A Tensor Approach for Local Structure Analysis in Multi–Dimensional Images

Horst Haußecker and Bernd Jähne

Interdisciplinary Center for Scientific Computing, IWR, University of Heidelberg
Im Neuenheimer Feld 368, D-69120 Heidelberg, Germany
Email: horst.haussecker@iwr.uni-heidelberg.de

Abstract

Using the n–dimensional *structure tensor*, linear symmetries within local neighborhoods of multi–dimensional image structures can be detected. Depending on the image content this information yields a a variety of features, such as velocity, local orientation, texture and structure information that can be used for visualization. The proposed technique is well suited to be applied within a multi–resolution framework as well as for multi–component images and is not restricted in the dimensionality of the image content. However, the structure tensor requires a careful interpretation with increasing dimensionality. In this paper the implementation of the tensor approach will be introduced and the application to 3D image analysis and motion estimation in in spatio-temporal images will be discussed in detail.

1 Introduction

Analyzing textured images the structure of the grey value distribution within a local neighborhood of a pixel plays an important role. The performance of the human visual system in separating textured patterns results in its ability to identify coherent structures of arbitrary shape on multiple scales. This allows to interpret the characteristic features of image structures independently from their absolute grey values. One possible approach to local structure analysis is the search for *local symmetries* in the grey value distribution and as a basic feature the search for linear symmetry, i. e. *local orientation*.

Local orientation denotes the property of a local neighborhood within a n-dimensional image to contain lower-dimensional grey value structures with a distinct orientation. This orientation can be expressed by an *orientation angle* in the n-dimensional image space. The orientation analysis itself consists of assigning an orientation angle to every point within the image together with a measure of distinctness or *coherence* which quantifies the presence of this local orientation. Local orientation in 2D images can reach from a slight anisotropy in the grey value structure or faint lines up to very expressed lines or periodical structures that dominate the image. In 3D images local orientation does not only include extruded structures such as edges of objects but also layered structures or 2D sub-surfaces in the 3D space e. g. faces of objects. A lack of local orientation is given if the local neighborhood contains a constant grey value or distributed 3D textures in all directions such as e. g. random noise and corners of 3D objects. This enumeration of possible linear symmetries in images shows the complexity of the problem. Not only the kind of linear symmetry has to be determined but also its coherence and orientation in n dimensions. An image sequence constitutes a 3D spatio-temporal image. In this case local orientation reveals different types of motion since the angle of orientation is directly related to the velocity. Not only apparent motion but also the presence of an aperture problem can be extracted from 3D spatio-temporal images with the help of the structure tensor method.

This paper proposes a unified method to identify and separate these different local structures in n-dimensional images and to simultaneously assign a quantitative measure of coherence. Starting with the mathematical basis of the structure tensor method we focus on its practical implementation using efficient filter techniques and its application to motion analysis in 3D spatio temporal images.

In Proceedings "3D Image Analysis and Synthesis '96", Erlangen (Germany), November 18-19, 1996
edited by G. Girod, H. Niemann and H.-P. Seidel, pp. 171-178, 1997, Sankt Augustin: infix
doi: 10.5281/zenodo.16444

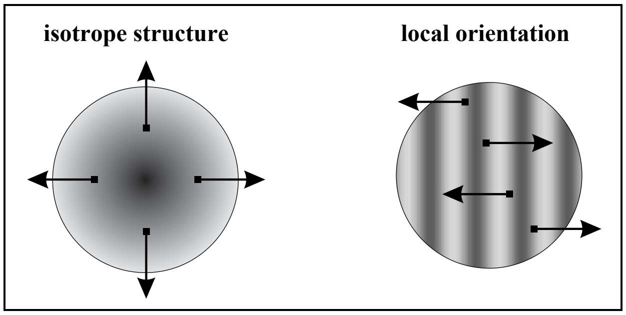

Figure 1: Schematic visualization of the gradient of isotropic and ideal oriented 2D image structures. The arrows indicate the direction of the gradient vector at different positions.

2 From gradients to tensors

In order to introduce the structure tensor we focus on 2D images and show how the mathematical formalism can be extended to n–dimensional images. Searching for an appropriate description of local orientation we start with the gradient $\vec{\nabla} g(\vec{x})$ of the image:

$$\vec{\nabla} g(\vec{x}) = (g_x(\vec{x}) \ , \ g_y(\vec{x})) = \left(\frac{\partial g(\vec{x})}{\partial x} \ , \ \frac{\partial g(\vec{x})}{\partial y} \right),$$

with $g(\vec{x})$ denoting the two-dimensional image structure and $\partial/\partial x$ and $\partial/\partial y$ the first order differential operators in horizontal and vertical direction, respectively.

Calculating the gradient transforms the image into a vectorial image structure with two components which constitutes a more abstract representation of the image content. From differential geometry it is well known, that the gradient always points into the direction normal to the steepest edges. The norm $\| \vec{\nabla} g \|$ of the gradient quantifies the steepness of the edge and therefore expresses the distinctness of the oriented structure. This leads to the suggestion that the gradient image already contains all information about the presence of local orientation and that the direction of the gradient represents the angle of orientation.

Figure 1 shows a schematic visualization of the gradient field for two different grey value structures. In order to compute a measure for local orientation the information at isolated pixels has to be averaged over a local neighborhood U in an appropriate way. Averaging the gradient fields in Figure 1 yields the average value

$$\overline{\vec{\nabla} g(\vec{x})} = \left(\int\limits_U g_x(\vec{x}) \, d\vec{x} \ , \ \int\limits_U g_y(\vec{x}) \, d\vec{x} \right) = \vec{0}.$$

in both cases. The grey value structure in the left illustration contains orientations in all possible directions, i. e. within the circular neighborhood no distinct direction is detectable. The right illustration in Figure 1 shows an ideal oriented grey value structure. In both cases, however, the average gradient field yields the zero vector. The reason for this shortcoming of the image gradient is the appearance of different signs in the gradient at increasing and decreasing edges. The orientation vector of the structure in the right part of Figure 1 should not change if the image is rotated by $180\,^\circ$.

In order to get an appropriate representation of local orientation that is invariant under rotations by $180\,^\circ$ a quadratic function in the gradient has to be found which does not depend on the sign of the gradient. The norm of the gradient

$$\| \vec{\nabla} g \| = \left(\vec{\nabla}^T g \, \vec{\nabla} g \right)^{1/2} = \left(g_x^2 + g_y^2 \right)^{1/2} \qquad (1)$$

fulfills this condition but constitutes a scalar value that does not allow to extract any information about the angle of local orientation.

The requirements for an ideal operator for local orientation analysis can be summarized as follows:

1. The angle of local orientation has to be computed
2. The orientation angle has to be invariant under a rotation by $180\,^\circ$
3. An additional measure of coherence has to be extracted

The prerequisite of a coherence measure is essential for local orientation analysis. If only the angle of coherence is computed it is not possible to distinguish between a faint anisotropic in the grey value structure and an ideal orientation. This kind of error estimate is often underrepresented in digital image processing as well as error estimate in general. A mathematical operator always extracts a result out of an image even if the necessary requirements on the image content are not given. After the computation the result has to be subject to an interpretation of possible errors. It will be shown later on, that the structure tensor approach does not only allow to distinguish different possible cases of the image content but also to detect the quality of the computation.

A method to describe local orientation that combines all summarized requirements is given by the construction of a *tensor*. Let $\vec{v}(\vec{x})$ be a vector at the point $\vec{x}$ of the image that contains the orientation information. A possible example of such a vector is the gradient $\vec{\nabla} g(\vec{x})$. The goal is to estimate the mean orientation $\vec{r}(\vec{x})$ within a local neighborhood U with no regard to the sign of $\vec{v}$. From the components of $\vec{v}$ a symmetric tensor $\mathbf{J}$ can be constructed in the following way:

$$\mathbf{J}(\vec{x}) = \int\limits_U \vec{v}(\vec{x}')\, \vec{v}^T(\vec{x}')\, d\vec{x}', \tag{2}$$

with the components

$$\mathrm{J}_{pq} = \int\limits_U v_p(\vec{x}')\, v_q(\vec{x}')\, d\vec{x}'. \tag{3}$$

The tensor $\mathbf{J}$ is a quadratic function of $\vec{v}$ and therefore invariant under rotations by $180°$. In contrast to the inner product $\|\vec{v}\|$ of the vector $\vec{v}$ the orientation information is not lost. It can be shown that the vector $\vec{r}(\vec{x})$ can directly be computed by an eigenvalue analysis of $\mathbf{J}$. It is given as the eigenvector of $\mathbf{J}$ that corresponds to the largest eigenvalue.

The projection of a vector $\vec{v}(\vec{x})$ onto the direction $\vec{r}(\vec{x})$ is represented by the inner product

$$\vec{v}^T\, \vec{r} = v_x\, r_x + v_y\, r_y. \tag{4}$$

If both vectors are perpendicular to each other the inner product vanishes and if they are parallel or point into opposite directions it reaches a maximal or minimal value. The searched vector $\vec{r}(\vec{x})$ constitutes an estimate for the mean orientation within the local neighborhood U that maximizes the following expression:

$$S = \int\limits_U \left(\vec{v}^T(\vec{x}')\, \vec{r}(\vec{x})\right)^2\, d\vec{x}'. \tag{5}$$

Under this condition the vector $\vec{r}(\vec{x})$ is as much parallel or anti-parallel as possible to all vectors $\vec{v}(\vec{x}')$ within the local neighborhood U surrounding the point $\vec{x}$. With the already introduced tensor (2) this expression can be written as

$$S = \vec{r}^T\, \mathbf{J}\, \vec{r} \;\rightarrow\; \text{Maximum}. \tag{6}$$

Basic linear algebra tells us that this expression reaches a maximum if the vector $\vec{r}$ is given by the *eigenvector* of the the tensor $\mathbf{J}$ to the *maximum eigenvalue* [13]. The search for local orientation therefore reduces to an eigenvalue analysis of the orientation tensor $\mathbf{J}$.

This relationship is independent from the dimensionality of the vector space and therefore yields the same solution for multi-dimensional image structures without change. In the next chapter the structure tensor is proposed as a possible realization of the general orientation tensor and the practical implementation of the computation will be introduced.

3 Definition and implementation of the structure tensor

So far we have shown that in order to describe linear symmetry of the grey value structure within a local neighborhood of a multi–dimensional image, a symmetric tensor proves to be the most adequate representation of the image content. One possible realization is the *structure tensor* which was initially introduced by [9] for the two–dimensional case to compute the angle of *local orientation*. Replacing the vector $\vec{v}(\vec{x})$ in equation (2) by the n–dimensional gradient vector $\vec{\nabla} g(\vec{x})$ at a point $\vec{x}$ the general form of the n–dimensional structure tensor can be expressed as follows [7]:

$$\begin{aligned} \mathbf{J}(\vec{x}) &= \int\limits_U \vec{\nabla} g(\vec{x}')\, \vec{\nabla} g^T(\vec{x}')\, d\vec{x}' \\ &= \int\limits_{-\infty}^{\infty} h(\vec{x} - \vec{x}')\, \vec{\nabla} g(\vec{x}')\, \vec{\nabla} g^T(\vec{x}')\, d\vec{x}', \end{aligned} \tag{7}$$

with the components

$$J_{pq} = \int\limits_{-\infty}^{\infty} h(\vec{x} - \vec{x}')\, \frac{\partial g(\vec{x}')}{\partial x_p}\, \frac{\partial g(\vec{x}')}{\partial x_q}\, d\vec{x}'. \tag{8}$$

With $g(\vec{x})$ we denote the n–dimensional image structure and $\partial g(\vec{x})/\partial x_q$ represents the partial derivation along the direction of the q-axis. The information within a local neighborhood U around the central point $\vec{x}$ is weighted by a window–function $h(\vec{x} - \vec{x}')$. In practical applications the size of the local neighborhood U represents the area over which the orientation information is averaged. The whole computation of the structure tensor components therefore reduces to a point-wise product of the partial derivatives along the corresponding coordinate directions and a following convolution of the result.

The implementation of the tensor components can be carried out very efficiently by standard image processing operators. Identifying the convolution in (8) with a n–dimensional smoothing of the product of partial derivatives, each component of the structure tensor can be computed as

$$J_{pq} = \mathcal{B}\,(\mathcal{D}_p \cdot \mathcal{D}_q), \tag{9}$$

with the n–dimensional smoothing operator $\mathcal{B}$ and the differential operator $\mathcal{D}_p$ in the direction of the coordinate x_p. Using a binomial operator the smoothing can be performed very efficiently on a multi-grid data structure [4]. A more critical point is the choice of an appropriate differential operator. It can be shown that special optimized recursive filters reduce the error up to an order of magnitude compared to standard differential operators [11].

4 Eigenvalue analysis in two and three dimensions

After the structure tensor has been computed an eigenvalue analysis has to be performed to extract the parameters of local orientation. The two-dimensional case will be treated separately from the 3D-case since the complexity of the problem increases with growing dimensionality. In two dimensions all necessary parameters can be computed at once, whereas in three and more dimensions the eigenvalue problem can only be solved iteratively.

4.1 2D images

In two dimensions the structure tensor (8) has the structure

$$\mathbf{J} = \begin{bmatrix} \mathrm{J}_{xx} & \mathrm{J}_{xy} \\ \mathrm{J}_{xy} & \mathrm{J}_{yy} \end{bmatrix}. \tag{10}$$

The eigenvalue analysis reduces to a single rotation of the coordinate system into the direction of the principal axes system, which can be performed directly without iteration. In this coordinate system the two axes point into the directions of the eigenvectors of maximum and minimum eigenvalues, respectively. The three independent components of (10) can be transformed into the desired parameters of local orientation [7]: the *orientation vector* $\vec{o}$

$$\vec{o} = \begin{bmatrix} J_{yy} - J_{xx} \\ 2\,J_{xy} \end{bmatrix} \tag{11}$$

and a measure of *coherence* C

$$C = \frac{(J_{yy} - J_{xx})^2 + (2\,J_{xy})^2}{(J_{xx} + J_{yy})^2}. \tag{12}$$

These quantities still constitute a three-component information that is necessary to sufficiently describe linear symmetry of a local neighborhood within the 2D image structure. The orientation vector $\vec{o}$ represents the orientation of the principal axes system and the coherence C quantifies the presence of local orientation. This allows in two dimensions to directly compute the *angle of local orientation* together with a *measure of certainty*. The coherence measure C relates the length of the orientation vector to the length of the gradient vector and ranges between $C = 0$ for no apparent orientation and $C = 1$ for ideal orientation. With this definition the coherence gets independent from the absolute brightness of the image.

The orientation angle Θ which represents the angle of rotation between the image coordinate system and the principle axes system of the structure tensor can be computed from the orientation vector as

$$\tan 2\Theta = \frac{o_y}{o_x} = \frac{2\,J_{xy}}{J_{yy} - J_{xx}}. \tag{13}$$

Comparing (13) with the general expression for the angle Φ between an arbitrary vector $\vec{k} = (k_x, k_y)$ and the x-axis of the coordinate system

$$\tan \Phi = \frac{k_y}{k_x}, \tag{14}$$

shows that the angle between the orientation vector $\vec{o}$ and the x-axis of the image coordinate system is twice the orientation angle Θ. This angle doubling is a consequence of the invariance of local orientation under rotations of 180° and automatically follows out of the eigenvalue analysis of $\mathbf{J}$ [7].

4.2 3D image structures

In three dimensions the problem gets more complex. The 3D structure tensor has the structure

$$\mathbf{J} = \begin{bmatrix} \mathrm{J}_{xx} & \mathrm{J}_{xy} & \mathrm{J}_{xz} \\ \mathrm{J}_{xy} & \mathrm{J}_{yy} & \mathrm{J}_{yz} \\ \mathrm{J}_{xz} & \mathrm{J}_{yz} & \mathrm{J}_{zz} \end{bmatrix} \qquad (15)$$

and

$$\mathbf{J} = \begin{bmatrix} \mathrm{J}_{xx} & \mathrm{J}_{xy} & \mathrm{J}_{xt} \\ \mathrm{J}_{xy} & \mathrm{J}_{yy} & \mathrm{J}_{yt} \\ \mathrm{J}_{xt} & \mathrm{J}_{yt} & \mathrm{J}_{tt} \end{bmatrix} \qquad (16)$$

for 3D volumetric images and 3D spatio-temporal images respectively.

Besides the fact, that the symmetric 3×3 structure tensor contains six independent components that have to be interpreted, the eigenvalue analysis can only be performed iteratively. Nevertheless the six components yield a deep insight into the structure of 3D images or the dynamics of motion in image sequences.

4.2.1 Rank analysis

The different classes of structures can be identified without explicitly solving the eigenvalue problem. The structure tensor contains the entire information on the first–order structure of the grey value function in a local neighborhood. By analyzing the *rank of the matrix* four different cases of spatial or spatio–temporal structures can be distinguished. In Table 1 these cases are compared for 3D images and 3D spatio–temporal images.

A constant grey value in one or more directions that are linear independent from each other reduces the dimensionality of the grey value distribution by the number of these directions. This is directly revealed by the rank of the tensor since the structure tensor contains partial derivations of the grey value structure along all different coordinate directions. The smallest and the largest possible rank represent a constant grey value or grey value changes in all possible directions. Both cases show no apparent orientation and therefore no orientation angle or orientation vector can be assigned to these image points.

As an input for visualization techniques a rank analysis of the 3D structure tensor therefore yields information about the presence of constant brightness, 2D–sub-surfaces, edges and corners of the grey value distribution at any point within a

rank	3D image textures
0	constant grey value
1	layered textures
2	extruded textures
3	distributed 3D textures
rank	**3D images of simple objects**
0	constant grey value
1	faces of objects
2	edges of objects
3	corners of objects
rank	**3D spatio–temporal images**
0	constant grey value, no motion
1	spatial orientation and constant motion (moving edge)
2	distributed spatial structure and constant motion
3	no coherent motion (motion discontinuities, fast accelerations, ...)

Table 1: Rank of the structure tensor for different cases of 3D image structures. For the 3D spatio-temporal images the grey value structure within the 2D images and the different types of motion are classified.

3D image structure. In 3D spatio-temporal images the different cases of grey value structures directly correspond to different kinds of motion and image content of the scenes (Table 1). In section 5 the topic of motion analysis will be discussed in more detail.

4.2.2 Eigenvalue analysis

As already mentioned the full 3D eigenvalue analysis can only be carried out iteratively. It still consists of a rotation of the coordinate system into the principle axes system but the rotation has to be done in 3D space. A standard procedure in numerical eigenvalue analysis is the *Jacobi transformation*. This method consists of a sequence of orthogonal similarity transformations where each transformation (a *Jacobi rotation*) is just a plane rotation designed to annihilate one of the off-diagonal matrix-elements of the structure tensor. Successive transformations undo previously set zeros, but the off-diagonal elements nevertheless get smaller, until the matrix is diagonal to machine precision [12]. The eigenvectors are computed iteratively and the eigenvalues are the corresponding diagonal elements of the diagonalized matrix. The Jacobi method is absolutely foolproof for all real symmetric matrices which the structure tensor can be assured to be. This is very advantageous because it does not depend on the image content.

Although the Jacobi method gets slow for matrices of the order greater than about 10, the algorithm is sufficiently fast for $n = 3$ and much simpler than the more efficient methods. Rather than trying to speed up the whole eigenvalue analysis, a major decrease in computation time can be achieved by preselecting interesting image regions. One possible method that proved to accelerate the computation without loosing important information is to analyze the *trace* of the matrix **J**

$$\text{trace}\,(\mathbf{J})(\vec{x}) = \sum_{k=1}^{n} \mathrm{J}_{kk}(\vec{x}) \qquad (17)$$

for each point before starting the Jacobi transformation. Since the trace of a matrix is invariant under orthogonal similarity transformations this can be done in advance after computing the structure tensor. The case of constant grey value, i. e. $\text{rank}\,(\mathbf{J}) = 0$ (ref. Table 1) corresponds to $\text{trace}\,(\mathbf{J}) = 0$ and these areas of no apparent orientation can be excluded from the eigenvalue analysis by just adding up the diagonal elements and thresholding the result. At these points no local orientation can be computed and therefore the 3D orientation vector is set zero:

$$\vec{o}(\vec{x}) = 0 \quad \text{if} \quad \text{trace}\,(\mathbf{J})(\vec{x}) < \text{threshold}.$$

At the remaining points the 3D eigenvalue analysis can be carried out. Depending on the rank of the matrix these points correspond to extruded or layered structures (for 3D textures), to object faces or edges (for simple 3D objects) and generally have a lower dimensionality than the image. The direction of the 3D eigenvector that corresponds to the largest eigenvalue points perpendicular to directions of constant grey value. For layered textures this direction is normal to the layers and for 3D objects normal to the object faces. For extruded textures, on the other hand, the eigenvector that corresponds to the smallest eigenvalue points into the direction of constant grey value which is unique in 3D space and represents the direction of ideal 3D orientation, i. e. the direction of an object edge. All other cases, including noise and isotropic structures such as object corners, can be identified by having three independent eigenvalues of similar values.

4.2.3 Type measure

In order to identify the different types of structure or motion we introduce a *type measure* T which relates the magnitude of the three eigenvalues E_x, E_y and E_t of the structure tensor:

$$T = \frac{(E_x - E_y)^2 + (E_x - E_t)^2 + (E_y - E_t)^2}{E_x^2 + E_y^2 + E_t^2}.$$

With the type measure T all possible types of 3D image structure or motion are quantified by a scalar value that ranges between 0 and 2. In Table 2 the possible values are summarized together with the corresponding rank of the structure tensor (ref. Table 1).

rank	0	1	2	3
T	0	2	1	0

Table 2: Type measure T compared to the rank of the structure tensor.

Comparing the type measure T to the rank of the structure tensor shows not much similarity between them. For practical applications, however, the type measure T proved to be very useful. It maps both cases where no linear symmetry or motion estimation is possible to the value zero. Therefore it behaves similar to the coherence measure C of the 2D structure tensor. For those cases in which linear symmetry is present the value lies within the range of 1 to 2 and enables to distinguish between the cases for $\text{rank}\,(\mathbf{J}) = 1$ and $\text{rank}\,(\mathbf{J}) = 2$. This allows e. g. to switch between (20), (21) and (19) in order to appropriately calculate the optical flow.

In real images it is not always possible to distinguish between the different possible cases unambiguously. For those cases that constitute a mixture between two types of structure or motion the type measure T lies in between the values for both cases. This property of the type measure differs from the rank of the matrix, which can only take the integer values 0, ... , 3. This ‘fuzzy’ behavior of the type measure T allows to deal with intermediate cases in terms of *fuzzy logic.*

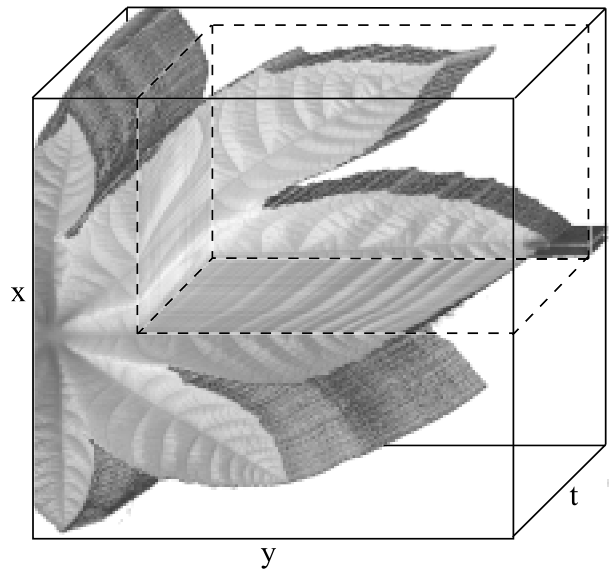

Figure 2: Visualization of a 3D spatio-temporal image. The image sequence shows a leaf of a castor-oil plant that grows over time. In order to visualize the interior structure of the image the upper front corner of the image cube has been cut along the dashed lines [10].

5 Motion analysis

As a special - and very important - case of 3D image analysis we want to focus in this section on the application of the structure tensor method to motion analysis in 3D spatio-temporal images, i. e. extended image sequences.

The 3D spatio-temporal image structure results from both the structure of objects in single images as well as from their apparent motion in the image plane. Depending on the shape and dynamics of objects the spatio-temporal structure can be rather difficult. One example of such a spatio-temporal image is visualized in Figure 2. In this special case the dynamics of growth processes in plants have to be extracted from image sequences showing plant leafs over a time span of some days [10].

The image in Figure 2 directly reveals a basic property of spatio-temporal images: The displacement within consecutive images yields inclined image structures with respect to the temporal axis. The relation between the orientation angle and the optical flow is given by

$$\vec{u} = -\begin{bmatrix} \tan\varphi_1 \\ \tan\varphi_2 \end{bmatrix}, \tag{18}$$

where $\vec{u} = (u_x, u_y)$ denotes the optical flow on the image plane and the angles φ_1 and φ_2 define the angles between the plane normal to the lines of constant grey value and the x_1 and x_2 axes [6].

Since motion appears as orientation in space-time images the concepts of orientation analysis can be extended to motion analysis. In 2D *xt*-images the concept is straightforward, because only one component of the velocity can be extracted which corresponds to the orientation angle. The coherence gives an estimate for the certainty of motion computation. In 3D images the problem gets more complex.

Table 1 summarizes the different types of motion that are possible. Again the two extreme cases of $\text{rank}(\mathbf{J}) = 0$ and $\text{rank}(\mathbf{J}) = 3$ represent no apparent linear motion. In the first case the constant grey value does not allow to extract any information. For the case of an isotropic grey value distribution no coherent motion is detectable. The two interesting cases are those with $\text{rank}(\mathbf{J}) = 1$ and $\text{rank}(\mathbf{J}) = 2$.

In the case of $\text{rank}(\mathbf{J}) = 1$ an already oriented image structure moves with a constant velocity. In this case the spatio-temporal grey value structure has the shape of a plane with a certain inclination in 3D space. From the two components of the optical flow only the projection $v_\perp$ of the 2D velocity $\vec{u}$ on the direction perpendicular to the local orientation in the 2D image can be extracted. This is the well known *aperture problem* in optical flow computation. Only one of the three eigenvectors has an eigenvalue larger than zero. This eigenvector $\vec{e}_l = (e_{l,x}, e_{l,y}, e_{l,t})$ points normal to the plane in 3D space and can be used to compute the normal velocity $v_\perp$:

$$v_\perp = -\frac{e_{l,t}}{\sqrt{e_{l,x}^2 + e_{l,y}^2}}. \tag{19}$$

For $\text{rank}(\mathbf{J}) = 2$ an isotropic grey value structure moves with a constant velocity. This is the optimal case for motion analysis, corresponding to an ideal oriented structure. The spatio-temporal grey value structure lies on a line in 3D space. The orientation of this line yields the two components u_x and u_y of the optical flow. With the eigenvector $\vec{e}_s = (e_{s,x}, e_{s,y}, e_{s,t})$ to the smallest eigenvalue pointing into the direction of the line, these components can be computed as:

$$v_x = \frac{e_{s,x}}{e_{s,t}} \tag{20}$$

and

$$v_y = \frac{e_{s,y}}{e_{s,t}}. \tag{21}$$

This detailed discussion of possible cases shows that the structure tensor method is able to distinguish between different types of motion and to simultaneously compute the relevant motion parameters. The aperture problem cannot be solved in general, since it constitutes a physical shortcoming, but it can be identified by the rank of the structure tensor. This identification is very important in order to interpret the result of the motion analysis appropriately. If it could not be identified the projected motion $v_\perp$ would be treated as the real motion $\vec{u}$ which can lead to tremendous errors.

6 Conclusions

In this paper we introduced a unique approach to local structure analysis in multi-dimensional images. The structure tensor method yields both the type of image structure together with a measure of coherence and the angle of orientation. In spatio-temporal images it can be used to distinguish different types of motion and to compute the optical flow. This paper focuses on the theoretical foundations of the structure tensor approach and its practical application. Results of this technique will be presented at the oral presentation.

Acknowledgments

We gratefully acknowledge financial support by the 'Deutsche Forschungsgemeinschaft', DFG, within the frame of the research initiative 'Image Sequence Analysis to Investigate Dynamic Processes'.

References

[1] Bigün, J. and G. H. Granlund, "Optimal Orientation Detection of Linear Symmetry," Proc. First Intern. Conf. on Computer Vision, ICCV, London, England, June 8-11, 1987.

[2] Bigün, J., "Local Symmetry Features in Image Processing," Linköping Studies in Science and Technology, Dissertations No. 179, Linköping, Schweden, 1988.

[3] Bigün, J., and H. M. du Buf, 'N-folded Symmetries by Complex Moments in Gabor Space and their Application to Unsupervised Texture Segmentation," IEEE Trans. on Pattern Analysis and Machine Intell., Vol. 16, No. 1, pp. 80 - 87, 1994.

[4] Haußecker, H. and B. Jähne, "Effiziente Filterverfahren auf Mehrgitter–Datenstrukturen," Bildverarbeitung 95': Forschen, Entwickeln, Anwenden, Symposium, 29. 11. – 01. 12. 95, pp. 43 - 57, Technische Akademie Esslingen, R. J. Ahlers, Hrsg., 1995.

[5] Haußecker, H., "Lokale Orientierung als Basis für die Texturanalyse," 3. Heidelberger Bildverarbeitungsforum: Texturanalyse, 2. Juli 1996, Forschungszentrum ABB AG, Heidelberg, 1996.

[6] Jähne, B., "Digital Image Processing: Concepts, Algorithms and Scientific Applications," 3rd ed., Springer, Heidelberg, 1995.

[7] Jähne, B., "Spatio-Temporal Image Processing, Theory and Applications," Lecture Notes in Computer Science, *Vol. 751*, Springer, Berlin, 1993.

[8] Kass, M. und A. Witkin, "Analyzing Oriented Patterns," Computer Vision, Graphics and Image Processing, vol. 37, pp. 362 - 385, 1987.

[9] Knutsson, H., "Representing Local Structure using Tensors," 6th Scandinavian Conf. Image Analysis, Oulu, Finland, pp. 244 - 251, 1989.

[10] Lell M., "Ortsaufgelöste Bestimmung von Blattwachstum durch Strukturanalyse von Bildsequenzen aus dem nahen Infrarot," Diplomarbeit, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, IWR, Universität Heidelberg, 1996.

[11] Leue C., F. Hering, P. Geißler and B. Jähne, "Segmentierung von Partikelbildern in der Strömungsvisualisierung," 18. DAGM–Symposium, 11. – 13. September 1996, Heidelberg, 1996.

[12] Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, "Numerical Recipes in C: The Art of Scientific Computing," Cambridge University Press, New York, 1992.

[13] Rao, A. R., "A Taxonomy for Texture Description and Identification," Springer Verlag, New York, 1990.