2

Adaptive Structure Tensors and their Applications

Thomas Brox[1], Rein van den Boomgaard[2], François Lauze[3], Joost van de Weijer[2], Joachim Weickert[1], Pavel Mrázek[1], and Pierre Kornprobst[4]

[1] Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science, Building 27, Saarland University, 66041 Saarbrücken, Germany
{brox,weickert,mrazek}@mia.uni-saarland.de
[2] Intelligent Sensory Information Systems Group, Informatics Institute, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
rein@science.uva.nl
[3] IT University of Copenhagen, Glentevej 67, DK-2400 Copenhagen, Denmark
francois@itu.dk
[4] Odyssée Project, INRIA Sophia-Antipolis, 2004 Route des Lucioles, 06902 Sophia Antipolis, France
Pierre.Kornprobst@sophia.inria.fr

Summary. The structure tensor, also known as second moment matrix or Förstner interest operator, is a very popular tool in image processing. Its purpose is the estimation of orientation and the local analysis of structure in general. It is based on the integration of data from a local neighborhood. Normally, this neighborhood is defined by a Gaussian window function and the structure tensor is computed by the weighted sum within this window. Some recently proposed methods, however, adapt the computation of the structure tensor to the image data. There are several ways how to do that. This chapter wants to give an overview of the different approaches, whereas the focus lies on the methods based on robust statistics and nonlinear diffusion. Furthermore, the data-adaptive structure tensors are evaluated in some applications. Here the main focus lies on optic flow estimation, but also texture analysis and corner detection are considered.

2.1 Introduction

Orientation estimation and local structure analysis are tasks that can be found in many image processing and early vision applications, e.g. in fingerprint analysis, texture analysis, optic flow estimation, and in geo-physical analysis of soil layers. The classical technique to estimate orientation is to look at the set of luminance gradient vectors in a local neighborhood. This leads to a very popular operator for orientation estimation, the matrix field of the so-called *structure tensor* [4, 10, 16, 20, 38].

The concept of the structure tensor is a consequence of the fact that one can only describe the local structure at a point by considering also the data of its neighborhood. For instance, from the gradient at a single position, it is not possible to distinguish a corner from an edge, while the integration of the gradient information in the neighborhood of the pixel gives evidence about whether the pixel is occupied by an edge or a corner. Further on, the consideration of a local neighborhood becomes even more important as soon as the data is corrupted by noise or other disturbing artifacts, so that the structure has to be estimated before the background of unreliable data.

The structure tensor therefore extends the structure information of each pixel, which is described in a first order approximation by the gradient at that pixel, by the structure information of its surroundings weighted with a Gaussian window function. This comes down to the convolution of the structure data with a Gaussian kernel, i.e. Gaussian smoothing.

Note however, that the smoothing of gradients can lead to cancellation effects. Consider, for example, a thin line. At one side of the line there appears a positive gradient, while at the other side the gradient is negative. Smoothing the gradients will cause them to mutually cancel out. This is the reason why in the structure tensor, the gradient is considered in form of its outer product. The outer product turns the gradient vector ∇I of an image I into a symmetric positive semi-definite matrix, which we will refer to as the *initial matrix field*

$$J_0 := \nabla I \nabla I^\top = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} . \tag{2.1}$$

Subscripts thereby denote partial derivatives. The structure tensor can be easily generalized from scalar-valued data to vector-valued data. As with the matrix representation it is possible to sum up gradient information, the structure information from all channels of a vector-valued image $\mathbf{I} = (I_1, \ldots, I_N)$ can be integrated by taking the sum of all matrices [8]:

$$J_0 := \sum_{i=1}^{N} \nabla I_i \nabla I_i^\top . \tag{2.2}$$

The structure tensor for a certain neighborhood of scale ρ is then computed by convolution of the components of J_0 with a Gaussian kernel K_ρ:

$$J_\rho = K_\rho * J_0 . \tag{2.3}$$

The smoothing, i.e. the integration of neighborhood information, has two positive effects on orientation estimation. Firstly, it makes the structure tensor robust against noise or other artifacts, and therefore allows a more reliable estimation of orientation in real-world data. Secondly, it distributes the information about the orientation into the areas between edges. This is a very important effect, as it allows to estimate the dominant orientation also at those points in the image where the gradient is close to zero. The dominant

orientation can be obtained from the structure tensor as the eigenvector to the largest eigenvalue. An operator which is closely related to the structure tensor is the boundary tensor discussed in Chap. 4 by Köthe.

There are many applications for the structure tensor in the field of image processing. One popular application is optic flow estimation based on the local approach of Lucas and Kanade [21]. In optic flow estimation one searches for the spatio-temporal direction with least change in the image, which is the eigenvector to the *smallest* eigenvalue of the structure tensor [4, 15].

Another application for orientation estimation is texture analysis. Here the dominant orientation extracted from the structure tensor can serve as a feature to discriminate textures [4, 28]. The dominant local orientation is also used in order to drive anisotropic diffusion processes, which enhance the coherence of structures [39]. Often the structure tensor is also used as a feature detector for edges or corners [10]. An application apart from image processing is a structure analysis for grid optimization in the scope of fluid dynamics [34].

Although the classic structure tensor has proven its value in all these applications, it also holds a drawback. This becomes apparent as soon as the orientation in the local neighborhood is not homogeneous like near the boundary of two different textures or two differently moving objects. In these areas, the local neighborhood induced by the Gaussian kernel integrates ambiguous structure information that actually does not belong together and therefore leads to inaccurate estimations.

There are two alternatives to remedy this problem. One is to adapt the neighborhood to the data. A classical way of doing so is the Kuwahara-Nagao operator [2, 18, 25]. At a certain position in an image this operator searches for a nearby neighborhood where the response (the orientation) is more homogeneous than it is at the border. That response is then used at the point of interest. In this way the neighborhoods are not allowed to cross the borders of the differently oriented regions. In [36] it was shown that the classic Kuwahara-Nagao operator can be interpreted as a 'macroscopic' version of a PDE image evolution that combines linear diffusion (smoothing) with morphological sharpening (a shock filter in PDE terms). A very similar approach is to use adaptive Gaussian windows [23, 26] for choosing the local neighborhood. Also by nonlinear diffusion one can perform data-adaptive smoothing that avoids the integration of ambiguous data [7, 41].

A second possibility to enhance local orientation estimation is to keep the non-adaptive window, but to clearly choose one of the ambiguous orientations by means of robust statistics [37]. This chapter will describe both approaches and will show their performance in the most common applications also in comparison to the conventional structure tensor. Note that for a data-adaptive structure tensor to reveal any advantages, discontinuities or mixed data must play a role for the application. Some applications where this is the case are optic flow estimation, texture discrimination, and corner detection.

Chapter organization. The chapter is organized as follows. In the next section we give an overview on data-adaptive structure tensors. The approaches using robust statistics and nonlinear diffusion are described in detail and relations between methods are examined. In Sect. 2.3 – Sect. 2.5 the structure tensor is applied to optic flow estimation, texture analysis, and corner detection. Some experiments show the superiority of adaptive structure tensors in comparison to the classic structure tensor and differences between the methods. The chapter is concluded by a brief summary in Sect. 2.6.

2.2 Data-adaptive Structure Tensors

An early approach to data-adaptive structure tensors is the *gray value local structure tensor* of Nagel and Gehrke [26], which has been designed for its use in spatio-temporal optic flow estimation. Instead of using a fixed isotropic Gaussian kernel K_ρ for smoothing the structure tensor, a space-dependent Gaussian

$$G(x) = \frac{1}{\sqrt{(2\pi)^3 |\Sigma(x)|}} e^{-\frac{1}{2} x^\top \Sigma(x)^{-1} x} \tag{2.4}$$

is employed, which is parameterized by the covariance matrix $\Sigma(x)$. This covariance matrix is locally adapted to the image by setting

$$\Sigma(x) = U(x) \begin{pmatrix} \sigma_{min} + \frac{\sigma_{max}^2}{1+\sigma_{max}^2 \lambda_1(x)} & 0 & 0 \\ 0 & \sigma_{min} + \frac{\sigma_{max}^2}{1+\sigma_{max}^2 \lambda_2(x)} & 0 \\ 0 & 0 & \sigma_{min} + \frac{\sigma_{max}^2}{1+\sigma_{max}^2 \lambda_3(x)} \end{pmatrix} U^\top(x) \tag{2.5}$$

where $\lambda_i(x), i \in \{1, 2, 3\}$ are the eigenvalues of the resulting structure tensor and U holds its eigenvectors. Initially, $\Sigma(x)$ is set to an arbitrary diagonal matrix. The parameters σ_{min} and σ_{max} are for restricting the anisotropy and the size of the Gaussian. This concept of using a data-adaptive Gaussian for the convolution with the structure tensor has been further investigated in the works of Middendorf and Nagel [22, 23]. See also Chap. 3 by Nagel for the estimation of an adaptive Gaussian.

Another data-adaptive structure tensor has been proposed by Köthe [17] for the purpose of corner detection. For corner detection one uses the fact that the coherence of the orientation measured by the structure tensor becomes small when two edges meet. To achieve an accurate localization of these points, it is favorable to smooth the structure tensor mainly along edges in the image. Köthe has therefore proposed to use an hourglass-shaped filter for the convolution with the structure tensor. The orientation of the filter is thereby adapted to the orientation of the edges, so it is a data-adaptive smoothing.

Note that though the two previous structure tensors are data-adaptive, they are still linear operators, as they imply a convolution operation (which is linear) based on the initial image data. The adaptation quality can be further improved by nonlinear operators, which use the updated data in a kind of feedback loop for the adaptation. Two such nonlinear operators have been proposed for the structure tensor, firstly the concept based on robust statistics by van den Boomgaard and van de Weijer [37], and secondly the techniques based on nonlinear diffusion, proposed by Weickert and Brox [7, 41]. These methods will now be explained in more detail.

2.2.1 Structure Tensors Based on Robust Statistics

Before describing data-adaptive structure tensors based on robust statistics it will be shown that the classic structure tensor is the result of least squares estimation procedures for local orientation. For illustration consider also the texture in Fig. 2.1(a). The histogram of the gradient vectors in this texture patch is shown in Fig. 2.1(b). Let $\mathbf{v}$ be the true orientation vector of the patch, i.e. the vector perpendicular to the stripes. In an ideal image patch every gradient vector should be parallel to the orientation $\mathbf{v}$. In practice they will not be parallel. The error of a gradient vector $\mathbf{g}(\mathbf{y}) := \nabla I(\mathbf{y})$ observed in a point $\mathbf{y}$ with respect to the orientation $\mathbf{v}(\mathbf{x})$ of an image patch centered at location $\mathbf{x}$ is defined as:

$$e(\mathbf{x}, \mathbf{y}) = \|\mathbf{g}(\mathbf{y}) - (\mathbf{g}(\mathbf{y})^\top \mathbf{v}(\mathbf{x}))\mathbf{v}(\mathbf{x})\|$$

The difference $\mathbf{g}(\mathbf{y}) - (\mathbf{g}(\mathbf{y})^\top \mathbf{v}(\mathbf{x}))\mathbf{v}(\mathbf{x})$ is the projection of $\mathbf{g}$ on the normal to $\mathbf{v}$. The error $e(\mathbf{x}, \mathbf{y})$ thus measures the perpendicular distance from the gradient vector $\mathbf{g}(\mathbf{y})$ to the orientation vector $\mathbf{v}(\mathbf{x})$. Integrating the squared error over all positions $\mathbf{y}$ using a soft Gaussian aperture for the neighborhood definition we define the total error:

$$\epsilon(\mathbf{x}) = \int_\Omega e^2(\mathbf{x}, \mathbf{y}) K_\rho(\mathbf{x} - \mathbf{y}) d\mathbf{y} \tag{2.6}$$

The error measure can be rewritten as

$$\epsilon = \int_\Omega \mathbf{g}^\top \mathbf{g} K_\rho d\mathbf{y} - \int_\Omega \mathbf{v}^\top (\mathbf{g}\mathbf{g}^\top) \mathbf{v} K_\rho d\mathbf{y} \; .$$

where we have omitted the arguments of the functions. Minimizing the error thus is equivalent with maximizing

$$\int_\Omega \mathbf{v}^\top (\mathbf{g}\mathbf{g}^\top) \mathbf{v} K_\rho d\mathbf{y} \; ,$$

subject to the constraint that $\mathbf{v}^\top \mathbf{v} = 1$. Note that $\mathbf{v}$ is not dependent on $\mathbf{y}$ so that we have to maximize:

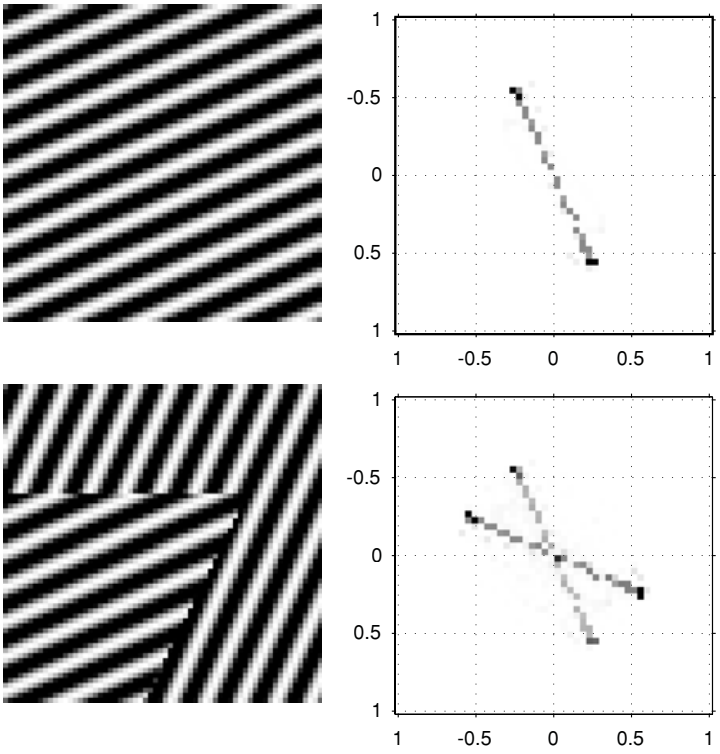

Fig. 2.1. Histograms of gradient vector space. In (**a**) an image (64×64) is shown with in (**b**) the histogram of all gradient vectors (where darker shades indicate that those gradient vectors occur more often in the image. In (**c**) a composition of two differently oriented patterns is shown with the corresponding histogram in (**d**)

$$\mathbf{v}^\top \left(\int_\Omega (\mathbf{g}\mathbf{g}^\top) K_\rho d\mathbf{y} \right) \mathbf{v} = \mathbf{v}^\top J_\rho \mathbf{v}$$

where J_ρ is the structure tensor.

Using the method of Lagrange multipliers to maximize $\mathbf{v}^\top J_\rho \mathbf{v}$ subject to the constraint that $\mathbf{v}^\top \mathbf{v} = 1$, we need to find an extremum of

$$\lambda(1 - \mathbf{v}^\top \mathbf{v}) + \mathbf{v}^\top J_\rho \mathbf{v} \; .$$

Differentiating with respect to $\mathbf{v}$ (remember that $\frac{d}{d\mathbf{v}}(\mathbf{v}^\top A \mathbf{v}) = 2A\mathbf{v}$ in case $A = A^\top$) and setting the derivative equal to zero results in:

$$J_\rho \mathbf{v} = \lambda \mathbf{v} \; . \tag{2.7}$$

The 'best' orientation thus is an eigenvector of the structure tensor J_ρ. Substitution in the quadratic form then shows that we need the eigenvector corresponding to the largest eigenvalue.

The least squares orientation estimation works well in case all gradients in the set of vectors in an image neighborhood all belong to the same oriented pattern. In case the image patch shows two oriented patterns the least squares estimate will mix the two orientations and give a wrong result.

A robust estimator is constructed by introducing the Gaussian error norm:

$$\psi(e) = 1 - \exp\left(\frac{e^2}{2m^2}\right)$$

as depicted in Fig. 2.2. In a robust estimator large deviations from the model (what is considered 'large' is determined by the value of m) are not taken into account very heavily. In our application large deviations from the model are probably due to the mixing of two different linear textures (see Fig. 2.1(c-d)).

The error, (2.6), can now be rewritten as (we will omit the spatial arguments):

$$\epsilon = \int_\Omega \psi\left(\sqrt{\mathbf{g}^\top\mathbf{g} - \mathbf{v}^\top(\mathbf{g}\mathbf{g}^\top)\mathbf{v}}\right) K_\rho d\mathbf{y} \ .$$

Again we use a Lagrange multiplier method to minimize the error subject to the constraint that $\mathbf{v}^\top\mathbf{v} = 1$:

$$\frac{d}{d\mathbf{v}}\left(\lambda(1 - \mathbf{v}^\top\mathbf{v}) + \int_\Omega \psi\left(\sqrt{\mathbf{g}^\top\mathbf{g} - \mathbf{v}^\top(\mathbf{g}\mathbf{g}^\top)\mathbf{v}}\right) K_\rho d\mathbf{y}\right) = 0 \ .$$

This leads to

$$J_\rho^m(\mathbf{v})\mathbf{v} = \lambda\mathbf{v} \tag{2.8}$$

where

$$J_\rho^m(\mathbf{v}) = \int_\Omega \mathbf{g}\mathbf{g}^\top K_m(\mathbf{g}^\top\mathbf{g} - \mathbf{v}^\top(\mathbf{g}\mathbf{g}^\top)\mathbf{v}) K_\rho d\mathbf{y} \tag{2.9}$$

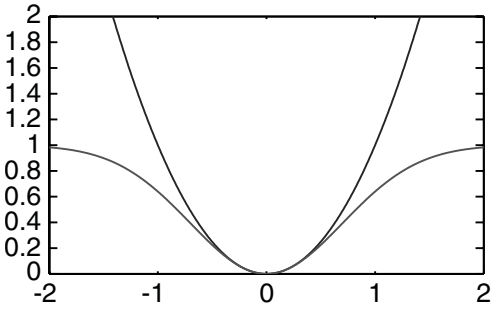

Fig. 2.2. Quadratic versus (robust) Gaussian error norm. The Gaussian error norm is of 'scale' $m = 0.7$

with $K_m(e^2) = \exp\left(e^2/2m^2\right)$. The big difference with the least squares estimator is that now the matrix $J_\rho^m(\mathbf{v})$ is dependent on $\mathbf{v}$ (and on $\mathbf{x}$ as well). Note that $J_\rho^m(\mathbf{v})$ can be called a *'robustified' structure tensor* in which the contribution of each gradient vector is weighted not only by its distance to the center point of the neighborhood, but also weighted according to its 'distance' to the orientation model.

Note that the 'robustification' of the structure tensor is dependent on the model that is fitted to the data, so there is no unique robust structure tensor. The structure tensor is a local averaging of the gradient product $\mathbf{g}\mathbf{g}^\top$, but whereas in the classical case each point in the neighborhood contributes in an equal amount to this average, in the robust formulation the weight is dependent on the plausibility of the gradient observation $\mathbf{g}$ given the model.

A *fixed point* iteration scheme is used to find a solution. Let $\mathbf{v}^i$ be the orientation vector estimate after i iterations. The estimate is then updated as the eigenvector $\mathbf{v}^{i+1}$ of the matrix $J_\rho^m(\mathbf{v}^i)$ corresponding to the largest eigenvalue, i.e. one solves:

$$J_\rho^m(\mathbf{v}^i)\mathbf{v}^{i+1} = \lambda \mathbf{v}^{i+1}$$

The proposed scheme is a generalization of the well-known fixed point scheme (also called *functional iteration*) to find a solution of the equation $v = F(v)$.

Note that the iterative scheme does not necessarily lead to the *global* minimum of the error. In fact one is often not even interested in that global minimum. Consider for instance the situation of a point in region A (with orientation α) that is surrounded by many points in region B (with orientation β). It is not too difficult to imagine a situation where the points of region B outnumber those in region A. Nevertheless the algorithm is to find the orientation α whereas the global minimum would correspond with orientation β. Because the algorithm starts in the initial orientation estimate and then finds the local minimum nearest to the starting point it hopefully ends up in the desired *local* minimum: orientation α. The choice for an initial estimate of the orientation vector is thus crucial in a robust estimator in case the image patch shows two (or more) orientations.

2.2.2 Structure Tensors Based on Nonlinear Diffusion

In the preceding subsection it has been shown that a least squares estimate of the local orientation comes down to solving an eigenvalue problem of the structure tensor smoothed with the Gaussian kernel K_ρ which determines the local neighborhood. We have also seen a more general technique than least squares that introduces an additional weighting dependent on the data. Now the question may arise if there is on the other hand also a more general smoothing approach than Gaussian convolution, and indeed there is one.

The generalization of Gaussian smoothing, which is equivalent to diffusion with a constant diffusivity, is nonlinear diffusion. In contrast to Gaussian convolution, nonlinear diffusion reduces the amount of smoothing in the presence of discontinuities in the data, so it is a data-adaptive smoothing method. Being a nonlinear approach, discontinuities are determined iteratively in the updated, smoothed data and therefore one can integrate data from an arbitrarily shaped neighborhood, as illustrated in Fig. 2.3. Thus nonlinear diffusion seems very appropriate to replace the Gaussian convolution of the classic structure tensor in order to bring in data-adaptive neighborhoods for the integration.

Nonlinear diffusion has been introduced by Perona and Malik [27]. With the initial condition $u(t=0) = I$, the PDE

$$\partial_t u = \operatorname{div}\left(g(|\nabla u|^2)\nabla u\right) \tag{2.10}$$

evolves a scalar-valued data set, such as a gray value image, where I is the initial image. The so-called *diffusivity function* g correlates the amount of smoothing to the gradient magnitude and thereby prevents smoothing across edges. For smoothing the structure tensor, a good choice for this diffusivity function is

$$g(|\nabla u|) = \frac{1}{\sqrt{|\nabla u|^2 + \epsilon^2}} \tag{2.11}$$

where ϵ is a small positive constant only introduced in order to prevent unlimited diffusivities. Diffusion with this diffusivity is called *total variation (TV) flow* [1], which is the diffusion filter corresponding to TV regularization [32].

Since the structure tensor is not a scalar-valued but a matrix-valued data set, one needs an extension of (2.10) to matrix-valued data. Such an extension has been provided in [35]:

$$\partial_t u_{ij} = \operatorname{div}\left(g\left(\sum_{k,l=1}^{N} |\nabla u_{kl}|^2\right)\nabla u_{ij}\right) \qquad i,j = 1,\ldots,N\,. \tag{2.12}$$

Details can also be found in Chap. 25 by Weickert et al. When setting the initial condition to $u_{ij}(t=0) = J_{0,ij}$ (cf. (2.1) and (2.2)), this PDE provides the nonlinear structure tensor J_t for some diffusion time t. Here, N is the

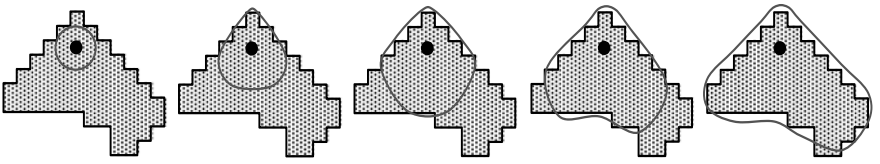

Fig. 2.3. Illustration of how the local neighborhood is adapted by an increasing amount of nonlinear diffusion

number of rows/columns of the structure tensor (which is symmetric), i.e. $N = 2$ for the spatial structure tensor and $N = 3$ for its spatio-temporal version. Note that all matrix channels are coupled in this scheme. They are smoothed with a joint diffusivity taking into account the edges of all channels. Consequently, a discontinuity in one matrix channel inhibits also smoothing in the others.

There exists also an anisotropic counterpart to this scheme, which has been introduced in [7, 41]. In the anisotropic case not only the *amount* of diffusion is adapted locally to the data but also the *direction* of smoothing. This has positive effects for instance in the application of corner detection where one is interested in smoothing mainly along edges in the image.

$$\partial_t u_{ij} = \operatorname{div}\left(D\left(\sum_{k,l=1}^{N} \nabla u_{kl} \nabla u_{kl}^\top\right) \nabla u_{ij}\right) \qquad i,j = 1,\dots,N \tag{2.13}$$

The matrix D is the so-called *diffusion tensor* that replaces the scalar-valued diffusivity g and which we define in the spatial case, where $N = 2$, as

$$D = U \begin{pmatrix} g(\lambda_1) & 0 \\ 0 & 1 \end{pmatrix} U^\top \tag{2.14}$$

The diffusivity function g is the same as in the isotropic setting and λ_1 denotes the larger eigenvalue of the matrix $\sum_{i,j=1}^{N} \nabla u_{ij} \nabla u_{ij}^\top$ while U holds its eigenvectors. Simply speaking, the diffusion tensor reduces the amount of smoothing in gradient direction depending on the gradient magnitude, while it employs the full amount of smoothing in the direction perpendicular to the gradient. For detailed information about anisotropic diffusion in general, we refer to [38]. Anisotropic nonlinear matrix diffusion is also a topic of Chap. 25 by Weickert et al.

By applying a Gaussian convolution with a kernel K_ρ to the matrix $\sum_{k,l=1}^{N} \nabla u_{kl} \nabla u_{kl}^\top$ that determines the diffusion tensor D, one can even emphasize the smoothing along discontinuities in the data [40]. With such a nonlinear diffusion process, one obtains the anisotropic structure tensor $J_{t,\rho}$.

2.2.3 Relations

After the description of these approaches to data-adaptive structure tensors, one might wonder how they are related. Are they basically all the same, or are there significant differences?

Let us first consider the gray value local structure tensor of Nagel and Gehrke and the nonlinear structure tensor based on diffusion. Both methods perform a smoothing operation on the structure tensor, using a neighborhood that is adapted to the data, so one would expect that both methods do approximately the same. However, despite the similarities, there are some significant differences.

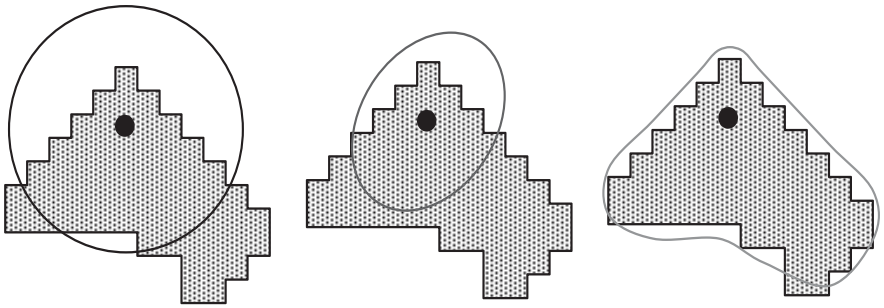

Fig. 2.4. *Left*: Neighborhood of a non-adaptive isotropic Gaussian *Center*: Neighborhood of a data-adaptive anisotropic Gaussian. *Right*: Neighborhood obtained with an iterative diffusion process

Figure 2.4 visualizes these differences between the approach of Nagel and Gehrke and the classic as well as the nonlinear structure tensor. The classic structure tensor uses a fixed isotropic Gaussian kernel for smoothing the data, thus it is not data-adaptive at all. The method proposed by Nagel and Gehrke parameterizes the neighborhood by an anisotropic Gaussian and adapts the parameters locally to the data. Although this approach is more precise than the classic structure tensor, one can see that in many situations the Gaussian cannot fully cover the region of interest without also integrating ambiguous information. The iterative diffusion process involved in the nonlinear structure tensor is more flexible and can therefore cover a neighborhood with arbitrary shape.

Furthermore, the nonlinear structure tensor is based on a nonlinear smoothing operation, i.e. the operation works on the *updated* data, while the method of Nagel and Gehrke is still a linear operation as it smooths the *initial* data.

The robust structure tensor described in Sect. 2.2.1 is also based on a nonlinear process, so let us consider its relations to the nonlinear structure tensor. From (2.9) one can see that dependent on how well the values fit to the currently estimated orientation, their influence is decreased. This is similar to the concept of the nonlinear structure tensor, where the further expansion of the local neighborhood is reduced if the new values do not fit well to the values of the current neighborhood. Note that the weighting function $\psi'(s^2)$ in (2.9) is one of the diffusivity functions used by Perona and Malik when they introduced nonlinear diffusion (cf. [27]). Thus both the nonlinear structure tensor and the robust structure tensor make the integration of further data dependent on whether it fits to the already gathered data. One can even choose the same weighting function for this selection process.

The difference between both approaches is that the nonlinear structure tensor applies this selection process in order to determine the local neighborhood and then uses a simple least squares approach within this neighborhood,

while the structure tensor based on robust statistics first gathers the data from the simple fixed Gaussian neighborhood K_ρ and applies the nonlinear weighting process afterwards. Thus the nonlinear structure tensor assumes that the values needed for a good estimation are connected, whereas the robust statistics ignore the aspect of connectivity. Consequently, it can be expected that in situations where the assumption of connected data holds, the nonlinear structure tensor is better suited, while in situations where the assumption is false, robust statistics should be advantageous.

Relations between robust statistics, nonlinear diffusion, and other data-adaptive smoothing approaches are also dealt with in [24].

2.3 Optic Flow Estimation

A well-known application of the structure tensor is optic flow estimation. In optic flow estimation one searches for the displacement field $(u(x, y), v(x, y))$ that says for each pixel (x, y) of one image $I(x, y, t)$ to which position it has moved in a second image $I(x, y, t + 1)$.

In Bigün et al. [4] optic flow estimation has been regarded as the search for the spatio-temporal orientation where there is the least change in the image sequence. This immediately leads to an orientation estimation problem that can be solved by computing the eigenvector $\mathbf{w} = (w_1, w_2, w_3)$ to the smallest eigenvalue of the structure tensor. The optic flow vector can then be computed by normalizing the last component of $\mathbf{w}$ to 1, which leads to $u = w_1/w_2$ and $v = w_2/w_3$.

Although this has been the first explicit usage of the structure tensor for optic flow estimation, the structure tensor is also implicitly present in the early method of Lucas and Kanade [21]. In this method the assumptions of the optic flow estimation problem become more explicit. Furthermore, the method of Lucas-Kanade is an ordinary least squares approach, while the method of Bigün estimates the flow vector by means of total least squares. For optic flow estimation in practice, it turns out that a simple least squares approach is more robust, so we will stick here to the method of Lucas-Kanade.

2.3.1 Lucas-Kanade with the Conventional Structure Tensor

The assumption that is most frequently used in optic flow estimation is the assumption that the displacement of pixels does not alter their gray values. This can be expressed by the well-known optic flow constraint (OFC) [14]

$$I_x u + I_y v + I_z = 0 \,. \tag{2.15}$$

The optic flow is not uniquely determined by this constraint, since this is only one equation for two flow components. This is also called the *aperture problem*. In order to obtain a unique solution, Lucas and Kanade proposed to

assume the optic flow vector to be constant within some neighborhood, e.g. a Gaussian window K_ρ.

With this second assumption, it is possible to estimate the optic flow at each point by the minimizer of the local energy function

$$E(u,v) = \frac{1}{2} K_\rho * \left((I_x u + I_y v + I_z)^2 \right) \ . \tag{2.16}$$

A minimum (u,v) of E must satisfy $\partial_u E = 0$ and $\partial_v E = 0$, what leads to the 2×2 linear system

$$\begin{pmatrix} K_\rho * I_x^2 & K_\rho * I_x I_y \\ K_\rho * I_x I_y & K_\rho * I_y^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -K_\rho * I_x I_z \\ -K_\rho * I_y I_z \end{pmatrix} \ . \tag{2.17}$$

Note that it is possible to use instead of a purely spatial neighborhood also a spatio-temporal neighborhood where the assumption of constant flow is extended to hold also over time. Since the spatio-temporal version has access to more data, it leads in general to more accurate results. However, for simplicity we considered only spatial neighborhoods in the experiments.

2.3.2 Lucas-Kanade with the Nonlinear Structure Tensor

One can easily observe that the entries of this linear system are five of the six different components of the spatio-temporal structure tensor

$$J_\rho = K_\rho * \left(\nabla I \nabla I^\top \right) = K_\rho * \begin{pmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{pmatrix} \ . \tag{2.18}$$

Thus it is possible to replace these entries by the components of one of the data-adaptive structure tensors. Such a replacement means that the fixed neighborhood of the original method is replaced by an adaptive neighborhood which prefers those pixels that fit the assumption of constant optic flow.

As already discussed in Sect. 2.2, one can obtain a good adaptation of the neighborhood by nonlinear diffusion. Thus with the nonlinear structure tensor [7, 41] described in Sect. 2.2.2 and determined by the nonlinear diffusion process given by (2.12), the assumption of constant flow holds much more often than in the case of the conventional structure tensor. Consequently, there are less estimation errors, in particular near motion boundaries.

2.3.3 Robust Structure Tensor for Optic Flow Estimation

Compared to Sect. 2.2.1, with optic flow estimation the orientation estimation task has changed a bit. We now search for the orientation with *least* change in a *spatio-temporal* space. Since the robustified structure tensor selects the data according to how well it fits to the model, a new robust structure tensor has

to be derived due to the change of the model. In order to see the relations to the derivation in Sect. 2.2.1 we adapt to the same notation and write $\mathbf{g} = \nabla I$. The optic flow vector (u, v) will be written as the estimated orientation $\mathbf{v}$. Further on, the Lucas-Kanade approach will be interpreted as a least squares estimation procedure first, before the generalized robust estimation procedure is described.

Least squares estimation. As stated above, the optic flow constraint (2.15) has two unknowns: the two components of the optic flow vector $\mathbf{v}$, and a way to get an expression for a unique solution for $\mathbf{v}$ is to come up with more equations each describing the same vector $\mathbf{v}$. This is achieved with the assumption of Lucas-Kanade that within a local neighborhood of a point $\mathbf{x}$ the optical flow vector is constant. Like in Sect. 2.2.1 a Gaussian aperture is selected to define the local neighborhood. Let $\mathbf{v}(\mathbf{x})$ be the optical flow vector at $\mathbf{x}$ then the error towards the optic flow constraint is given as:

$$\epsilon(\mathbf{x}) = \int_{\Omega} \left(I_z(\mathbf{y}) + \mathbf{v}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{y})\right)^2 K_\rho(\mathbf{x} - \mathbf{y}) d\mathbf{y} \tag{2.19}$$

If we now select the vector $\mathbf{v}^*$ that minimizes the above expression then the OFC expression $I_z + \mathbf{v} \cdot \mathbf{g}$ is minimized on average in the local neighborhood of a point $\mathbf{x}$:

$$\mathbf{v}^* = \text{argmin}_{\mathbf{v}} \epsilon(\mathbf{x})$$

The optimal value is found by solving for $\mathrm{d}_v \epsilon = 0$:

$$\mathrm{d}_v \epsilon = 2 \int_{\Omega} \big(I_z(\mathbf{y}) + \mathbf{v}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{y})\big) \mathbf{g}(\mathbf{y}) K_\rho(\mathbf{x} - \mathbf{y}) d\mathbf{y}$$

Here we use the convention used throughout this chapter that the integration of a matrix/vector equation is to be done for each of the matrix/vector components individually. Consider the term $(\mathbf{v} \cdot \mathbf{g})\mathbf{g}$, where we have omitted the spatial arguments for clarity. This can be rewritten as $(\mathbf{g}\mathbf{g}^\top)\mathbf{v}$. Note that $\mathbf{g}\mathbf{g}^\top$ is a 2×2 matrix which, when integrated over a spatial neighborhood, is the structure tensor $J(\mathbf{x})$. Using this we can rewrite the above equation as:

$$\left(\int_{\Omega} \mathbf{g}(\mathbf{y}) \mathbf{g}^\top(\mathbf{y}) \; K_\rho(\mathbf{x} - \mathbf{y}) d\mathbf{y}\right) \mathbf{v}(\mathbf{x}) = -\int_{\Omega} I_z(\mathbf{y}) \, \mathbf{g}(\mathbf{y}) \, K_\rho(\mathbf{x} - \mathbf{y}) d\mathbf{y} \tag{2.20}$$

or

$$J(\mathbf{x}) \mathbf{v}(\mathbf{x}) = -\int_{\Omega} I_z(\mathbf{y}) \, \mathbf{g}(\mathbf{y}) \, K_\rho(\mathbf{x} - \mathbf{y}) d\mathbf{y}$$

After integration the structure tensor can be assumed to be non-singular and thus:

$$\mathbf{v}(\mathbf{x}) = -J^{-1}(\mathbf{x}) \int_{\Omega} I_z(\mathbf{y}) \, \mathbf{g}(\mathbf{y}) \, K_\rho(\mathbf{x} - \mathbf{y}) d\mathbf{y}$$

This is the well-known linear least squares estimator of the optical flow vector. Like many local structure calculations it suffers from the fact that *all* points

in the neighborhood are used in the calculation. At motion boundaries the above expression is known to give the wrong answers.

Robust estimation. Robustifying optical flow calculations can be found e.g. in [5]. Here we emphasize that a robust estimator of the optical flow vector nicely fits into the framework for robust local structure calculations as set up in this chapter.

The squared error of (2.19) is replaced with a robust error measure:

$$\epsilon(\mathbf{x}) = \int_{\Omega} \psi\left(I_z(\mathbf{y}) + \mathbf{v}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{y})\right) K_\rho(\mathbf{x}-\mathbf{y}) d\mathbf{y} \tag{2.21}$$

leading to the following expression for the derivative $\mathrm{d}_{\mathbf{v}}\epsilon$:

$$\mathrm{d}_{\mathbf{v}}\epsilon = \int_{\Omega} \psi'\big(I_z(\mathbf{y}) + \mathbf{v}(\mathbf{x})\,\mathbf{g}(\mathbf{y})\big)\,\mathbf{g}(\mathbf{y})\,K_\rho(\mathbf{x}-\mathbf{y}) d\mathbf{y}$$

Like in Sect. 2.2.1 we select the Gaussian error norm for ψ, leading to:

$$\mathrm{d}_{\mathbf{v}}\epsilon = \int_{\Omega} \frac{I_z(\mathbf{y}) + \mathbf{v}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{y})}{m^2} \exp\left(-\frac{\left(I_z(\mathbf{y}) + \mathbf{v}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{y})\right)^2}{2m^2}\right) \mathbf{g}(\mathbf{y}) K_\rho(\mathbf{x}-\mathbf{y}) d\mathbf{y}$$

This can be rewritten as:

$$\mathrm{d}_{\mathbf{v}}\epsilon = \int_{\Omega} \big(I_z(\mathbf{y}) + \mathbf{v}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{y})\big)\mathbf{g}(\mathbf{y}) K_m(I_z(\mathbf{y}) + \mathbf{v}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{y})) K_\rho(\mathbf{x}-\mathbf{y}) d\mathbf{y}$$

Solving for $\mathrm{d}_{\mathbf{v}}\epsilon = 0$ we obtain:

$$\begin{aligned}\left(\int_{\Omega} \mathbf{g}(\mathbf{y})\mathbf{g}(\mathbf{y})^\top K_m(\dots) K_\rho(\mathbf{x}-\mathbf{y}) d\mathbf{y}\right)\mathbf{v} \\ = -\int_{\Omega} I_z(\mathbf{y})\,\mathbf{g}(\mathbf{y}) K_m(\dots) K_\rho(\mathbf{x}-\mathbf{y}) d\mathbf{y}\end{aligned}$$

Compared with the linear least squares estimator, a new term $K_m(\dots)$ has been added that can be interpreted as the *model error penalty*. This equation is the 'robustified' equivalent of 2.20.

Again we obtain a 'robustified' structure tensor. Carefully note that the model error penalty term is different from the one we have derived in a previous section where we looked for the local orientation of *maximum* change in a purely *spatial* neighborhood. Here we arrive at the equation

$$J_\rho^m(\mathbf{v})\mathbf{v} = \mathbf{l}(\mathbf{v})$$

where

$$J_\rho^m(\mathbf{v}) = \int_{\Omega} \mathbf{g}(\mathbf{y})\mathbf{g}(\mathbf{y})^\top K_m(I_z(\mathbf{y}) + \mathbf{v}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{y})) K_\rho(\mathbf{x}-\mathbf{y}) d\mathbf{y}$$

and

$$\mathbf{l}(\mathbf{v}) = -\int_{\Omega} I_z(\mathbf{y})\,\mathbf{g}(\mathbf{y}) K_m(\dots) K_\rho(\mathbf{x}-\mathbf{y}) d\mathbf{y}\ .$$

And again we can solve this through a fixed point procedure:

$$\mathbf{v}^{i+1} = -\left(J_\rho^m\right)^{-1}(\mathbf{v}^i)\mathbf{l}(v^i)$$

with $\mathbf{v}^0$ some initial estimate of the optical flow vector (the linear least squares estimate is an obvious choice for this).

2.3.4 Adapting the Neighborhood with a Coherence Measure

As stated above, the assumption of constant flow field over a neighborhood is used in order to disambiguate the optic flow constraint equation. This leads to the idea that diffusion should be reduced at those areas where the aperture problem is already reasonably solved [19]. In regions with non-constant smooth motion fields, this will avoid oversmoothing the tensor field and then preserve small motion differences. The aperture problem is solved as soon as the two larger eigenvalues of the structure tensor are large enough compared to the smallest one, i.e. that the ellipsoid associated to the tensor is flat. In order to quantify the flatness of a tensor, we use a slightly changed version of the *coherence* or *corner* measure proposed in [13]:

$$c_m(J) = \left(\frac{\lambda_1 - \lambda_3}{\lambda_1 + \lambda_3 + \epsilon}\right)^2 - \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \epsilon}\right)^2 \tag{2.22}$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ are the eigenvalues of the structure tensor J and ϵ is a small positive constant for regularization purposes. If $\lambda_2 \approx \lambda_3$, this measure yields a value close to 0, while if $\lambda_1 \approx \lambda_2 > \lambda_3$, the value is close to 1. This measure can be use to steer the diffusion of the structure tensor through a matrix-valued nonlinear diffusion scheme, written in a continuous formulation as

$$\partial_t J_{ij} = \operatorname{div}\ (g(c_m)\nabla J_{ij}) \tag{2.23}$$

where g is decreasing, $g(0) = 1$, $g(1) = 0$. Note that the continuous formulation is problematic if g is not smooth. However, an associated discrete scheme will be generally well defined. It can be written as

$$J_s^{n+1} = J_s^n + \tau \sum_{r \in N(s)} \beta_r g\left(c_m(J_r^n + J_s^n)\right)(J_r^n - J_s^n) \tag{2.24}$$

where $s = (i, j)$ denotes a image location (and not the tensor component), $N(s)$ a discrete neighborhood, τ an evolution step and the β_r are positive values that depend on the neighborhood (but not on the tensors), with reflecting boundary conditions. The diffusivity function used here

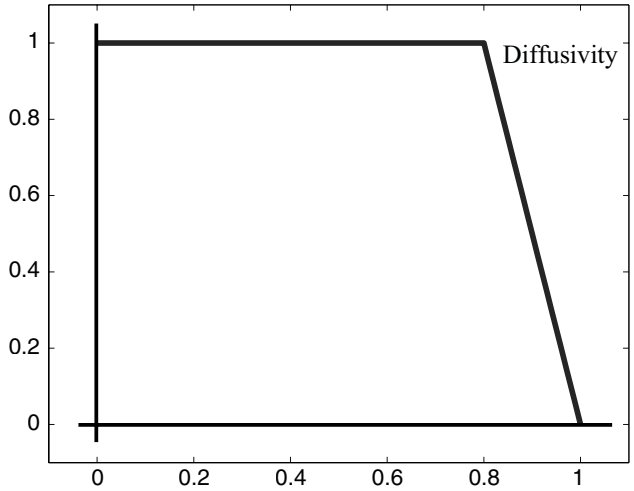

Fig. 2.5. The diffusivity function $g(c_m)$

$$g(c_m) = \begin{cases} 1 & \text{if } c_m < \alpha - \eta \\ (\alpha + \eta - c_m)/2\eta & \text{if } \alpha - \eta \leq c_m < \alpha - \eta \\ 0 & \text{if } c_m \geq \alpha - \eta \end{cases} \tag{2.25}$$

is depicted in Fig. 2.5. The thresholds α and η have been set respectively to 0.9 and 0.1 in the experiments. This diffusion is an alteration of the linear diffusion and possesses the same stability properties. It behaves well in presence of small structures with high curvatures, but has the same drawback that the linear diffusion with respect to motion discontinuities. Indeed, as it can be seen from the discrete formulation (2.24), if J_s and J_r are neighboring tensors with different orientations, their sum will become isotropic and their coherence measure small, so a maximal diffusivity of 1 will be assigned in the corresponding term of (2.24).

2.3.5 Comparison

Figures 2.6–2.8 shows three well-known test sequences for optic flow estimation and the results obtained with the methods described above[1]. The visualization of both the orientation and the magnitude of the flow vector is achieved by using color plots where the hue is determined by the orientation and the intensity corresponds to the magnitude of the flow vector.

[1] The Yosemite sequence with clouds was created by Lynn Quam and is available at `ftp://ftp.csd.uwo.ca/pub/vision`. The version without clouds is available at `http://www.cs.brown.edu/people/black/images.html`. The original, uncropped, street sequence has been published in [11] and is available at `http://www.cs.otago.ac.nz/research/vision`.

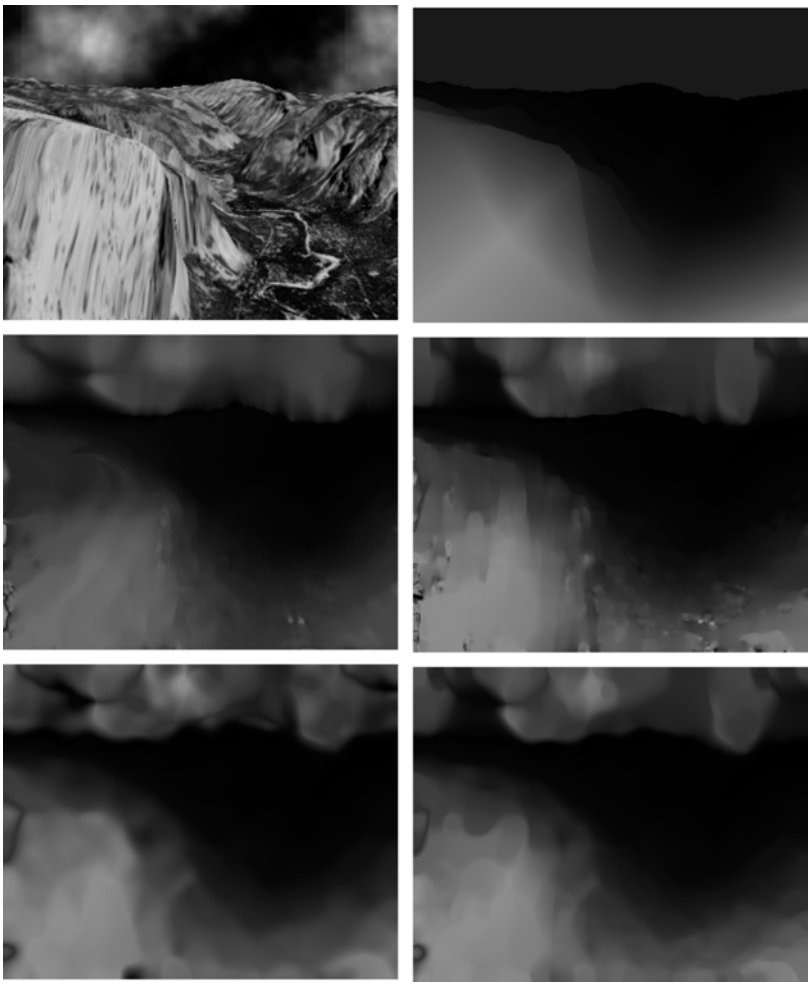

Fig. 2.6. Yosemite sequence ($316 \times 252 \times 15$). From *Left* to *Right*, *Top* to *Bottom*: (**a**) Frame 8. (**b**) Ground truth. (**c**) Classic structure tensor. (**d**) Nonlinear structure tensor. (**e**) Robust structure tensor. (**f**) Coherence based smoothing. See colour plates

In all sequences, one can see a clear qualitative difference between the Lucas-Kanade method based on the classical structure tensor and the methods based on its data-adaptive versions. While the classic structure tensor causes blurring artifacts at motion discontinuities, leading to bad estimates in these areas, the data-adaptive structure tensors avoid mixing the data from the different regions and therefore yield much more accurate results.

For the test sequences used here, there is also the ground truth available, so it becomes possible to compare the methods by a quantitative measure. The standard measure used in the literature is the average angular error (AAE)

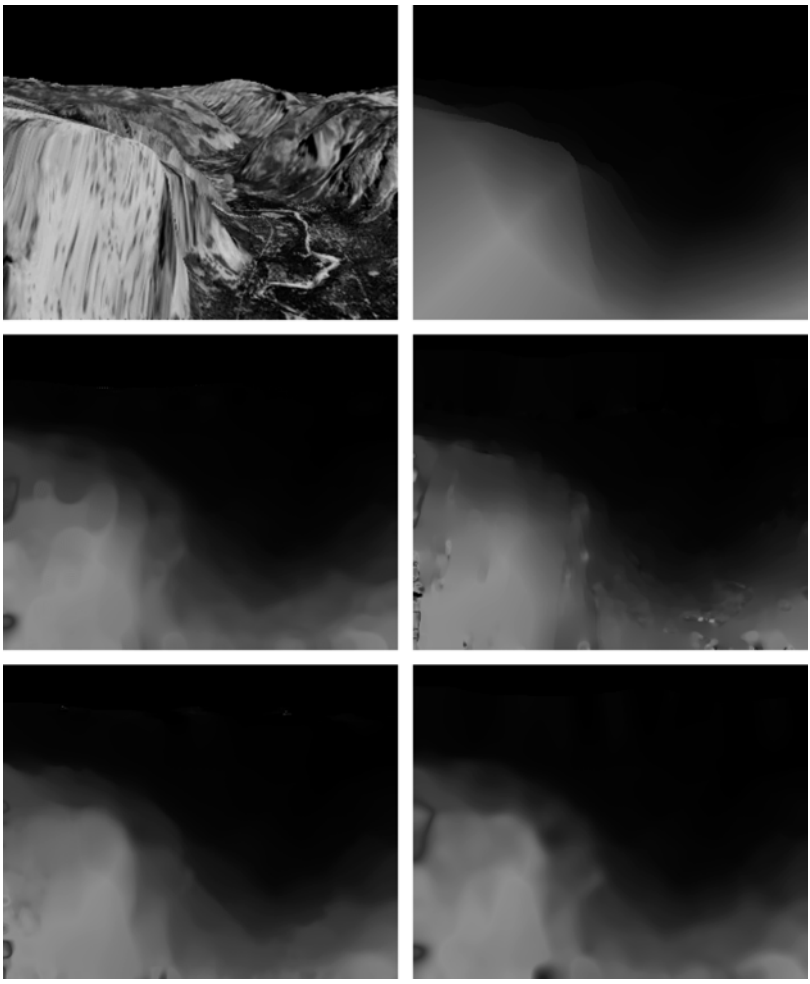

Fig. 2.7. Yosemite sequence without clouds (316×252×15). From *Left* to *Right*, *Top* to *Bottom*: (**a**) Frame 8. (**b**) Ground truth. (**c**) Classic structure tensor. (**d**) Nonlinear structure tensor. (**e**) Robust structure tensor. (**f**) Coherence based smoothing. See colour plates

introduced in [3]. Given the estimated flow field (u_e, v_e) and ground truth (u_c, v_c), the AAE is defined as

$$aae = \frac{1}{N} \sum_{i=1}^{N} \arccos \left(\frac{u_{ci} u_{ei} + v_{ci} v_{ei} + 1}{\sqrt{(u_{ci}^2 + v_{ci}^2 + 1)(u_{ei}^2 + v_{ei}^2 + 1)}} \right) \tag{2.26}$$

where N is the total number of pixels. Against its indication, this quality measure not only measures the angular error between the estimated flow vector

Fig. 2.8. Street sequence (cropped) ($145 \times 100 \times 20$). From *Left* to *Right*, *Top* to *Bottom*: (**a**) Frame 10. (**b**) Ground truth. (**c**) Classic structure tensor. (**d**) Nonlinear structure tensor. (**e**) Robust structure tensor. (**f**) Coherence based smoothing. See colour plates

and the correct vector, but also differences in the magnitude of both vectors, since it measures the angular error of the *spatio-temporal* vector $(u, v, 1)$.

Table 2.1 compares the errors of the different methods. It can be observed that all data-adaptive approaches show a higher performance than the conventional method in all sequences. Between the data-adaptive methods there are some differences, however, there is no clear winner.

Table 2.1. Comparison between results. In all cases the flow fields are dense. AAE = average angular error

Yosemite sequence without clouds.	
Technique	AAE
Classic structure tensor	3.80°
Nonlinear structure tensor	3.74°
Robust structure tensor	3.21°
Coherence based structure tensor	3.43°
Yosemite sequence with clouds.	
Technique	AAE
Classic structure tensor	8.78°
Nonlinear structure tensor	7.67°
Robust structure tensor	8.01°
Coherence based structure tensor	8.21°
Street sequence.	
Technique	AAE
Classic structure tensor	10.54°
Nonlinear structure tensor	7.75°
Robust structure tensor	7.08°
Coherence based structure tensor	9.79°

2.4 Texture Analysis

2.4.1 Robust Orientation Estimation

An important property of texture is its dominant orientation. In Sect. 2.2.1 it was shown that the dominant orientation of an line pattern can be estimated using a linear least squares estimator. The resulting orientation turns out to be the eigenvector of the structure tensor belonging to the largest eigenvalue.

In Fig. 2.9(a) an oriented pattern is shown and in (b) the scatter diagram of the gradient vectors observed at small scale in a neighborhood in the image at the border of the two differently oriented regions. It is evident that a least squares estimator cannot distinguish between the two oriented patterns and will 'smooth' the orientation.

A robust estimation of orientation greatly improves this. We start again with the estimator for the orientation that is based on the error measure as given in (2.6):

$$\epsilon = \int_{\Omega} K_\rho \, \psi(\sqrt{\mathbf{g} - (\mathbf{g}^\top \mathbf{v})\mathbf{v}}) d\mathbf{x}$$

where we have replaced the quadratic error norm with a robust error norm ψ. The local orientation is then found by minimizing the above error measure for $\mathbf{v}$ under the constraint that $\mathbf{v}^\top \mathbf{v} = 1$. Using a Lagrange multiplier we have to minimize $\epsilon + \lambda(1 - \mathbf{v}^\top \mathbf{v})$. Setting $\partial\epsilon/\partial\mathbf{v} = 0$ and solving for $\mathbf{v}$ we arrive at:

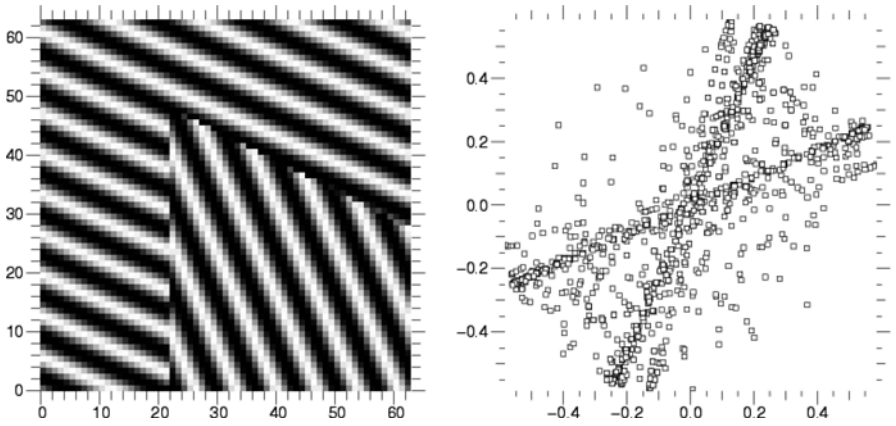

Fig. 2.9. Gradient histogram with two differently oriented textures

$$J_\rho^m(\mathbf{v})\mathbf{v} = \lambda\mathbf{v}$$

where

$$J_\rho^m(\mathbf{v}) = \int_\Omega \mathbf{g}\mathbf{g}^\top K_m(\mathbf{g} - (\mathbf{g}^\top\mathbf{v})\mathbf{v})K_\rho d\mathbf{x}$$

is the 'robustified' structure tensor. Note that the structure tensor $J_\rho^m(\mathbf{v})$ depends on the orientation $\mathbf{v}$ and thus we have to solve for the optimal orientation in an iterative fixed point manner. Starting with an initial estimate $\mathbf{v}^0$, calculate the structure tensor $J_\rho^m(\mathbf{v}^0)$ and calculate a new orientation estimate as the eigenvector of largest eigenvalue. This iterative procedure in practice needs very few iterations to converge (typically 3 to 5 iterations).

In Fig. 2.10 the robust orientation estimation is compared with the linear least squares estimation. It can be clearly observed that whereas the linear estimator 'gently' changes from the one orientation to the second, the robust estimator shows a sharp transition. A pattern with only slight variation in orientation is shown in Fig. 2.11. Again the robust estimator is capable of clearly detecting the edges between areas of different orientation.

2.4.2 Texture Segmentation

The three different components of the structure tensor can also directly be integrated as features into a segmentation method, like the one proposed in [6, 31]. This segmentation framework computes a two region segmentation given a suitable feature vector. In our case this is the vector composed of the three different components of the structure tensor and the image gray value. The components of the structure tensor are normalized to the same range as the image gray value in order to ensure a fair weighting between the

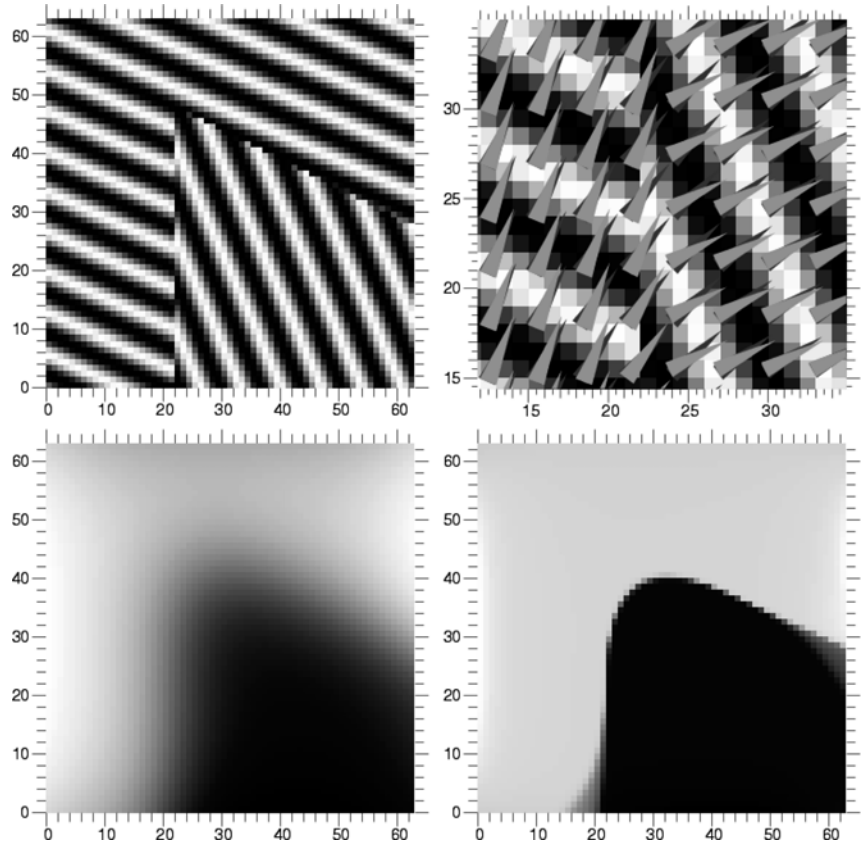

Fig. 2.10. Comparison between least squares and robust orientation estimation. See also colour plates

channels. Figure 2.12 reveals that with a data-adaptive approach, the segmentation can benefit from the reduced blurring effects in the feature channels and yields a higher accuracy at region boundaries. Note that although the components of the nonlinear structure tensor look almost unsmoothed, there is some smoothing that provides the dominant orientation also in the gaps between the stripes. For comparison, the segmentation result obtained with the unsmoothed structure tensor J_0 is depicted in Fig. 2.13.

2.5 Corner Detection

When looking for some important, distinguished locations of an image, one often considers points where two or more edges meet. Such locations have been named *corners* or *interest points*, and a range of possible approaches

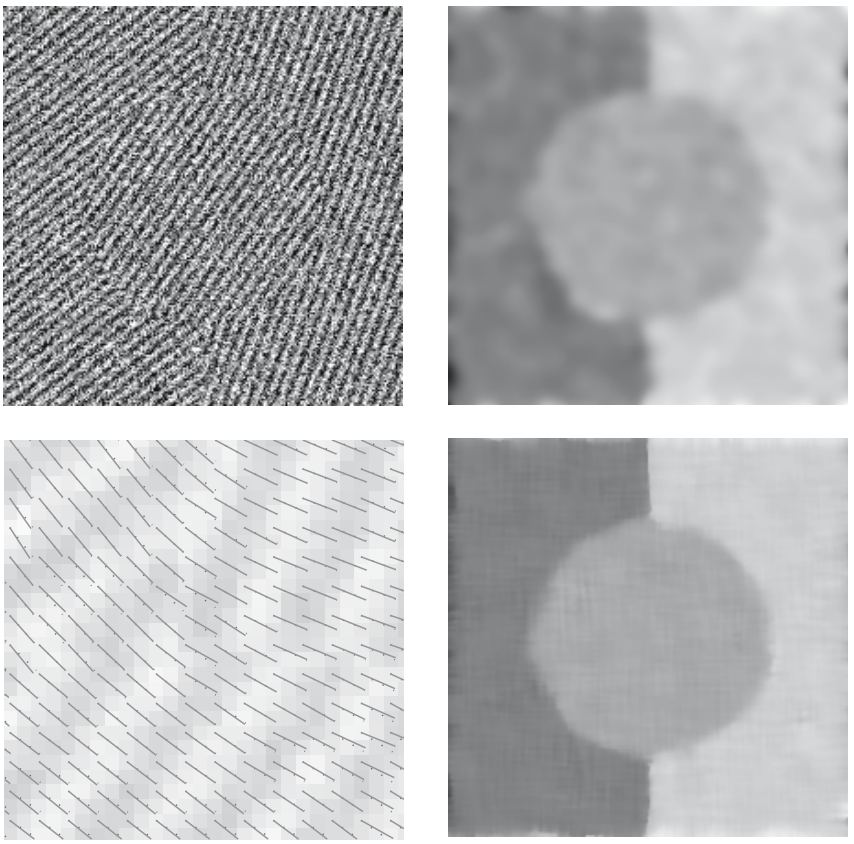

Fig. 2.11. Comparison between least squares and robust orientation estimation. See also colour plates

exists to detect them in an image, see e.g. the reviews in [30, 33]. Methods based on the structure tensor are well established in this field.

For detecting corners, the coherence information present in the structure tensor after integration is exploited. At zero integration scale, the structure tensor J_0 as introduced in (2.1) or (2.2) contains information on intrinsically 1-dimensional features of the image, i.e. edges. For gray-scale images, only one eigenvalue of the structure tensor J_0 may attain nonzero values (equal to the squared gradient magnitude), while its corresponding eigenvector represents the gradient direction.

Two-dimensional features of an image (corners) can be detected after integrating the local 1-D information of J_0 within some neighborhood, since the consideration of a local neighborhood makes additional information: that of the homogeneity, or coherence, of the surrounding orientation. If two differently oriented edges appear in the neighborhood, the smoothed structure

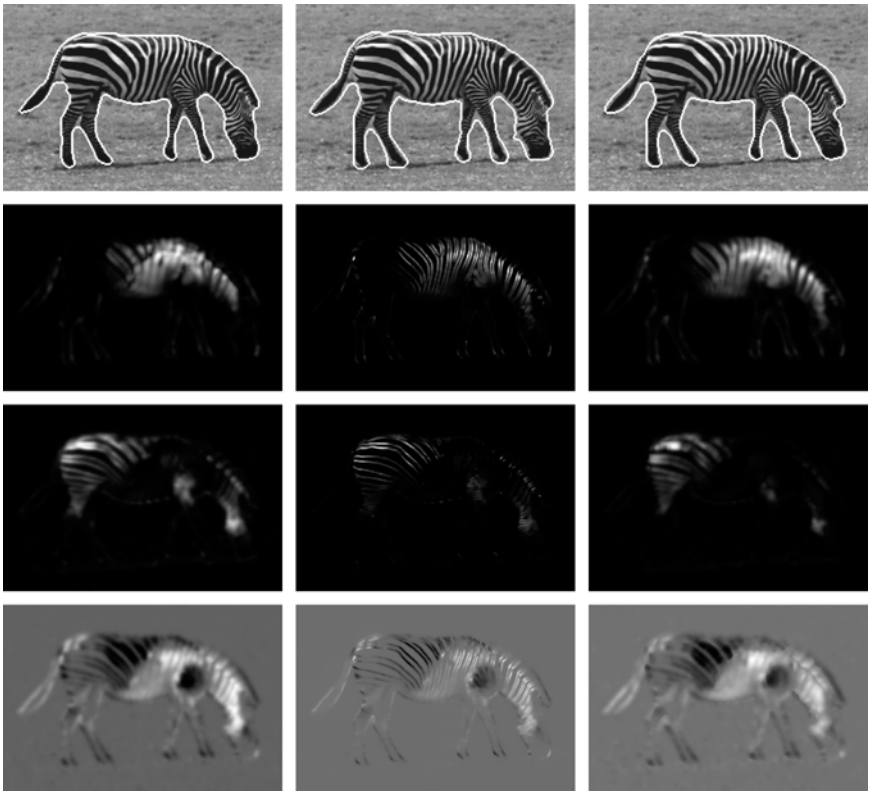

Fig. 2.12. *Left Column*: Segmentation with the classic structure tensor ($\rho = 2$). *Center Column:* Segmentation with the nonlinear structure tensor ($t = 25$). *Right Column:* Segmentation with the robust structure tensor ($\rho = 3$, $m = 0.05$). *From Top to Bottom.* (**a**) Segmented image (250×167). (**b**) Tensor component J_{11} based on I_x^2. (**c**) J_{22} based on I_y^2. (**d**) J_{12} based on $I_x I_y$

Fig. 2.13. Segmentation with the unsmoothed structure tensor J_0

tensor J will possess two nonzero eigenvalues λ_1, λ_2. An analysis of the eigenvalues can serve as a measure for the coherence of the surrounding structure. Three cases can be distinguished when regarding the eigenvalues $\lambda_1 \geq \lambda_2$ of the matrix:

- $\lambda_1 \approx \lambda_2 \approx 0$: homogeneous areas, almost no structure present

- $\lambda_1 > 0, \lambda_2 \approx 0$: edges, one dominant orientation
- $\lambda_1 > 0, \lambda_2 > 0$: corners, structure with ambiguous orientation

Several possibilities have been proposed to convert this information into a coherence measure or a measure of 'cornerness', e.g. by Förstner [9], Harris and Stephens [12], Rohr [29], or Köthe [17]. In our experiments on corner detection we employ the last approach, and detect corners at local maxima of the smaller eigenvalue of the smoothed structure tensor.

One should note that for this application of the structure tensor, it is necessary to allow the integration of ambiguous orientation, because one searches for exactly the points where these ambiguities attain a maximum. This is completely contrary to orientation estimation where ambiguities are to be avoided. It therefore seems contradictive on the first glance that a data-adaptive structure tensor could perform better than the classic one on this task. Indeed, the structure tensor based on robust statistics is not applicable here, since it uses the same neighborhood as the classic structure tensor but selects the weighting of the pixels in order to *minimize* the ambiguities.

With the nonlinear structure tensor, however, the situation is a bit different. The nonlinear diffusion process does not select the pixels in order to minimize the ambiguities, but it selects the neighborhood. Thus ambiguities in the orientation, though they are reduced, can still appear. Since the neighborhood is better adapted to the structures in the image, this even leads to advantages in comparison to the classic structure tensor, see Fig. 2.15 and Fig. 2.16. Corners remain well localized even for higher diffusion times when any possible noise or small-scale features would have been removed.

The better concept of data-adaptive smoothing in the case of corner detection, however, is the nonlinear diffusion process stated in (2.13). The anisotropic diffusion process propagates information along the edges. This leads to a very precise maximum in the second eigenvalue of the structure tensor at the position where two edges meet, see Fig. 2.15. A small diffusion time already suffices to produce significant corner features which are well localized. In Fig. 2.16 it can be observed that this kind of smoothing leads to the best performance.

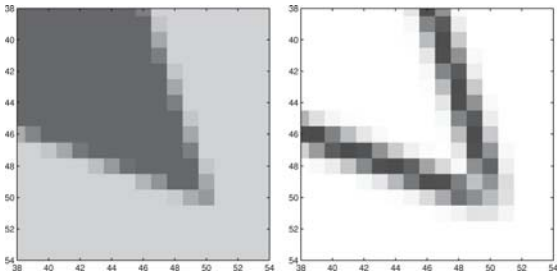

Fig. 2.14. *Left*: Detail of a test image with ideal corner position $(50, 50)$. *Right*: Larger eigenvalue of the unsmoothed structure tensor J_0

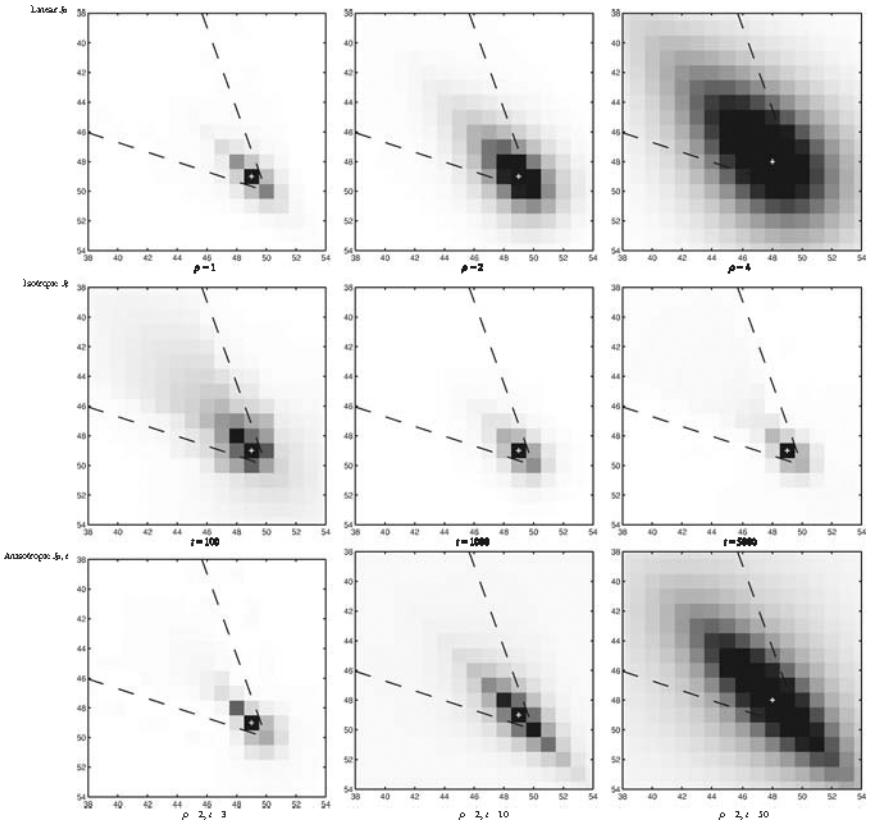

Fig. 2.15. Cornerness measured by the smaller eigenvalue of a smoothed structure tensor J, and the detected corner. *Top*: Linear smoothing. *Center*: Isotropic nonlinear diffusion with TV diffusivity. *Bottom*: Anisotropic nonlinear diffusion

It is also very closely related to the data-adaptive structure tensor proposed by Köthe [17]. In order to detect corners, Köthe also smoothes along edges, in his case using a linear, hourglass-shaped filter. This filter as well propagates information along edges and leads to a maximum in the second eigenvalue of the structure tensor at the position where edges meet.

2.6 Summary

In this chapter, we have juxtaposed several concepts for data-adaptive structure tensors. It has emerged that though the different techniques have the same basic motivation, there are quite important differences in detail. All data-adaptive structure tensors discussed here are to deal with the inaccuracies and blurring artifacts caused by the Gaussian neighborhood of the

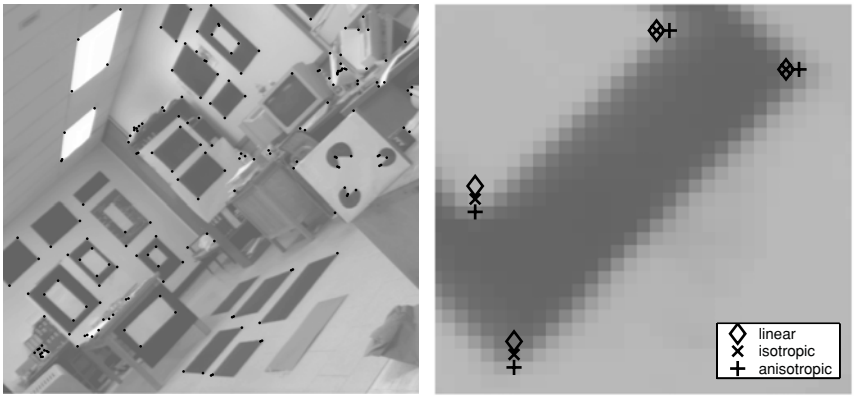

Fig. 2.16. *Left*: Corners detected in the 'lab' test image using the nonlinear structure tensor with anisotropic diffusion. *Right*: Comparison of the corners detected by the classic linear structure tensor and the nonlinear structure tensor with an underlying isotropic and anisotropic diffusion process, respectively

conventional structure tensor. However, the strategies how to choose an adaptive neighborhood are different. In some typical applications of the structure tensor, the data-adaptive structure tensors have shown their beneficial properties in comparison to the classic structure tensor. The differences between the data-adaptive structure tensors have been sometimes marginal, sometimes larger, depending on the application. This yields two messages: firstly, compared to the conventional structure tensor, the data-adaptive methods are in many cases worth the additional effort. Secondly, it is wise to choose a data-adaptive technique depending on the application. There is no clear winner that always performs better than the other techniques.

Acknowledgements

Our research was partly funded by the DFG project WE 2602/1-1 and the project *Relations between Nonlinear Filters in Digital Image Processing* within the DFG Priority Program 1114. This is gratefully acknowledged.

References

1. F. Andreu, C. Ballester, V. Caselles, and J. M. Mazón. Minimizing total variation flow. *Differential and Integral Equations*, 14(3):321–360, March 2001.
2. P. Bakker, L.J. van Vliet, and P.W. Verbeek. Edge preserving orientation adaptive filtering. In M. Boasson, J.A. Kaandorp, J.F.M. Tonino, and M.G. Vosselman, editors, *ASCI'99, Proc. 5th Annual Conference of the Advanced School for Computing and Imaging*, pp. 207–213, 1999.

3. J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, February 1994.
4. 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*, 13(8):775–790, August 1991.
5. M. J. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.
6. T. Brox, M. Rousson, R. Deriche, and J. Weickert. Unsupervised segmentation incorporating colour, texture, and motion. Technical Report 4760, INRIA Sophia-Antipolis, France, March 2003.
7. T. Brox, J. Weickert, B. Burgeth, and P. Mrázek. Nonlinear structure tensors. Technical report, Dept. of Mathematics, Saarland University, Saarbrücken, Germany, July 2004.
8. S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics and Image Processing*, 33:116–125, 1986.
9. W. Förstner. A feature based corresponding algorithm for image matching. *International Archive of Photogrammetry and Remote Sensing*, 26:150–166, 1986.
10. W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pp. 281–305, Interlaken, Switzerland, June 1987.
11. B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: an analysis of eight optical flow algorithms. In *Proc. 1998 British Machine Vision Conference*, Southampton, England, September 1998.
12. C. G. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Fouth Alvey Vision Conference*, pp. 147–152, Manchester, England, August 1988.
13. H. Haußecker and H. Spies. Motion. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook on Computer Vision and Applications. Vol. 2: Signal Processing and Pattern Recognition*, pp. 309–396. Academic Press, San Diego, 1999.
14. B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.
15. B. Jähne. *Spatio-Temporal Image Processing*, volume 751 of *Lecture Notes in Computer Science*. Springer, Berlin, 1993.
16. M. Kass and A. Witkin. Analyzing oriented patterns. *Computer Graphics and Image Processing*, 37:363–385, 1987.
17. U. Köthe. Edge and junction detection with an improved structure tensor. In B. Michaelis and G. Krell, editors, *Pattern Recognition. 25th DAGM Symposium*, number 2781 in Lecture Notes in Computer Science, pp. 25–32, Berlin, 2003. Springer.
18. M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita. Processing of riangiocardiographic images. In K. Preston and M. Onoe, editors, *Digital Processing of Biomedical Images*, pp. 187–202, 1976.
19. F. Lauze, P. Kornprobst, C. Lenglet, R. Deriche, and M. Nielsen. About some optical flow methods from structure tensors: review and contribution. In *RFIA 2004, Actes du* 14^e *Congrès Francophone AFRIF-AFIA*, volume 1, pp. 283–292, Toulouse, January 2004. LAAS-CNRS. In French.

20. T. Lindeberg and J. Garding. Shape from texture from a multi-scale perspective. In *Proc. 4th International Conference on Computer Vision*, pp. 683–691, Berlin, Germany, May 1993.
21. B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Seventh International Joint Conference on Artificial Intelligence*, pp. 674–679, Vancouver, Canada, August 1981.
22. M. Middendorf and H.-H. Nagel. Estimation and interpretation of discontinuities in optical flow fields. In *Proc. Eighth International Conference on Computer Vision*, volume 1, pp. 178–183, Vancouver, Canada, July 2001. IEEE Computer Society Press.
23. M. Middendorf and H.-H. Nagel. Empirically convergent adaptive estimation of grayvalue structure tensors. In L. van Gool, editor, *Proc. 24th DAGM Symposium*, volume 2449 of *Lecture Notes in Computer Science*, pp. 66–74, Zürich, Switzerland, September 2002. Springer.
24. P. Mrázek, J. Weickert, and A. Bruhn. On robust estimation and smoothing with spatial and tonal kernels. Technical Report 51, Series SPP-1114, Department of Mathematics, University of Bremen, Germany, June 2004.
25. M. Nagao and T. Matsuyama. Edge preserving smoothing. *Computer Graphics and Image Processing*, 9(4):394–407, April 1979.
26. H.-H. Nagel and A. Gehrke. Spatiotemporally adaptive estimation and segmentation of OF-fields. In H. Burkhardt and B. Neumann, editors, *Computer Vision – ECCV '98*, volume 1407 of *Lecture Notes in Computer Science*, pp. 86–102. Springer, Berlin, 1998.
27. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:629–639, 1990.
28. A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53:157–185, 1991.
29. K. Rohr. Modelling and identification of characteristic intensity variations. *Image and Vision Computing*, 10(2):66–76, 1992.
30. K. Rohr. Localization properties of direct corner detectors. *Journal of Mathematical Imaging and Vision*, 4:139–150, 1994.
31. M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 699–704, Madison, WI, June 2003.
32. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
33. S. M. Smith and J. M. Brady. SUSAN – a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, May 1997.
34. I. Thomas. Anisotropic adaptation and structure detection. Technical Report F11, Institute for Applied Mathematics, University of Hamburg, Germany, August 1999.
35. D. Tschumperlé and R. Deriche. Diffusion tensor regularization with contraints preservation. In *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pp. 948–953, Kauai, HI, December 2001. IEEE Computer Society Press.
36. R. van den Boomgaard. The Kuwahara-Nagao operator decomposed in terms of a linear smoothing and a morphological sharpening. In H. Talbot and R. Beare,

editors, *Mathematical Morphology, Proceedings of the 6th International Symposium on Mathematical Morphology*, pp. 283–292, Sydney, Australia, April 2002. CSIRO Publishing.

37. R. van den Boomgaard and J. van de Weijer. Robust estimation of orientation for texture analysis. In *Proc. Texture 2002, 2nd International Workshop on Texture Analysis and Synthesis*, Copenhagen, June 2002.
38. J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.
39. J. Weickert. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision*, 31(2/3):111–127, April 1999.
40. J. Weickert. Coherence-enhancing diffusion of colour images. *Image and Vision Computing*, 17(3–4):199–210, March 1999.
41. J. Weickert and T. Brox. Diffusion and regularization of vector- and matrix-valued images. In M. Z. Nashed and O. Scherzer, editors, *Inverse Problems, Image Analysis, and Medical Imaging*, volume 313 of *Contemporary Mathematics*, pp. 251–268. AMS, Providence, 2002.