. 2 Illustrations of orientation score construction from images. (**a**) Image with crossing lines. (**b**) Sketch of the corresponding three-dimensional orientation score. Since the lines have different orientations they are separated in the orientation score. Notice that the orientation dimension, which is displayed vertically, is actually 2π-periodic. (**c**) Image with a circle. (**d**) The corresponding response in the orientation score is confined to a spiral

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 (Franken et al. 2006), and MRI images of bifurcating blood vessels. At a position with a crossing, methods like coherence-enhancing diffusion do not enhance the elongated structures appropriately, as illustrated in Fig. 1(b). One would resolve this problem if one diffuses anisotropically in the directions of the different oriented structures, independent on the angles and number of elongated structures that cross. An interesting approach to handle crossing elongated structures is proposed by Scharr (2006). The method is similar to coherence-enhancing diffusion, but the gradient operator in both the structure tensor and the diffusion PDE is substituted by the second order jet operator. However, the drawback of this method is that the order of the PDE is 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", i.e. 2 curves that cross with a large angle.

In this paper we try to resolve the problem of crossing elongated structures by using *orientation scores* of images. An orientation score of a two-dimensional image is a three-dimensional function on spatial position (x, y) and orientation θ, i.e. a function on the *2D Euclidean motion group*, which provides an overview of all local orientations in the corresponding image. This is illustrated in Fig. 2. The concept of orientation scores first occurred in the field of perceptual grouping (Walters 1987; Heitger and von der Heydt 1993; Mumford 1994; Williams and Jacobs 1997; Zweck and Williams 2004) and is also applied for segmenting crossing structures (Chen et al. 2000; August 2001; Van Ginkel 2002) and estimating local orientation (Felsberg et al. 2006). The main advantage of using an orientation score is that crossing curves are separated (see Figs. 2(a) and (b)), since at the position where two lines cross they have a different orientation. This means that it is not needed anymore to take special care for crossings e.g. by explicitly detecting them. Figure 1(c) shows a typical result that can be obtained using orientation scores.

Kalitzin et al. (1997) proposed the *invertible* orientation score, which makes it possible to reconstruct the original image from the orientation score in a well-posed way. The invertibility of the transformation is essential in order to use orientation scores (OS) for image processing and enhancement, using the following chain of operations

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

An example of a simple orientation score processing operation is to take a certain power or taking nonlinear combinations of derivatives in the orientation score, leading to enhancement of lines in the reconstructed image (Kalitzin et al. 1997).

Duits et al. (2004, 2007), Duits (2005) developed a theory on the robustness of the invertible orientation score transformation, which is a multi-orientation wavelet transform. In wavelet literature, several authors have investigated wavelet transforms that are both *multi-orientation* and *multi-scale*. The *curvelet transform* (Candès and Donoho 1999a, 1999b; Starck et al. 2002) is a discrete multi-scale and multi-orientation wavelet transform. Antoine et al. propose continuous *directional wavelets* (Antoine and Murenzi 1996; Antoine et al. 1999) and establish the wavelet theory on the *similitude group*. The orientation score wavelet transform leaves out scaling, while it is still possible to obtain a stable image reconstruction.[1] In fact, an orientation score wavelet picks up all scales simultaneously, i.e. all scales within a reasonable range determined by the sampling and the size of the image. A practical advantage is the reduction of storage requirements.[2]

The notion of scale does come into play in orientation scores when considering processing operations *on* orientation scores. Instead of applying soft thresholds on the wavelet domain, which is common practice in wavelet literature, Duits (2005) explicitly employs the group structure in the wavelet domain by considering left-invariant convection-diffusion equations on the orientation score, which can be regarded as *scale spaces* on the Euclidean motion group. The linear left-invariant convection-diffusion equation correspond to stochastic processes for *enhancement* and *completion* of curves (Duits 2005; Van Almsick 2007). The major differences with the earlier work in this area, i.e. the approach for curve enhancement by Citti and Sarti (2006) and curve completion by Mumford (1994) and Zweck and Williams (2004), is that the invertibility of the orientation score is used. Furthermore, the linear evolution equations are solved by means of an $SE(2)$-*convolution* on the orientation score with the corresponding Green's function, which have been derived by Duits and van Almsick (2008), Duits and Franken (2007, 2009a).

This paper goes one step further and describes *nonlinear* diffusion equations on the two-dimensional Euclidean motion group to enhance crossing elongated structures. 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. We describe how to make the diffusion tensor adaptive to three features that describe the local structure in the orientation score: orientation confidence, curvature, and deviation from horizontality. These features are estimated using a Hessian matrix and they establish a *gauge frame* for the diffusion process at each position in the domain of the orientation score. The idea of a gauge frame in image processing (Florack et al. 1993) is to define, at each spatial position independently, a data-dependent orthogonal coordinate frame that spans the tangent space, such that the basis vectors are in alignment with some local feature of interest in the image.

This paper extends earlier work presented at conferences (Franken et al. 2007a, 2007b) with new results, particularly the gauge frame, stability bounds of the numerical schemes, and the concept of deviation from horizontality. In this paper we focus on the applied image processing and algorithmic part of our work. The essential mathematical theory will be presented in such a way that it should be understandable without too many prerequisites. In the companion papers (Duits and Franken 2009a, 2009b), which are written by the same authors, we focus on the mathematical theory and underlying differential geometry of both linear and nonlinear left-invariant diffusion equations on orientation scores.

1.1 Structure of the Paper

This paper starts with the introduction of orientation scores in details in Sect. 2. We will provide the necessary theory: group structure, left invariance, the tangent space at group elements, the left-invariant diffusion equation, and the gauge frame. Then, in Sect. 3 we describe how to construct invertible orientation scores in practice, and in Sect. 4 we describe how to operationalize regularized derivatives in orientation scores. All these results are used for the next step in Sect. 5: estimating local features in orientation scores. These features are used next in Sect. 6 to come to a nonlinear diffusion model, which corresponds to coherence-enhancing diffusion in the orientation score domain. Then, in Sect. 7 we describe

[1] The admissibility constraint in the wavelet theory on the similitude group (Antoine et al. 1999) can not be satisfied by proper wavelets in the Euclidean motion group. Therefore, we need a totally different admissibility constraint to construct proper wavelets in case of the Euclidean motion group (Duits et al. 2004), which will be briefly addressed in this paper.

[2] Another reason to leave out scaling is that the admissibility conditions for multi-scale multi-orientation wavelet transforms (Antoine et al. 1999) require the oriented wavelet to oscillate along its radial direction, which is an undesirable effect that will cause problems for the diffusion processes in the wavelet domain that we will aim at in this paper.

the numerical schemes that are used, including an analysis on stability. Finally, in Sect. 8 we show experimental results on the quality of curvature estimation and results of the coherence-enhancing diffusion in the orientation score, and we draw conclusions in Sect. 9.

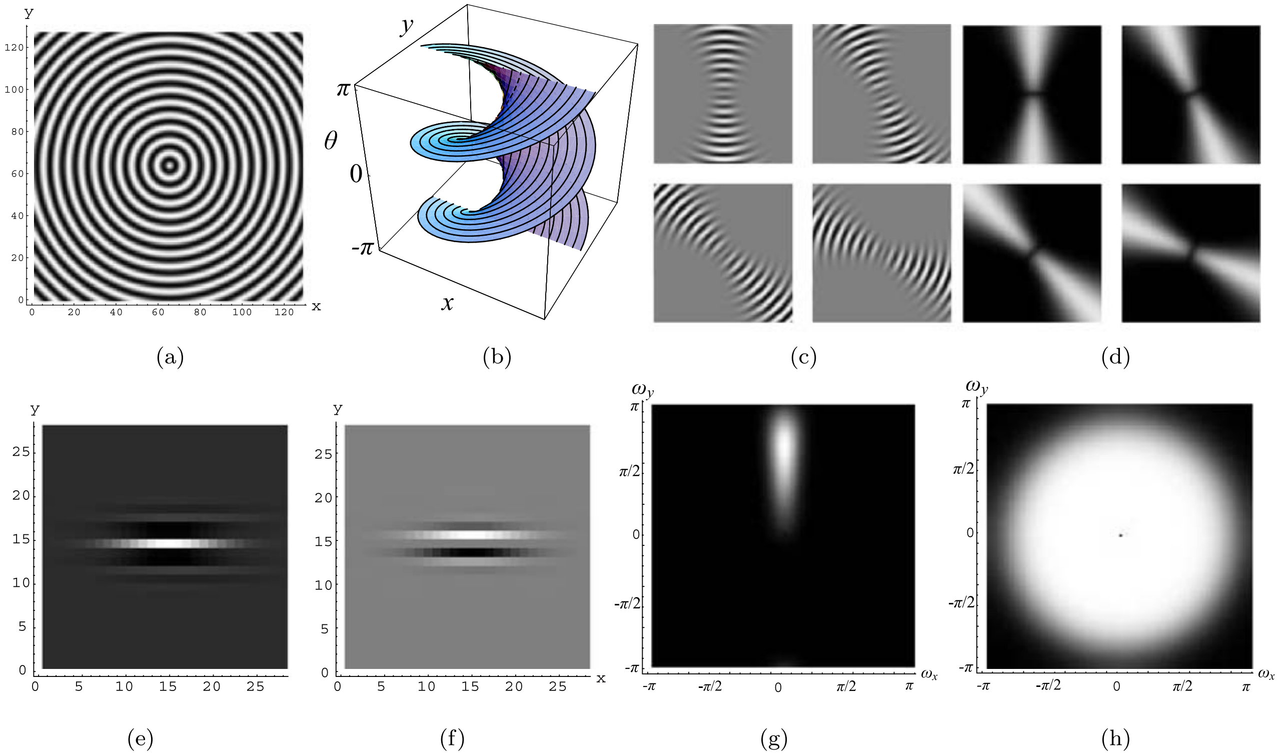
2 Theory of Orientation Scores

In this section we will describe the theory that is essential for the rest of the paper. We will introduce 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 understand and 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 (Duits and van Almsick 2008; Duits et al. 2007).

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 elongated 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* we add a 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, i.e. $\mathbb{T} = \{e^{i\theta} | \theta \in \mathbb{R}\}$. 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 Figs. 3(a, b).

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



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3 (**a**) Example of an image (128 × 128 pixels) with concentric circles. (**b**) Sketch of an iso-surface of the absolute value of the corresponding orientation score cf. (d), showing that 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 value $|U_f|$ yields a phase-invariant response displayed for 4 orientations. (**e**) Real part of the kernel cf. (45), used to generate (c) and (d), with $\theta = 0$ and parameter values $k = 2$, $q = 8$, $t = 400$, $s = 10$, $s_\theta = 64$. (**f**) Imaginary part. (**g**) Fourier transform of the kernel depicted in (e) and (f). (**h**) M_ψ (see (6)), which can be seen as the Fourier transform of the effective operation if no correction is applied, i.e. if reconstruction equation (11) is used

On the image domain, it is straightforward to develop image processing operations that are *translation invariant*, 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 (Florack et al. 1993). With invertible orientation 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

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

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, instead of 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

$$(\mathcal{U}_g f)(\mathbf{y}) = f(\mathbf{R}_\theta^{-1}(\mathbf{y} - \mathbf{x})), \quad f \in \mathbb{L}_2(\mathbb{R}^2), \tag{2}$$
$$\text{where} \quad g = (\mathbf{x}, \theta) \in SE(2), \quad \mathbf{y} \in \mathbb{R}^2, \quad \text{and}$$
$$(\mathcal{L}_g U)(h) = U(g^{-1}h), \quad U \in \mathbb{L}_2(SE(2)),\ g, h \in SE(2). \tag{3}$$

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

2.3 Invertible Orientation Scores

In this section we briefly discuss the theory of invertible orientation score transformations. A more mathematical treatment of invertible orientation scores can be found in the work by Duits et al. (2007), who developed a generalization of the wavelet theory.

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

$$U_f(\mathbf{x}, \theta) = (\overline{\psi^\theta} \star f)(\mathbf{x}) = \int_{\mathbb{R}^2} \overline{\psi(\mathbf{R}_\theta^{-1}(\mathbf{x}' - \mathbf{x}))}\, f(\mathbf{x}')\mathrm{d}\mathbf{x}', \tag{4}$$

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. The correlation kernel is related to the convolution kernel by $\check{\psi}(\mathbf{x}) = \psi(-\mathbf{x})$. 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. Note, that (4) can also be expressed using representation $\mathcal{U}$ cf. (2), making the group structure explicit, i.e. $U_f(\mathbf{x}, \theta) = (\mathcal{U}_g\psi, f)_{\mathbb{L}_2(\mathbb{R}^2)}$ where $(\cdot, \cdot)_{\mathbb{L}_2(\mathbb{R}^2)}$ denotes the $\mathbb{L}_2$-inner product.

The exact reconstruction equation accompanying (4) is given by

$$f = \mathcal{F}^{-1}\left[M_\psi^{-1}\mathcal{F}\left[\mathbf{x} \mapsto \int_0^{2\pi} (\check{\psi}^\theta \star U_f(\cdot, \theta))(\mathbf{x})\mathrm{d}\theta\right]\right], \tag{5}$$

where $\mathcal{F}$ denotes the unitary Fourier transform on $\mathbb{R}^2$ and $M_\psi : \mathbb{R}^2 \to \mathbb{R}^+$ is calculated by

$$M_\psi = \int_0^{2\pi} \overline{\mathcal{F}[\psi^\theta]}\mathcal{F}[\psi^\theta]\,\mathrm{d}\theta = \int_0^{2\pi} \left|\mathcal{F}[\psi^\theta]\right|^2 \mathrm{d}\theta. \tag{6}$$

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 (5). It can be verified that the construction/reconstruction equations (4) and (5) fulfill the following Plancherel's formula

$$\|f\|^2_{\mathbb{L}_2(\mathbb{R}^2)} = \|U_f\|^2_{M_\psi}, \tag{7}$$

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

$$\|U_f\|^2_{M_\psi} = \int_{\mathbb{R}}^{2} \int_0^{2\pi} |\mathcal{F}_{\mathbb{R}^2}[U_f](\boldsymbol{\omega}, \theta)|^2 \mathrm{d}\theta \frac{1}{M_\psi(\boldsymbol{\omega})} \mathrm{d}\boldsymbol{\omega}, \tag{8}$$

where $\mathcal{F}_{\mathbb{R}^2}$ denotes the Fourier transform on the spatial coordinates only. Note that we have $\mathbb{L}_2$-norm preservation, i.e. $\|f\|^2_{\mathbb{L}_2(\mathbb{R}^2)} = \|U_f\|^2_{M_\psi} = \|U_f\|^2_{\mathbb{L}_2(SE(2))}$, 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. This is a natural choice for bandlimited images: because of finite sampling we can assume images to be bandlimited anyway, where the bandwidth coincides with the well-known Nyquist frequency.

Invertible orientation scores were first proposed by Kalitzin et al. (1999). They propose a specific choice for an oriented wavelet which falls in class of proper wavelets (Duits et al. 2007). This kernel, however, has practical disadvantages, which are explained in details in Duits (2005, p. 141, Sect. 4.6.2). The oriented wavelets we use in this paper are also proper wavelets, but are of an essentially different type (Duits 2005, p. 141, Sect. 4.6.1). The idea is that if $M_\psi \approx 1$ we can approximate the reconstruction by

$$\hat{f}(\mathbf{x}) \approx \int_0^{2\pi} (\check{\psi}^\theta \star U_f(\cdot, \theta))(\mathbf{x})\, \mathrm{d}\theta. \tag{9}$$

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

$$M_\psi = \int_0^{2\pi} \left|\mathcal{F}[\psi^\theta]\right|^2 \mathrm{d}\theta \approx \int_0^{2\pi} \left|\mathcal{F}[\psi^\theta]\right| \mathrm{d}\theta, \tag{10}$$

where the reconstruction formula simplifies to integration over the orientation dimension

$$\hat{f}(\mathbf{x}) \approx \int_0^{2\pi} U_f(\mathbf{x}, \theta)\, \mathrm{d}\theta, \tag{11}$$

which has the practical advantage that the computation time of the reconstruction procedure is significantly reduced. Duits et al. (2007) first proposed proper wavelets that fulfill the latter reconstruction equation. We will use this type of kernel and introduce one particular choice in more detail in Sect. 3.

2.4 Left-invariant Operations in Orientation Scores

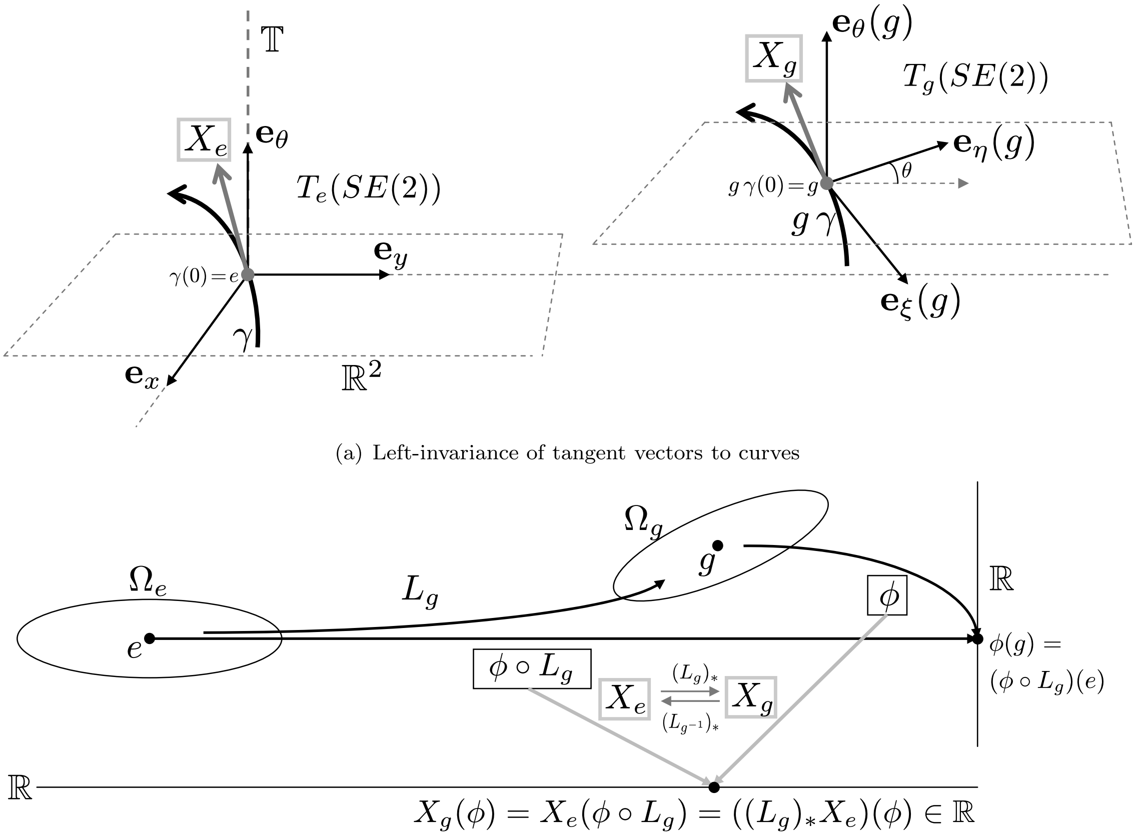
We want to perform operations on orientation scores of images, in order to enhance the corresponding image. Analogously to the fact that the Gabor transform of a signal makes it easier to perform operations that manipulate local frequencies, the orientation score transform makes it easier to apply anisotropic 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 scores exist, but not all of them are sensible. As already mentioned in Sect. 2.1, we want to ensure that the effective operation on the image is Euclidean invariant. It can be shown (Duits 2005, Theorem 2.1, p. 153) that this is only the case if the operator $\Phi : \mathbb{L}_2(SE(2)) \to \mathbb{L}_2(SE(2))$ that is applied to the orientation score is *left-invariant*. An operator Φ is left-invariant iff $\mathcal{L}_g \Phi U = \Phi \mathcal{L}_g U$, for all $g \in G$ and for all $U \in \mathbb{L}_2(SE(2))$, where $\mathcal{L}_g$ is defined in (3). In other words, left-invariance means that the operator commutes with the left-regular representation $\mathcal{L}$ of the group.

Figure 4 illustrates left-invariance in two different ways: for tangent vectors and for differential operators. In Fig. 4(a), it is shown how a tangent vector $X_e = c^\xi \mathbf{e}_x + c^\eta \mathbf{e}_y + c^\theta \mathbf{e}_\theta \in T_e(SE(2))$, 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 $g\gamma$ at position g. Here, the curve $g\gamma$ is the original curve γ that is left-multiplied with $g = (\mathbf{x}, \theta) \in SE(2)$, so that the curve is translated over $\mathbf{x}$ and rotated over θ. The tangent space at the unity element is spanned by $T_e(G) = \mathrm{span}\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_\theta\}$. We transport 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

$$\begin{aligned}&\{\mathbf{e}_\xi(g), \mathbf{e}_\eta(g), \mathbf{e}_\theta(g)\} \\ &\quad = \{\cos\theta\, \mathbf{e}_x + \sin\theta\, \mathbf{e}_y, -\sin\theta\, \mathbf{e}_x + \cos\theta\, \mathbf{e}_y, \mathbf{e}_\theta\}.\end{aligned} \tag{12}$$

This basis for tangent vectors in g has the property that $X_g = c^\xi \mathbf{e}_\xi + c^\eta \mathbf{e}_\eta + c^\theta \mathbf{e}_\theta$ for all g, so the vector components $(c^\xi, c^\eta, c^\theta)$ 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 $g = (\mathbf{x}, \theta)$.



(a) Left-invariance of tangent vectors to curves

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

Fig. 4 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. $X_g = c^\theta \partial_\theta|_g + c^\xi \partial_\xi|_g + c^\eta \partial_\eta|_g$ 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 the text (Sect. 2.4) for details

In Fig. 4(b) it is shown how X_e and X_g can also be viewed as differential operators, acting on a function $U : SE(2) \to \mathbb{R}$, e.g. an orientation score. In the figure, the codomain of U is the vertical $\mathbb{R}$-axis. X_g can be viewed as an operator that calculates the derivative of U at g, i.e.

$$\begin{aligned} X_g(U) &= (c^\xi \partial_\xi|_g + c^\eta \partial_\eta|_g + c^\theta \partial_\theta|_g)U \\ &= (c^\xi(\cos\theta\,\partial_x + \sin\theta\,\partial_y) \\ &\qquad + c^\eta(-\sin\theta\,\partial_x + \cos\theta\,\partial_y) + c^\theta \partial_\theta)U, \end{aligned} \tag{13}$$

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 U over g, i.e. $U \circ L_g$, such that the original neighborhood Ω_g is shifted to neighborhood Ω_e. That is,

$$X_g(U) = X_e(U \circ L_g) = (c^\xi \partial_x + c^\eta \partial_y + c^\theta \partial_\theta)(U \circ L_g), \tag{14}$$

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.

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. Furthermore, we need to define inner products and norms on these spaces.

A vector in the tangent space $T_g(SE(2))$ is denoted in a basis-independent way by $c^\xi \partial_\xi|_g + c^\eta \partial_\eta|_g + c^\theta \partial_\theta|_g \in T_g(SE(2))$, with $(c^\xi, c^\eta, c^\theta) \in \mathbb{R}^3$. We will consistently use the left-invariant basis $\{\partial_\xi, \partial_\eta, \partial_\theta\}$ further on, i.e. we will work with the vector components $\mathbf{c} = (c^\xi, c^\eta, c^\theta)^{\mathrm{T}}$ expressed in this basis. The physical dimensions of the vector components are (length, length, 1) respectively.

Similarly, a *covector* is denoted by $c_\xi \mathrm{d}\xi|_g + c_\eta \mathrm{d}\eta|_g + c_\theta \mathrm{d}\theta|_g \in T_g^*(SE(2))$, where $\mathrm{d}\xi$, $\mathrm{d}\eta$, and $\mathrm{d}\theta$ 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

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

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

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

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

Note that 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 μ with physical dimension 1/length and define as inner product

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

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

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

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

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

Consequently, the inner product between two covectors is given by

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

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

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

For a theoretical motivation of this left-invariant (and not right-invariant) inner product on each tangent space $T_g(SE(2))$ see (Duits and Franken 2009b).

2.6 Left-invariant Derivatives

As mentioned in Sect. 2.4, $\{\partial_\xi, \partial_\eta, \partial_\theta\}$ are left-invariant differential operators, which are appropriate to use instead of the set $\{\partial_x, \partial_y, \partial_\theta\}$. They 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 $\{\partial_\theta, \partial_\xi, \partial_\eta\}$ 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

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

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

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

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* at $t \in \mathbb{R}$ iff

$$\theta(t) = \angle \left(\frac{\mathrm{d}x(t)}{\mathrm{d}t}, \frac{\mathrm{d}y(t)}{\mathrm{d}t} \right), \tag{25}$$

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, for all tangent vectors $\mathbf{c} = (c^\xi, c^\eta, c^\theta)^{\mathrm{T}}$ over the curve we know that $c^\eta = 0$, i.e. an equivalent formulation for horizontality of q is

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

On a horizontal curve, ∂_ξ is always the spatial derivative tangent to the curve and ∂_η is always orthogonal to the curve.

A curve $(x(t), y(t))$ in an image would render a perfectly horizontal response if the at each spatial position along the curve the response in the orientation dimension would be a δ-spike, i.e. $U_f(x(t), y(t), \theta) = \delta(\arg(\frac{\mathrm{d}x(t)}{\mathrm{d}t} + i\,\frac{\mathrm{d}y(t)}{\mathrm{d}t}) - \theta)$. In this case we can say that the curve renders an exactly *horizontal response* in the orientation score. However, by construction, cf. Sect. 2.3, the response in the orientation score caused by a curve in the corresponding image always exhibits some uncertainty in orientation. Therefore, in practice, the orientation score transformation cf. Sect. 2.3 renders orientation scores in which image curves render *approximately* horizontal responses.

2.8 Exponential Curves

An *exponential curve* is a curve $\gamma_{\mathbf{c}} : \mathbb{R} \to SE(2)$ for which the components of the tangent vector expressed in the left-invariant basis $\{\mathbf{e}_\xi, \mathbf{e}_\eta, \mathbf{e}_\theta\}$ are constant over the entire parametrization (if necessary after reparametrization of t), i.e.

$$\frac{\mathrm{d}}{\mathrm{d}t}\gamma_{\mathbf{c}}(t) = c^\xi \mathbf{e}_\xi(\gamma_{\mathbf{c}}(t)) + c^\eta \mathbf{e}_\eta(\gamma_{\mathbf{c}}(t)) + c^\theta \mathbf{e}_\theta(\gamma_{\mathbf{c}}(t)), \tag{27}$$

for all $t \in \mathbb{R}$. These curves are analogous to *straight* lines in $\mathbb{R}^n$, which also have a constant tangent vector relative to the Cartesian basis $\{\mathbf{e}_x, \mathbf{e}_y\}$.

An exponential curve is obtained by an exponential mapping of Lie algebra elements, which explains why they are called "exponential". That is, an exponential curve passing through the identity element $e \in SE(2)$ at $t = 0$ can be written as

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

and an exponential curve passing through point $g_0 \in SE(2)$ can be obtained by left-multiplication with g_0, i.e. $g_0\,\gamma_{\mathbf{c}}(t)$. For the case $c^\theta \neq 0$ this results in

$$g_0\,\gamma_{\mathbf{c}}(t) = \begin{pmatrix} x_0 + \frac{c^\xi}{c^\theta}\,\mu(tc^\theta, \theta_0) + \frac{c^\eta}{c^\theta}\,\nu(tc^\theta, \theta_0) \\ y_0 + \frac{c^\xi}{c^\theta}\,\nu(tc^\theta, \theta_0) + \frac{c^\eta}{c^\theta}\,\mu(tc^\theta, \theta_0) \\ tc^\theta + \theta_0 \end{pmatrix}, \tag{29}$$

$$\begin{aligned}\text{with}\quad &\mu(tc^\theta, \theta_0) = \sin(tc^\theta + \theta_0) - \sin(\theta_0),\\ &\nu(tc^\theta, \theta_0) = \cos(tc^\theta + \theta_0) - \cos(\theta_0).\end{aligned}$$

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

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

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

Horizontal exponential curves are exponential curves that are horizontal cf. (25) for all $t \in \mathbb{R}$. They form the subset of all exponential curves with $c^\eta = 0$, see Fig. 5.

2.9 Curvature and Deviation from Horizontality

From the tangent vector $\mathbf{c} = (c^\xi, c^\eta, c^\theta)$ of an exponential curve we define two features with a clear geometrical interpretation: *curvature* and *deviation from horizontality*, see Fig. 6.

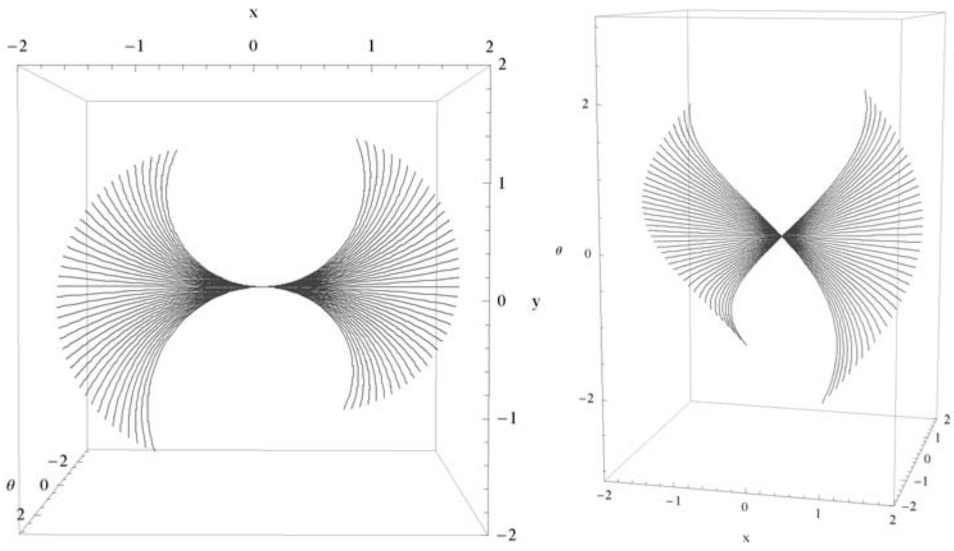


Fig. 5 Horizontal exponential curves in $SE(2)$ for a range of different curvature values, shown from two different perspectives. The *left-sided image* shows that these curves are circular arc when projected onto the spatial plane

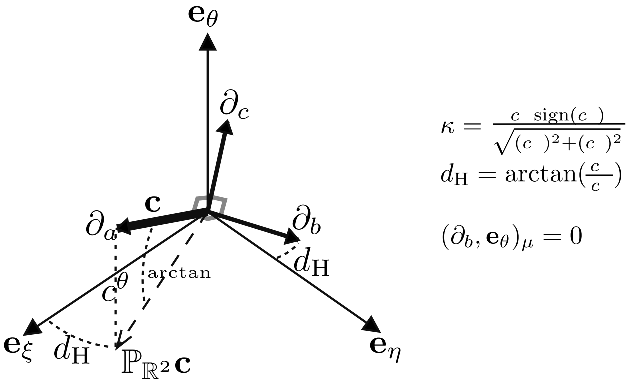

$\mathbf{e}_\theta$
∂_c
$\kappa = \frac{c \;\; \mathrm{sign}(c \;\;)}{\sqrt{(c \;\;)^2+(c \;\;)^2}}$
$d_{\mathrm{H}} = \arctan(\frac{c}{c})$
$(\partial_b, \mathbf{e}_\theta)_\mu = 0$
$\mathbf{c}$
∂_a
∂_b
d_{H}
arctan
c^θ
$\mathbf{e}_\xi$
d_{H}
$\mathbb{P}_{\mathbb{R}^2}\mathbf{c}$
$\mathbf{e}_\eta$

Fig. 6 Illustration of curvature and deviation from horizontality (Sect. 2.9) and of the gauge frame (Sect. 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 $\|\partial_c\|_\beta = \|\partial_a\|_\beta = \|\partial_b\|_\beta = \mu$

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

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

where t must be the arc length parametrization in the spatial plane, that is $\|\frac{\mathrm{d}}{\mathrm{d}t}(\mathbb{P}_{\mathbb{R}^2}\gamma(t))\| = 1$ for all $t \in \mathbb{R}$ should hold. The signed norm of the curvature vector is

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

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 Fig. 6. 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 $\gamma_{\mathbf{c}}g_0(t)|_{c^\eta=0}$. For a non-horizontal exponential curve, we also need the *deviation from horizontality* d_{H} given by

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

i.e. d_{H} is the angle that the projection of the exponential curve onto $\mathbb{R}^2$ makes with the horizontal direction $\mathbf{e}_\xi$, see Fig. 6.

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

The diffusion equation on $\mathbb{R}^n$, which is commonly used in image processing, is given by

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

where f is the input image, u the evolving image, the ∇ operator is defined 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 $SE(2)$.

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

where $g \in SE(2)$ and $t \geq 0$, ∇ 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 $U(\cdot,t) = e^{t(\nabla\cdot\mathbf{D}\nabla)}U_f$.

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

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

The equation is "μ-isoptropic" since $\|\partial_\xi\|_\mu = \|\partial_\eta\|_\mu = \mu^2\|\partial_\theta\|_\mu = \mu$. We use this convention for μ-isotropy because it ensures that the result of a β-isotropic diffusion process at time t can be identified one-to-one with the standard isotropic diffusion process on the image (cf. (35)) at the same time t on the image. This makes it more convenient to set parameter μ. In the next section we will discuss useful choices for the diffusion tensor $\mathbf{D}$ for *anisotropic* diffusion on $SE(2)$.

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

The idea of gauge coordinates in image processing (Florack et al. 1993) is to define a data-dependent orthogonal coordinate frame, for the tangent space $T_{\mathbf{x}}(\mathbb{R}^N)$ at each position $\mathbf{x}$, such that 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 to orientation scores. We want to establish a μ-*orthogonal* gauge frame $\{\partial_a, \partial_b, \partial_c\}$, where μ-orthogonality is defined as

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

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)

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

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

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

In terms of κ and d_{H}, as defined in Sect. 2.9, this renders the following gauge frame

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

where

$$\tilde{\mathbf{R}}_{d_{\mathrm{H}}} = \begin{pmatrix} \cos d_{\mathrm{H}} & -\sin d_{\mathrm{H}} & 0 \\ \sin d_{\mathrm{H}} & \cos d_{\mathrm{H}} & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{Q}_{\kappa,\mu} = \begin{pmatrix} \frac{\mu}{\sqrt{\mu^2+\kappa^2}} & 0 & \frac{\kappa}{\sqrt{\mu^2+\kappa^2}} \\ 0 & 1 & 0 \\ \frac{-\kappa}{\sqrt{\mu^2+\kappa^2}} & 0 & \frac{\mu}{\sqrt{\mu^2+\kappa^2}} \end{pmatrix}. \tag{42}$$

This gauge frame is illustrated in Fig. 6. Note that it actually involves two rotations, since $\mathbf{Q}_{\kappa,\mu}$ 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 diffusion equation without mixed terms

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

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

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

This diffusion equation will be used in the rest of the paper, where we will make D_{bb} dependent on the local differential structure of U. Figure 7 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 our algorithm. In this section we describe the practical design of an invertible orientation score transformation, based on (Duits 2005, Sect. 4.6.2).

The invertibility conditions described in Sect. 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. The orientation score should be constructed for a finite number (N_{o}) of orientations. This requirement is obvious from an implementational point of view.
2. It should be possible to achieve perfect reconstruction by summing all orientations, i.e. $f(\mathbf{x}) = \sum_{i=0}^{N_{\mathrm{o}}-1} U_f(\mathbf{x}, i s_\theta)$, where s_θ is the orientation sample distance in radians, i.e. $s_\theta = \frac{2\pi}{N_{\mathrm{o}}}$ if the periodicity of the orientation score 2π.
3. The kernel should be strongly directional and minimize the uncertainty principle on the Euclidean motion group. These requirements are fulfilled if the kernel is a compactly supported convex cone in the Fourier domain (Antoine and Murenzi 1996; Antoine et al. 1999).
4. The kernel should be polar separable 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 $\boldsymbol{\omega} = (\omega_x, \omega_y) = (\rho \cos\varphi, \rho \sin\varphi)$.

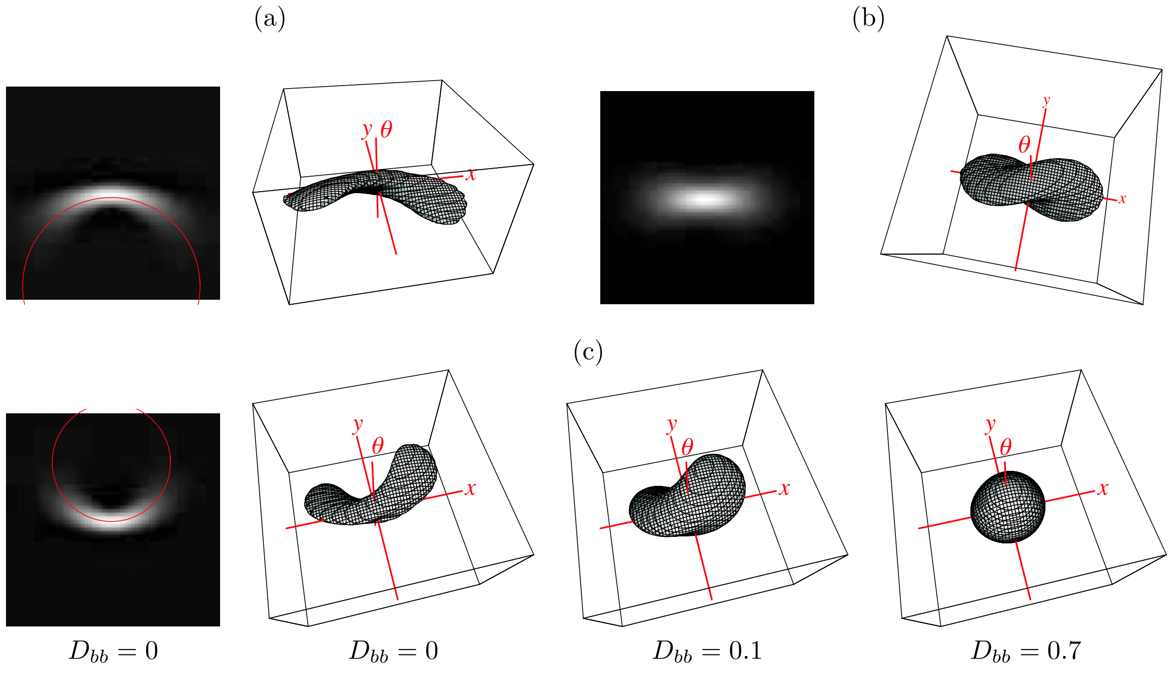


Fig. 7 Illustrations of Green's functions for different parameter values, obtained using an explicit iterative numerical scheme (Sect. 7.2) with end time $t = 70$. (**a**) Shows the effect of nonzero κ. Parameters $\kappa = -0.04$, $D_{bb} = 0$, $D_{aa} = 1$, and $\mu = 1$. *Left*: Greens functions in 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_{bb}. Parameters $D_{bb} = 0.003$, $\kappa = 0$, $D_{aa} = 1$, and $\mu = 1$. (**c**) Shows the effect of varying D_{bb}. Parameters $\kappa = 0.06$, $D_{aa} = 1$, and $\mu = 0.1$. As D_{bb} increases from 0 to 1, the resulting Green's function becomes more and more isotropic

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 (Granlund and Knutsson 1995). More details can be found in Sect. 3.1.

These requirements are similar to the requirements in Van Ginkel (2002), except for the reconstruction.

Based on these requirements we propose the following kernel

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

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

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

The 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.

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

The function G_s in (45) is a Gaussian kernel with scale s, which ensures spatial locality. Figure 3 shows an example of this orientation score transformation.

3.1 Quadrature Property and Hilbert Transform

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

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

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

$$\psi_{\text{odd}} = \mathcal{H}^{\mathbf{e}_y}_{\mathbb{R}^2}[\psi_{\text{even}}], \qquad (49)$$

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

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

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.

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

The complex-valued orientation score is simply found by

$$\begin{aligned}U_f&=(\psi^\theta_{\mathrm{even}}\star f)-i(\psi^\theta_{\mathrm{odd}}\star f))\\&=\mathrm{Re}[U_f](\mathbf{x},\theta)-i\mathcal{H}_{SE(2)}[\mathrm{Re}[U_f]].\end{aligned}\tag{53}$$

Moreover since $\psi^\theta_{\mathrm{even}}=\psi^{\theta+\pi}_{\mathrm{even}}$ and $\psi^\theta_{\mathrm{odd}}=-\psi^{\theta+\pi}_{\mathrm{odd}}$, we have the relation $U_f(\mathbf{x},\theta+\pi)=\overline{U_f(\mathbf{x},\theta)}$.

4 Regularized Derivatives

In Sect. 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. To ensure that the left-invariance is preserved, we use left-invariant diffusion described in Sect. 2.10 for the regularization. Thus, 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 the left-invariant diffusion. Note that 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 (Duits and Franken 2009a). Here, for simplicity and computational efficiency we restrict ourselves to the μ-isotropic of (37), which can be written as

$$\begin{aligned}\partial_t W&=\left(\left(\partial_\xi^2+\partial_\eta^2\right)+\mu^2\partial_\theta^2\right)W\\&=\left(\left(\partial_x^2+\partial_y^2\right)+\mu^2\partial_\theta^2\right)W.\end{aligned}\tag{54}$$

Since the operators ∂_x, ∂_y, and ∂_θ, 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

$$G_{t_s,t_o}(x,y,\theta)=\frac{1}{8\sqrt{\pi^3t_s^2t_o}}e^{-\frac{x^2+y^2}{4t_s}-\frac{\theta^2}{4t_o}},\tag{55}$$

with $t_s=t$ and $t_o=\mu^2t$. 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

$$\partial_x^i\partial_y^j\partial_z^k e^{t(\partial_x^2+\partial_y^2+\partial_z^2)}f=\partial_x^i e^{t\,\partial_x^2}\partial_y^j e^{t\,\partial_y^2}\partial_z^k e^{t\,\partial_z^2}f.\tag{56}$$

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

$$\begin{aligned}&\partial_\xi^i\,\partial_\eta^j\,\partial_\theta^k\,e^{t_o\partial_\theta^2+t_s(\partial_\xi^2+\partial_\eta^2)}=\partial_\xi^i\,\partial_\eta^j\,e^{t_s(\partial_x^2+\partial_y^2)}\,\partial_\theta^k\,e^{t_o\partial_\theta^2},\\&\partial_\theta^k\,\partial_\xi^i\,\partial_\eta^j\,e^{t_o\partial_\theta^2+t_s(\partial_\xi^2+\partial_\eta^2)}\neq\partial_\theta^k\,e^{t_o\partial_\theta^2}\,\partial_\xi^i\,\partial_\eta^j\,e^{t_s(\partial_x^2+\partial_y^2)},\end{aligned}\tag{57}$$

we conclude that we always should ensure a certain *ordering* of the derivative operators. That is, 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 (13). 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 scalar-valued features at each position $g\in SE(2)$ to which the diffusion is adapted. The two features *curvature* $\kappa(g)$ and *deviation from horizontality* $d_{\mathrm{H}}(g)$ have already been introduced in Sect. 2.9. The third important feature is *orientation confidence* $s(g)$: a scalar number indicating the confidence that locally an

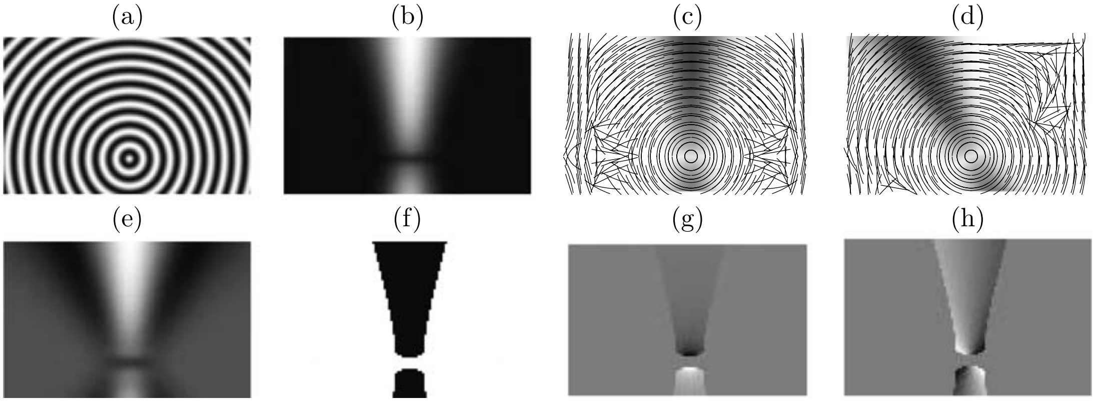


Fig. 8 Example of feature estimation in an orientation score. (**a**) Input image f. (**b**) $|U_f|$ at $\theta = 0$, used for feature estimation. (**c**) and (**d**) estimated tangent vectors at orientations $\theta = 0$ and $\theta = \frac{5\pi}{32}$. The tangent vectors are displayed as *circular 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**) Orientation confidence s, cf. (68) at $\theta = 0$. (**f**) D_{bb} calculated using (72). 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 $D_{bb} \approx 0$, 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 $D_{bb} \approx 0$. 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

elongated structure is present. In Fig. 8 we show an example of these three local orientation score features.

For curvature and orientation confidence we need an estimate for the tangent vector $\mathbf{c}(g)$ at each position g. Therefore, after discussing which orientation score to use for feature estimation, 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 (32) and (34) respectively. Finally, we will introduce a measure for orientation confidence.

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 aim to treat edges and ridges in a uniform manner, i.e. curvature, deviation from horizontality, and orientation confidence 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 Fig. 3(d).

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 Fig. 3(a)), results in a flat plane response in W (see Fig. 3(d)). This means that the tangent vector ∂_a 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 Sect. 5.2.1, or by using a different orientation score for feature estimation in that case.

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 nonlinear 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 $\nabla W = (\partial_\xi, \partial_\eta, \partial_\theta)^{\mathrm{T}} W$ 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 at position g_0, leading to

$$\mathbf{c}^* = \arg\min_{\mathbf{c}} \left\{ \left\| \frac{\mathrm{d}}{\mathrm{d}s} (\nabla W(g_0 \gamma_{\mathbf{c}}(s)))\Big|_{s=0} \right\|_\mu^2 \,\middle|\, \|\mathbf{c}\|_\mu = 1 \right\}, \tag{58}$$

where $\mathbf{c} = (c^\xi, c^\eta, c^\theta)$, the norm $\|\cdot\|_\mu$ is defined in Sect. 2.5, and $g_0\gamma_{\mathbf{c}}(s) = g_0 \exp(s(c^\theta \partial_\theta + c^\xi \partial_x + c^\eta \partial_y))$, recall (28).

The minimizing equation in (58) is a norm of a covector and can be rewritten as

$$
\begin{aligned}
&\left\| \frac{\mathrm{d}}{\mathrm{d}s} (\nabla W(g_0 \gamma_{\mathbf{c}}(s)))\Big|_{s=0} \right\|_{\mu}^{2} \\
&\quad = \left\| (\nabla(\nabla W(g_0 \gamma_{\mathbf{c}}(s)))\, \dot{\gamma}_{\mathbf{c}}(s))\big|_{s=0} \right\|_{\mu}^{2} = \|\mathcal{H} W \mathbf{c}\|_{\mu}^{2} \\
&\quad = (\mathcal{H} W \mathbf{c}, \mathcal{H} W \mathbf{c})_{\mu} = (\mathbf{M}_{\mu} \mathcal{H} W \mathbf{c}, \mathbf{M}_{\mu} \mathcal{H} W \mathbf{c})_{1} \\
&\quad = (\mathbf{c}, (\mathcal{H} W)^{\mathrm{T}} \mathbf{M}_{\mu}^{2} (\mathcal{H} W) \mathbf{c})_{1}, \qquad (59)
\end{aligned}
$$

where $\mathbf{M}_{\mu} = \mathrm{diag}\{1/\mu, 1/\mu, 1\}$ and $(\cdot,\cdot)_1$ denotes the normal $\mathbb{R}^3$ inner product. The Hessian $\mathcal{H}$ which we implicitly assume to be calculated at position g_0, is defined by

$$
\mathcal{H} W = \nabla(\nabla W) = \begin{pmatrix} \partial_{\xi}^{2} W & \partial_{\eta} \partial_{\xi} W & \partial_{\theta} \partial_{\xi} W \\ \partial_{\xi} \partial_{\eta} W & \partial_{\eta}^{2} W & \partial_{\theta} \partial_{\eta} W \\ \partial_{\xi} \partial_{\theta} W & \partial_{\eta} \partial_{\theta} W & \partial_{\theta}^{2} W \end{pmatrix}. \qquad (60)
$$

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

$$
\|\mathbf{c}\|_{\mu} = (\mathbf{M}_{\mu}^{-1} \mathbf{c}, \mathbf{M}_{\mu}^{-1} \mathbf{c})_{1} = (\mathbf{c}, \mathbf{M}_{\mu}^{-2} \mathbf{c})_{1}. \qquad (61)
$$

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

$$
(\mathcal{H} W)^{\mathrm{T}} \mathbf{M}_{\mu}^{2} (\mathcal{H} W) \mathbf{c}^{*} = \lambda \mathbf{M}_{\mu}^{-2} \mathbf{c}^{*}. \qquad (62)
$$

This can be rewritten as

$$
(\mathbf{M}_{\mu} \mathcal{H} W \mathbf{M}_{\mu})^{\mathrm{T}} (\mathbf{M}_{\mu} \mathcal{H} W \mathbf{M}_{\mu})\, \tilde{\mathbf{c}}^{*} = \lambda\, \tilde{\mathbf{c}}^{*}, \qquad (63)
$$

where $\tilde{\mathbf{c}}^* = \mathbf{M}_{\mu}^{-1} \mathbf{c}^*$ which amounts to eigensystem analysis of the symmetric 3×3 matrix $(\mathbf{M}_{\mu} \mathcal{H} W \mathbf{M}_{\mu})^{\mathrm{T}} (\mathbf{M}_{\mu} \mathcal{H} W \mathbf{M}_{\mu})$, 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 $\mathbf{c}^* = \mathbf{M}_{\mu} \tilde{\mathbf{c}}^*$.

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

$$
\left(G_{\rho_s, \rho_o} * (\mathbf{M}_{\mu} \mathcal{H} W \mathbf{M}_{\mu})^{\mathrm{T}} (\mathbf{M}_{\mu} \mathcal{H} W \mathbf{M}_{\mu}) \right) \tilde{\mathbf{c}}^{*} = \lambda\, \tilde{\mathbf{c}}^{*}, \qquad (64)
$$

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 (see Sect. 2.7), 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 (58). In the minimization term, $(\mathcal{H} W \mathbf{c})$ can now be rewritten as

$$
\mathcal{H} W \mathbf{c}\big|_{c^{\eta}=0} = \mathcal{H}_{\mathrm{hor}} W \mathbf{c}_{\mathrm{hor}} = \begin{pmatrix} \partial_{\xi} \partial_{\eta} W & \partial_{\theta} \partial_{\eta} W \\ \partial_{\xi} \partial_{\theta} W & \partial_{\theta}^{2} W \\ \partial_{\xi}^{2} W & \partial_{\theta} \partial_{\xi} W \end{pmatrix} \begin{pmatrix} c^{\xi} \\ c^{\theta} \end{pmatrix}. \qquad (65)
$$

Now the Euler-Lagrange equation gives

$$
(\mathbf{M}_{\mu}\, \mathcal{H}_{\mathrm{hor}} W\, \mathbf{M}_{\mu,\mathrm{hor}})^{\mathrm{T}} (\mathbf{M}_{\mu}\, \mathcal{H}_{\mathrm{hor}} W\, \mathbf{M}_{\mu,\mathrm{hor}})\, \tilde{\mathbf{c}}_{\mathrm{hor}}^{*} = \lambda\, \tilde{\mathbf{c}}_{\mathrm{hor}}^{*}, \qquad (66)
$$

where $\tilde{\mathbf{c}}_{\mathrm{hor}}^{*} = \mathbf{M}_{\mu,\mathrm{hor}} \mathbf{c}_{\mathrm{hor}}^{*}$ and $\mathbf{M}_{\mu,\mathrm{hor}} = \mathrm{diag}\{1/\mu, 1\}$. 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 computational advantage of this approach.

5.2.2 *Structure Tensor Approach*

A closely related approach to the tangent vector estimation described above, is to use the structure tensor instead of the Hessian, as was proposed by Van Ginkel (2002) for the purpose of curvature estimation. In this approach one simply replaces the Hessian by the structure tensor, defined by

$$
\begin{aligned}
&\mathcal{S} W(\mathbf{x}, \theta) \\
&\quad = \tilde{\mathbf{R}}_{\theta} \left\{ G_{\rho_s, \rho_o} * \left(\begin{pmatrix} \partial_x W \\ \partial_y W \\ \partial_{\theta} W \end{pmatrix} \cdot \begin{pmatrix} \partial_x W \\ \partial_y W \\ \partial_{\theta} W \end{pmatrix}^{\mathrm{T}} \right) \right\} (\mathbf{x}, \theta) \cdot \tilde{\mathbf{R}}_{\theta}^{\mathrm{T}}, \qquad (67)
\end{aligned}
$$

where the derivatives 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 (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 $(\tilde{\mathcal{H}} W)^{\mathrm{T}} (\tilde{\mathcal{H}} W)$. In the algorithm by Van Ginkel, horizontality is not enforced, although one could enforce horizontality in this approach as well.

In the evaluation in Sect. 8.1 we will compare the Hessian approaches with this structure tensor approach, since it is the only alternative approach for tangent vector estimation in orientation scores.

5.3 Orientation Confidence

In a 3D image, a commonly used measure for orientation confidence 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 choose a similar approach. An oriented structure in $W = |U_f|$ renders a convex hill in the intensity landscape of the orientation score, yielding negative second order derivatives in the plane orthogonal to the tangent vector of the line. Therefore, as measure for orientation confidence s we take the Laplacian in the plane orthogonal to the line, yielding

$$s = -\Delta_{\mathrm{orth}} W = -(\mathbf{e}_{o1}^{\mathrm{T}} \mathcal{H}W \mathbf{e}_{o1} + \mathbf{e}_{o2}^{\mathrm{T}} \mathcal{H}W \mathbf{e}_{o2}), \tag{68}$$

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 (17), i.e.

$$\begin{aligned} &(\mathbf{c}, \mathbf{e}_{o1})_\mu = 0, \quad (\mathbf{c}, \mathbf{e}_{o2})_\mu = 0, \quad (\mathbf{e}_{o1}, \mathbf{e}_{o2})_\mu = 0, \\ &(\mathbf{e}_{o1}, \mathbf{e}_{o1})_\mu = 1, \quad (\mathbf{e}_{o2}, \mathbf{e}_{o2})_\mu = 1. \end{aligned} \tag{69}$$

Note that we added a minus sign in (68) in order to get a positive value for oriented structures.

Note that one can also use the two other eigenvectors of the matrix $(\mathbf{M}_\mu \mathcal{H}W\mathbf{M}_\mu)^{\mathrm{T}}(\mathbf{M}_\mu \mathcal{H}W\mathbf{M}_\mu)$. 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 orientation confidence is given by

$$s = -\left(\tilde{\mathbf{e}}_{o1}^{\mathrm{T}} \mathbf{M}_\mu \mathcal{H}W\mathbf{M}_\mu \tilde{\mathbf{e}}_{o1} + \tilde{\mathbf{e}}_{o2}^{\mathrm{T}} \mathbf{M}_\mu \mathcal{H}W\mathbf{M}_\mu \tilde{\mathbf{e}}_{o2}\right). \tag{70}$$

6 Nonlinear Diffusion in the Orientation Score

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

$$\begin{cases} \partial_t U(g,t) = \big(\partial_a \partial_a + \partial_b D_{bb}(U)(g,t)\partial_b \\ \qquad\qquad + \partial_c D_{bb}(U)(g,t)\partial_c\big) U(g,t), \\ U(g, t=0) = U_f(g). \end{cases} \tag{71}$$

Here, 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 high orientation confidence, we only want to diffuse tangent to this structure, so D_{aa} should be large and $D_{bb} = D_{cc}$ should be small. If there is no strong orientation, the diffusion should be μ-isotropic, so $D_{aa} = D_{bb} = D_{cc}$ should be large. Notice that in the resulting μ-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 $D_{bb}(U)$ we have different choices. We propose to use

$$D_{bb}(U)(g,t) = \begin{cases} \exp\left(-\frac{s(U)(g,t)}{c}\right), & s(U)(g,t) \geq 0, \\ 1, & \text{otherwise}, \end{cases} \tag{72}$$

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

Figure 9 shows how all different parts are connected. 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.

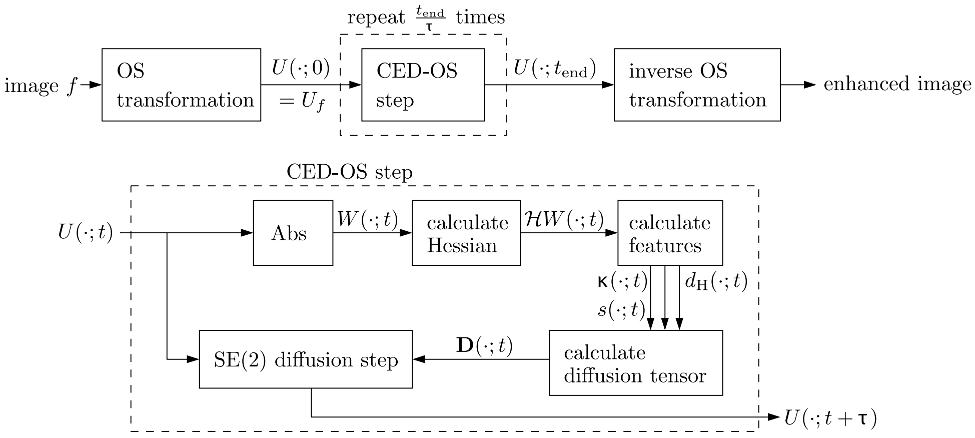


Fig. 9 Flow chart of the CED-OS (Coherence Enhancing Diffusion in Orientation Score) method

7 Numerical Scheme for Nonlinear Diffusion

In this section we will propose two explicit finite difference schemes to solve diffusion equation (71). 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 (Weickert 1999), are therefore discarded due to their poor rotation invariance (see Fig. 10). The LSAS scheme (Welk et al. 2006) 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 μ-isotropy and use LSAS but then one would be restricted to the case $s_\theta = \mu$, and there is no good reason for this restriction.

7.1 Simple Explicit Finite Difference Scheme

We aim to develop a simple scheme for solving the nonlinear diffusion PDE given by (71). It is obtained by rewriting the PDE to ∂_x, ∂_y, and ∂_θ derivatives, and using centered finite differences to calculate the first order derivatives. For second order accurate finite differences this yields

$$\begin{aligned}
\partial_x U^k(\mathbf{x}, l) &\approx \frac{1}{2}\Big(U^k(\mathbf{x}+\mathbf{e}_x, l) - U^k(\mathbf{x}-\mathbf{e}_x, l)\Big),\\
\partial_y U^k(\mathbf{x}, l) &\approx \frac{1}{2}\Big(U^k(\mathbf{x}+\mathbf{e}_y, l) - U^k(\mathbf{x}-\mathbf{e}_y, l)\Big),\\
\partial_\theta U^k(\mathbf{x}, l) &\approx \frac{1}{2s_\theta}\Big(U^k(\mathbf{x}, l+1) - U^k(\mathbf{x}-\mathbf{e}_x, l-1)\Big),
\end{aligned} \tag{73}$$

where $k \in \mathbb{N}$ denotes the discrete time and $l \in [0, N_o - 1]$ denotes the sampled orientation axis where $\theta = l \cdot s_\theta$ with orientation sample distance s_θ. In time direction we use the first order forward finite difference $(U^{k+1} - U^k)/\tau$ where τ denotes 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 $\partial_x^2 U = \frac{1}{4}(U(\mathbf{x}+2\mathbf{e}_x, l) - 2U(\mathbf{x}, l) + U(\mathbf{x}-2\mathbf{e}_x, l))$, 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 (Weickert and Scharr 2002).

7.1.1 Stability Analysis

For the stability analysis, we consider the linear diffusion equation (71) assuming constant D_{bb} 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 nonlinear diffusion equation of Sect. 6, i.e. $D_{bb} = D_{cc} \leq D_{aa} = 1$. In $\{\partial_x, \partial_y, \partial_\theta\}$ coordinates, the diffusion tensor components are given by

$$\begin{aligned}
D_{xx} &= \cos^2(\alpha)\cos^2(\theta) + D_{bb}\Big(\cos^2(\theta)\sin^2(\alpha) + \sin^2(\theta)\Big),\\
D_{yy} &= D_{bb}\cos^2(\theta) + \Big(\cos^2(\alpha) + D_{bb}\sin^2(\alpha)\Big)\sin^2(\theta),\\
D_{\theta x} &= \frac{1}{q}(D_{bb} - 1)\cos(\alpha)\cos(\theta)\sin(\alpha),
\end{aligned} \tag{74}$$

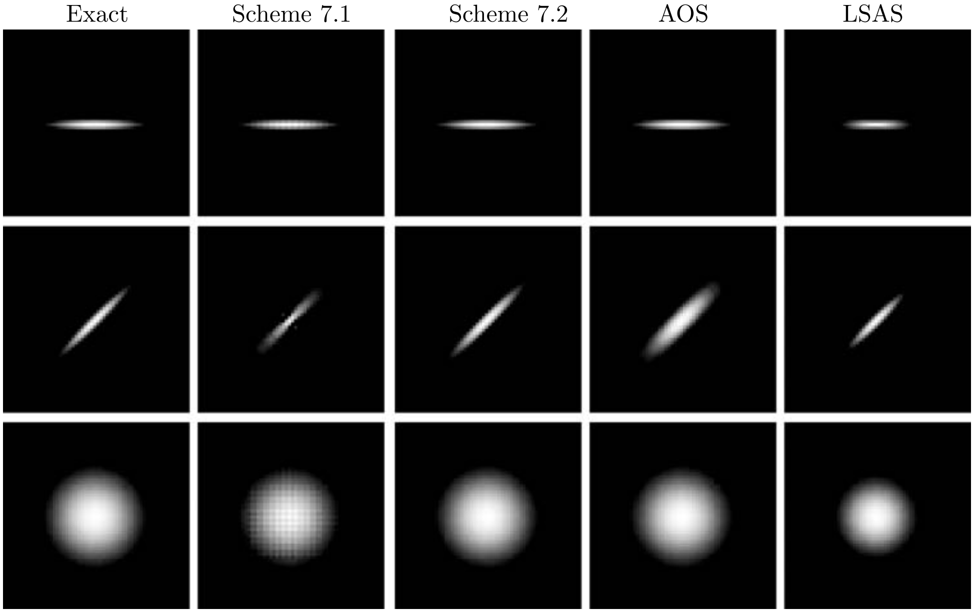


Fig. 10 Comparison of rotation invariance of different numerical anisotropic diffusion schemes. The input to all algorithms is an image with dimensions 60 × 60 with a single Gaussian blob with scale 0.9 pixels, linear diffusion is applied with diffusion tensor $\mathbf{D} = \mathbf{R}_\theta \,\mathrm{diag}(D_{xx}, D_{yy})\, \mathbf{R}_\theta^{-1}$, $\tau = 0.25$, 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)

$$D_{\theta y} = \frac{1}{q}(D_{bb}-1)\cos(\alpha)\sin(\alpha)\sin(\theta),$$
$$D_{xy} = (1-D_{bb})\cos^2(\alpha)\cos(\theta)\sin(\theta),$$
$$D_{\theta\theta} = \frac{1}{q^2}\left(D_{bb}\cos^2(\alpha)+\sin^2(\alpha)\right),$$

where $q=\frac{s_\theta}{\mu}$, $\sin\alpha = \frac{\kappa}{\sqrt{\mu^2+\kappa^2}}$, and $\cos\alpha = \frac{\mu}{\sqrt{\mu^2+\kappa^2}}$. The first order finite differences are defined in (73), rendering the following stencils for the second order finite differences that are applied in all three dimensions

$$\mathbf{S}_{ii} = \frac{1}{4}\begin{pmatrix}1 & 0 & -2 & 0 & 1\end{pmatrix}, \qquad \mathbf{S}_{ij|i\neq j} = \frac{1}{4}\begin{pmatrix}-1 & 0 & 1\\ 0 & 0 & 0\\ 1 & 0 & -1\end{pmatrix}. \tag{75}$$

One numerical iteration can be written as

$$U^{k+1} = U^k + \tau\mathbf{A}(U^k)U^k = \underbrace{(\mathbf{I}+\tau\mathbf{A}(U^k))}_{\mathbf{M}(U^k)}U^k, \tag{76}$$

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 (Gerschgorin 1931) we find that all eigenvalues are situated in a circle with center C and radius R

$$\begin{aligned} C &= 1-\frac{\tau}{2}(D_{\theta\theta}+D_{xx}+D_{yy})\\ R &= \frac{\tau}{2}(|D_{\theta\theta}|+|D_{xx}|+|D_{yy}|+4|D_{\theta x}|+4|D_{\theta y}|\\ &\quad +4|D_{xy}|). \end{aligned} \tag{77}$$

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

$$\begin{aligned}\tau \leq 4q^2\big\{&(1+q^2)+D_{bb}(1+3q^2)\\ &+\cos(2\alpha)(D_{bb}-1)(1-q^2)\\ &+2q\,|(D_{bb}-1)\sin(2\alpha)|\cdot(|\cos\theta|+|\sin\theta|)\\ &+q^2\,|(1-D_{bb})\sin(2\theta)(\cos(2\alpha)+1)|\big\}^{-1}.\end{aligned} \tag{78}$$

Note that the second inequality $C+R\leq 1$ never holds as (77) 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 under all circumstances. For the sine and cosine terms in the denominator, these worst case values are

$$\begin{aligned} &\cos(2\alpha)=1, && \text{for } \alpha=\frac{\pi}{4};\\ &\sin(2\alpha)=1, && \text{for } \alpha=0;\\ &|\cos\theta|+|\sin\theta|=\sqrt{2}, && \text{for } \theta=\frac{\pi}{4}. \end{aligned} \tag{79}$$

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

$$\tau \leq \frac{4q^2}{1+2\sqrt{2}q+3q^2-|1-q^2|}. \tag{80}$$

For a typical value of $q=\frac{s_\theta}{\mu}=\frac{\pi/32}{0.1}$ 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 in (71) 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 Fig. 11. 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 (1999) 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 Fig. 7 and all experiments in the next section are obtained with this numerical scheme.

The main advantage of this scheme is the improved rotation invariance, see Fig. 10. The drawback, however, is the computational speed and the some additional blurring caused by the interpolation scheme.

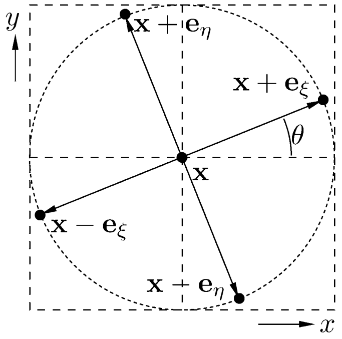


Fig. 11 Illustration of the spatial part of the stencil of the numerical scheme of Sect. 7.2. The *horizontal* and *vertical dashed lines* indicate the sampling grid, which is aligned with $\{\mathbf{e}_x, \mathbf{e}_y\}$. The stencil points, indicated by the *black dots*, are aligned with the rotated coordinate system cf. (12) with $\theta = l\, s_\theta$

$$\partial_\theta U \approx \frac{1}{2s_\theta}\left(U(\mathbf{x}, l+1) - U(\mathbf{x}, l-1)\right)$$

$$\partial_\theta^2 U \approx \frac{1}{s_\theta^2}\left(U(\mathbf{x}, l+1) - 2U(\mathbf{x}, l) + U(\mathbf{x}, l-1)\right)$$

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

$$\partial_\xi^2 U \approx U(\mathbf{x}+\mathbf{e}_\xi^l, l) - 2U(\mathbf{x}, l) + U(\mathbf{x}-\mathbf{e}_\xi^l, l)$$

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

$$\partial_\eta^2 U \approx U(\mathbf{x}+\mathbf{e}_\eta^l, l) - 2U(\mathbf{x}, l) + U(\mathbf{x}-\mathbf{e}_\eta^l, l)$$

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

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

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 $D_{bb} = D_{bb} \leq D_{aa} = 1$. In $\{\partial_\xi, \partial_\eta, \partial_\theta\}$ coordinates, the diffusion tensor is given by

$$\begin{aligned}
D_{\xi\xi} &= \cos^2(\alpha) + D_{bb}\sin^2(\alpha), \\
D_{\eta\eta} &= D_{bb}, \\
D_{\theta\xi} &= \frac{1}{q}(D_{bb} - 1)\cos(\alpha)\sin(\alpha), \\
D_{\theta\eta} &= D_{\xi\eta} = 0, \\
D_{\theta\theta} &= \frac{1}{q^2}D_{bb}\cos^2(\alpha) + \sin^2(\alpha),
\end{aligned} \tag{81}$$

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

$$\mathbf{S}_{ii} = \begin{pmatrix} 1 & -2 & 1 \end{pmatrix}, \qquad \mathbf{S}_{ij|i\neq j} = \frac{1}{4}\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix}. \tag{82}$$

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

$$\chi = \max_{\Delta\in[-1/2,+1/2]} \sum_{x=-\infty}^{+\infty} |C_2(x+\Delta)| = \sqrt{2}, \tag{83}$$

where C_2 is the cardinal spline (Unser 1999) corresponding to the second order B-spline, which is in fact the net convolution kernel that is used for spline interpolation.

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

$$\begin{aligned}
C &= 1 - 2\tau\Bigg(\left(\frac{D_{bb}}{q^2} + 1\right)\cos^2(\alpha) \\
&\quad + \left(D_{bb} + \frac{1}{q^2}\right)\sin^2(\alpha) + D_{bb}\Bigg), \\
R &= 2\tau\Bigg(\sqrt{2}D_{bb} + \sqrt{2}\left|\frac{(D_{bb}-1)\cos(\alpha)\sin(\alpha)}{q}\right| \\
&\quad + \left(D_{bb}q^{-2} + \sqrt{2}\right)\cos^2(\alpha) \\
&\quad + \left(q^{-2} + \sqrt{2}D_{bb}\right)\sin^2(\alpha)\Bigg).
\end{aligned} \tag{84}$$

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

$$\begin{aligned}
\tau \leq q^2\Bigg\{&\Bigg(\frac{1}{\sqrt{2}}\left|\frac{(1-D_{bb})\sin(2\alpha)}{q}\right| + \frac{1}{2}\left(1+\sqrt{2}\right) \\
&\times (3D_{bb} + (1-D_{bb})\cos(2\alpha) + 1)\Bigg)q^2 \\
&+ D_{bb} + (D_{bb}-1)\cos(2\alpha) + 1\Bigg\}^{-1}.
\end{aligned} \tag{85}$$

By inserting the “worst case values” for all sine and cosine terms, see (79), and by setting isotopic diffusion $D_{bb} = 1$, which in this case is the worst-case for stability since it gives the largest 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 $0 \leq D_{bb} \leq 1$

$$\tau \leq \frac{2q^2}{4 + 4(1+\sqrt{2})q^2}. \tag{86}$$

For a typical value of $q = \frac{s_\theta}{\mu} = \frac{\pi/32}{0.1}$ 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. We will also 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 (Fig. 12). This is a useful test image since it contains a wide range of circle radii (in this case 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 $\mu = 0.08$. We compare our method, with and without enforcing horizontality, with the structure tensor approach (see Sect. 5.2.2 and (Van Ginkel 2002)), with parameters $t_s = 2.5$, $\rho_s = 7.5$, and $\mu = 0.08$. 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 (Fig. 12) we do know the ground truth orientation θ_{true} and curvature κ_{true}. To evaluate the results, for each spatial position we take 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

$$\text{Error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\kappa_{\text{est},i} - \kappa_{\text{true}}}{\kappa_{\text{true}}}\right)^2}, \tag{87}$$

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 13 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 14 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 15 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. (71) with results obtained by the normal coherence enhancing diffusion (CED) approach (Weickert 1999). The CED method solves diffusion equation on $\mathbb{R}^2$, cf. (35), where the diffusion tensor is given by

$$\begin{aligned}
&\mathbf{D}(\mathbf{x}) = \lambda_1\, \mathbf{e}_1(\mathbf{x})\, \mathbf{e}_1(\mathbf{x})^{\mathrm{T}} + \lambda_2(\mathbf{x})\, \mathbf{e}_2(\mathbf{x})\, \mathbf{e}_2(\mathbf{x})^{\mathrm{T}}, \\
&\quad \text{with } \lambda_1 = \alpha, \quad \text{and} \\
&\lambda_2(\mathbf{x}) = \begin{cases} \alpha & \text{if } \mu_1(\mathbf{x}) = \mu_2(\mathbf{x}), \\ \alpha + (1-\alpha)\exp\left(\frac{-C}{(\mu_1(\mathbf{x}) - \mu_2(\mathbf{x}))^2}\right) & \text{otherwise,} \end{cases}
\end{aligned} \tag{88}$$

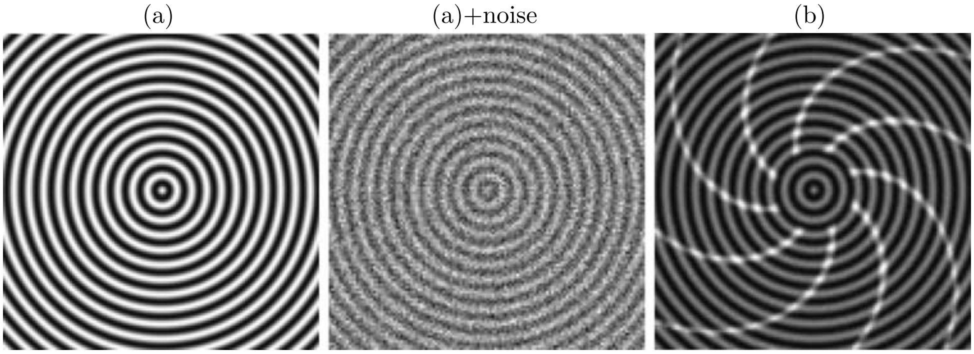


Fig. 12 Curvature estimation test images

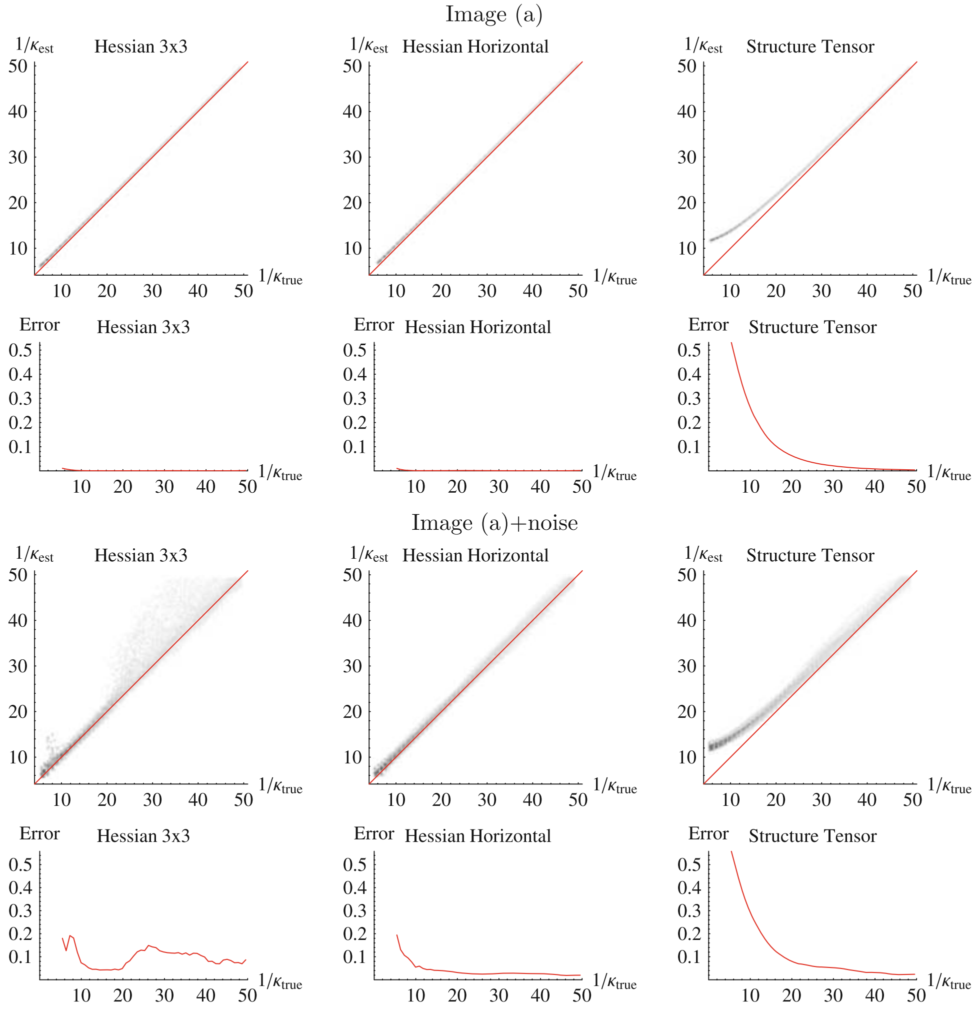


Fig. 13 Curvature estimation results on *image* (**a**) and *image* (a) + *noise* (see Fig. 12) for the three methods. For both images, the *first row* shows the density plot of true curvature κ_{true} against estimated curvature κ_{est}. The *second row* shows the relative ℓ_2-error, cf. (87) as function of the different curvatures

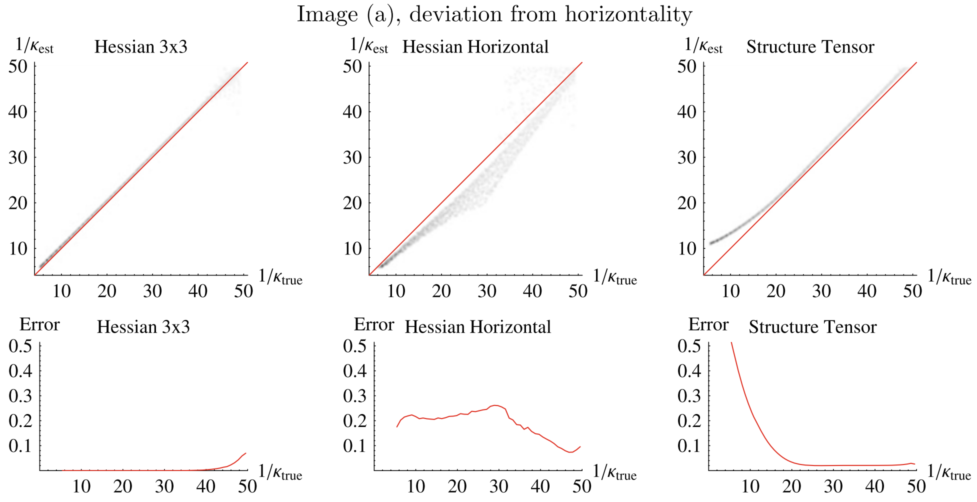


Fig. 14 The effect of deviating from horizontality on *image* (**a**) (see Fig. 12). For these graphs, the curvature estimation results are obtained at orientation $\theta_{\text{true}} + \pi/8$

where $C > 0$ controls the nonlinear behavior, $0 < \alpha \ll 1$ is a small parameter that ensures that the diffusion tensors are positive definite, $\mu_1(\mathbf{x})$ and $\mu_2(\mathbf{x})$ are the eigenvalues of the structure tensor with $\mu_1(\mathbf{x}) > \mu_2(\mathbf{x})$, and $\mathbf{e}_1$ and $\mathbf{e}_2$ are the corresponding eigenvectors. The structure tensor is given by

$$\mathcal{S}f(\mathbf{x}) = \left\{ G_\rho * \left(\begin{pmatrix} \partial_x(f * G_\sigma) \\ \partial_y(f * G_\sigma) \end{pmatrix} \cdot \begin{pmatrix} \partial_x(f * G_\sigma) \\ \partial_y(f * G_\sigma) \end{pmatrix}^{\mathrm{T}} \right) \right\}(\mathbf{x}). \tag{89}$$

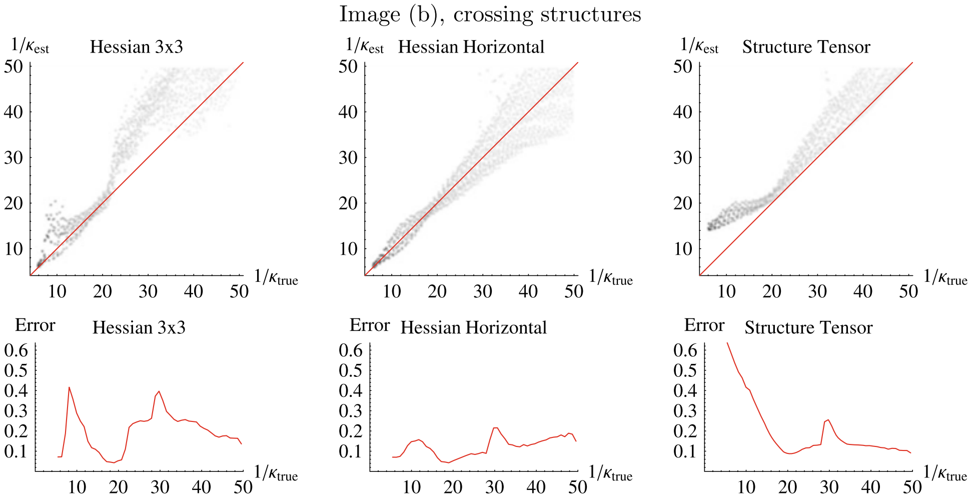


Fig. 15 Shows curvature estimation results on *image* (**b**) (see Fig. 12) with crossing elongated structures

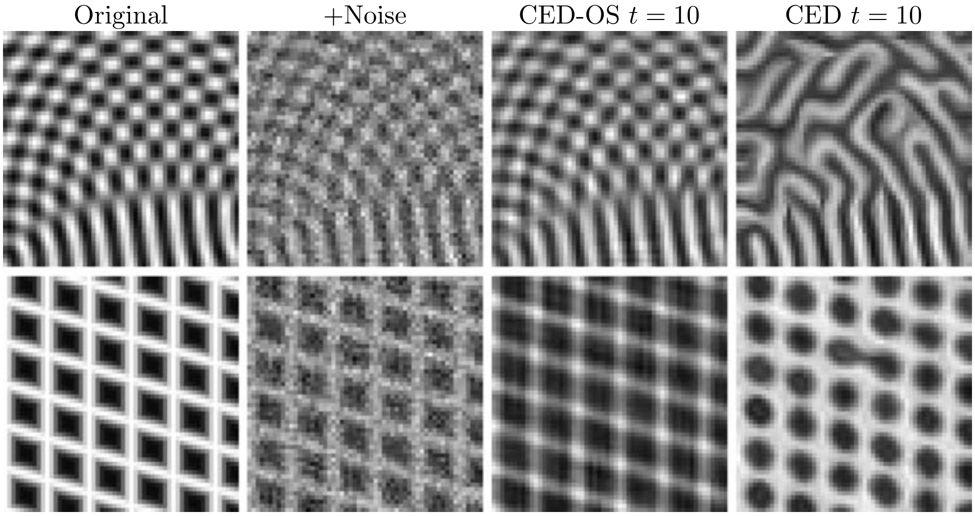


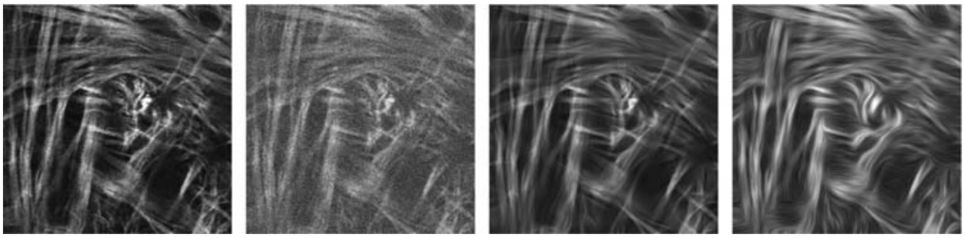
Fig. 16 Shows the typical different behavior of CED-OS compared to CED. In CED-OS crossing structures are better preserved

We use the LSAS numerical scheme (Welk et al. 2006) to solve the diffusion equation numerically.

The following parameters are used for the orientation score transformation (Sect. 3): $k = 2$, $q = 8$, $t = 1.6(\frac{X}{2\pi})^2$ where X is the number of pixels of the image in x-direction, $s = 200$, and number of orientations $N_o = 32$. 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 have a range of pixel values from 0 to 255. To ensure numerical stability, in the experiments where we use the numerical scheme of Sect. 7.1 we use $\tau = 0.25$ and for the numerical scheme of Sect. 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.

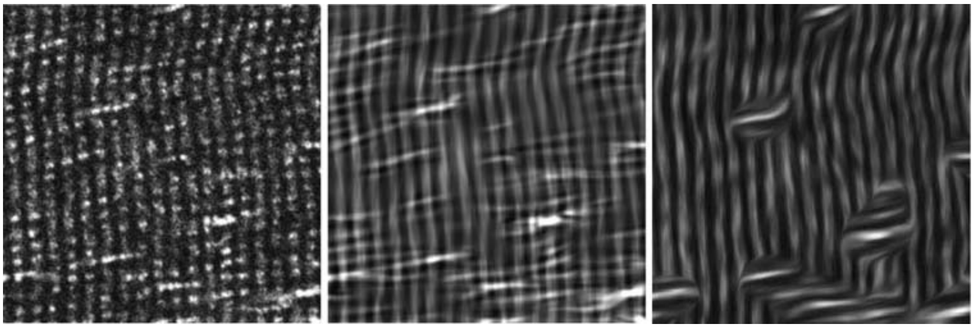
Figure 16 shows the effect of CED-OS compared to CED on artificial images with crossing line structures. The upper image shows an additive superimposition of two images with concentric circles. Our method is able to preserve this structure, while CED cannot. The same holds for the lower image with crossing 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 Sect. 7.2 is used. The nonlinear diffusion parameters for CED-OS are: $t_s = 12$, $\rho_s = 0$, $\mu = 0.058$, and $c = 0.08$. The parameters that we used for CED are (see Weickert (1999)): $\sigma = 1$, $\rho = 1$, $C = 1$, 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 a microscopy image of collagen fibres, where two rotated version of the image have been superimposed for the sake of illustration of our algorithm. These kind of images are acquired in tissue engineering research, where the goal is to create artificial heart valves (Rubbens et al. 2008). All parameters during these experiments were set the same as the artificial images mentioned above except for CED para-



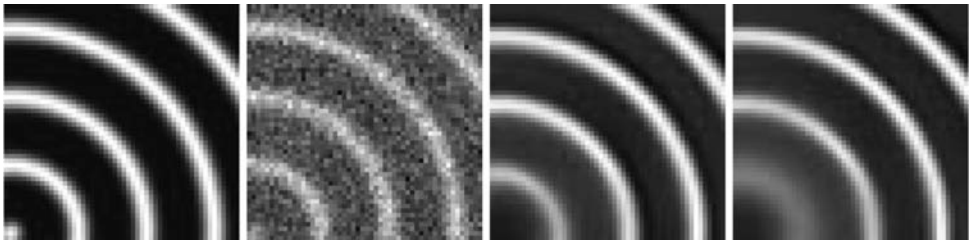
(a) Original (b) +Noise (c) CED-OS $t = 30$ (d) CED $t = 30$

Fig. 17 Result of CED-OS and CED on microscopy images of bone tissue. Additional Gaussian noise is added to verify the behaviour on noisy images



(a) Original (b) CED-OS $t = 30$ (c) CED $t = 30$

Fig. 18 Result of CED-OS and CED on a microscopy image of a muscle cell



(a) Original (b) +Noise (c) CED-OS $t = 30$ with curvature (d) CED-OS $t = 30$ without curvature

Fig. 19 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

meter $\rho = 6$. The stopping time is $t = 30$ and the image size is 160×160 pixels.

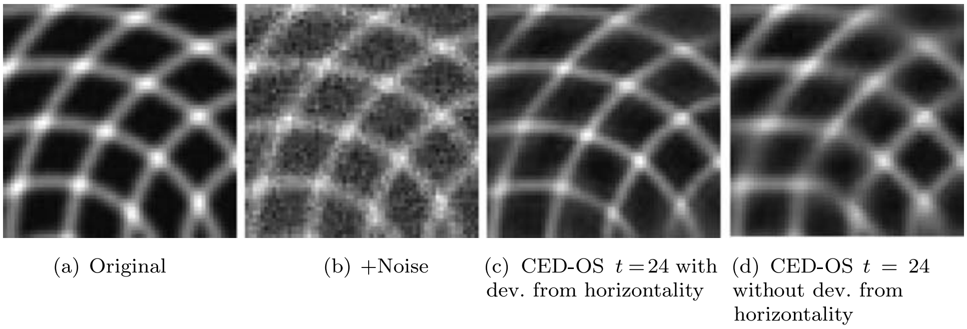
Figures 17 and 18 show examples of the method on two microscopy images, see Foolen et al. (2008), resp. Shaw et al. (2008) for more information on the biomedical relevance of these images. The same parameters are used as above except for $t_s = 25$ in Fig. 18. Clearly, the curve enhancement and noise 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 19 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_o, because with a low number of orientations and without d_H, the elongated structures in the resulting images show strong biases towards the angles $l\frac{\pi}{N_o}$ with $l \in \mathbb{Z}$, while with d_H this problem does not occur. In Fig. 20 this is illustrated. We set $N_o = 4$, $t_s = 5$, $\rho_s = 0$, $\mu = 0.058$, and $c = 0.1$ and use the numerical scheme of Sect. 7.1. Clearly we observe that 4 orientations 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 crossings of two lines. Figure 21 shows the same effect on a microscopy image of bone tissue, where the parameters were set to $N_o = 4$, $t_s = 2$, $\rho_s = .5$, $\mu = 0.11$, and $c = 0.01$.

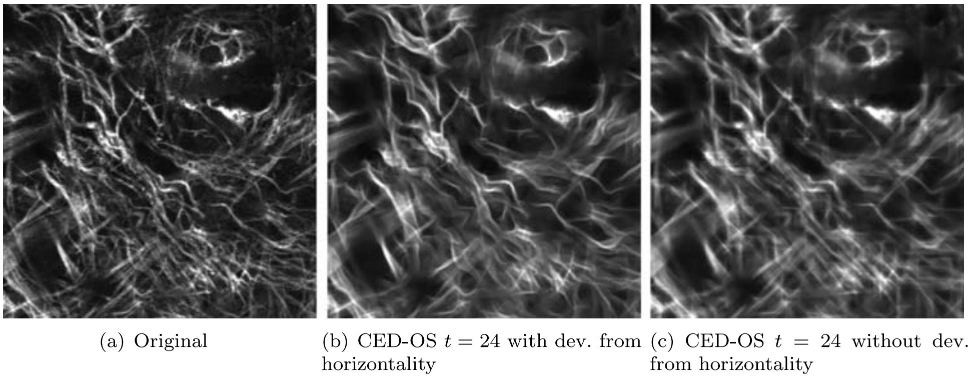
This means that including deviation from horizontality can make CED-OS more efficient, since we can get good results with a very low number of orientations. Note, however that if one wants to handle crossings of more than 2 lines, or if the angle between the crossing lines is small, it is still necessary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

Although CED-OS is clearly advantageous for handling crossings, CED has the advantage that it is faster. In our C++



(a) Original (b) +Noise (c) CED-OS $t = 24$ with dev. from horizontality (d) CED-OS $t = 24$ without dev. from horizontality

Fig. 20 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 $N_o = 4$ we are able to handle crossings correctly



(a) Original (b) CED-OS $t = 24$ with dev. from horizontality (c) CED-OS $t = 24$ without dev. from horizontality

Fig. 21 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 $N_o = 4$

implementation, which is not fully optimized, a single iteration of CED on 128×128 roughly took 0.004 seconds while an iteration of CED-OS with the scheme of Sect. 7.1 and with $N_o = 32$ orientations took 0.5 seconds on a 2.4 GHz single core CPU. The large difference in computation times is caused by the fact that CED applies numerical diffusion on a 2D dataset while CED-OS applies numerical diffusion to a 3D dataset, resulting in a multiplication of algorithmic complexity with a factor N_o. Furthermore, the LSAS scheme that we used for CED is a very efficient scheme and our CPU implementation was highly optimized, while our current implementations for CED-OS can be optimized much further.

9 Conclusions

In this paper we introduced nonlinear 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 process 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 orientation confidence s. The features κ and d_H uniquely determine the optimal gauge frame. The nonlinear diffusion is aligned with this optimal gauge frame and the orientation confidence s controls whether locally the diffusion is isotropic or anisotropic.

We proposed two explicit numerical schemes to apply the nonlinear 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. Furthermore we showed 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. The adaptation of the diffusion to 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_o. Therefore, including deviation from horizontality enables a large reduction in computations and storage. The diffusion shows the typical nonlinear 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 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 (Duits and Burgeth 2007) 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. Finally, it is interesting to apply the same techniques in the 3D Euclidean motion group, to enhance elongated structures in three-dimensional images. This is especially useful for enhancing and segmenting fibers in High Angular Resolution Diffusion Imaging (HARDI) data. This is currently being investigated by the authors of this paper (Franken 2008, Chap. 7).

Acknowledgements The authors would like to thank the anonymous referees whose comments and suggestions led to a significant improvement of the presentation and organization of this paper. The project was financially supported by the Dutch BSIK program entitled Molecular Imaging of Ischemic heart disease (project number BSIK 03033). The Netherlands Organisation for Scientific Research (NWO) is gratefully acknowledged for financial support.
Furthermore, we would like to thank Mirjam Rubbens (Technische Universiteit Eindhoven) for providing the source image for Fig. 1(a), Jasper Foolen (Technische Universiteit Eindhoven) for providing the images in Figs. 17(a) and 21(a), and Christopher Shaw (University of Birmingham, UK) for providing the image in Fig. 18(a).

References

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

Antoine, J.-P., Murenzi, R., & Vandergheynst, P. (1999). Directional wavelets revisited: Cauchy wavelets and symmetry detection in patterns. *Applied and Computational Harmonic Analysis*, *6*(3), 314–345.

August, J. (2001). *The curve indicator random field*. Ph.D. thesis, Yale University.

Candès, E. J., & Donoho, D. L. (1999a). Curvelets—a surprisingly effective nonadaptive representation for objects with edges. In A. Cohen, C. Rabut, & L. L. Schumaker (Eds.), *Curve and surface fitting: Saint-Malo 1999*. Nashville: Vanderbilt University Press.

Candès, E. J., & Donoho, D. L. (1999b). Ridgelets: the key to high dimensional intermittency? *Philosophical Transactions of the Royal Society of London A*, *357*, 2495–2509.

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

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

Cottet, G.-H., & Germain, L. (1993). Image processing through reaction combined with nonlinear diffusion. *Mathematics of Computation*, *61*, 659–667.

Duits, R. (2005). *Perceptual organization in image analysis*. Ph.D. thesis, Technische Universiteit Eindhoven. http://www.bmi2.bmt.tue.nl/Image-Analysis/People/RDuits/THESISRDUITS.pdf.

Duits, R., & Burgeth, B. (2007). Scale spaces on Lie groups. In F. Sgallari, A. Murli, & N. Paragios (Eds.), *Lecture notes in computer science: Vol. 4485. Scale space and variational methods in computer vision: proceedings of the first international conference, SSVM 2007* (pp. 300–312). Ischia, Italy, May–June 2007. Berlin: Springer.

Duits, R., & Franken, E. M. (2007). 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. http://arxiv.org/abs/0711.0951. Also available as CASA report nr. 35, Eindhoven University of Technology.

Duits, R., & Franken, E. M. (2009a, accepted). Left-invariant parabolic evolutions on $SE(2)$ and contour enhancement via invertible orientation scores—part I: Linear left-invariant diffusion equations on $SE(2)$. *Quarterly on Applied Mathematics*.

Duits, R., & Franken, E. M. (2009b, accepted) Left-invariant parabolic evolutions on $SE(2)$ and contour enhancement via invertible orientation scores—part II: Nonlinear left-invariant diffusions on invertible orientation scores. *Quarterly on Applied Mathematics*.

Duits, R., & van Almsick, M. A. (2008). The explicit solutions of linear left-invariant second order stochastic evolution equations on the 2D-Euclidean motion group. *AMS Quarterly of Applied Mathematics*, *66*, 27.

Duits, R., van Almsick, M. A., Duits, M., Franken, E. M., & Florack, L. M. J. (2004). Image processing via shift-twist invariant operations on orientation bundle functions. In N. Zhuralev et al. (Eds.), *7th International conference on pattern recognition and image analysis (PRIA-7-2004)* (pp. 193–196). St. Petersburg, October 2004.

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

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

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

Foolen, J., van Donkelaar, C., Nowlan, N., Murphy, P., Huiskes, R., & Ito, K. (2008). Collagen orientation in periosteum and perichondrium is aligned with preferential directions of tissue growth. *Journal of Orthopaedic Research: official publication of the Orthopaedic Research Society*.

Franken, E. M. (2008). *Enhancement of crossing elongated structures in images*. Ph.D. thesis, Eindhoven University of Technology, Department of Biomedical Engineering, Eindhoven, The Netherlands.

Franken, E. M., Rongen, P., van Almsick, M. A., & ter Haar Romeny, B. M. (2006). Detection of electrophysiology catheters in noisy fluoroscopy images. In *Lecture notes in computer science: Vol. 4191. Proceedings of the 9th international conference on medical image computing and computer-assisted intervention—MICCAI 2006* (pp. 25–32). Copenhagen, Denmark, 1–6 October 2006. Berlin: Springer.

Franken, E. M., Duits, R., & ter Haar Romeny, B. M. (2007a). Nonlinear diffusion on the 2D Euclidean motion group. In F. Sgallari, A. Murli, & N. Paragios (Eds.), *Lecture notes in computer science: Vol. 4485. Scale space and variational methods in computer vision: proceedings of the first international conference, SSVM 2007* (pp. 461–472). Ischia, Italy, May–June 2007. Berlin: Springer.

Franken, E. M., Duits, R., & ter Haar Romeny, B. M. (2007b). Curvature estimation for enhancement of crossing curves. In W. Niessen, C.-F. Westin, & M. Nielsen (Eds.), *Proceedings of the 8th IEEE Computer Society workshop on mathematical methods in biomedical image analysis, held in conjunction with the IEEE international conference on computer vision*, Rio de Janeiro, Brazil, 14–20 October 2007. Omnipress, Digital proceedings.

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

Granlund, G. H., & Knutsson, H. (1995). *Signal processing for computer vision*. Dordrecht: Kluwer Academic.

Heitger, F., & von der Heydt, R. (1993). A computational model of neural contour processing. In *Proceedings of the 4th international conference on computer vision* (pp. 32–40). Berlin, Germany, 20–23 June 1993. Washington: IEEE Computer Society Press.

Kalitzin, S. N., ter Haar Romeny, B. M., & Viergever, M. A. (1997). Invertible orientation bundles on 2D scalar images. In B. M. ter Haar Romeny, L. M. J. Florack, J. Koenderink, & M. Viergever (Eds.), *Scale space theory in computer vision* (pp. 77–88).

Kalitzin, S. N., ter Haar Romeny, B. M., & Viergever, M. A. (1999). Invertible apertured orientation filters in image analysis. *International Journal of Computer Vision*, *31*(2–3), 145–158.

Manniesing, R., & Niessen, W. J. (2005). Multiscale vessel enhancing diffusion in ct angiography noise filtering. In *Lecture notes in computer science: Vol. 3565. Information processing in medical imaging* (pp. 138–149). Berlin: Springer.

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

Mumford, D. (1994). Elastica and computer vision. In C. L. Bajaj (Ed.), *Algebraic geometry and its applications* (pp. 491–506). New York: Springer.

Nitzberg, M., & Shiota, T. (1992). Nonlinear image filtering with edge and corner enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*, 826–833.

Rubbens, M. P., Mol, A., Boerboom, R. A., Bank, R. A., Baaijens, F. P. T., & Bouten, C. V. C. (2008, in press). Intermittent straining accelerates the development of tissue properties in engineered heart valve tissue. *Tissue Engineering Part A, 14*. doi:10.1089/ten.tea.2007.0396.

Scharr, H. (2006). Diffusion-like reconstruction schemes from linear data models. In *Lecture notes in computer science: Vol. 4174. Pattern recognition: 28th DAGM symposium* (pp. 51–60). Berlin, Germany, 12–14 September 2006. Berlin: Springer.

Shaw, C. S., Jones, D. A., & Wagenmakers, A. J. M. (2008). Network distribution of mitochondria and lipid droplets in human muscle fibres. *Histochemistry and Cell Biology*, *129*(1), 65–72.

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

Tschumperlé, D. (2006). Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *International Journal of Computer Vision*, *68*(1), 65.

Tuch, D. S., Weisskoff, R. M., Belliveau, J. W., & Wedeen, V. J. (1999). High angular resolution diffusion imaging of the human brain. In *Proc. of the 7th annual meeting of ISMRM*, Philadelphia (p. 321).

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

Van Almsick, M. A. (2007). *Context models of lines and contours*. Ph.D. thesis, Eindhoven University of Technology, Department of Biomedical Engineering, Eindhoven, The Netherlands.

Van Ginkel, M. (2002). *Image analysis using orientation space based on steerable filters*. Ph.D. thesis, Technische Universiteit Delft, The Netherlands.

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

Weickert, J. A. (1998). Anisotropic diffusion in image processing. In *European Consortium for Mathematics in Industry series*. Stuttgart: Teubner.

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

Weickert, J., & Scharr, H. (2002). A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance. *Journal of Visual Communication and Image Representation*, *13*(1–2), 103–118.

Welk, M., Weickert, J., & Steidl, G. (2006). From tensor-driven diffusion to anisotropic wavelet shrinkage. In A. Leonardis, H. Bischof, & A. Prinz (Eds.), *Lecture notes in computer science: Vol. 3951–3954. Proceedings of the ninth European conference on computer vision* (pp. 391–403). Graz, Austria, May 2006. Berlin/Heidelberg: Springer.

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

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