Estimation of Multiple Orientations and Multiple Motions in Multi-Dimensional Signals

Ingo Stuke[1], Erhardt Barth[2]
[1] Institute for Signal Processing
[2] Institute for Neuro- and Bioinformatics
University of Luebeck, Germany
{stuke@isip, barth@inb}.uni-luebeck.de

Cicero Mota
Departamento de Matematica
Universidade Federal do Amazonas
69077-000 Manaus, Brazil
mota@ufam.edu.br

Abstract

The estimation of multiple orientations in multi-dimensional signals is a strongly non-linear problem to which a two-step solution is here presented. First, the problem is linearized by introducing the so-called mixed-orientation parameters *as a unique, albeit implicit, descriptor of the orientations. Second, the non-linearities are decomposed such as to find the individual orientations. For two-dimensional signals, e.g., images, this decomposition step is solved by simply determining the roots of a polynomial. For multi-dimensional signals, the nD decomposition problem is solved by reducing it to a cascade of 2D decomposition problems. In this way, a full solution for the estimation of any number of orientations in any dimension is achieved for the first time.*

Key words: multiple orientations, multiple motions, transparency, occlusion.

1. Introduction

The analysis of local orientation in images and multidimensional signals is an essential step in, e.g., directional filtering [1, 2], directional interpolation [9], feature extraction, tracking, motion estimation [17, 13, 11, 5], pattern analysis [8, 12, 5], compression [7, 15] and the understanding of the human visual system [25]. Multiple orientations appear in non-opaque imagery like X-ray, ultrasound, and computer tomography. Moreover, they characterize image features like corners, crossings and bifurcations. In such applications, orientation estimation seeks to find locally one-dimensional (1D) structures such as lines in images [4, 1, 2] or multi-variate image data [9, 5]. Corners and junctions are a rich source of information: L- and Y-junctions represent object corners, T-junctions occur at occluding object boundaries, X-junctions at object crossings, while ψ-junctions are caused by bending object surfaces. In such signals more than one single orientation is present, and these multiple orientations cannot be described accurately by assuming a single orientation. Multiple orientations have been an active subject of research in image processing and vision science, see [21, 6, 19, 17] and the references therein. The problem of estimating N multiple orientations in 3D and 2D can be divided into a linear and a non-linear part [17]. In the linear part, the *mixed-orientation parameters (MOP)* are estimated by standard linear techniques such as least squares or singular value decomposition. Since the nonlinearities are hidden in the mixed-orientation parameters, these do not provide the orientations explicitly. In the nonlinear part, the orientations are obtained from the MOP by solving for the roots of an $N-$degree polynomial. This approach has been extended to an hierarchical algorithm [17], which successively tests for zero, single, double or more orientations in a local neighborhoods and then solves for the appropriate number of orientations.

In this paper, we present a more general framework for estimating multiple orientations in multi-dimensional signals. We model multiple orientation signals as an additive superposition of a set of one-dimensional oriented signals, but occluded orientations can be approached in the same manner [3, 22, 16, 14]. As a major contribution, we then present a general solution for decomposing the MOP into the individual orientations, and thereby overcome previous limitations in either the dimension of the signal or the number of orientations [20, 21, 17, 3, 22, 16, 14]. This paper is organized as follows. In Section 2, we review decomposition methods for superimposed double orientations in images. Next, we present a robust method for decomposing *any* number of orientations (that are intermingled in the MOP) for the 2D case. We then deal, in Section 4, with the problem of transparent motions; and in Section 5, we show how the estimation of orientations in 3D can be performed by reducing the problem to the transparent motion case. A general solution for the decomposition of the MOP involv-

XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'06)
0-7695-2686-1/06 $20.00 © 2006 IEEE

ing *any* number of orientations in spaces of *arbitrary* numbers of dimensions is presented in Section 6. The solution is obtained by transforming the general case of multiple orientations in multi-dimensional signals to a cascade of decomposition problems for the 2D case. Finally, in Section 7, we present some applications of the technique both for the analysis of orientations in images and the estimation of multiple motions.

2. Estimation of two orientations in 2D

We start with one special case of orientations in 2D, namely the problem of estimating the velocities of overlaid one-dimensional waves.

One-dimensional motion. Consider a wave $g : \mathbb{R} \to \mathbb{R}$ moving with constant velocity v. At each position in space-time, we can observe the value $f(x,t) = g(x - tv)$. We wish to determine the velocity of propagation v. The hypothesis of constant velocity is equivalent to the property that the observed space-time signal $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is constant along the lines $x - tv = a$. By taking derivatives, we find the well known *flux equation* $vf_x + f_t = 0$, where f_x, f_t are the partial derivatives of f. From the flux equation, the velocity can easily be estimated.

Now consider two waves g_1, g_2 simultaneously propagating with constant velocities u and v. The observed signal is, therefore, $f(x,t) = g_1(x - tu) + g_2(x - tv)$. i.e., the overlaid superposition of two single oriented patterns in space time. The constraint equation for the velocities becomes

$$uvf_{xx} + (u+v)f_{xt} + f_{tt} = 0. \qquad (1)$$

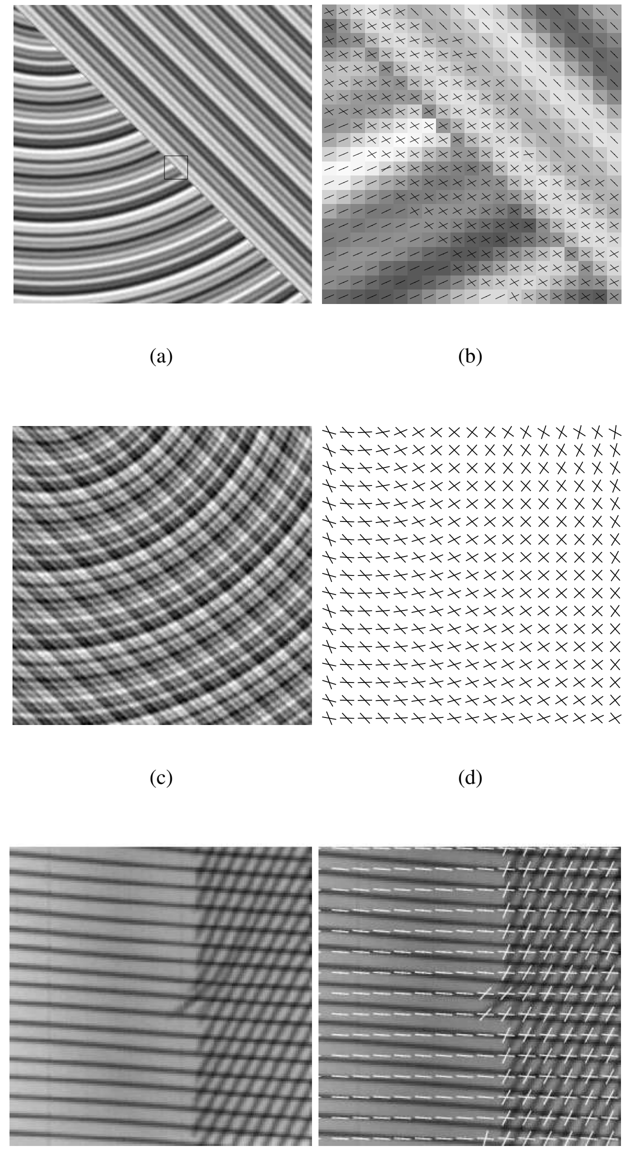
Equation (1) is non-linear in the motion parameters themselves, but linear in the so-called *mixed motion parameters* (MMP): $c_{xx} = uv$, $c_{xt} = u + v$, $c_{tt} = 1$, which, therefore, can be estimated by standard linear techniques. Once the MMP are known, the velocities can be recovered as the roots of $Q_2(z) \equiv (z-u)(z-v) = z^2 - c_{xt}z + c_{xx}$. The generalization of this approach to the case of N overlaid waves is straightforward.

Multiple orientations in images. An image $f : \mathbb{R}^2 \to \mathbb{R}$ is said to be oriented along the direction $\mathbf{u}$ in a region Ω if

$$f(\mathbf{x}) = f(\mathbf{x} + s\mathbf{u}) \qquad (2)$$

$\forall \mathbf{x}, \mathbf{x} + s\mathbf{u} \in \Omega$. Two collinear vectors $\mathbf{u}, \mathbf{v}$ can represent the same orientation. A unit vector $\mathbf{u} = (\cos\theta, \sin\theta)^T$ describes the orientation of $f(\mathbf{x})$ by the angle θ, which is conventionally restricted to lie in the interval $(-\pi/2, \pi/2]$. Equation 2 is equivalent to the following constraint that is valid within Ω, [24, 5, 12].

$$\frac{\partial f}{\partial \mathbf{u}} = 0. \qquad (3)$$



(a) (b)

(c) (d)

(e) (f)

Figure 1. Synthetic examples: (a) T-junction, (b) region with superimposed orientations; (c) X-junction, (d) estimated orientations. Real example: (e) X-Ray image of rubber product, (f) single and double orientations superimposed.

Double-oriented patterns can be modeled by

$$f(\mathbf{x}) = g_1(\mathbf{x}) + g_2(\mathbf{x}). \qquad (4)$$

Since the two components g_1 and g_2 are assumed to be ideally oriented in the directions $\mathbf{u} = (u_x, u_y)^T$ and $\mathbf{v} = (v_x, v_y)^T$ respectively, both components obey Equation (3). Therefore, the composite image $f(\mathbf{x})$ satisfies the nonlinear equation

$$\frac{\partial^2 f}{\partial \mathbf{u} \partial \mathbf{v}} = c_{xx} f_{xx} + c_{xy} f_{xy} + c_{yy} f_{yy} = 0, \tag{5}$$

where the nonlinearities are hidden in the *mixed-orientation parameters (MOP)*

$$c_{xx} = u_x v_x, \; c_{xy} = u_x v_y + u_y v_x \text{ and } c_{yy} = u_y v_y. \tag{6}$$

As shown in [17], the mixed-orientation parameters can be estimated from Equation (5) with a least-squares approach. The resulting mixed-orientation parameter vector $\boldsymbol{c} = (c_{xx}, c_{xy}, c_{yy})^T$ is an unambiguous descriptors of double orientation neighborhoods, thus providing a feature that could be used, e.g., for tracking. The MOP vector, however, does not explicitly provide the orientations. In the following, we review different approaches to decompose the MOP into the orientations.

MOP decomposition by normalization. A simple way to decompose the MOP into the orientation components assumes that $u_x = v_x = 1$ ($c_{xx} = 1$). This can be enforced by dividing all mixed-orientation parameters by c_{xx} as long as $c_{xx} \neq 0$. (In case of $c_{xx} = 0$ we set $\mathbf{u} = (1, 0)$ and the problem reduces to the one orientation case [22]). With the above assumption, Equation (6) becomes

$$c_{xx} = 1, \; c_{xy} = u_y + v_y \text{ and } \; c_{yy} = u_y v_y, \tag{7}$$

and the y-components of the orientation vectors $\mathbf{u}$ and $\mathbf{v}$ are obtained as the roots of the polynomial

$$Q_2(z) = (z - u_y)(z - v_y) = z^2 - c_{xy} z + c_{yy}. \tag{8}$$

This approach has two main drawbacks: (i) we have to choose a threshold parameter to decide whether $c_{xx} = 0$, and (ii) unnecessary errors are introduced in the estimates if c_{xx} is small because either u_x or v_x is close to zero. However, the approach can easily be extended to the general case of more that two orientations in images as shown in [18].

Rotation of the coordinate system. Alternatively, we can deal with the case $c_{xx} = c_{yy} = 0$ by rotating the coordinate system with a randomly chosen rotation matrix $\mathbf{R}$. The MOP can be represented as the symmetric 2-tensor [20, 10, 14]

$$\mathbf{C} = \frac{1}{2}(\mathbf{u} \otimes \mathbf{v} + \mathbf{v} \otimes \mathbf{u}) = \begin{pmatrix} c_{xx} & c_{xy}/2 \\ c_{xy}/2 & c_{yy} \end{pmatrix}. \tag{9}$$

A transformation of the coordinate system, say $\mathbf{x}' = \mathbf{R}\mathbf{x}$, yields

$$\mathbf{C}' = \mathbf{R}\mathbf{C}\mathbf{R}^T \tag{10}$$

as the tensor representation of the MOP in the new coordinate system. Fortunately, because $c_{xx} \neq 0$ and $c_{yy} \neq 0$ in almost every coordinate system, we can always find an appropriate system for solving the decomposition problem. Note that this procedure is applicable for improving the estimates in cases where c_{xx} is small.

Decomposition matrix. An alternative decomposition method which circumvents the above difficulties [3, 16] without any further exceptions determines the orientation vectors as the rows and columns of the tensor

$$\mathbf{u} \otimes \mathbf{v} = \begin{pmatrix} u_x v_x & u_x v_y \\ u_y v_x & u_y v_y \end{pmatrix} = \begin{pmatrix} c_{xx} & z_1 \\ z_2 & c_{yy} \end{pmatrix}. \tag{11}$$

The unknown matrix elements z_1 and z_2 are obtained as the roots of the polynomial

$$Q_2(z) = z^2 - c_{xy} z + c_{xx} c_{yy} \tag{12}$$

and thus, the desired (yet not normalized) orientation vectors are given by

$$\begin{aligned} \mathbf{u} &= (c_{xx}, z_2)^T = (z_1, c_{yy})^T \\ \mathbf{v} &= (c_{xx}, z_1)^T = (z_2, c_{yy})^T. \end{aligned} \tag{13}$$

Eigensystem-based decomposition. Alternatively, it is possible to decompose the MOP by an eigensystem analysis of the symmetric 2-tensor in Equation (9) [20, 10]. Denoting the eigenvalues of $\mathbf{C}$ by λ_1 and $-\lambda_2$ and the corresponding eigenvectors by $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, the (again not normalized) orientations vectors are given by

$$\begin{aligned} \mathbf{u} &= \sqrt{\lambda_1} \boldsymbol{e}_1 + \sqrt{\lambda_2} \boldsymbol{e}_2 \\ \mathbf{v} &= \sqrt{\lambda_1} \boldsymbol{e}_1 - \sqrt{\lambda_2} \boldsymbol{e}_2 \end{aligned} \tag{14}$$

Note that this decomposition method is more general because it can also be used to solve the decomposition problem of double oriented higher dimensional signals [21, 16].

3. Generalization of MOP decomposition in 2D

Here we present a robust way for finding the orientation parameters that does not require a threshold.

3.1. The case of 3 orientations

First, we demonstrate the general idea for the decomposition for the case of three orientations. In this case the observed signal is modeled as

$$f(\mathbf{x}) = g_1(\mathbf{x}) + g_2(\mathbf{x}) + g_3(\mathbf{x}) \tag{15}$$

where g_1, g_2, g_3 have orientations $\mathbf{u}, \mathbf{v}, \mathbf{w}$. The MOP are given by

$$\begin{aligned} c_{xxx} &= u_x v_x w_x \\ c_{xxy} &= u_x v_x w_y + u_x v_y w_x + u_y v_x w_x \\ c_{xyy} &= u_x v_y w_y + u_y v_x w_y + u_y v_y w_x \\ c_{yyy} &= u_y v_y w_y \end{aligned} \tag{16}$$

Note that if one of the vectors is known, say $\mathbf{w}$, the problem reduces to the problem of separating two orientations. In fact,

$$\begin{array}{lclclcl} c_{xxx} & = & c_{xx} w_x & & & & \\ c_{xxy} & = & c_{xx} w_y & + & c_{xy} w_x & & \\ c_{xyy} & = & & & c_{xy} w_y & + & c_{yy} w_x \\ c_{yyy} & = & & & & & c_{yy} w_y \end{array} \tag{17}$$

Note that the matrix $\mathbf{B}$ defined by the coefficients in the right-hand side of the system of equations above is band diagonal with the two main diagonals formed by shifted repetitions of $\mathbf{w}$. Therefore, $\mathbf{B}$ has full rank, since $\mathbf{w}$ is a nonzero vector. Thus, $\mathbf{c}_2 = (c_{xx}, c_{xy}, c_{yy})$ can be recovered, at least by least squares or singular value decomposition. Once we have $\mathbf{c}_2$, the MOP can be separated either by repeating the above reduction scheme or by one of the methods discussed in Section 2.

Now we proceed to show how one of the orientation vectors needed in the reduction step can be determined. By inspecting the first two equations in (16), we note that the problem is essentially solved if we can determine $a = u_x v_x w_y$, $b = u_x v_y w_x$, and $c = u_y v_x w_x$ since

$$\begin{aligned} (c_{xxx}, a) &= u_x v_x \mathbf{w} \\ (c_{xxx}, b) &= u_x w_x \mathbf{v} \\ (c_{xxx}, c) &= v_x w_x \mathbf{u} \end{aligned} \tag{18}$$

To find $\mathbf{w}$, for example, we normalize (c_{xxx}, a) to unit length. This procedure removes the scaling factor $u_x v_x$. To find a, b, c, we build a polynomial having those numbers as roots. If

$$Q(z) \equiv (z-a)(z-b)(z-c) \equiv z^3 - A_2 z^2 + A_1 z - A_0 \tag{19}$$

then

$$A_2 = c_{xxy} \quad A_1 = c_{xyy} c_{xxx} \quad A_0 = c_{yyy} c_{xxx}^2 . \tag{20}$$

The three vectors in Equation (18) are the desired orientations, unless one of the orientations is parallel to the $y-$axis, i.e., $c_{xxx} = 0$, and all three represent the orientation $\mathbf{w} = (0, 1)$. For two orientations close to the $y-$axis, the three numbers $u_x v_x$, $u_x w_x$, $v_x w_x$ may actually be small and therefore lead to unreliable estimates. To account for this problem, we compute the vectors

$$u_y v_y \mathbf{w}, \quad u_y w_y \mathbf{v}, \quad v_y w_y \mathbf{u} \tag{21}$$

by using the two last equations in (16). Now, at least one of the six vectors is reliable since small u_x, v_x will produce a large $u_y v_y$. We choose the largest vector, assign it to $\mathbf{w}$ and solve Equation (17). The generalization for N orientations is straightforward as will be shown in the next section.

3.2. The general case of N orientations in 2D

In this, case the observed signal is modeled as

$$f(\mathbf{x}) = g_1(\mathbf{x}) + \cdots + g_N(\mathbf{x}) \tag{22}$$

where $g_1, \ldots, g_N$ have orientations

$$\mathbf{u}_1 = (u_{1\,x}, u_{1\,y}), \ldots, \mathbf{u}_N = (u_{N\,x}, u_{N\,y}), \tag{23}$$

respectively. Let $u_{\mathbf{x}\,j} = \prod_{i \neq j} u_{i\,x}$, and $a_j = u_{j\,y} u_{\mathbf{x}\,j}$, for $j = 1, \ldots, N$, i.e.,

$$c_{x\cdots xy} = a_N + a_{N-1} + \cdots + a_1. \tag{24}$$

The polynomial

$$Q(z) = z^N - A_{N-1} z^{N-1} + \cdots + (-1)^N A_0, \tag{25}$$

that has the parameters a_j as roots, must have the coefficients

$$\begin{aligned} A_{N-1} &= c_{x\cdots xy} \\ A_{N-2} &= c_{x\cdots yy} c_{x\cdots x} \\ &\vdots \\ A_0 &= c_{y\cdots y} c_{x\cdots x}^{N-1} . \end{aligned} \tag{26}$$

Solving for the roots of $Q_N(z)$, we find, for each j,

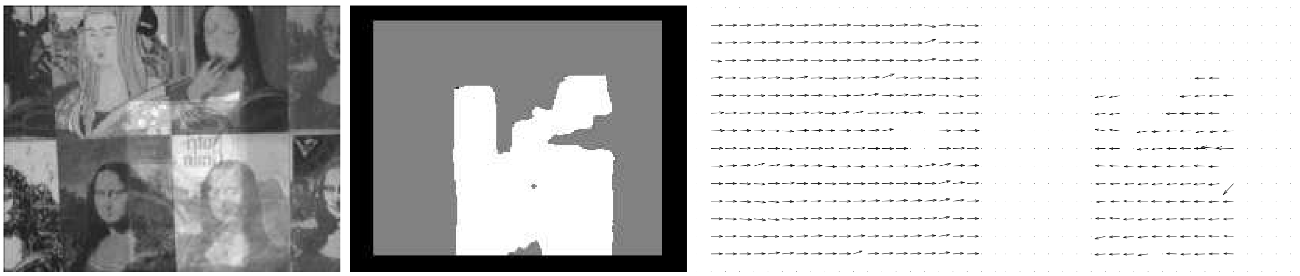
$$(c_{x\cdots x}, a_j) = u_{\mathbf{x}\,j} \mathbf{u}_j \tag{27}$$

and similarly we find

$$u_{\mathbf{y}\,j} \mathbf{u}_j \, . \tag{28}$$

The hypothesis that the orientations are pairwise distinct assures that at least one $u_{\mathbf{x}\,j}$ is non-zero (the same holds for $u_{\mathbf{y}\,j}$). As in the case of three orientations, we choose as $\mathbf{u}_N$, the largest of the $2N$ candidates in Equations (27) and (28).

The procedure does not need any threshold parameters. Nevertheless, to accelerate the separation, we may choose to threshold $|c_{x\cdots x}|$ by some parameter ϵ as follows. We suppose $|c_{x\cdots x}| \geq |c_{y\cdots y}|$ (and reorder the axes if necessary). If $|c_{x\cdots x}| \geq \epsilon$, all the orientations are obtained from Equation (27). Otherwise we assign to $\mathbf{u}_N$ the largest vector in Equation (27), and reduce the problem by one orientation. By doing so we avoid any fine tuning of threshold parameters. In the next sections, we will describe a particular case of multiple orientations in 3D. The solution in this particular case will then lead to a general solution in 3D.



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

Figure 2. A real example of multiple transparent motion estimation: the Mona Lisa sequence.

4. Multiple motions and mixed-motion parameters in 3D

We model two additive superimposed motions by

$$f(\mathbf{x},t) = g_1(\mathbf{x} - t\mathbf{u}) + g_2(\mathbf{x} - t\mathbf{v}), \tag{29}$$

with the time variable t. The layers $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are moving with the velocities $\mathbf{u} = (u_x, u_y)^T$ and $\mathbf{v} = (v_x, v_y)^T$, respectively. The sequences $g_1(\mathbf{x} - t\mathbf{u})$ and $g_2(\mathbf{x} - t\mathbf{v})$ satisfy the optical flow constraints

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} g_1(\mathbf{x} - t\mathbf{u}) &= \hat{\mathbf{u}} \cdot \nabla g_1 = 0 \\ \frac{\mathrm{d}}{\mathrm{d}t} g_2(\mathbf{x} - t\mathbf{v}) &= \hat{\mathbf{v}} \cdot \nabla g_2 = 0\,, \end{aligned} \tag{30}$$

where $\nabla = (\partial_x, \partial_y, \partial_t)$ is the gradient operator and $\hat{\mathbf{u}} = (u_x, u_y, 1)^T$ and $\hat{\mathbf{v}} = (v_x, v_y, 1)^T$ are three-dimensional space-time orientation vectors. These equations state that the gray levels remain unchanged along the trajectories of motion. Thus, the directional derivative in direction of motion must vanish. Evidently, motion analysis is a particular case of orientation estimation in three dimensional space with the additional constraint that the temporal components $u_t = v_t = 1$. As in Equation (5), $f(\mathbf{x}, t)$ satisfies the constraint

$$\begin{aligned} \frac{\partial^2 f}{\partial \hat{\mathbf{u}} \partial \hat{\mathbf{v}}} = c_{xx} f_{xx} + c_{xy} f_{xy} + c_{xt} f_{xt} \\ + c_{yy} f_{yy} + c_{yt} f_{yt} + c_{tt} f_{tt} = 0, \end{aligned} \tag{31}$$

with the *mixed-motion parameters (MMP)*

$$\begin{aligned} c_{xx} = u_x v_x,\ c_{yy} = u_y v_y,\ c_{xy} = (u_x v_y + u_y v_x) \\ c_{xt} = (u_x + v_x),\ c_{yt} = (u_y + v_y),\ c_{tt} = 1. \end{aligned} \tag{32}$$

In order to estimate the mixed-motion parameters we can choose one of the methods proposed in [17, 23]. Then the nonlinear problem is solved by decomposing the MMP into the individual motion components.

Decomposition with complex polynomials. A general solution for an arbitrary number of superimposed motions was proposed in [17]. We now sketch the idea for the case of only two motions. The interpretation of motion vectors as complex numbers $\mathbf{v} = v_x + iv_y$ allows to determine the motion vectors as the roots of the complex polynomial

$$\begin{aligned} Q(z) = (z - \mathbf{u})(z - \mathbf{v}) = z^2 - (c_{xt} - ic_{yt})z \\ + (c_{xx} - c_{yy} + ic_{xy}), \end{aligned} \tag{33}$$

whose coefficients are expressed in terms of the MOP. In the next section we will show how multiple orientations in 3D can be projected to multiple motions and thus be decomposed according to Equation (33).

5. Decomposition of the mixed-orientation parameters in 3D

The model for multiple oriented signals in 3D is the same as in Equation (22), except that now $\mathbf{x}$ and the $\mathbf{u}$'s belong to the three-dimensional space.

Transforming the MOP to the MMP. We will first reduce the problem of decomposing the N mixed orientations in the MOP into an equivalent problem of separating N mixed motion vectors in the MMP. For simplicity we first consider the case of two orientations $N = 2$. By setting $\mathbf{u} = \mathbf{u}_1$ and $\mathbf{v} = \mathbf{u}_2$, the equations for the MOP are

$$\begin{aligned} c_{xx} &= u_x v_x \\ c_{xy} &= u_x v_y + u_y v_x \\ c_{yy} &= u_y v_y \\ c_{xt} &= u_x v_t + u_t v_x \\ c_{yt} &= u_y v_t + u_t v_y \\ c_{tt} &= u_t v_t\,. \end{aligned} \tag{34}$$

First, we reorder the axes to assure $|c_{tt}| \geq |c_{jj}|$, $j = x, y$. We have $c_{tt} \neq 0$ if and only if $u_t\, v_t \neq 0$. In this case, we set $u_t = v_t = 1$, and System (34) reduces to System (32), which is MMP for two motions. Therefore, the orientations can be separated by solving a second degree complex polynomial. If $c_{tt} = 0$, we simply apply a random rotation to the coordinate system, such that the MOP transform accordingly. It follows that in the new coordinate system (x', y', t') we have $c_{t't'} \neq 0$. The generalization for *any* number or orientations is straightforward.

Projection in 2D. We will present another approach for the separation of two orientations in 3D, which will lead to the general solution for any number of orientations in any dimension. Equation (34) reveals that by considering the pairs (x, y), (y, t) and (x, t), the problem of finding two orientations in 3D can be split into three problems of finding two orientations in $2D$:

$$\begin{aligned} c_{xx} &= u_x v_x \\ c_{xy} &= u_x v_y + u_y v_x \\ c_{yy} &= u_y v_y \,, \end{aligned} \tag{35}$$

$$\begin{aligned} c_{yy} &= u_y v_y \\ c_{yt} &= u_y v_t + u_t v_y \\ c_{tt} &= u_t v_t \,, \end{aligned} \tag{36}$$

and

$$\begin{aligned} c_{xx} &= u_x v_x \\ c_{xt} &= u_x v_t + u_t v_x \\ c_{tt} &= u_t v_t \,. \end{aligned} \tag{37}$$

Now, we can benefit from one of the decomposition techniques discussed in Sections 2 and 3 to obtain three pairs of (unnormalized) vectors

$$\alpha_t \mathbf{u}_t = (u_x, u_y, 0) \text{ and } \alpha_t \mathbf{v}_t = (v_x, v_y, 0) \tag{38}$$

and

$$\alpha_x \mathbf{m}_x = (0, m_y, m_t) \text{ and } \alpha_x \mathbf{n}_x = (0, n_y, n_t) \tag{39}$$

We could now use the middle equation in (37) to find the correspondence, but instead, we proceed to obtain the geometrical insight for the general solution by computing

$$\alpha_y \mathbf{r}_y = (r_y, 0, r_t) \text{ and } \alpha_y \mathbf{s}_y = (s_y, 0, s_t). \tag{40}$$

We have now found all the projections of the desired orientations onto the coordinate planes xy, yt and xt. However, we still have an assignment problem to solve: which of the computed 2D vectors are projections of $\mathbf{u}$, and which are projections of $\mathbf{v}$? If we could assume that $\mathbf{u}_t$, $\mathbf{m}_x$, $\mathbf{r}_y$ are projections of $\mathbf{u}$, we could recover $\mathbf{u}$. This assumption, however, holds if (and only if) the matrix

$$\begin{pmatrix} u_x & 0 & -r_x \\ -u_y & m_y & 0 \\ 0 & -m_t & r_t \end{pmatrix} \tag{41}$$

is rank deficient. This means that we can recover the orientations by trying all possible combinations of the vectors in Equations (38), (39) and (40) until the matrix becomes rank deficient.

6. Multiple orientations in any dimension

In what follows $\mathbf{x} = x_1\mathbf{e}_1 + \cdots + \mathbf{x}_p\mathbf{e}_p$ is a point in $\mathbb{R}^p$ and $\mathbf{u}_1, \ldots, \mathbf{u}_N$ are the desired orientations. We compute the following projections of these orientations onto the planes $x_p x_1$, $x_j x_{j+1}$, $j = 1, \ldots, p-1$:

$$\begin{aligned} \mathbf{u}_p^k &= u_1^k \mathbf{e}_1 + u_p^k \mathbf{e}_p \\ \mathbf{u}_j^k &= u_j^k \mathbf{e}_j + u_{j+1}^k \mathbf{e}_{j+1} \quad j = 1, \ldots, p-1. \end{aligned} \tag{42}$$

$k = 1, \ldots, N$ by one the methods discussed in Sections 2 and 3. The sequence of projections $\mathbf{u}_1^{k(1)}, \ldots, \mathbf{u}_p^{k(p)}$ represents the same orientation if and only if the matrix

$$\begin{pmatrix} u_1^{k(1)} & 0 & \cdots & -u_1^{k(p)} \\ -u_2^{k(1)} & u_1^{k(2)} & \cdots & 0 \\ 0 & -u_2^{k(2)} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & u^{k(2)} \end{pmatrix} \tag{43}$$

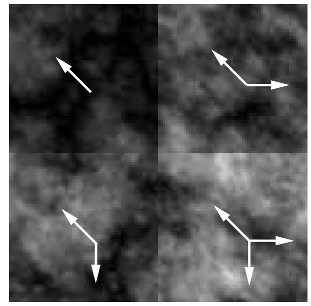
is rank deficient. We can, therefore, recover all the orientations based on all possible sequences $\mathbf{u}_j^{k(j)}$, $j = 1, \ldots, p$.

7. Examples

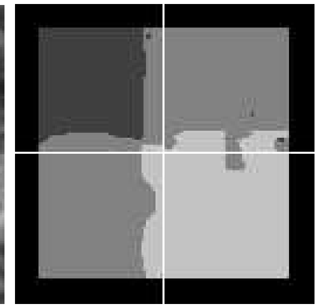
In this section, we present some application examples for multiple motion and orientation analysis.

Figure 1 shows both synthetic and real examples. Results for orientation estimation at junctions are in the top and middle rows. Panel (a) depicts a T-junction formed by two sinusoidal patterns. The estimated orientations for the marked region are shown in (b) and fit the image orientations well. Panel (c) depicts an example for X-junctions, where two sinusoidal patterns are additively superimposed. The estimated orientations are shown in (d) and comply well with the individual orientations of the patterns. A real example is shown in the bottom row. Panel (e) shows an X-ray projection of a security critical rubber component of a car. Panel (f) depicts the estimated orientation vectors superimposed for every tenth pixel. Note that the estimated orientations fit the orientations in the image well, and we can distinguish between single- and double-oriented regions. To compute the MOP, we used the hierarchical method described in [22] in combination with the decomposition method in Section 2.

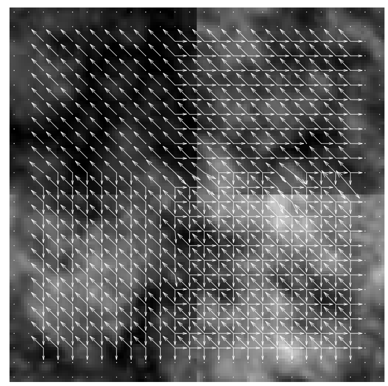
Figure 3 shows an example in which up to three motions occur. Three additively superimposed textured stripes are moving in this sequence such that the center frame (a) consists of four quadrants with different numbers of motions: one with one, two with two and one with three motions. The arrows in (a) illustrate the number and direction of the motions for each individual quadrant. Image (b) depicts the estimated number of motions for each pixel by a color cod: black means no, dark gray one, gray two and light gray three motions. The white cross marks the true border of the regions. The estimates are blurred across the borders due to an integration window (see [17] for further details). The estimated motion vectors according to section 4 are shown (superimposed) in (c). Finally, in Figure 2 we show a result for the Mona Lisa sequence [1]. Image (a) depicts one frame, for which we compute the motion vectors. In this sequence we see the reflection of a box in a glass plate in front of a poster. The box is moving to the left and the poster to the right. In the area of the box, we observe two motions, one for the box and the other for the poster. Panel (b) show the number of estimated motions at each pixel by a color code: black no, gray one and white two. The estimated motion vectors for each layer are shown in (c) and (d), respectively. The MMP were estimated according to the algorithm proposed in [17].



(a)



(b)



(c)

Figure 3. Synthetic example for estimation of one to three transparent superimposed motions.

8. Summary and Conclusions

In this paper we derived a general theory for the estimation of multiple orientations in multidimensional signals. We first solved the MOP decomposition problem for any number of orientations in 2D. We then showed how the general nD decomposition problem can be reduced to a cascade of decompositions in 2D. MOP decomposition is obtained by searching for the roots of a polynomial. The degree of the polynomial equals the number of orientations. Hence, in any dimension, the solution is analytical for up to four orientations. For more than four orientations numerical methods are necessary. The signal superposition model applies also for multi-spectral images and occlusions, thus making it applicable to a wide range of problems. Finally, we have shown application examples for multiple motion and multiple orientation analysis in images. However, we expect that our solutions will lead to further new kinds of applications.

Acknowledgments

Work supported by the Deutsche Forschungsgemeinschaft under Ba-1176/7-2 (to E. Barth) and CNPq/FAPEAM (to C. Mota). We thank M. Boehme for comments on the manuscript.

1 http://www.ks.informatik.uni-kiel.de/~wy/motiondemo.html

References

[1] T. Aach and D. Kunz. Anisotropic spectral magnitude estimation filters for noise reduction and image enhancement. In *Proc. IEEE Int. Conf. Image Processing*, pages 335–8, Lausanne, Switzerland, Sept. 16–19, 1996. IEEE Signal Processing Soc.

[2] T. Aach and D. Kunz. A lapped directional transform for spectral image analysis and its application to restoration and enhancement. *Signal Processing*, 80:2347–64, 2000.

[3] T. Aach, I. Stuke, C. Mota, and E. Barth. Estimation of multiple local orientations in image signals. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, volume 3, pages 553–6, Montreal, Canada, May 17–21, 2004.

[4] J. Bigün. Frequency and orientation sensitive texture measures using linear symmetry. *Signal Process.*, 29(1):1–16, 1992.

[5] J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with application to texture anal-

ysis and optical flow. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(8):775–90, 1991.

[6] T. Darrell and E. Simoncelli. Nulling filters and the separation of transparent motions. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 738–9, New York, June 14–17, 1993. IEEE Computer Press.

[7] C. Z. Erhardt Barth, Terry Caelli. Image encoding, labeling, and reconstruction from differential geometry. *CVGIP: Graphical Model and Image Processing*, 55(6):428–46, Nov. 1993.

[8] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[9] J. Hladůvka and E. Gröller. Direction-driven shape-based interpolation of volume data. In T. Ertl, B. Girod, H. Niemann, and H.-P. Seidel, editors, *Proc. Vision, Modeling, and Visualization*, pages 113–20, Stuttgart, Germany, Nov. 21–23, 2001. Aka GmbH.

[10] T. Iso and M. Shizawa. Detecting L-, T-, and X-Junctions from low-order image derivatives. In *Proc. SPIE Conference on Visual Communications and Image Processing 93*, volume 2094, pages 1185–1197, 1993.

[11] B. Jähne. Image sequence analysis in environmental and live sciences. In B. Michaelis and G. Krell, editors, *Pattern Recognition, 25th DAGM Symposium*, volume 2781 of *Lecture Notes in Computer Science*, pages 608–17, Magdeburg, Germany, Sept. 10–12, 2003. Springer.

[12] M. Kass and W. Witkin. Analyzing oriented patterns. *Computer Vision, Graphics, and Image Processing*, 37:362–85, 1987.

[13] R. Mester. Some steps towards a unified motion estimation procedure. In *Proc. 45th IEEE MidWest Symposium on Circuits and Systems*, Tulsa, OK, Aug. 4–7, 2002.

[14] 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)*, pages 2665–2668, Singapore, Oct. 24–27 2004. IEEE.

[15] C. Mota and E. Barth. On the uniqueness of curvature features. In G. Baratoff and H. Neumann, editors, *Dynamische Perzeption*, volume 9 of *Proceedings in Artificial Intelligence*, pages 175–8, Köln, 2000. Infix Verlag.

[16] C. Mota, I. Stuke, T. Aach, and E. Barth. Estimation of multiple orientations at corners and junctions. In *DAGM04: 26th Pattern Recognition Symposium*, volume 3175 of *LNCS*, pages 163–170, Tuebingen, Aug. 30 – Sept. 1, 2004. Springer Verlag.

[17] C. Mota, I. Stuke, and E. Barth. Analytic solutions for multiple motions. In *Proc. IEEE Int. Conf. Image Processing*, volume II, pages 917–20, Thessaloniki, Greece, Oct. 7–10, 2001. IEEE Signal Processing Soc.

[18] M. Mühlich and T. Aach. A theory of multiple orientation estimation. In H. Bischof and A. Leonardis, editors, *9th European Conference on Computer Vision (ECCV)*, page to appear, Graz, May 7-13 2006. Springer Lecture Notes on Computer Science.

[19] N. Qian, R. A. Andersen, and E. H. Adelson. Transparent motion perception as detection of unbalanced motion signals. iii. modeling. *J Neuroscience*, 14(12):7381–92, 1994.

[20] M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume I, pages 274–8, Atlantic City, NJ, June 1990. IEEE Computer Press.

[21] M. Shizawa and K. Mase. A unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 289–95, Maui, HI, June 1991. IEEE Computer Press.

[22] I. Stuke, T. Aach, E. Barth, and C. Mota. Analysing superimposed oriented patterns. In *Proc. IEEE Southwest Symp. Image Analysis and Interpretation*, pages 133 – 137, Lake Tahoe, NV, 2004.

[23] I. Stuke, T. Aach, C. Mota, and E. Barth. Estimation of multiple motions: regularization and performance evaluation. In B. Vasudev, T. R. Hsing, A. G. Tescher, and T. Ebrahimi, editors, *Image and Video Communications and Processing 2003*, volume 5022 of *Proceedings of SPIE*, pages 75–86, May 2003.

[24] S. D. Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33:116–25, Jan. 1986.

[25] C. Zetzsche and E. Barth. Fundamental limits of linear filters in the visual processing of two-dimensional signals. *Vision Research*, 30:1111–7, 1990.

A. Effect of rotating the coordinate system on the MOP

If the orientations are represented by the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_N$. The MOP are up to scaling factors the components of the tensor

$$\mathbf{C} = \frac{1}{N!} \sum_{\sigma} \mathbf{u}_{\sigma(1)} \otimes \mathbf{u}_{\sigma(2)} \cdots \otimes \mathbf{u}_{\sigma(N)}, \qquad (44)$$

where the sum runs over all permutations σ of $1, \ldots, N$ [21, 16]. The actual relation is

$$c_w = N(w) C_w \qquad (45)$$

where $w = w_1 w_2 ... w_N$ is an ordered word with characters $w_j \in \{1, 2, \ldots, N\}$, C_w are the components of $\mathbf{C}$, and $N(w)$ is the number of possible words with the same characters as w. If the coordinate system changes by a rotation, say

$$\mathbf{x}' = \mathbf{R}\mathbf{x}, \qquad (46)$$

the components of $\mathbf{C}$, and therefore, the MOP rotate accordingly.