



---

# Analysis of Multiple Orientations

Matthias Mühlich and Til Aach

in: IEEE Transactions Image Processing. See also  $\text{BIB}_{\text{T}}\text{E}_\text{X}$  entry below.

---

## $\text{BIB}_{\text{T}}\text{E}_\text{X}$ :

```
@article{MuehlichAach2009TIP,  
  author    = {Matthias M\{"u\}hlich and Til Aach},  
  title     = {Analysis of Multiple Orientations},  
  journal   = {IEEE Transactions Image Processing},  
  publisher = {IEEE},  
  volume   = {18},  
  number   = {7},  
  pages    = {1424--1437},  
  year     = {2009}  
}
```

© 2009 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

# Analysis of Multiple Orientations

Matthias Mühlich and Til Aach, *Senior Member, IEEE*

**Abstract**—Estimation of local orientations in multivariate signals is an important problem in image processing and computer vision. This general problem formulation also covers optical flow estimation, which can be regarded as orientation estimation in space-time-volumes. Modelling a signal using only a *single* orientation, however, is often too restrictive, since occlusions and transparencies occur frequently, thus necessitating the modelling and analysis of *multiple orientations*.

We therefore develop a unifying mathematical model for multiple orientations: Beyond describing an arbitrary number of orientations in scalar- and vector-valued image data such as color image sequences, it allows the unified treatment of *additively* and *occludingly* superimposed oriented structures as well as of combinations of these. Based on this model, we describe estimation schemes for an arbitrary number of *additively* or *occludingly* superimposed orientations in images. We confirm the performance of our framework on both synthetic and real image data.

## I. INTRODUCTION

Local orientations are an important low level feature for analyzing multivariate data. Local orientations are closely linked to gradient directions [1], [2], [3], [4], [5], [6], [7]. The relevance of orientations therefore rests on what has been coined the *local simplicity hypothesis* [8], [9], [6, Chapter 6], which states that in many local regions, the variation of the gradient *directions* is generally much lower than the variation of the multivariate signal itself. Locally linear slopes, for instance, result in a constant gradient field. More important, however, is that gradient directions are invariant to signal changes occurring strictly in the gradient directions. This hypothesis is illustrated in Fig. 1: Its left-hand side shows a local region with a line superimposed onto a background signal which changes parabolically orthogonal to the line. While the signal thus changes everywhere except on the line's ridge, all gradients in this region are parallel or anti-parallel. In other words, the bivariate signal is locally univariate.

Analysis of gradients led to the concept of *local orientations* which can be used to describe image features like edges or lines. However, this concept is limited to *linear structures*. For instance, corners or crossings – which are especially suited for applications like motion estimation [10] or tracking [11] – are in fact characterized by *sharp* local gradient variations at the most interesting point, namely the corner or crossing, as also shown in Fig. 1. In such regions, the local simplicity hypothesis as stated above does not hold anymore.

Figure 2 shows some idealized low level image features which go beyond single local orientation. These junctions are often denoted by a letter: L- and Y-junctions represent object

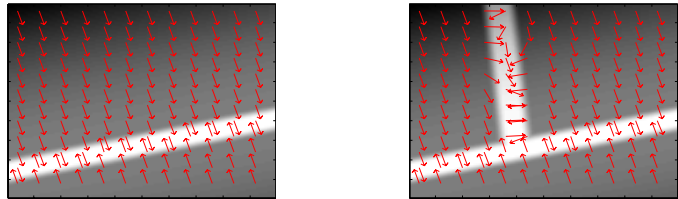


Fig. 1. Image regions of size  $99 \times 99$  pixels, showing a line (left) and a junction (right). The overlaid arrows represent the *directions* of the locally computed gradients. On the left-hand side, the line runs across a background which varies orthogonal to the line; all gradients are therefore (anti-)parallel. The T-junction (right) results in sharply changing gradient directions.

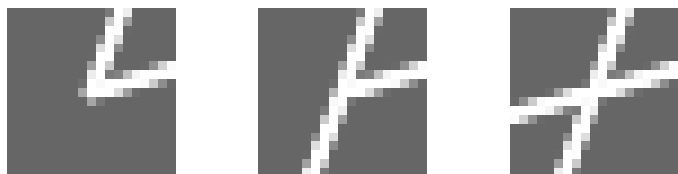


Fig. 2. Corners, junctions, and crossings are prominent low-level image features which can be described as composition from two oriented signals. These features are also known as L-, T-, and X-junctions. The size of the regions shown is  $19 \times 19$  pixels.

corners, T-junctions occur at occluding object boundaries, and X-junctions at object crossings [12], [13], [14], [15].

The framework of *double* orientation estimation [16], [17] allows a unified treatment of such features. However, modelling a signal as superposition of two oriented signals is still too restrictive for many real signals. As an example, every bifurcation of a line structure, for instance of a blood vessel or of a plant root, generates a “Y-junction” and implies that three independent orientations have to be modelled. Figure 3 shows a variety of both gray level and color (“RGB”) examples.

Multiple orientations appear in X-ray projection imaging [18], or locally as corners and junctions. Multiple superimposed orientations also occur ubiquitously in Radon space (“sinograms” in Computer Tomography [19, Fig. 3.8]), where sine curves of different objects cross. Additionally, multiple orientations allow to describe and model multioriented textures and fabrics in a more explicit manner in comparison to the more implicit features provided by, e.g., local filtering, [20], [21], [22], [23], eigenfilters [24] or co-occurrence matrices [25], [26]. In this paper, we show how limitations of the single orientation concept can be overcome by a unifying framework of multiple signal orientations. We first review single orientation estimation. We then develop models and formulate constraint equations for two types of multiple superimposed orientations, viz. additive and occluding superpositions. Subsequently, we reformulate the constraints in tensor form, and introduce the properties of the tensors involved. We

The authors are with the Institute of Imaging and Computer Vision, RWTH Aachen University, 52056 Aachen, Germany.

Parts of this work were published at the European Conference on Computer Vision (ECCV 2006), May 7–13, 2006, Graz, Austria.

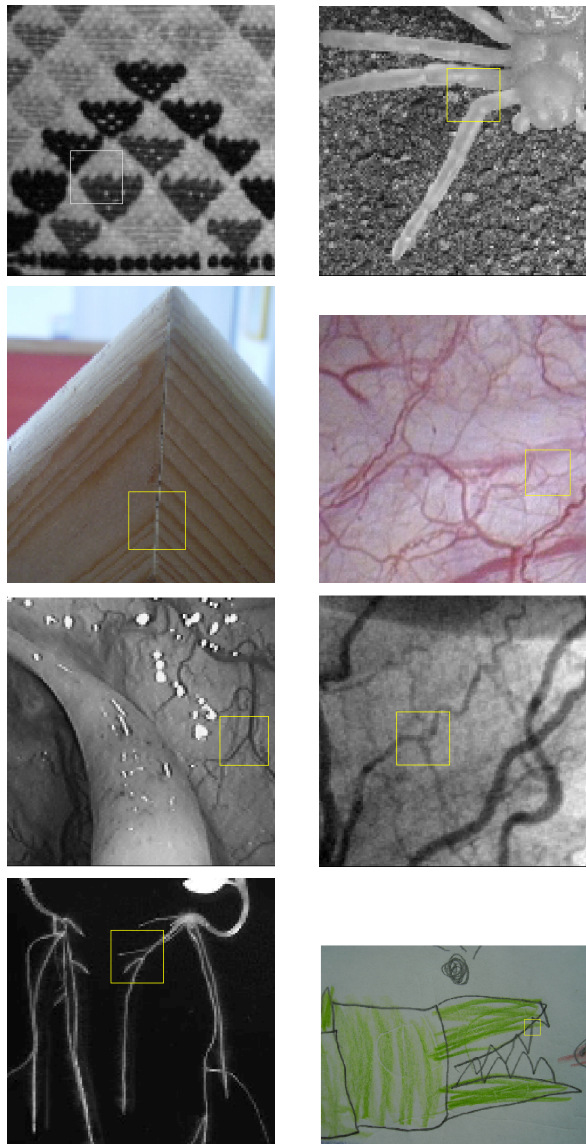


Fig. 3. Parts of images ( $128 \times 128$  pixels unless otherwise indicated) with multi-oriented neighborhoods highlighted by a square ( $25 \times 25$  pixels). From left to right and top to bottom: carpet pattern, part of a toy spider, wooden gable of a doll house ( $117 \times 130$  pixels, RGB), bladder wall of a male patient as seen through a rigid cystoscope (RGB,  $150 \times 150$  pixels, orientations in ROI only weakly pronounced), endoscopic image from inside the colon, X-ray angiography of the coronary vessel tree, plant roots in a nutrient solution. Bottom right: Picture of a crocodile ( $320 \times 240$  pixels, RGB, multi-orientations in ROI generated by dark-gray pencil and by green crayon) drawn by a five-year old child.

extend the tensor formulation to cover also multispectral, i.e., vector-valued data, thus providing for a unified treatment of occluding and additive orientations in scalar and vector-valued data. Based on factorization algorithms [27], we then describe solutions to estimate additive and occluding orientations in images, as well as combinations of these. We conclude by providing results for synthetic images as well as for a variety of real images from various applications such as medicine, biology, and inspection.

### A. Single Orientations: The structure tensor

Let  $\mathbf{x} \in \mathbb{R}^N$  be a vector in  $N$ -dimensional space. We denote  $\mathbf{x}$  by  $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$  or, when  $N = 2$ , by  $\mathbf{x} = [x, y]^T$ . A multivariate signal  $s(\mathbf{x})$ ,  $s : \mathbb{R}^N \rightarrow \mathbb{R}$ , is called locally oriented in a region  $\Omega$  if it is constant along parallel lines, i.e.

$$s(\mathbf{x} + \lambda \mathbf{u}) = s(\mathbf{x}) \quad \text{for all } \lambda \in \mathbb{R} \text{ and } \mathbf{x}, \mathbf{x} + \lambda \mathbf{u} \in \Omega \quad (1)$$

with the unit vector  $\mathbf{u}$  denoting the orientation.

Equation (1) states that a given signal is locally constant with respect to  $\mathbf{u}$  if its directional derivative  $\frac{\partial s}{\partial \mathbf{u}} = \langle \mathbf{g}, \mathbf{u} \rangle$ , i.e., the scalar product between signal gradient  $\mathbf{g}$  and  $\mathbf{u}$ , is zero for all gradients computed in the neighborhood  $\Omega$ . The gradients span a subspace in which the signal is not oriented. Consequently, the orthogonal complement of this subspace is the sought orientation. This orientation is uniquely determined if we can find  $N - 1$  linearly independent gradients. More precisely, it is an orientation axis which is only determined up to the sign; in images, it is an angle which can be mapped to the range  $[0, \pi)$ . In practice, because of non-ideal compliance of  $s(\mathbf{x})$  with the orientation model as well as because of noise, the gradients in  $\Omega$  will span the entire  $N$ -dimensional space. Minimizing the squared residual error  $\sum_{\Omega} |\langle \mathbf{g}, \mathbf{u} \rangle|^2$  of (1) over  $\Omega$  can then be understood as yielding the orientation subspace in a least-squares sense.

The introduction given so far is one of several approaches leading to the so-called *structure tensor* approach for orientation estimation found in the pioneering work of Förstner [28], Bigün et. al. [1], Di Zenzo [4], Kass and Witkin [5], and others. For bivariate image data ( $N = 2$ ; generalization to arbitrary  $N$  is straightforward), we first compute the discrete derivatives of the signal with respect to  $x$  and  $y$  using convolution with filters  $f_x$  and  $f_y$  by  $s_x = f_x * s$  and  $s_y = f_y * s$ . With the image gradient  $\mathbf{g} = \nabla s = (s_x, s_y)^T$ , we now define the (standard) structure tensor  $\mathbf{S}^{(1)} : \mathbb{R}^N \rightarrow \mathbb{R}^{N \times N}$  as local integration over the outer product of the gradient:

$$\mathbf{S}^{(1)} = f' * (\mathbf{g}\mathbf{g}^T) \quad (2)$$

where  $f'$  is an averaging filter with support  $\Omega$ . While  $\mathbf{g}\mathbf{g}^T$  is always of rank one, the structure tensor obtained by averaging may well be of full rank. Widely used choices for  $f_x$ ,  $f_y$ , and  $f'$  are horizontal and vertical derivatives of Gaussian filters resp. Gaussian lowpass filters (cf. [29]). If a  $N$ -variate signal complies perfectly with the single orientation model in (1) over  $\Omega$ , the structure tensor  $\mathbf{S}^{(1)}$  has one zero eigenvalue. The corresponding eigenvector represents the sought orientation. For noisy data or model violations, the eigenvector corresponding to the smallest eigenvalue defines the orientation in which the signal is “most constant”.

### B. Related approaches

The structure tensor  $\mathbf{S}^{(1)}$  is not the only possibility to analyse single-oriented structures. Higher order directional derivatives also vanish along the orientation [30]:

$$\frac{\partial s}{\partial \mathbf{u}} = 0 \quad \text{and} \quad \frac{\partial^2 s}{\partial \mathbf{u}^2} = 0 \quad \text{and} \quad \frac{\partial^3 s}{\partial \mathbf{u}^3} = 0 \quad \text{and} \quad \dots \quad (3)$$

which allows to design a wide class of approaches based on combinations of different order derivatives<sup>1</sup>. This freedom can be used for filter design. For instance, Granlund and Knutsson [6] give a slightly different definition of the *orientation* concept: the *invariance requirement* states that an entity which characterizes orientation must not depend on the signal variations orthogonal to the sought orientation. This defines a much stronger concept of orientation than the one in (1). For instance, in bivariate signals such as images, it forces us to design an orientation estimator such that it makes no distinction between two especially important types of oriented linear structures: “lines” (a light-dark-light signal orthogonal to the orientation direction or vice versa; also called “ridges”) and “edges” (only a single light-dark transition). A detector that reacts uniformly to these two types of structures is called *phase invariant* and can be realized with *quadrature filters* [6] (cf. also [31], [32], [22]).

If, on the contrary, we are interested in detecting line structures only (and not edges), the second order directional derivative defines an appropriate and widely used filter; this can be traced back to [33]. A recent paper [34] also discusses line-specific and edge-specific orientation estimation in the context of steerable filters. An extension of steerability towards multi-steerable filters, which are steerable to different simultaneous orientations, is described in [35], [36].

Two generalizations of the structure tensor to phase-invariant feature detectors are the *2D energy tensor* defined by Felsberg and Granlund in [14] and the *boundary tensor* by Köthe in [37]. The connections between energy tensor and boundary tensor are analyzed recently by both authors in a joint paper [38]. A variant of the energy tensor, the *gradient energy tensor*, can be found in [39]. All these approaches are based on higher (up to fourth) order derivatives.

We summarize that the standard structure tensor approach can be extended with combinations of higher order derivatives to obtain certain desired properties. Odd order filters can be optimized for edge detection, even order filters for lines, and mixed order filters for phase invariant behavior.

## II. MODELLING AND ESTIMATION OF MULTIPLE ORIENTATIONS

In the context of orientation estimation, higher order derivatives also appear in the analysis of *multiple* orientations. Despite its ability to characterize various important low level image features like lines or edges, the single orientation model underlying the above discussion is too restrictive for many real signals. For instance, the presence of two oriented textures in a region  $\Omega$  calls for an extended mathematical model. This observation led to the study of *double-oriented* signals. We will denote the multiplicity of orientations by  $M$ , so double orientation estimation means  $M = 2$ .

For image sequences, double orientation estimation means the analysis of two independent optical flows; this is the area where double orientation estimation appeared first in the beginning of the 90s in pioneering work of Shizawa and

<sup>1</sup>We assume throughout this paper that the signal  $s$  is sufficiently regular for the derivatives to exist.

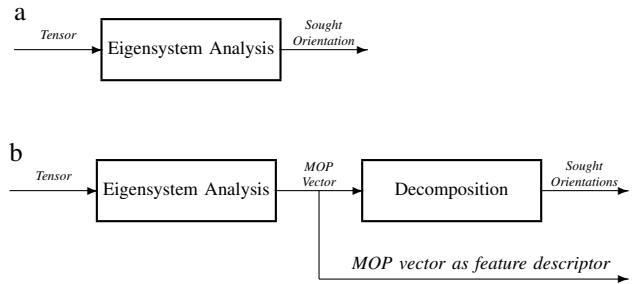


Fig. 4. While standard single orientation estimation (a) consists of only a single estimation step, multiple orientation estimation (b) is more complicated. The first step, viz., the estimation of the *mixed orientation parameters (MOP) vector*, is analogous to the single orientation case. This vector can already be used as a feature vector. However, if the orientations are explicitly needed, an additional decomposition step is required.

Mase [40], [16] (additive superposition model, gray value image sequences), followed by Shizawa and Iso [41] (additive superposition, gray value images, connection to steerable filters). More recent results can be found in [42], [43] (additive model, images; first theoretical steps towards higher multiplicity of signals beyond double orientations) and [13] (occluding model). An extensive discussion of these double-orientation approaches can be found in [17].

However, the algorithms above are still limited to the estimation of double orientations ( $M = 2$ ) in image or volume data ( $N \leq 3$ ). Here, we extend the theory of multiple orientation estimation into ‘orthogonal’ directions: *model* (additive or occluding superposition, and both of these combined), *scalar- or vector-valued signals* (gray value or multispectral), and *multiplicity  $M$*  (double orientations or  $M > 2$ ).

### A. Outline of multiple orientation estimation

The main difference between single orientation estimation, either via the standard structure tensor or via advanced variants composed from higher order derivatives, and multiple orientation estimation is that the latter is a two-step procedure. As illustrated in Fig. 4, the mathematical core of single orientation estimation is an eigensystem analysis. The resulting eigenvector (usually corresponding to the smallest eigenvalue) is a unit vector which directly represents the sought orientation.

For multiple orientation estimation, on the other hand, the eigensystem analysis is only a first step, from which a *single* eigenvector of higher dimensionality is retained, viz., the eigenvector corresponding to the lowest eigenvalue. This eigenvector *encodes* the sought *set of multiple orientations*. This vector is called the *mixed orientation parameters (MOP) vector*. Therefore, we have to apply a subsequent decomposition step after the eigensystem analysis, as shown in Fig. 4b.

In some specific applications, though, such as segmentation or classification, the MOP vector itself can directly be used as a feature vector without further decomposition, as also shown Fig. 4b.

### B. Additive and occluding orientations

Two different ways of combining  $M$  oriented signals  $s_i$ ,  $i = 1, \dots, M$ , to form a new signal  $s$  can be found in the literature. These are the *occluding orientations model* (OOM) and the *additive orientations model* (AOM):

Multiple orientation models:

$$\begin{aligned} \text{OOM : } \quad s(\mathbf{x}) &= s_i(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega_i \\ \text{AOM : } \quad s(\mathbf{x}) &= \sum_{i=1}^M \alpha_i s_i(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega . \end{aligned} \quad (4)$$

The first model states that we take the first oriented signal if the point  $\mathbf{x}$  is in some region  $\Omega_1$ , the second signal  $s_2(\mathbf{x})$  in region  $\Omega_2$ , and so on. All regions must be distinct and sum up to the entire analysis region  $\Omega$ . For instance, this model is applicable with  $M = 2$  if the region  $\Omega_1$  corresponds to some object which occludes another object observable in  $\Omega_2$ , provided that both objects can be modelled reasonably well as single-oriented structures. As another example, a model using three occluding orientations can be used to describe bifurcations of blood vessels in medical images: without any separate junction detection step, we directly detect the presence of three vessel orientations and estimate these.

The second model, AOM, assumes that all basic signals are present over the entire region  $\Omega$ , and that we observe their superposition, possibly weighted with some constants  $\alpha_i$ . Computing directional derivatives, we obtain the constraints:

Multiple orientation constraints, derivative forms:

$$\begin{aligned} \text{OOM : } \quad \prod_{i=1}^M \frac{\partial s}{\partial \mathbf{u}_i} &= 0 \quad \forall \mathbf{x} \in \Omega \\ \text{AOM : } \quad \frac{\partial^M s}{\partial \mathbf{u}_1 \cdots \partial \mathbf{u}_M} &= 0 \quad \forall \mathbf{x} \in \Omega . \end{aligned} \quad (5)$$

Note that the OOM needs only first order derivatives, since for each region  $\Omega_i$ , the  $i$ -th factor in the constraint is zero, while the AOM constraint needs  $M$ -th order derivatives.

Both models can be summarized under the more general approach  $s(\mathbf{x}) = \sum_i \alpha_i(\mathbf{x}) s_i(\mathbf{x})$ . This model is structurally similar to the AOM model, but we now allow the weights to depend on the position  $\mathbf{x}$ . Computing the  $M$ -th order directional derivatives w.r.t. all  $M$  orientations, each summand decomposes into  $2^M$  summands because of the product rule. Half of these summands contain  $\frac{\partial s_i}{\partial \mathbf{u}_i}$  (or derivatives of it), which vanish for the sought orientations. However,  $2^{(M-1)}$  summands remain in which the signal-dependent part does not vanish. In total, we obtain  $M 2^{(M-1)}$  summands containing derivatives of the weight-functions  $\alpha_i(\mathbf{x})$ . Consequently, they can only vanish for special choices of the weight functions.

As an example, we examine the general additive model with weight functions for  $M = 2$ , i.e.  $s(\mathbf{x}) = \alpha_1(\mathbf{x}) s_1(\mathbf{x}) + \alpha_2(\mathbf{x}) s_2(\mathbf{x})$ . The second order directional derivative then is

$$\frac{\partial^2 s}{\partial \mathbf{u}_1 \partial \mathbf{u}_2} = \frac{\partial^2 \alpha_1}{\partial \mathbf{u}_1 \partial \mathbf{u}_2} s_1 + \frac{\partial^2 \alpha_2}{\partial \mathbf{u}_1 \partial \mathbf{u}_2} s_2 + \frac{\partial \alpha_1}{\partial \mathbf{u}_1} \frac{\partial s_1}{\partial \mathbf{u}_2} + \frac{\partial \alpha_2}{\partial \mathbf{u}_2} \frac{\partial s_2}{\partial \mathbf{u}_1}$$

The last two summands can only vanish if either the signal  $s_i$  is constant (the only case where it can be oriented in both directions  $\mathbf{u}_1$  and  $\mathbf{u}_2$  simultaneously) or, which is the more interesting case, if we impose restrictions on the weight functions  $\alpha_i$ . If they are constant, then all derivatives vanish and we arrive at (5) for the AOM. A slightly relaxed requirement is piece-wise constancy. Then the constraint holds *almost everywhere*, i.e., everywhere except at discontinuities. The OOM case is one specific realization for this, namely where the whole region  $\Omega$  is divided into distinct regions  $\Omega_i$  with weights  $\alpha_i(\mathbf{x}) = 1$  if  $\mathbf{x} \in \Omega_i$  and 0 else. This shows that the orientations of signals generated from the OOM model can also be estimated using the AOM constraint.

With  $(\mathbf{u}_i)_j$  denoting the  $j$ -th component of vector  $\mathbf{u}_i$ , the directional derivative is defined as

$$\langle \mathbf{g}, \mathbf{u}_i \rangle = \frac{\partial s}{\partial \mathbf{u}_i} = \langle \nabla s, \mathbf{u}_i \rangle = \sum_{j=1}^N \frac{\partial s}{\partial x_j} (\mathbf{u}_i)_j . \quad (6)$$

Inserting this into the OOM constraint (5) yields

$$\prod_{i=1}^M \left( \sum_{j=1}^N \frac{\partial s}{\partial x_j} (\mathbf{u}_i)_j \right) = 0 . \quad (7)$$

The left hand side is a product of  $M$  factors which are sums consisting of  $N$  summands each. Multiplying out leads to a sum of  $N^M$  summands. By introducing the *tensor scalar product* between order- $M$  tensors as

$$\langle \mathcal{O}, \mathcal{U} \rangle := \sum_{k_1, \dots, k_M=1}^N (\mathcal{O})_{k_1 \dots k_M} (\mathcal{U})_{k_1 \dots k_M} , \quad (8)$$

we can now encapsulate the entire dependency on the sought orientations in the tensor

$$\mathcal{U} = \mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_M \quad (9)$$

where “ $\otimes$ ” denotes the tensor product operator. Then we express (7) as

Multiple occluding orientations constraint:

$$\langle \mathcal{O}, \mathcal{U} \rangle = 0 \quad \text{with} \quad (\mathcal{O})_{k_1 \dots k_M} = \prod_{i=1}^M \frac{\partial s}{\partial x_{k_i}} . \quad (10)$$

Each point in  $\Omega$  yields one *data tensor*  $\mathcal{O}$  and from all these tensors, we have to estimate the sought orientation tensor  $\mathcal{U}$  which is orthogonal to the given data tensors, i.e., leads to the scalar products being zero. Both  $\mathcal{O}$  and  $\mathcal{U}$  are  $M$ th-order  $N \times \cdots \times N$  tensors, since each index  $k_i$ ,  $i = 1, \dots, M$ , runs from 1 to  $N$ . Analogously, we find the

Multiple additive orientations constraint:

$$\langle \mathcal{A}, \mathcal{U} \rangle = 0 \quad \text{with} \quad (\mathcal{A})_{k_1 \dots k_M} = \frac{\partial^M s}{\partial x_{k_1} \cdots \partial x_{k_M}} . \quad (11)$$

We emphasize the structural similarity of both models: the tensors constructed from signal derivatives are different – product of first derivatives in  $\mathcal{O}$  versus higher order derivatives in  $\mathcal{A}$  –, but once we have constructed the data tensor, the

computation of the sought orientations, i.e., the estimation of an orientation tensor  $\mathcal{U}$  being orthogonal to it, is *exactly the same*.

For vector-valued signals  $\mathbf{s}(\mathbf{x})$ ,  $\mathbf{s} : \mathbb{R}^N \rightarrow \mathbb{R}^P$ , for instance color images, the models extend as follows: Under the AOM, the scalar AOM-constraint in (5) must hold for all signal components, while under the OOM, the scalar OOM-constraint must hold for all combinations of signal components and directional derivatives. The multispectral version of (5) thus is

$$\begin{aligned} \text{OOM} : \quad & \frac{\partial \mathbf{s}}{\partial \mathbf{u}_1} \otimes \cdots \otimes \frac{\partial \mathbf{s}}{\partial \mathbf{u}_M} = \mathbf{0} \otimes \cdots \otimes \mathbf{0} \quad \forall \mathbf{x} \in \Omega \\ \text{AOM} : \quad & \frac{\partial^M (\mathbf{s})_\ell}{\partial \mathbf{u}_1 \cdots \partial \mathbf{u}_M} = 0 \quad \forall \mathbf{x} \in \Omega, \ell = 1, \dots, P \end{aligned} \quad (12)$$

where  $(\mathbf{s})_\ell$  is the  $\ell$ -th component of  $\mathbf{s}$ , and  $\partial \mathbf{s} / \partial \mathbf{u}_i = [\mathbf{u}_i^T \nabla (\mathbf{s})_1, \dots, \mathbf{u}_i^T \nabla (\mathbf{s})_P]^T$ . For the AOM, we thus obtain  $P$  data tensors of size  $\mathbb{R}^{N \times \cdots \times N}$ :

$$(\mathcal{A})_{k_1 \cdots k_M}^\ell = \frac{\partial^M (\mathbf{s})_\ell}{\partial x_{k_1} \cdots \partial x_{k_M}} \quad (13)$$

and  $P$  constraints  $\langle \mathcal{A}^{(\ell)}, \mathcal{U} \rangle = 0$ ,  $\ell = 1, \dots, P$ , where  $\mathcal{A}^{(\ell)}$  denotes the tensor in (13) evaluated for a specific  $\ell$ . Analogously, the multispectral OOM constraint is given by  $\langle \mathcal{O}^{(\ell_1 \cdots \ell_M)}, \mathcal{U} \rangle = 0$  for all  $\ell_i = 1, \dots, P$ ,  $i = 1, \dots, M$ , and with  $\mathcal{O}^{(\ell_1 \cdots \ell_M)}$  denoting a specific slice of the tensor  $\mathcal{O} \in \mathbb{R}^{N \times \cdots \times N \times P \cdots \times P}$ :

$$(\mathcal{O})_{k_1 \cdots k_M}^{\ell_1 \cdots \ell_M} = \frac{\partial (\mathbf{s})_{\ell_1}}{\partial x_{k_1}} \cdots \frac{\partial (\mathbf{s})_{\ell_M}}{\partial x_{k_M}} \quad (14)$$

In subsection II-F, we will see that the computation of the multispectral structure tensors requires only one additional tensor contraction over  $\ell$  in the AOM case, and  $M$  tensor contractions over  $\ell_1, \dots, \ell_M$  for the OOM. The resulting structure tensors are then all of the same size, regardless of whether they are computed from scalar or vector-valued data under OOM or AOM. Orientation estimation for all these cases can thus be done in exactly the same way.

### C. Geometric interpretations of OOM and AOM

In general, all constraints for orientation estimation are orthogonality constraints in an  $N \times \cdots \times N$ -dimensional space, where a set of data tensors — evaluated at all points inside the region  $\Omega$  — are given and an orientation tensor orthogonal to it is sought. For scalar-valued signals, we can also find geometric interpretations in  $N$ -dimensional space. With (6), the OOM constraint in (5) becomes

$$\langle \mathcal{O}, \mathcal{U} \rangle = \langle \mathbf{g}, \mathbf{u}_1 \rangle \cdots \langle \mathbf{g}, \mathbf{u}_M \rangle = 0 \quad \forall \mathbf{x} \in \Omega, \quad (15)$$

i.e., the constraint factorizes into  $M$  scalar products, each containing the *same* gradient vector  $\mathbf{g}$  for a given  $\mathbf{x} \in \Omega$ . Expanding this constraint shows that it is a homogeneous polynomial of degree  $M$  in the components of the gradient  $\mathbf{g}$ . All gradients measured inside the analysis region  $\Omega$  must thus lie on a union of  $M$  hyperplanes in  $N$ -dimensional space,

each of which is described by a normal vector  $\mathbf{u}_i$ . For the AOM, the constraint in (5) can be turned into a similar form by expressing it in the Fourier domain, yielding

$$\langle \mathcal{A}, \mathcal{U} \rangle = (\langle \omega, \mathbf{u}_1 \rangle \cdots \langle \omega, \mathbf{u}_M \rangle) S_\Omega(\omega) = 0 \quad \forall \omega, \quad (16)$$

where  $S_\Omega(\omega)$  is the spectrum of  $s(\mathbf{x})$  taken over  $\Omega$ , and  $\omega$  the spatial frequency vector. If  $s(\mathbf{x})$  satisfies the AOM in  $\Omega$ ,  $S_\Omega(\omega)$  must vanish everywhere except on the  $M$  hyperplanes with normals  $\mathbf{u}_i$ ,  $i = 1, \dots, M$ . The structure of (16) is thus the same as that of (15), with the only difference that the gradients  $\mathbf{g}(\mathbf{x})$ ,  $\mathbf{x} \in \Omega$ , are replaced by the vectors  $\omega S_\Omega(\omega)$ .

The normal vectors  $\mathbf{u}_i$  can be estimated by generalized principal component analysis (GPCA, [27], [44]). Its basic idea is to first compute the coefficients of the expanded version of (15), yielding the equivalent of the MOP vector in Fig. 4. The resulting coefficients are then decomposed into the subspace normals  $\mathbf{u}_i$ ,  $i = 1, \dots, M$  by polynomial differentiating and evaluation at distinct data points  $\mathbf{g}$  [44, Sec. 3.4].

### D. Symmetry properties of the data tensors

The commutativity in the definitions of (10) and (11) is the key to the understanding of multiple orientations. The data tensors are invariant against any arbitrary permutation of indices and therefore have some very pronounced symmetry properties. For  $M = 2$ , the data tensors  $\mathcal{O}$  and  $\mathcal{A}$  are symmetric  $N \times N$  matrices, but for higher  $M$ , we cannot rely on concepts from matrix algebra anymore. We therefore define the *space of fully symmetric  $M$ -tensors*<sup>2</sup> as

$$\mathbb{R}_{\oplus}^{N \times \cdots \times N} = \left\{ \mathcal{T} \in \mathbb{R}^{N \times \cdots \times N} \mid (\mathcal{T})_{i_1 \cdots i_M} = (\mathcal{T})_{P(i_1 \cdots i_M)} \right\} \quad (17)$$

with  $P(i_1 \cdots i_M)$  denoting any arbitrary permutation of the indices  $i_1 \cdots i_M$ .<sup>3</sup>

While the data tensors are fully symmetric, the orientation tensor  $\mathcal{U} = \mathbf{u}_1 \otimes \cdots \otimes \mathbf{u}_M$  clearly is not. But the symmetry of the left operand in some scalar product, like  $\langle \mathcal{A}, \mathcal{U} \rangle$ , always implies that the value of the scalar product does not change if the same symmetry transformations are applied to the second operand (here:  $\mathcal{U}$ ). Hence, if  $\langle \mathcal{A}, \mathcal{U} \rangle = 0$ , then  $\langle \mathcal{A}, \mathcal{U}' \rangle = 0$  with  $\mathcal{U}'$  denoting any arbitrary permutation of the order of orientations in the tensor product  $\mathcal{U}$ . The interpretation is that we can only estimate a *set* of  $M$  orientations, but cannot identify them as “first”, “second”, “ $M$ -th” orientation; they are interchangeable. As a consequence, the original tensor  $\mathcal{U}$  *cannot* be recovered uniquely, any linear combination of permuted tensors is consistent with the constraint.

However, it *is* possible to describe a set of  $M$  orientations with a unique order- $M$  tensor. The key is symmetrization: among all possible orientation tensors  $\mathcal{U}$  which are orthogonal to  $\mathcal{A}$ , i.e., for which  $\langle \mathcal{A}, \mathcal{U} \rangle = 0$  holds, there is — up to a non-zero scale factor — only a single fully symmetric one.

<sup>2</sup>Fully symmetric in order to allow the term symmetric also for invariance against *special* permutations only, for instance the exchange of indices 1 and 2. *Fully symmetric* as used here means “invariant under the permutation group”.

<sup>3</sup>The space of fully symmetric tensors is a subset of general  $M$ -tensors. The sets are identical only for vectors, i.e.,  $M = 1$ , as there is no possible permutation for a single index. Therefore the statement that the data tensors are fully symmetric tensors is consistent with single orientation estimation.

It is given by the sum over all possible permuted tensors with *equal* weights. This implies that, instead of estimating a general  $\mathcal{U} \in \mathbb{R}^{N \times \dots \times N}$ , we have to estimate the orientation tensor subject to  $\mathcal{U} \in \mathbb{R}_{\oplus}^{N \times \dots \times N}$  to obtain a unique solution.

From (9) follows that the sought orientation tensor  $\mathcal{U}$  is even restricted to reside in a subset of  $\mathbb{R}_{\oplus}^{N \times \dots \times N}$ , namely the set of tensors which are elements of

$$\mathbb{R}_{\oplus}^{N \times \dots \times N} = \left\{ \sum_{P(i_1 \dots i_M)} \mathbf{u}_{i_1} \otimes \dots \otimes \mathbf{u}_{i_M} \mid \mathbf{u}_{i_1}, \dots, \mathbf{u}_{i_M} \in \mathbb{R}^N \setminus \{\mathbf{0}\} \right\}. \quad (18)$$

We will call this space of *symmetrized* outer products the *minimal fully symmetric tensors* from now on. However, it is difficult to enforce this additional constraint directly in the estimation step. Therefore, we will only estimate the orientation tensor subject to the constraint that it is an element of  $\mathbb{R}_{\oplus}^{N \times \dots \times N}$ , and impose the restriction to  $\mathbb{R}_{\oplus}^{N \times \dots \times N}$  later on in the decomposition step.

All tensor scalar products can be converted to standard scalar products by stacking the tensor elements to form a vector. The symmetry properties of both operands, however, call for a modified version of vectorization. Our fully symmetric tensors have

$$k = \sum_{i=1}^N \binom{N+M-1-i}{M-1} = \binom{N+M-1}{M} \quad (19)$$

different elements or degrees of freedom (DOF). Therefore, the space  $\mathbb{R}_{\oplus}^{N \times \dots \times N}$  can be mapped to  $\mathbb{R}^k$ . We thus define as follows:

*Definition 1:* Let  $\mathcal{T} \in \mathbb{R}_{\oplus}^{N \times \dots \times N}$  denote a fully symmetric tensor of order  $M$ . Then we define the mapping  $\text{VecSymm}(\cdot) : \mathbb{R}_{\oplus}^{N \times \dots \times N} \rightarrow \mathbb{R}^k$  with  $k$  defined in (19) as stacking all independent elements in some arbitrary but fixed order. Furthermore, we define  $\text{VecSymmN}(\cdot) : \mathbb{R}_{\oplus}^{N \times \dots \times N} \rightarrow \mathbb{N}^k$  as counting the number of appearances or index permutations of each independent element in the tensor.

Note that the  $\text{VecSymmN}(\cdot)$  operation only depends on the dimensionality of the argument, not on the entries. Thus, every element of  $\mathbb{R}_{\oplus}^{N \times \dots \times N}$  produces the same  $\text{VecSymmN}(\cdot)$  result.

As an example, we consider the estimation of two orientations in bivariate images; then all data tensors are  $2 \times 2$  matrices. For the OOM, we obtain

$$\mathbf{O} = \begin{pmatrix} s_x^2 & s_x s_y \\ s_x s_y & s_y^2 \end{pmatrix}$$

(we use  $\mathbf{O}$  instead of  $\mathcal{O}$  since the tensor is a matrix here, and  $s_x, s_y$  to denote the partial derivatives), which is then vectorized to

$$\tilde{\mathbf{g}}_{\text{OOM}} = (s_x^2, s_x s_y, s_y^2)^T.$$

and the ‘appearance counter’ yields

$$\text{VecSymmN}(\mathbf{O}) = (1, 2, 1)^T.$$

In the same way, we obtain for the AOM the matrix

$$\mathbf{A} = \begin{pmatrix} s_{xx} & s_{xy} \\ s_{xy} & s_{yy} \end{pmatrix}$$

which is vectorized to

$$\tilde{\mathbf{g}}_{\text{AOM}} = (s_{xx}, s_{xy}, s_{yy})^T,$$

and where  $s_{xx}, s_{xy}$  and  $s_{yy}$  are second-order partial derivatives. Applying these definitions to (10) and (11) now allows to generalize the single orientation constraint  $\langle \mathbf{g}, \mathbf{u} \rangle = 0$  that the gradient must be orthogonal to the sought orientation to  $\langle \tilde{\mathbf{g}}, \tilde{\mathbf{u}} \rangle = 0$  with the *generalized gradient* vector

$$\text{OOM} : \tilde{\mathbf{g}}_{\text{OOM}} = \text{VecSymm}(\mathcal{O}) \quad (20)$$

resp.

$$\text{AOM} : \tilde{\mathbf{g}}_{\text{AOM}} = \text{VecSymm}(\mathcal{A}) \quad (21)$$

and the mixed orientation parameters (MOP) vector

$$\tilde{\mathbf{u}} = \text{VecSymmN}(\mathcal{U}) \cdot \text{VecSymm}(\mathcal{U}) \quad (22)$$

with “ $\cdot$ ” indicating element-by-element multiplication. The vectors  $\tilde{\mathbf{g}}$ , where, for  $P = 1$ ,  $\tilde{\mathbf{g}} : \mathbb{R}^N \rightarrow \mathbb{R}^k$ , are the multiple orientation counterparts of the gradients. In analogy to the single orientation structure tensor (2), we can therefore define the double [triple,  $M$ -] orientation structure tensor  $\mathbf{S}^{(2)}$  [ $\mathbf{S}^{(3)}$ ,  $\mathbf{S}^{(M)}$ ] as spatial integration, i.e., convolution with averaging filter  $f'$ , over the outer product of  $\tilde{\mathbf{g}}$  with itself:

$$\mathbf{S}^{(M)} = f' * (\tilde{\mathbf{g}} \tilde{\mathbf{g}}^T). \quad (23)$$

For our example of  $M = 2$ , the structure tensor  $\mathbf{S}^{(2)}$  thus is

$$\mathbf{S}^{(2)} = f' * \tilde{\mathbf{g}}_{\text{OOM}} \tilde{\mathbf{g}}_{\text{OOM}}^T = f' * \begin{pmatrix} s_x^4 & s_x^3 s_y & s_x^2 s_y^2 \\ s_x^3 s_y & s_x^2 s_y^2 & s_x s_y^3 \\ s_x^2 s_y^2 & s_x s_y^3 & s_y^4 \end{pmatrix}$$

for the OOM, and

$$\mathbf{S}^{(2)} = f' * \tilde{\mathbf{g}}_{\text{AOM}} \tilde{\mathbf{g}}_{\text{AOM}}^T = f' * \begin{pmatrix} s_{xx}^2 & s_{xx} s_{xy} & s_{xx} s_{yy} \\ s_{xx} s_{xy} & s_{xy}^2 & s_{xy} s_{yy} \\ s_{xx} s_{xy} & s_{xy} s_{yy} & s_{yy}^2 \end{pmatrix}$$

for the AOM.

For  $M = 3$  orientations, we obtain in the same way  $\tilde{\mathbf{g}}_{\text{OOM}} = (s_x^3, s_x^2 s_y, s_x s_y^2, s_y^3)^T$  and  $\tilde{\mathbf{g}}_{\text{AOM}} = (s_{xxx}, s_{xxy}, s_{xyy}, s_{yyy})^T$ , yielding structure tensors  $\mathbf{S}^{(3)}$  of size  $4 \times 4$ .

The eigenvector of  $\mathbf{S}^{(M)}$  corresponding to the smallest eigenvalue — which, ideally, is equal to zero — will then yield the MOP vector for  $M$  orientations, thus completing the first step of Fig. 4. Depending on  $M$  and  $N$ , though, the MOP vector is overparameterized in the sense that it has clearly too many DOF:  $k$  is a polynomial with leading term  $N^M$ , whereas  $M$  unit vectors only have  $M(N-1)$  DOF. Still, the MOP vector itself — without decomposition into the underlying orientations — could already be used as a feature for applications like texture classification or segmentation.

### E. Polynomial representation via the Veronese Map

Under the OOM, the entries in the generalized gradient vector  $\tilde{\mathbf{g}}_{\text{OOM}}$  as defined in (20) are all monomials of degree  $M$  which can be formed from the gradient components  $\partial s / \partial x_i$ . Similarly under the AOM, the entries in  $\tilde{\mathbf{g}}_{\text{AOM}}$  as defined in (21) are all derivatives of the order  $M$  which can be formed with respect to the coordinate axes  $x_i$ . For the latter, this can formally be interpreted as first composing all monomials of degree  $M$  of the components of the gradient operator  $\nabla$  and applying the result to  $s$ , generating higher-order derivatives

instead of products of first-order derivatives. The mapping between a vector (here:  $\nabla s$  resp.  $\nabla$  itself) and the set of all degree- $M$  monomials which can be constructed from its components is the *Veronese map* [45]. These monomials form a basis for the set of homogeneous polynomials of the same degree [45], [44].

A homogeneous polynomial of degree  $M$  is a polynomial  $p(\mathbf{x})$  such that  $p(\lambda\mathbf{x}) = \lambda^M p(\mathbf{x})$  for all  $\lambda \in \mathbb{R}$ . As already discussed in section II-C, the scalar OOM constraints (5) and (15) are homogeneous polynomials. The homogeneous polynomials of degree  $M$  in  $N$  variables form a vector space of dimension

$$D_M(N) = \binom{M+N-1}{N-1} = \binom{M+N-1}{M} \quad (24)$$

which is the same number as  $k$  in (19).

A basis for this space is given by the set of all monomials of degree  $M$  in  $N$  variables. We denote this basis as  $\mathbf{x}^I = x_1^{m_1} x_2^{m_2} \dots x_N^{m_N}$  with  $0 \leq m_j \leq M$  for  $j = 1, \dots, N$ , and  $\sum_{j=1}^N m_j = M$  [44]. Each homogeneous polynomial can then be written as a linear combination of the monomials  $\mathbf{x}^I$  with some coefficient vector  $\mathbf{c} \in \mathbb{R}^{D_M(N)}$  as

$$p(\mathbf{x}) = \mathbf{c}^T \mathbf{v}_M(\mathbf{x}) = \sum c_{m_1, m_2, \dots, m_N} x_1^{m_1} x_2^{m_2} \dots x_N^{m_N}, \quad (25)$$

where  $\mathbf{v}_M(\mathbf{x}) : \mathbb{R}^N \rightarrow \mathbb{R}^{D_M(N)}$  is the Veronese map of degree  $M$ , defined as  $\mathbf{v}_M(\mathbf{x}) : (x_1, x_2, \dots, x_N)^T \mapsto (\dots, \mathbf{x}^I, \dots)$  with  $I$  chosen in degree-lexicographic order. For instance, the Veronese map of degree 2 of the vector  $(x, y)^T \in \mathbb{R}^2$  is the vector  $(x^2, xy, y^2)^T \in \mathbb{R}^3$ .

Using these definitions, one can also define the generalized gradients for either of the two data models without reference to tensor-valued entities and vectorization of such tensors. We find

$$\tilde{\mathbf{g}}_{\text{OOM}} = \mathbf{v}_M(\nabla s) \quad (26)$$

and

$$\tilde{\mathbf{g}}_{\text{AOM}} = \mathbf{v}_M(\nabla) s \quad (27)$$

where  $\mathbf{v}_M(\nabla)$  has to be understood as applying the Veronese mapping to the nabla operator, yielding operators of higher order derivatives. The OOM- and AOM constraints in (5) can then alternatively be expressed as

$$\tilde{\mathbf{u}}^T \mathbf{v}_M(\nabla s) = 0 \quad \forall \mathbf{x} \in \Omega \quad (28)$$

and

$$\tilde{\mathbf{u}}^T \mathbf{v}_M(\nabla) s = 0 \quad \forall \mathbf{x} \in \Omega \quad (29)$$

respectively, with  $\tilde{\mathbf{u}}$  being the mixed orientations parameter vector defined in (22). For the OOM, the constraint equation (28) is a homogeneous polynomial in the gradient components, in fact, (28) is the expanded version of (15). As discussed in section II-C, this constraint can also be derived by first calculating all gradients in  $\Omega$ , and then requiring each one to lie in a subspace formed by a union of  $M$  hyperplanes, showing that both the polynomial and tensor representation are thus appropriate under the OOM. Although slightly different for the AOM, the tensor formulation allows the unified treatment of both models via the structure tensors  $\mathbf{S}^{(M)}$  given in (23). Also, it allows to distinguish between the spaces

$\mathbb{R}_{\oplus}^{N \times \dots \times N}$  of all fully symmetric tensors, which is equivalent to all potential MOP *estimates*, and  $\mathbb{R}_{\otimes}^{N \times \dots \times N}$  which contains only those fully symmetric tensors which can be formed by tensor products of orientation vectors. In the polynomial formulation, this corresponds to distinguishing between the space of homogeneous polynomials which comply with (28), and the subset of these which can be factorized into the normals  $\mathbf{u}_i$  in (15). In the next section, we show how structure tensors for vector-valued data can be computed.

### F. Vector-valued signals

Based on work by Di Zenzo [4] and Förstner [46] on multi-band gradients, we now derive multiple orientations structure tensors for vector-valued signals  $\mathbf{s}(\mathbf{x})$ ,  $\mathbf{s} : \mathbb{R}^N \rightarrow \mathbb{R}^P$ . For a vector-valued signal with  $P$  components, the gradient is a  $N \times P$  matrix containing the gradients of the components as columns, the generalized gradients  $\tilde{\mathbf{g}}_{\text{OOM}}$  and  $\tilde{\mathbf{g}}_{\text{AOM}}$  become  $k \times P$  matrices, and the directional derivatives are therefore  $P$ -dimensional vectors. Products of scalars then become scalar products of vectors, i.e., sums over the products of their components. Let

$$\tilde{\mathbf{g}}_{\text{AOM}}^\ell, \quad \ell = 1, \dots, P \quad (30)$$

denote the vectorized version of  $\mathcal{A}^{(\ell)}$  in (13), i.e., the  $\ell$ -th column of the matrix  $\tilde{\mathbf{g}}_{\text{AOM}}$ , and let

$$\tilde{\mathbf{g}}_{\text{OOM}}^{\ell_1 \dots \ell_M}, \quad \ell_i = 1, \dots, P, \quad i = 1, \dots, M \quad (31)$$

denote the vectorized  $\ell_1 \dots \ell_M$ -slice of the multispectral version of  $\mathcal{O}$  defined in (14). Then structure tensor computation for vector-valued signals turns into tensor contraction over all  $P$ -dimensional indices, i.e., local integration over

$$\sum_{\ell=1}^P \tilde{\mathbf{g}}_{\text{AOM}}^\ell \otimes \tilde{\mathbf{g}}_{\text{AOM}}^\ell \quad (32)$$

resp.

$$\sum_{\ell_1=1}^P \dots \sum_{\ell_M=1}^P \tilde{\mathbf{g}}_{\text{OOM}}^{\ell_1 \dots \ell_M} \otimes \tilde{\mathbf{g}}_{\text{OOM}}^{\ell_1 \dots \ell_M}. \quad (33)$$

For instance, multispectral AOM orientation estimation for  $N = 2$  and  $M = 2$  yields the tensor

$$\mathbf{S}^{(2)} = f' * \begin{pmatrix} \langle \mathbf{s}_{xx}, \mathbf{s}_{xx} \rangle & \langle \mathbf{s}_{xx}, \mathbf{s}_{xy} \rangle & \langle \mathbf{s}_{xx}, \mathbf{s}_{yy} \rangle \\ \langle \mathbf{s}_{xx}, \mathbf{s}_{xy} \rangle & \langle \mathbf{s}_{xy}, \mathbf{s}_{xy} \rangle & \langle \mathbf{s}_{xy}, \mathbf{s}_{yy} \rangle \\ \langle \mathbf{s}_{xx}, \mathbf{s}_{yy} \rangle & \langle \mathbf{s}_{xy}, \mathbf{s}_{yy} \rangle & \langle \mathbf{s}_{yy}, \mathbf{s}_{yy} \rangle \end{pmatrix}. \quad (34)$$

where  $\mathbf{s}_{xx}$ ,  $\mathbf{s}_{xy}$  and  $\mathbf{s}_{yy}$  denote the vector-valued derivatives. Their scalar products realize the summation over  $\ell$  in (32). For the AOM, it is worth mentioning that convolution (local integration) and the sum over the signal bands commute. It is therefore mathematically equivalent to compute structure tensors for all  $P$  bands individually and then sum up these tensors in the end. For the OOM,  $\mathbf{S}^{(2)}$  is with (33) given as local integration over

$$\begin{pmatrix} \langle \mathbf{s}_x, \mathbf{s}_x \rangle^2 & \langle \mathbf{s}_x, \mathbf{s}_x \rangle \langle \mathbf{s}_x, \mathbf{s}_y \rangle & \langle \mathbf{s}_x, \mathbf{s}_y \rangle^2 \\ \langle \mathbf{s}_x, \mathbf{s}_x \rangle \langle \mathbf{s}_x, \mathbf{s}_y \rangle & \frac{1}{2} \langle \mathbf{s}_x, \mathbf{s}_x \rangle \langle \mathbf{s}_y, \mathbf{s}_y \rangle + \frac{1}{2} \langle \mathbf{s}_x, \mathbf{s}_y \rangle^2 & \langle \mathbf{s}_x, \mathbf{s}_y \rangle \langle \mathbf{s}_y, \mathbf{s}_y \rangle \\ \langle \mathbf{s}_x, \mathbf{s}_y \rangle^2 & \langle \mathbf{s}_x, \mathbf{s}_y \rangle \langle \mathbf{s}_y, \mathbf{s}_y \rangle & \langle \mathbf{s}_y, \mathbf{s}_y \rangle^2 \end{pmatrix}. \quad (35)$$



In summary, we can thus compute structure tensors for an arbitrary number  $M$  of orientations, occludingly or additively superposed, in arbitrary  $P$ -dimensional and  $N$ -variate signals.

### III. DECOMPOSITION OF THE MOP VECTOR

Once an estimate  $\hat{\mathbf{u}}$  for the MOP vector is computed, we first reverse (22) by dividing each component by the corresponding number of permutations, i.e., by the corresponding entry of  $\text{VecSymmN}(\cdot)$ . Then the mapping itself is reversed, thus producing an estimate  $\hat{\mathbf{U}}$  which is a fully symmetric tensor, i.e., an element of  $\mathbb{R}_{\oplus}^{N \times \dots \times N}$ . However, this tensor is in general not an element of the space  $\mathbb{R}_{\otimes}^{N \times \dots \times N}$  of minimal fully symmetric tensors defined in (18). Therefore, our estimate does not generally represent a valid set of  $M$  orientations. Going back to vector representation, this implies that the set of *valid MOP vectors* is a *subset* of  $\mathbb{R}^k$ . The outward sign of this discrepancy is the overparametrization of the MOP vector estimates discussed at the end of section II-D. Consequently, the above estimated MOP vector does therefore not necessarily represent a vector which can be decomposed into the sought orientations, i.e., into the unit vectors  $\mathbf{u}_i$ , since its decomposability was nowhere enforced. This problem does not arise in single orientation estimation, because there  $\mathbb{R}_{\otimes}^N \equiv \mathbb{R}_{\oplus}^N \equiv \mathbb{R}^N$  holds – there are no indices which allow permutations. The tensor notation allows a convenient characterization of the valid subspace  $\mathbb{R}_{\otimes}^{N \times \dots \times N}$  within  $\mathbb{R}_{\oplus}^{N \times \dots \times N}$ . In the equivalent vector representation, this subset is some highly complicated  $M(N-1)$ -dimensional manifold within  $\mathbb{R}^k$ , which is homeomorphic to the set of homogeneous polynomials of degree  $M$  in  $N-1$  variables that factorize as a product of linear factors. In the following, we discuss decomposition techniques where we focus on images ( $N=2$ ), but also point out one approach for higher-dimensional data.

#### A. Multiple orientation estimation for images

For bivariate images, i.e.  $N=2$ , we find with (19) that the MOP vectors exhibit  $k = \binom{2+M-1}{M} = \binom{M+1}{M} = M+1$  components. A MOP vector is the eigenvector corresponding to the lowest eigenvalue of  $\mathbf{S}^{(M)} = f' * (\tilde{\mathbf{g}}\tilde{\mathbf{g}}^T)$  in (23) which, in the case of  $M$  ideal orientations being present in  $\Omega$ , is equal to zero. The MOP vector is therefore a so-called homogeneous vector which can only be determined up to scale and sign [15, section 13.1]. Its length can therefore be normalized to one, and it thus has  $k-1 = M$  DOF. Simultaneously,  $M$  unit orientation vectors in bivariate data together have  $M$  DOF, too. Therefore, the problem of overparameterized MOP vectors does not appear in images, regardless of the number of orientations sought. This means that we have to qualify the above considerations: for images, and – apart from univariate signals where orientation cannot be defined – only for images, the MOP vector is in fact a minimal description of the sought parameters. For  $M=2$ , decomposition techniques via roots of a polynomial can be found in [27], [47], [42], [43], [17]. For  $M > 2$ , the approach can be extended as follows:  $M=3$ ,

```

1 function solution = solveMultiOri(vec)
2 % test degree of polynomial
3 if abs(vec(1)) < 1e-10
4 % at least one vertical orientation : recursion
5 verticalOri = [0; 1];
6 solution = [verticalOri solveMultiOri(vec(2:end))]
7 else
8 % solve polynomial for orientation vector slopes
9 len = length(vec);
10 temp = [1 -1 1 -1 1 -1];
11 firstRow = vec(2:len)' / vec(1) .* temp(1:len-1);
12 M = [firstRow; eye(len-2) zeros(len-2,1)];
13 temp = eig(M);
14 solution = [ones(1,len-1); temp'];
15 % normalize solutions
16 for i = 1:len-1
17     temp = solution(:,i);
18     solution(:,i) = temp/norm(temp);
19 end
20 end

```

Listing L-1. Decomposition of the MOP vector for bivariate data and arbitrary  $M$ .

for instance, yields for the components of the MOP vector

$$\begin{aligned}
(\tilde{\mathbf{u}})_1 &= (\mathbf{u}_1)_x (\mathbf{u}_2)_x (\mathbf{u}_3)_x \\
(\tilde{\mathbf{u}})_2 &= (\mathbf{u}_1)_x (\mathbf{u}_2)_x (\mathbf{u}_3)_y + (\mathbf{u}_1)_x (\mathbf{u}_2)_y (\mathbf{u}_3)_x + (\mathbf{u}_1)_y (\mathbf{u}_2)_x (\mathbf{u}_3)_x \\
(\tilde{\mathbf{u}})_3 &= (\mathbf{u}_1)_x (\mathbf{u}_2)_y (\mathbf{u}_3)_y + (\mathbf{u}_1)_y (\mathbf{u}_2)_x (\mathbf{u}_3)_y + (\mathbf{u}_1)_y (\mathbf{u}_2)_y (\mathbf{u}_3)_x \\
(\tilde{\mathbf{u}})_4 &= (\mathbf{u}_1)_y (\mathbf{u}_2)_y (\mathbf{u}_3)_y ;
\end{aligned}$$

extension to arbitrary  $M$  is straightforward. We thus have to solve an equation system of  $k = M+1$  equations, where the vector elements on the left-hand sides are taken from the MOP vector normalized to unit length; this reduces the number of independent equations to  $M=3$ . The system possesses  $2M$  unknowns, because every orientation vector consists of two unknowns  $(\mathbf{u}_i)_x$  and  $(\mathbf{u}_i)_y$ . With the additional  $M$  constraints  $((\mathbf{u}_i)_x)^2 + ((\mathbf{u}_i)_y)^2 = 1$ , solving this equation system is feasible. We reformulate the problem as an unconstrained problem: If  $(\tilde{\mathbf{u}})_1 = 0$ , then one of the sought orientations, say  $\mathbf{u}_M$ , is  $(0, 1)^T$ , and the degree of the problem is reduced by 1. Otherwise, we form the new unknowns  $z_i := \frac{(\mathbf{u}_i)_y}{(\mathbf{u}_i)_x}$ . The set of  $M$  values for the  $z_i$  – which can be regarded as the slopes of the orientation vectors – can now be found as roots of a polynomial; for  $M=3$ , the polynomial is

$$(\tilde{\mathbf{u}})_1(z-z_1)(z-z_2)(z-z_3) = (\tilde{\mathbf{u}})_1 z^3 - (\tilde{\mathbf{u}})_2 z^2 + (\tilde{\mathbf{u}})_3 z - (\tilde{\mathbf{u}})_4 = 0 . \quad (36)$$

For  $M$  arbitrary, the polynomial is

$$\sum_{j=0}^M (-1)^j (\tilde{\mathbf{u}})_{j+1} z^{M-j} = 0 . \quad (37)$$

In all cases, the polynomial can directly be formed from the given components  $(\tilde{\mathbf{u}})_{j+1}$  of the MOP vector. Note that the actual length of the MOP vector cancels out when calculating  $p_i$ , thus, it has no influence on the solution. The  $M$  orientation vectors are finally given by normalizing the vectors  $(1, z_i)^T$ ,  $i = 1, \dots, M$ , combined with the vertical orientation  $(0, 1)$  if applicable. Listing L-1 shows the implementation of this algorithm in MATLAB.

#### B. Double orientation estimation in multivariate signals

For double orientation estimation, all tensors are matrices. Matrix algebra offers a convenient interpretation of the dif-

ference between the space of fully symmetric tensors  $\mathbb{R}_{\oplus}^{N \times N}$  and its subspace  $\mathbb{R}_{\otimes}^{N \times N}$ . While the first space is the space of symmetric  $N \times N$  matrices, the latter space is the space of  $N \times N$ -matrices formed by  $\mathbf{u}_1 \otimes \mathbf{u}_2 + \mathbf{u}_2 \otimes \mathbf{u}_1$ , i.e., the space of symmetric *rank-2* matrices of size  $N \times N$ . Methods to decompose such matrices into  $\mathbf{u}_1$  and  $\mathbf{u}_2$  can be found in, e.g., [40], and in [43], [48], and are outlined in the following. After estimating the MOP vector  $\tilde{\mathbf{u}}$ , it is mapped to the space of fully symmetric tensors, which is now equivalent to the space of symmetric  $N \times N$ -matrices, including the division by the permutation count according to (22). Let  $\mathbf{U}$  denote the result of this operation; we now have to find the two unit vectors  $\mathbf{u}_1$  and  $\mathbf{u}_2$  which fulfill

$$\mathbf{U} = c(\mathbf{u}_1 \otimes \mathbf{u}_2 + \mathbf{u}_2 \otimes \mathbf{u}_1) \quad (38)$$

for some nonzero scaling factor  $c \in \mathbb{R}$ . It can be shown that  $\mathbf{u}_1 + \mathbf{u}_2$  and  $\mathbf{u}_1 - \mathbf{u}_2$  are eigenvectors of  $\mathbf{U}$ . Being a rank-2 matrix,  $\mathbf{U}$  has only two non-zero eigenvalues, one of which is positive while the other one is negative. With  $\lambda_+, \lambda_-$  denoting the non-zero eigenvalues and  $\mathbf{x}_+, \mathbf{x}_-$  the corresponding eigenvectors of  $\mathbf{U}$ , the sought orientations can be found by

$$\begin{aligned} \mathbf{u}_1 &= \sqrt{\lambda_+} \mathbf{x}_+ + \sqrt{-\lambda_-} \mathbf{x}_- \quad \text{and} \\ \mathbf{u}_2 &= \sqrt{\lambda_+} \mathbf{x}_+ - \sqrt{-\lambda_-} \mathbf{x}_-. \end{aligned} \quad (39)$$

In practice, the estimate  $\mathbf{U}$  will exhibit  $N - 2$  eigenvalues which are close to zero, and which lie between  $\lambda_+$  and  $\lambda_-$ . Ignoring these intermediate eigenvalues corresponds to replacing  $\mathbf{U}$  by its closest symmetric rank-2 matrix in the sense of the Froebenius norm, from which the orientations are computed as specified in (39).

### C. Combining both strategies: Two occluding areas with double additive orientations

The unified treatment of additive and occluding orientations enables us to deal also with the simultaneous presence of both models within one region  $\Omega$ . In the following, we assume that we observe the occlusion of two signals each of which consists of an additive superposition of two oriented signals. Examples are given in Figs. 6 and 7. The model thus is

$$s(\mathbf{x}) = \begin{cases} s_1(\mathbf{x}) + s_2(\mathbf{x}) & \mathbf{x} \in \Omega_1 \\ s_3(\mathbf{x}) + s_4(\mathbf{x}) & \mathbf{x} \in \Omega_2 \end{cases} \quad (40)$$

Let  $\mathbf{u}_1$  to  $\mathbf{u}_4$  denote the corresponding orientation vectors. The combined OOM-AOM-model constraint is the product of two AOM constraints

$$\frac{\partial^2 s}{\partial \mathbf{u}_1 \partial \mathbf{u}_2} \cdot \frac{\partial^2 s}{\partial \mathbf{u}_3 \partial \mathbf{u}_4} = 0 \quad (41)$$

where, in region  $\Omega_1$ , the first factor is zero; otherwise, the second factor is zero. The first AOM-factor is with (11)

$$\frac{\partial^2 s}{\partial \mathbf{u}_1 \partial \mathbf{u}_2} = \langle \mathbf{A}, \mathbf{U}_{12} \rangle = \tilde{\mathbf{g}}_{\text{AOM}}^T \mathbf{c}_{12} \quad (42)$$

where  $\mathbf{A}$  is the matrix for additive double orientation estimation

$$\mathbf{A} = \begin{pmatrix} \frac{\partial^2 s}{\partial x_1^2} & \cdots & \frac{\partial^2 s}{\partial x_1 \partial x_N} \\ \vdots & & \vdots \\ \frac{\partial^2 s}{\partial x_1 \partial x_N} & \cdots & \frac{\partial^2 s}{\partial x_N^2} \end{pmatrix}, \quad (43)$$

and with  $\mathbf{U}_{12} = \frac{1}{2}(\mathbf{u}_1 \mathbf{u}_2^T + \mathbf{u}_2 \mathbf{u}_1^T)$  as in (9). From  $\mathbf{A}$ ,  $\tilde{\mathbf{g}}_{\text{AOM}}$  is formed as in (21) by  $\tilde{\mathbf{g}}_{\text{AOM}} = \text{VecSymm}(\mathbf{A})$ , while the vector  $\mathbf{c}_{12}$  is computed from the tensor  $\mathbf{U}_{12}$  as in (22) by  $\mathbf{c}_{12} = \text{VecSymmN}(\mathbf{U}_{12}) \cdot \text{VecSymm}(\mathbf{U}_{12})$ . Applying the same to the second factor in (41), the constraint can be rewritten to

$$\begin{aligned} \frac{\partial^2 s}{\partial \mathbf{u}_1 \partial \mathbf{u}_2} \cdot \frac{\partial^2 s}{\partial \mathbf{u}_3 \partial \mathbf{u}_4} &= \mathbf{c}_{12}^T \tilde{\mathbf{g}}_{\text{AOM}} \mathbf{c}_{34}^T \tilde{\mathbf{g}}_{\text{AOM}} \\ &= \langle \tilde{\mathbf{g}}_{\text{AOM}} \tilde{\mathbf{g}}_{\text{AOM}}^T, \mathbf{c}_{12} \mathbf{c}_{34}^T \rangle = 0. \end{aligned} \quad (44)$$

The constraint now exhibits the form of a double orientation constraint with the only exception that, instead of two orientation vectors, we estimate here the mixed orientation parameter vectors  $\mathbf{c}_{12}$  and  $\mathbf{c}_{34}$  for  $s_1(\mathbf{x}) + s_2(\mathbf{x})$  and  $s_3(\mathbf{x}) + s_4(\mathbf{x})$ , respectively. With the two techniques discussed in the previous subsections, we solve this problem in a two-step scheme: we first determine an estimate for the symmetrized form  $\mathbf{c}_{12} \mathbf{c}_{34}^T + \mathbf{c}_{34} \mathbf{c}_{12}^T$  by the eigensystem analysis of the structure tensor  $\mathbf{S}^2 = f' * (\tilde{\mathbf{g}}_{\text{AOM}} \tilde{\mathbf{g}}_{\text{AOM}}^T)$  formed from  $\tilde{\mathbf{g}}_{\text{AOM}}$  according to (23). As described in section III-B, this estimate is, if necessary, replaced by the rank-2 matrix closest to it, and decomposed into the two MOP vectors  $\mathbf{c}_{12}$  and  $\mathbf{c}_{34}$ . These, in turn, are then decomposed into the sought orientation vector pairs  $\mathbf{u}_1, \mathbf{u}_2$  and  $\mathbf{u}_3, \mathbf{u}_4$ , respectively.

The hierarchical two-step procedure is also applicable if more than two occluding regions with more than two additive orientations in each are present. In this case, each step solve for more than two orientations using correspondingly extended structure tensors  $\mathbf{S}^{(M)}$ , and MOP vector decomposition techniques as in section III-A.

## IV. RESULTS

Here, we provide a variety of results obtained for synthetic data and real images from applications such as medicine, texture analysis and biology. We use the following notation: AOM and OOM denote the model under which the corresponding result was obtained, RGB indicates that the structure tensor was calculated for color images. Partial derivatives in horizontal and vertical directions were calculated by convolution with finite filter kernels, where DoG1.2 denotes the derivative of a Gaussian of width  $\sigma = 1.2$ , while  $\text{D}(1,0,-1)/\text{L}(1,3,1)$  denotes a filter consisting of the finite-difference kernel  $[1,0,-1]$  combined with orthogonal smoothing by the lowpass  $[1,3,1]$ . The eigenvalues of the multi-orientation structure tensor  $\mathbf{S}^{(M)}$  are for now denoted by  $\lambda_i, i = 1, \dots, M + 1$ .

Fig. 5 shows results for synthetic local neighborhoods with three orientations in noise (SNR 6 dB), both obtained using the  $\text{D}(1,0,-1)/\text{L}(1,3,1)$  filters. For the additive superposition, the estimation errors were  $1.6^\circ, 1^\circ$  and  $0.7^\circ$ . The ratios of the eigenvalues were  $\lambda_2/\lambda_1 = 0.62, \lambda_3/\lambda_1 = 0.08, \lambda_4/\lambda_1 = 0.02$ . In the noise-free case (not shown), the errors reduced to  $0.5^\circ, 0.4^\circ$  and  $0.2^\circ$ , and the eigenvalues were  $\lambda_2/\lambda_1 =$

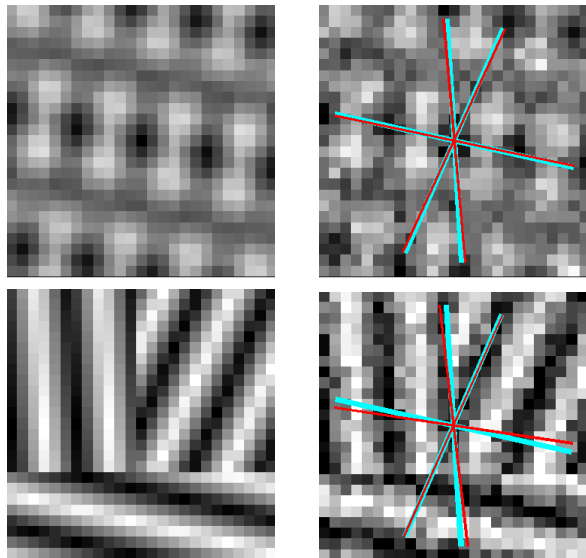


Fig. 5. Triple-oriented neighborhoods,  $25 \times 25$  pixel, generated synthetically from triangle-shaped waves. Upper row, left: additive superposition, noise-free; right: with additive Gaussian noise, SNR 6dB, true and estimated orientations (AOM) shown in light and dark gray, respectively. Lower row, left: occluding superposition, noise free; right: with additive Gaussian noise, SNR 6dB, and true and estimated orientations (OOM).

SNR	12dB	6dB	3dB	0dB
OOM	0.4	1.1	1.8	2.99
AOM	0.33	1.14	1.74	3.23

TABLE I

MEAN ABSOLUTE ESTIMATION ERROR (20 REALISATIONS) IN DEGREES OVER SNR FOR THE EXAMPLES IN FIG. 5.

0.6,  $\lambda_3/\lambda_1 = 0.07$ ,  $\lambda_4/\lambda_1 = 0.0004$ . Note that the last eigenvalue practically vanishes, indicating a practically perfect compliance of the neighborhood with the model. For the occluding superposition in noise, the estimation errors using the same filters were  $4.1^\circ$ ,  $3.3^\circ$  and  $0.5^\circ$ , with eigenvalues  $\lambda_2/\lambda_1 = 0.56$ ,  $\lambda_3/\lambda_1 = 0.07$ ,  $\lambda_4/\lambda_1 = 0.04$ . When using the DoG1.2 filter (not shown), the errors reduced to  $2.1^\circ$ ,  $0.6^\circ$  and  $0.2^\circ$ . In the noise-free occluding case (not shown), the errors were  $1.1^\circ$ ,  $0.5^\circ$  and  $0.1^\circ$ , with  $\lambda_4/\lambda_1 = 0.004$  again indicating a practically perfect model compliance. The mean absolute estimation errors for various SNRs are given in table I.

Fig. 6 shows the estimation result for a synthetically generated occluding superposition of two additive double-oriented signals, as described in section III-C, yielding a fourfold-oriented signal. Using the DoG1.2 filter, the errors in the noise-free case were  $2.7^\circ$ ,  $2.5^\circ$ ,  $2.1^\circ$  and  $1.5^\circ$ . With additive Gaussian noise at SNR 6dB (not shown), the errors became  $3.7^\circ$ ,  $3.6^\circ$ ,  $2.8^\circ$  and  $1.7^\circ$ . Fig. 7 shows the same for a fourfold-oriented image generated by combining a double-oriented texture occludingly with a rotated version of itself. The rotation angle was  $40^\circ$ , followed by nearest-neighbor interpolation. The upper right image shows the estimation result for the region indicated by the square. Since in this case no ground truth is available, the double orientations estimated

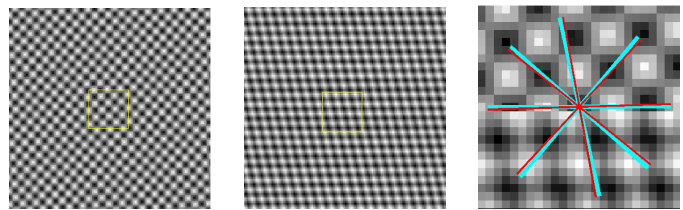


Fig. 6. Two synthetic additive double-orientation signals (left and center). Their highlighted neighborhoods are combined occludingly (right). The four orientations were estimated as described in section III-C, and are overlaid.

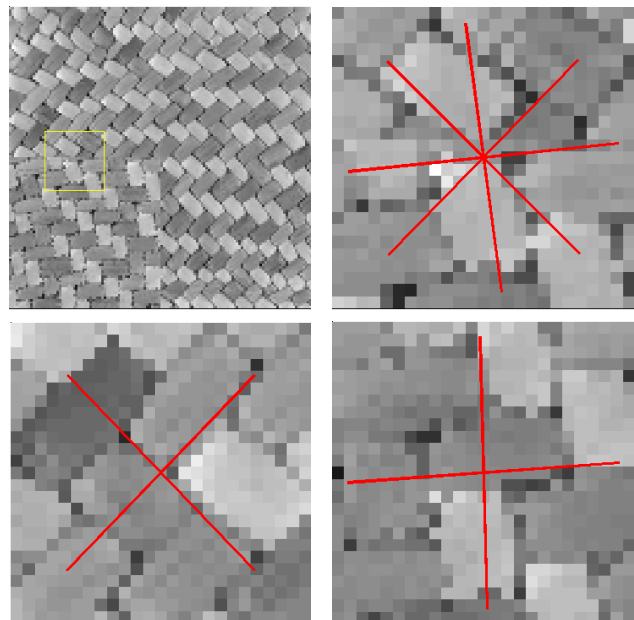


Fig. 7. Fourfold oriented neighborhoods generated by the occluding combination of a double-oriented texture signal with a rotated version of itself. From left to right and top to bottom: combined texture,  $128 \times 128$  pixel, with an occluding neighborhood inside the square; estimation result ( $D(1,0,-1)/L(1,8,1)$ ); double orientations estimated for the original texture alone; and double orientations estimated for the rotated texture alone.

for the two individual textures in regions close to the fourfold-neighborhood are provided in the lower row for comparison. Evidently, the four estimated orientations indeed reflect each of the individually estimated pairs.

Results for all examples shown in Fig. 3 are provided in Fig. 8. Results for the color images (doll house gable, bladder wall, crocodile) were computed by the AOM for  $P = 3$  as described in section II-F. In each case, the estimated orientations agree evidently well with the ones actually present. For the ROI in the bladder wall, the orientations were correctly identified by the multispectral algorithm even though they are only very weakly pronounced. In the crocodile drawing, two of the three detected orientations correspond to strokes by a pencil in dark gray, while the third corresponds to strokes by a green crayon. The latter is almost imperceptible in the corresponding luminance component.

Fig. 9 shows a part of an X-ray image of a tire, revealing its internal metal grating structure, which results in a varying number of orientations across the image. Both the number of orientations and the orientations themselves were estimated

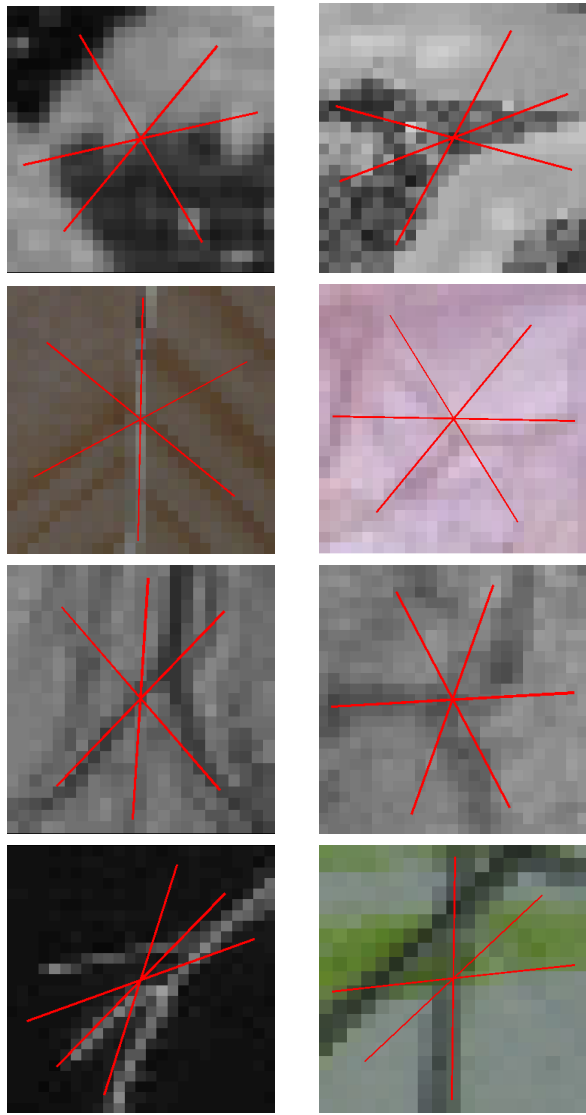


Fig. 8. Results for the highlighted regions of the images in Fig. 3. From left to right and top to bottom: carpet, OOM, DoG1.2,  $\lambda_2/\lambda_1 = 0.48$ ,  $\lambda_3/\lambda_1 = 0.07$ ,  $\lambda_4/\lambda_1 = 0.06$ ; toy spider: OOM, DoG1.2,  $\lambda_2/\lambda_1 = 0.28$ ,  $\lambda_3/\lambda_1 = 0.07$ ,  $\lambda_4/\lambda_1 = 0.01$ ; doll house gable: AOM-RGB, DoG1.2; bladder wall: AOM-RGB, DoG1.2; coloscopy: AOM, D(1,0,-1)/L(1,8,1),  $\lambda_2/\lambda_1 = 0.3$ ,  $\lambda_3/\lambda_1 = 0.07$ ,  $\lambda_4/\lambda_1 = 0.04$ ; angiography: AOM, DoG1.2,  $\lambda_2/\lambda_1 = 0.67$ ,  $\lambda_3/\lambda_1 = 0.09$ ,  $\lambda_4/\lambda_1 = 0.07$ ; plant roots: OOM, DoG1.2,  $\lambda_2/\lambda_1 = 0.3$ ,  $\lambda_3/\lambda_1 = 0.03$ ,  $\lambda_4/\lambda_1 = 0.003$ ; crocodile: AOM-RGB, DoG1.2.

within a sliding window  $\Omega$ , subject to the AOM (DoG0.8, with the integration over  $\Omega$  realized as a convolution with a Gaussian lowpass with  $\sigma_\Omega = 1.6$ ). The number of orientations was detected by hierarchical testing, starting with  $M = 1$  and increasing the multiplicity by one if the assumed model did not fit. Model validation exploits the fact that an assumed hypothesis of, say,  $M = 2$  orientations is correct if and only if the smallest eigenvalue  $\lambda_{M+1}$  of the data tensor  $\mathbf{S}^{(M)}$  is significantly smaller than all other eigenvalues. Let  $\lambda_i^{(M)}$  with  $i = 1, \dots, M+1$  denote the  $M+1$  eigenvalues of  $\mathbf{S}^{(M)}$ , where the superscript ( $M$ ) now indicates that each eigenvalue depends on the size  $(M+1) \times (M+1)$  of  $\mathbf{S}^{(M)}$ , i.e., the second eigenvalue  $\lambda_2^{(2)}$  of the  $3 \times 3$  tensor  $\mathbf{S}^{(2)}$  computed for two

orientations generally differs from the second eigenvalue  $\lambda_2^{(3)}$  of  $\mathbf{S}^{(3)}$  computed under the hypothesis of three orientations. Hierarchical testing is based on the ratios

$$s_M = \frac{\sqrt[M+1]{\lambda_1^{(M)} \dots \lambda_{M+1}^{(M)}}}{\sqrt[M]{\frac{1}{M+1} \sum_{i=1}^{M+1} \lambda_i^{(M)} \dots \lambda_{i-1}^{(M)} \lambda_{i+1}^{(M)} \dots \lambda_{M+1}^{(M)}}} \quad (45)$$

which are compared against predefined thresholds  $\varepsilon_M$ . Note that both numerator and denominator of  $s_M$  can be interpreted as mean eigenvalues, with the numerator being the geometric mean of all eigenvalues. Moreover, both numerator and denominator can be computed without eigensystem analysis from quantities such as trace, determinant, and minors of  $\mathbf{S}^{(M)}$ . The upper bound for  $s_M$  is one, which is reached when all eigenvalues are equal. The lower bound is zero, which is obtained when (at least) one eigenvalue vanishes. We therefore require the thresholds  $\varepsilon_M$  to lie in the range  $[0, 1]$ . For each region  $\Omega$ , our procedure for decision making is as follows:

- 1) compute the average norm of the image gradient, and mark region as homogeneous if it is lower than a threshold  $\varepsilon_0$ . Otherwise, set  $M = 1$  and continue.
- 2) compute the tensor  $\mathbf{S}^{(M)}$  and the value of  $s_M$
- 3) if  $s_M$  is lower than the threshold  $\varepsilon_M$ , mark the region as  $M$ -oriented and compute its orientation vectors for the  $M$  orientations model. Otherwise, increment  $M$  by one and go to previous step (provided that  $M$  is smaller or equal than some maximum value  $M_{\max}$ ).

Applying this procedure with  $M_{\max} = 3$  yields a segmentation of the image into areas with one, two, or three orientations, plus two classes for homogeneous neighborhoods and neighborhoods with more than 3 orientations. The latter ones are those with  $s_3 \geq \varepsilon_3$ . Fig. 9 shows the region map obtained for  $\varepsilon_0 = 0.0009$ ,  $\varepsilon_1 = 0.5$ ,  $\varepsilon_2 = 0.7$ , and  $\varepsilon_3 = 0.95$ . Areas in dark gray represent single orientations, areas in medium gray double orientations, and areas in light gray three orientations. Apart from isolated decision errors, the estimated region structure corresponds well to the original image. The bottom part shows an enlarged portion of the orientation map, with the orientations visualized in different gray levels by lines, crosses or stars, depending on the number of local orientations.

Figs. 10 and 11 show similar results for an image of a house. Evidently, the image content is clearly reflected in the region map with five different classes (homogeneous regions, one, two, and three orientations, more than three orientations). This demonstrates the relevance of (multiple) local orientations as low level image or texture features. Also shown is an enlargement of the region above the dormer window where the two roof parts meet. It is clearly visible that the estimated triple-orientation structure does not vary much within the same roof area, but differs between both halves of the image (the ‘stars’ indicating the three orientations roughly appear mirrored), thus allowing further segmentation within the regions having the same number of local orientations.

## V. SUMMARY AND CONCLUSIONS

We have presented a theory for modelling multivariate signals – such as textures – composed from multiple dom-

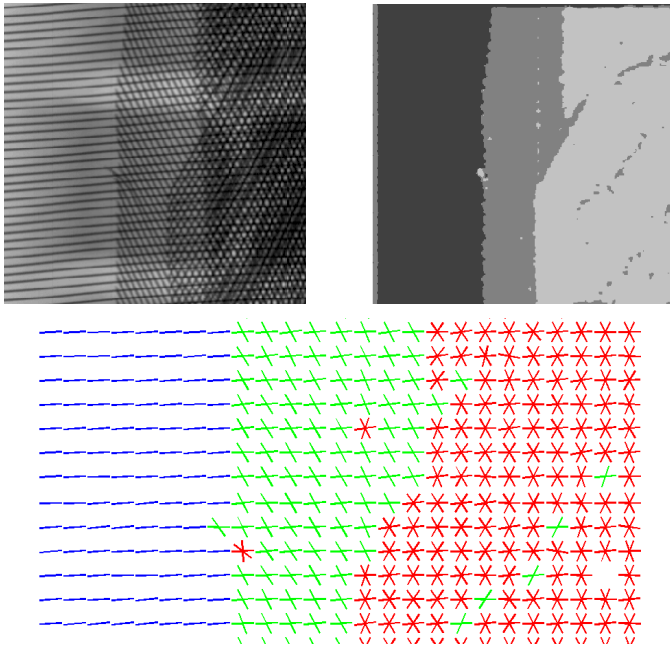


Fig. 9. Hierarchical orientation estimation for an X-ray image of tire ( $301 \times 301$  pixel). First row: original (left), detected number of orientations coded as gray levels (right). Bottom: enlarged part of the estimated orientation map.

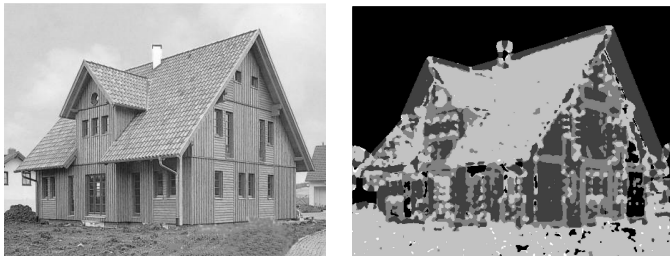


Fig. 10. Image of a house (left) and its color-coded region map (right).

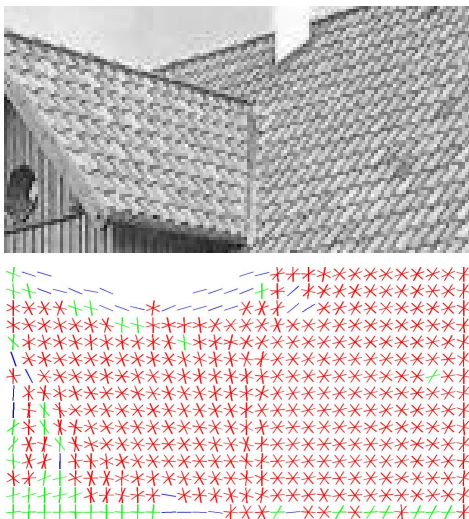


Fig. 11. Close-up of the region in the house image where two differently tilted roof parts meet (top), and the estimated orientations (bottom). Evidently, the roof tile texture is captured well by *three* local orientations. The orientation structure is homogeneous within each part of the roof, but differs between the two parts by a rotation.

inant orientations, extending the well-known structure tensor framework to a unified mathematical model for  $M$  orientations in  $N$ -variate and  $P$ -component signals  $s(\mathbf{x})$ . Generalization of the signal gradient to multiple directional derivatives led to tensor-valued entities. Depending on the assumed signal model, this generalization can be done under the occluding orientations model (OOM) or under the additive orientations model (AOM), and for both scalar-valued and vector-valued signals. In section II-B, we also discussed an even more general model allowing for mixed or intermediate forms between AOM and OOM; dealing with these in full depth could be a topic for future research. After the discussion of the different ways to generate data tensors, we derived a suitable mathematical representation for a set of  $M$  orientations, namely the space of *minimal fully symmetric tensors*  $\mathbb{R}_{\oplus}^{N \times \dots \times N}$ . From this tensor representation, vector descriptions for the sought  $M$  orientations were formed by appropriate vectorization, taking into account symmetry properties. An analogous vectorization procedure exists for the OOM and AOM data tensors, leading to generalized gradients. Alternatively, these generalized gradients could be formed as the Veronese map of order  $M$  of the standard gradient for the OOM respectively the standard gradient operator for the AOM. The tensor representation provided a convenient way of describing the space of admissible solutions. In particular, we have shown that the MOP vector – which we derived for general  $M$ ,  $N$  and  $P$  – can be a highly overparameterized representation of orientations. Based on this better mathematical understanding of multiple orientation estimation, we presented algorithms for the estimation of additively and occludingly superimposed multiple orientations, and for a combination of these. All algorithms are also applicable to vector-valued data, for instance color images.

In the experimental part, we applied the framework for multiple orientation estimation to both synthetic and real data. Apart from estimating the orientations themselves, it turned out that determining the number of orientations occurring in a local neighborhood may provide a valuable feature for tasks such as segmentation, classification or inspection. Note that the region maps shown in Figs. 9 and 10 were obtained purely data-driven, i.e., without prior assumptions on the region shapes as expressed, for instance, by Gibbs-Markov random fields (GMRFs) [49], [50]. Combining our framework with GMRFs within a Bayesian approach would likely reduce isolated segmentation errors. In addition, a statistical model for the effects of noise on the lowest eigenvalue  $\lambda_{M+1}$  of the data tensor could be integrated into the Bayesian approach (cf. [51]), thus allowing to tie the thresholds  $\varepsilon_i$  to error probabilities.

Software (Matlab) to reproduce most of these results (cf. [52]) is available for download at [www.lfb.rwth-aachen.de/de/highlights/tensor.html](http://www.lfb.rwth-aachen.de/de/highlights/tensor.html). The program allows to vary the number  $M$  of superposed signals in synthetic data, the basic signals themselves, their orientations, their respective weight functions in the combination process, the filters used for computing the discrete derivatives, the filter used for spatial integration of the structure tensors, the level and type of added noise, and

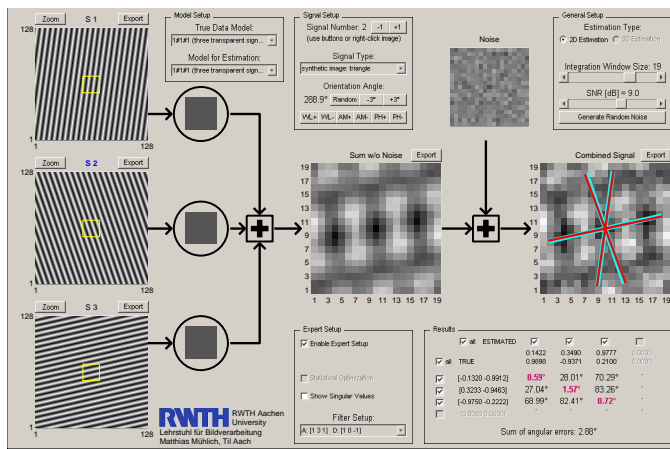


Fig. 12. The GUI of our multiple orientation synthesis & analysis software.

others. Its user interface is shown in Fig. 12.

## VI. ACKNOWLEDGEMENTS

The coloscopy image was provided by Jens Tischendorf and Christian Trautwein, University Hospital Aachen, the cystoscopic bladder image by Olympus Winter & Ibe GmbH, Hamburg, and the angiography by Philips Research Europe, Aachen. The plant root image was recorded and provided by Hanno Scharr, Research Centre Jülich. The crocodile picture was drawn by Linda Aach. The authors gratefully acknowledge the support.

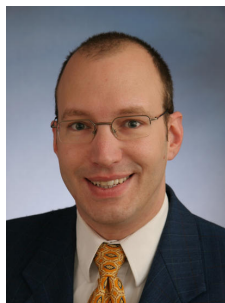
## REFERENCES

- [1] J. Bigün and G. H. Granlund, "Optimal orientation detection of linear symmetry," in *Proceedings IEEE First International Conference on Computer Vision*, London, June 1987, pp. 433–438.
- [2] J. Bigün, G. H. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 775–790, 1991.
- [3] J. Bigün, "Frequency and orientation sensitive texture measures using linear symmetry," *Signal Processing*, vol. 29, pp. 1–16, 1992.
- [4] S. Di Zenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics, and Image Processing*, vol. 33, pp. 116–125, 1986.
- [5] M. Kass and A. Witkin, "Analyzing oriented patterns," *Computer Vision, Graphics, and Image Processing*, vol. 37, pp. 362–385, 1987.
- [6] G. H. Granlund and H. Knutsson, *Signal Processing for Computer Vision*. Dordrecht: Kluwer, 1995.
- [7] J. Bigün, *Vision with Direction*. Berlin, Heidelberg, New York: Springer, 2006.
- [8] G. H. Granlund, "In search of a general picture processing operator," *Computer Graphics and Image Processing*, vol. 8, no. 2, pp. 155–178, 1978.
- [9] H. Knutsson, B. von Post, and G. H. Granlund, "Optimization of arithmetic neighborhood operations for image processing," in *First Scandinavian Conference on Image Analysis*, Linköping, January 1980, pp. –.
- [10] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [11] N. C. Stache, T. H. Stehle, M. Mühlich, and T. Aach, "Towards multiple-orientation based tensor invariants for object tracking," in *14th European Signal Processing Conference*, F. Gini, Ed. Firenze: EURASIP, September 4–8 2006, pp. 1–4 (CD-ROM).
- [12] E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision*. Upper Saddle River: Prentice-Hall, 1998.
- [13] C. Mota, I. Stuke, T. Aach, and E. Barth, "Estimation of multiple orientations at corners and junctions," in *DAGM04: 26th Pattern Recognition Symposium*, German Association for Pattern Recognition. Tuebingen: Springer Verlag: LNCS 3175, Aug. 30 – Sept. 1 2004, pp. 163–170.
- [14] M. Felsberg and G. Granlund, "POI detection using channel clustering and the 2D energy tensor," in *Proc. DAGM 2004*. Springer Verlag: LNCS 3175, 2004, pp. 103–110.
- [15] D. A. Forsyth and J. Ponce, *Computer Vision. A Modern Approach*. Upper Saddle River: Prentice-Hall, 2003.
- [16] M. Shizawa and K. Mase, "A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis," in *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition*. Maui: IEEE, June 3–6 1991, pp. 289–295.
- [17] T. Aach, C. Mota, I. Stuke, M. Mühlich, and E. Barth, "Analysis of superimposed oriented patterns," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3690–3700, 2006.
- [18] T. Aach, U. Schiebel, and G. Spekowius, "Digital image acquisition and processing in medical x-ray imaging," *Journal of Electronic Imaging*, vol. 8, no. Special Section on Biomedical Image Representation, pp. 7–22, 1999.
- [19] J. Hsieh, *Computed Tomography. Principles, Design, Artifacts, and Recent Advances*. Bellingham, Washington: SPIE Press, 2003.
- [20] K. I. Laws, "Textured image segmentation," Ph.D. dissertation, Image Processing Institute, Univ. of Southern California, 1980, rept. 940.
- [21] M. Unser, "Local linear transforms for texture measurements," *Signal Processing*, vol. 11, pp. 61–79, 1986.
- [22] T. Aach, A. Kaup, and R. Mester, "On texture analysis: Local energy transforms versus quadrature filters," *Signal Processing*, vol. 45, no. 2, pp. 173–181, 1995.
- [23] T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 291–309, 1999.
- [24] F. Ade, "Characterization of textures by 'eigenfilters'," *Signal Processing*, vol. 5, pp. 451–457, 1983.
- [25] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, pp. 786–804, 1979.
- [26] L. van Gool, P. Dewaele, and A. Oosterlinck, "Texture analysis anno 1983," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 336–357, 1985.
- [27] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Madison: IEEE, June 2003, pp. 621–628.
- [28] W. Förstner, "A feature based corresponding algorithm for image matching," *International Archive of Photogrammetry and Remote Sensing*, vol. 26, pp. 150–166, 1986.
- [29] T. Lindeberg and J. Garding, "Shape from texture from a multi-scale perspective," in *Proc. 4th Int. Conf. on Computer Vision*. IEEE Computer Society Press, 1993, pp. 683–691.
- [30] R. Mester, "The generalization, optimization and information-theoretic justification of filter-based and autocovariance based motion estimation," in *Proceedings IEEE International Conference on Image Processing (ICIP)*, I. 0-7803-7751-6, Ed. Barcelona: IEEE, September 14–17 2003, pp. –.
- [31] H. Knutsson and G. H. Granlund, "Texture analysis using two-dimensional quadrature filters," in *IEEE Workshop on Computer Architecture for Pattern Analysis and Image Data Base Management*. Pasadena, California: IEEE, October 1983.
- [32] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 55–73, 1990.
- [33] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach, "Ridges for image analysis," Dep. of Computer Science, Univ. of North Carolina, Chapel Hill, NC, USA, Tech. Rep. TR93-055, 1993.
- [34] M. Jacob and M. Unser, "Design of steerable filters for feature detection using Canny-like criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1007–1019, 2004.
- [35] M. Mühlich, T. Dahmen, and T. Aach, "Design of multi-steerable filters and their application for the detection of corners and junctions," in *IEEE International Conference on Image Processing (ICIP2007)*. San Antonio, TX: IEEE, September 16–19 2007, pp. IV–33 – IV–36.
- [36] —, "Doubly-rotated matched filtering," in *15th European Signal Processing Conference (EUSIPCO-2007)*. Poznan: EURASIP, September 3–7 2007, pp. 2169–2173.
- [37] U. Köthe, "Integrated edge and junction detection with the boundary tensor," in *Proc. ICCV03*, vol. 1, Nice, France, 2003, pp. 424–431.
- [38] U. Köthe and M. Felsberg, "Riesz-transforms versus derivatives: On the relationship between the boundary tensor and the energy tensor," in *Proc. Scale Space and PDE Methods in Computer Vision*. Springer Verlag: LNCS 3459, 2005, pp. 179–191.

- [39] M. Felsberg, "The GET operator," Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, Tech. Rep. LiTH-ISY-R-2633, October 2004.
- [40] M. Shizawa and K. Mase, "Simultaneous multiple optical flow estimation," in *Proceedings IEEE International Conference on Computer Vision and Pattern Recognition*. Atlantic City: IEEE, June 1990, pp. 274–278.
- [41] M. Shizawa and T. Iso, "Direct representation and detection of multi-scale, multi-orientation fields using local differentiation filters," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*. Piscataway: IEEE, June 1993, pp. 508–514.
- [42] T. Aach, I. Stuke, C. Mota, and E. Barth, "Estimation of multiple local orientations in image signals," in *Proceedings ICASSP-2004*. Montreal: IEEE, May 17–21 2004, pp. III 553–556.
- [43] C. Mota, T. Aach, I. Stuke, and E. Barth, "Estimation of multiple orientations in multi-dimensional signals," in *IEEE International Conference on Image Processing (ICIP)*. Singapore: IEEE, Oct. 24–27 2004, pp. 2665–2668, and on CD-ROM: ISBN 0–7803–8555–1.
- [44] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1–15, 2005.
- [45] J. Harris, *Algebraic Geometry*. Springer, 1992.
- [46] W. Förstner and E. Gülch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features," in *Workshop on Fast Processing of Photogrammetric Data*. Interlaken: ISPRS, June 1987.
- [47] I. Stuke, T. Aach, E. Barth, and C. Mota, "Analysing superimposed oriented patterns," in *6th IEEE Southwest Symposium on Image Analysis and Interpretation*. Lake Tahoe, NV: IEEE Computer Society, March 28–30 2004, pp. 133–137.
- [48] M. Mühlich and T. Aach, "A theory of multiple orientation estimation," in *9th European Conference on Computer Vision (ECCV)*, ser. LNCS, H. Bischof and A. Leonardis, Eds., no. 3952. Graz: Springer, May 7–13 2006, pp. (II) 69–82.
- [49] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 39–55, 1987.
- [50] T. Aach, U. Franke, and R. Mester, "Top-down image segmentation using object detection and contour relaxation," in *Proceedings ICASSP 89*, Glasgow, UK, May 1989, pp. 1703–1706.
- [51] T. Aach, L. Dümbgen, and R. Mester, "Bayesian illumination invariant change detection using a total least squares test statistic," in *Actes/Proceedings 18e Colloque GRETSI sur le Traitement du Signal et des Images*, Toulouse, France, Septembre 10–13 2001, pp. 587–590.
- [52] J. Kovacević, "From the Editor-in-Chief," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3625–3626, 2006.



Til Aach (M 1994 – SM 2002) received his Diploma and Doctoral degrees, both with honors, in EE from RWTH Aachen University in 1987 and 1993, respectively. While working towards his Doctoral Degree, he was a research scientist with the Institute for Communications Engineering, RWTH Aachen University, being in charge of several projects in image analysis, 3D-television and medical image processing. In 1984, he was an intern with Okuma Machinery Works Ltd., Nagoya, Japan. From 1993 to 1998, he was with Philips Research Labs, Aachen, Germany, where he was responsible for several projects in medical imaging, image processing and analysis. In 1996, he was also an independent lecturer with the University of Magdeburg, Germany. In 1998, he was appointed a Full Professor and Director of the Institute for Signal Processing, University of Luebeck. In 2004, he became Chairman of the Institute of Imaging and Computer Vision, RWTH Aachen University. His research interests are in medical and industrial image processing, signal processing, pattern recognition, and computer vision. He has authored or co-authored over 250 papers, and received several awards, among these the award of the German "Informationstechnische Gesellschaft" (ITG/VDE), for a paper published in the IEEE Transactions on Image Processing in 1998. Dr. Aach is a co-inventor for about 20 patents. From 2002 to 2008, he was an Associate Editor of the IEEE Transactions on Image Processing. He was a Technical Program Co-Chair for the IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI) in 2000, 2002, 2004, and 2006. He is a member of the Bio-Imaging and Signal Processing Committee (BISP-TC) of the IEEE Signal Processing Society.



Matthias Mühlich received the diploma and Ph.D. degree from Johann Wolfgang Goethe University Frankfurt, Germany, in 1999 and 2005, respectively. He is currently working at RWTH Aachen University, Aachen, Germany. His research interests are the estimation of orientations, motions and subspaces.