

# **Adaptation of Tensor Voting to Image Structure Estimation**

Rodrigo Moreno, Luis Pizarro, Bernhard Burgeth, Joachim Weickert,  
Miguel Angel Garcia and Domenec Puig

**Linköping University Post Print**

N.B.: When citing this work, cite the original article.

Original Publication:

Rodrigo Moreno, Luis Pizarro, Bernhard Burgeth, Joachim Weickert, Miguel Angel Garcia and Domenec Puig, Adaptation of Tensor Voting to Image Structure Estimation, 2012, in New Developments in the Visualization and Processing of Tensor Fields, eds David Laidlaw and Anna Vilanova, ISBN: 978-3-642-27342-1, pgs 29-50.

Copyright: Springer

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-79359>

# Adaptation of Tensor Voting to Image Structure Estimation

Rodrigo Moreno, Luis Pizarro, Bernhard Burgeth, Joachim Weickert, Miguel Angel Garcia, and Domenec Puig

**Abstract** Tensor voting is a well-known robust technique for extracting perceptual information from clouds of points. This chapter proposes a general methodology to adapt tensor voting to different types of images in the specific context of image structure estimation. This methodology is based on the structural relationships between tensor voting and the so-called structure tensor, which is the most popu-

---

Rodrigo Moreno

Linköping University, Center for Medical Image Science and Visualization (CMIV)  
Department of Medical and Health Sciences (IMH)  
Campus US, 58185 Linköping, Sweden  
e-mail: rodrigo.moreno@liu.se

Luis Pizarro

Imperial College London, Department of Computing  
180 Queen's Gate, SW7 2AZ, London, United Kingdom  
e-mail: luis.pizarro@imperial.ac.uk

Bernhard Burgeth

Saarland University, Faculty of Mathematics and Computer Science  
Building E2.4, 66041 Saarbrücken, Germany  
e-mail: burgeth@math.uni-sb.de

Joachim Weickert

Saarland University, Faculty of Mathematics and Computer Science  
Mathematical Image Analysis Group (MIA)  
Building E1.1, 66041 Saarbrücken, Germany  
e-mail: weickert@mia.uni-saarland.de

Miguel Angel Garcia

Autonomous University of Madrid, Department of Electronic and Communications Technology  
Francisco Tomas y Valiente 11, 28049 Madrid, Spain  
e-mail: miguelangel.garcia@uam.es

Domenec Puig

Rovira i Virgili University, Department of Computer Science and Mathematics  
Intelligent Robotics and Computer Vision Group (IRCV)  
Av. Països Catalans 26, 43007 Tarragona, Spain  
e-mail: domenec.puig@urv.cat

lar technique for image structure estimation. The problematic Gaussian convolution used by the structure tensor is replaced by tensor voting. Afterwards, the results are appropriately rescaled. This methodology is adapted to gray-valued, color, vector- and tensor-valued images. Results show that tensor voting can estimate image structure more appropriately than the structure tensor and also more robustly.

## 1 Introduction

Medioni and colleagues [26] proposed tensor voting as a robust technique for extracting perceptual information from a cloud of points. In 3D, tensor voting estimates saliency measurements of how likely a point lies on a surface, a curve, a junction, or it is an outlier. It is based on the propagation and aggregation of the most likely normal(s) encoded by means of second-order tensors through a convolution-like voting process, assuming that neighboring points belong to the same smooth surface. This technique has been proven versatile, since it has successfully been adapted to problems well beyond the ones to which it was originally applied with excellent results. For example, this method has already been applied to a variety of problems in image and video processing, such as perceptual organization [26, 43], image restoration [15], image segmentation [22, 29], video segmentation [36, 27], mesh analysis [17], 3D reconstruction [49] and dimensionality estimation [28]. Since the input data for most of these applications are not clouds of points, a common approach is to apply tensor voting as described in [26] to clouds of points derived from the original data. Although, in principle, it is more natural to apply tensor voting to the original data, that application requires extensions of tensor voting to different types of data, which, in most cases, have not been proposed so far.

Furthermore, extensions of tensor voting specifically tailored to applications have also been proved effective. They are based on the incorporation of additional application-dependent perceptual rules to the voting process. For example, the use of specifically designed inhibitory voting fields has been reported beneficial for perceptual organization in gray-scale images [25]. In addition, an extension of tensor voting specifically tailored to color image denoising [31], robust color edge detection [32] and color image segmentation [30] has yielded significantly good results. Related to tensor voting, a voting process specifically designed for detecting X- and T-shaped junctions has been proposed in [1].

In a different context, local image structure estimation methods aim at typifying the region around every pixel. These methods estimate similarity measurements of every local region with respect to certain patterns of interest, such as flat and textured regions, and regions that contain edges, lines or corners. These measurements can be used, for example, to steer image processing methods or to extract local features, such as edges, lines and corners, in a further step.

During the last decades, the use of tensors has allowed local image structure estimation methods to represent several types of local patterns with a same mathematical entity. The most popular of these methods is the structure tensor [10], which

is able to typify flat regions, regions with edges and regions with corners, through second-order tensors. It has been used in a multitude of applications, such as edge detection [11], corner detection [16, 39], texture analysis [38, 40], image filtering [44], image compression [14], optic flow estimation [24, 3], and detection of X- and T-shaped junctions [1]. It has gained popularity thanks to its robustness, efficiency and easiness of implementation. In addition, it depends on a single parameter, which is usually easy to tune.

The main hypothesis made by the structure tensor is that the orientation of the gradient changes slowly in regions with edges, and quickly in regions with corners. In addition, it assumes the size of the gradient to be small in flat regions, and large in both regions with edges and regions with corners. Thus, the structure tensor estimates local image structure by means of a weighted sum of gradients within a neighborhood. For a gray-scale image, the structure tensor,  $J$ , is defined as the convolution of a Gaussian with the tensorized gradient of the image [10]:

$$J = G_\rho * \nabla u \nabla u^T, \quad (1)$$

where  $G_\rho$  is the Gaussian with zero mean and standard deviation  $\rho$ ,  $\nabla u$  is the gradient of the image  $u$ , and  $\nabla u \nabla u^T$  represents the tensorized gradient at every pixel. A related approach based on quadrature filters has been proposed in [19, 12]. Furthermore, extensions using higher-order derivatives [9, 21], and extensions for curved structures [2] have also been proposed.

Despite its popularity, the structure tensor also has important shortcomings, such as detection of features in flat regions, loss of small features, detection of false corners, and misplacement of corners. These shortcomings are mainly related to the use of a Gaussian kernel, since it can propagate the gradient to pixels in flat regions. Thus, the structure tensor can yield similar tensors for flat regions, and regions with edges or corners, leading to errors in the extraction of features. This fact has encouraged researchers to propose alternatives to the structure tensor.

Most of the strategies intend to avoid the integration of different orientations of the gradient by adapting the neighborhood to the data in such a way that only neighbors with similar orientations of the gradient are taken into account in the summation. For example, Nagel and Gehrke [34] and Nath and Palaniappan [35] use adaptive Gaussians instead of a Gaussian convolution; Köthe [20] uses a hourglass-shaped kernel instead of the Gaussian; van de Weijer and van den Boomgaard [47] use robust statistics to choose one of the ambiguous orientations at every pixel; Brox *et al.* [4] and Hahn and Lee [13] propose non-linear diffusion processes in order to aggregate contributions of the neighbors.

Although tensor voting and the structure tensor have been proposed in different contexts, they have important similarities, as will be shown in Section 3. Thus, the aim of this chapter is twofold: First, to propose a general methodology to extend tensor voting to different types of images. This methodology is based on the similarities between the formulations of both tensor voting and the structure tensor. Second, to compare the performance of both methods in the specific context of image structure estimation for different types of images. It is important to remark that application-

dependent extensions of tensor voting are not considered in this chapter, since their formulation could not be related to the structure tensor.

Related to this work, two extensions of classical tensor voting in order to directly apply it to gray-scale images have been proposed. First, Tai *et al.* [42] encode curviness and regionness in the tensors before applying tensor voting. Unfortunately, discriminating edges from corners is not possible by applying this strategy, since both types of structure will yield high curviness using this approach. Second, Loss *et al.* [23] initialize the tensors with *ball* tensors (cf. Section 2) whose size depends on the gray-scale value of the pixel. However, this strategy cannot be used to extract corners.

The chapter is organized as follows. Section 2 summarizes the tensor voting formalism. Section 3 shows the relationships between tensor voting and the structure tensor. Section 4 describes a general methodology to extend tensor voting to different types of images, in particular to gray-scale, vector- and tensor-valued images in the specific application of image structure estimation. Section 5 shows some results of tensor voting applied to image structure estimation. Finally, Section 6 discusses the obtained results and makes some final remarks.

## 2 Tensor Voting

Medioni *et al.* [26] proposed tensor voting as a technique for extracting perceptual information from clouds of points, in particular in 3D. The method robustly estimates saliency measurements of how likely a point lies on a surface, a curve, a junction, or it is an outlier. It is based on the propagation and aggregation of the most likely normal(s) encoded by means of second-order tensors modeled by means of symmetric positive semidefinite matrices. In a first stage, a tensor is initialized at every point in the cloud either with a first estimation of the normal, or with a ball-shaped tensor if a priori information is not available. Afterwards, every tensor is decomposed into its three components: a *stick*, a *plate* and a *ball*. Every component casts votes, which are tensors that encode the most likely direction(s) of the normal at a neighboring point taking into account the information encoded by the voter in that component. Finally, the votes are summed up and analyzed in order to estimate surfaceness, curviness and junctionness measurements at every point. Points with low saliency are assumed to be outliers. More formally, the tensor voting at  $\mathbf{p}$ ,  $TV(\mathbf{p})$  is given by:

$$TV(\mathbf{p}) = \sum_{\mathbf{q} \in \text{neigh}(\mathbf{p})} SV(\mathbf{v}, S_{\mathbf{q}}) + PV(\mathbf{v}, P_{\mathbf{q}}) + BV(\mathbf{v}, B_{\mathbf{q}}), \quad (2)$$

where  $\mathbf{q}$  represents each of the points in the neighborhood of  $\mathbf{p}$ ,  $SV$ ,  $PV$  and  $BV$  are the *stick*, *plate* and *ball* tensor votes cast to  $\mathbf{p}$  by every component of  $\mathbf{q}$ ,  $\mathbf{v} = \mathbf{p} - \mathbf{q}$ , and  $S_{\mathbf{q}}$ ,  $P_{\mathbf{q}}$  and  $B_{\mathbf{q}}$  are the *stick*, *plate* and *ball* components of the tensor at  $\mathbf{q}$  respectively. These components are given by:

$$S_{\mathbf{q}} = (\lambda_1 - \lambda_2) (\mathbf{e}_1 \mathbf{e}_1^T), \quad (3)$$

$$P_{\mathbf{q}} = (\lambda_2 - \lambda_3) (\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T), \quad (4)$$

$$B_{\mathbf{q}} = \lambda_3 (\mathbf{e}_1 \mathbf{e}_1^T + \mathbf{e}_2 \mathbf{e}_2^T + \mathbf{e}_3 \mathbf{e}_3^T), \quad (5)$$

where  $\lambda_i$  and  $\mathbf{e}_i$  are the  $i$ -th largest eigenvalue and its corresponding eigenvector of the tensor at  $\mathbf{q}$ .

Saliency measurements can be estimated from an analysis of the eigenvalues of the resulting tensors in (2). Thus,  $s_1 = (\lambda_1 - \lambda_2)$ ,  $s_2 = (\lambda_2 - \lambda_3)$ , and  $s_3 = \lambda_3$  can be used as measurements of surfaceness, curviness and junctionness respectively. Points whose three eigenvalues are small are regarded as outliers. In addition, eigenvector  $\pm \mathbf{e}_1$  represents the most likely normal for points lying on a surface, whereas  $\pm \mathbf{e}_3$  represents the most likely tangent direction of a curve for points belonging to that curve.

Extensions of tensor voting to N-dimensions are straightforward. In this case, tensors are decomposed into a *stick*, a *ball* and N-2 *plate* components, which are processed through the N-D *stick*, N-D *ball* and N-D *plate* tensor voting respectively [43]. These processes are natural extensions of the 3D case. The next subsections describe how the *stick*, *plate* and *ball* votes are calculated in 3D.

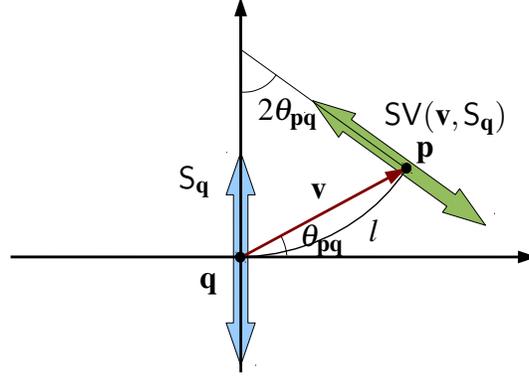
## 2.1 Stick Tensor Voting

*Stick* tensors are used by tensor voting in 3D to encode the orientation of the surface normal at a specific point. Tensor voting handles *stick* tensors through the so-called *stick* tensor voting, which aims at propagating surfaceness in a neighborhood by using the perceptual principles of proximity, similarity and good continuation borrowed from the Gestalt psychology [5]. The *stick* tensor voting is based on the hypothesis that surfaces are usually smooth. Thus, tensor voting assumes that normals of neighboring points lying on the same surface change smoothly. This process is illustrated in Figure 1. Given a known orientation of the normal at a point  $\mathbf{q}$ , which is encoded by  $S_{\mathbf{q}}$ , the orientation of the normal at a neighboring point  $\mathbf{p}$  can be inferred by tracking the change of the normal on a joining smooth curve. Although any smooth curve can be used to calculate *stick* votes, a circumference is usually chosen. A decaying function,  $w_s$ , is also used to weight the vote as defined below.

It is not difficult to show from Figure 1 that for a circumference:

$$SV(\mathbf{v}, S_{\mathbf{q}}) = w_s \left[ R_{2\theta_{\mathbf{pq}}} S_{\mathbf{q}} R_{2\theta_{\mathbf{pq}}}^T \right], \quad (6)$$

where  $\theta_{\mathbf{pq}}$  is shown in Figure 1 and  $R_{2\theta_{\mathbf{pq}}}$  represents a rotation with respect to the axis  $\mathbf{v} \times (S_{\mathbf{q}} \mathbf{v})$ , which is perpendicular to the plane that contains  $\mathbf{v}$  and  $S_{\mathbf{q}}$ . Let  $\lambda$  be the eigenvalue greater than zero of  $S_{\mathbf{q}}$ . Angle  $\theta_{\mathbf{pq}}$  is given by:



**Fig. 1** The *stick* tensor voting. A stick  $S_q$  casts a stick vote  $SV(\mathbf{v}, S_q)$  to  $\mathbf{p}$  that corresponds to the most likely tensorized normal at  $\mathbf{p}$ .

$$\theta_{pq} = \arcsin \left( \sqrt{\frac{\mathbf{v}^T S_q \mathbf{v}}{\lambda \mathbf{v}^T \mathbf{v}}} \right). \quad (7)$$

A point  $\mathbf{q}$  can only cast *stick* votes for  $\theta_{pq} \leq \pi/4$ , since the hypothesis that both points  $\mathbf{p}$  and  $\mathbf{q}$  belong to the same surface becomes more unlikely for larger values of  $\theta_{pq}$ . On the other hand, the weighting function  $w_s$  is used to reduce the strength of the vote with the arc length,  $l$ , given by:

$$l = \frac{\|\mathbf{v}\| \theta_{pq}}{\sin(\theta_{pq})}, \quad (8)$$

and with its curvature,  $\kappa$ , given by:

$$\kappa = \frac{2 \sin(\theta_{pq})}{\|\mathbf{v}\|}. \quad (9)$$

Thus,  $w_s$  is defined as [33]:

$$w_s(\mathbf{v}, S_q) = \begin{cases} e^{-\frac{l^2}{2\rho^2} - b\kappa^2} & \text{if } \theta_{pq} \leq \pi/4 \\ 0 & \text{otherwise,} \end{cases} \quad (10)$$

where  $\rho$  is a scale parameter and  $b$  can be adjusted to give more importance to the curvature. Following the methodology proposed in [33],  $b$  has been set to  $\|\mathbf{v}\|^2/4$  in the experiments.

## 2.2 The Plate Tensor Voting

A *plate* tensor is a tensor with  $\lambda_1 = \lambda_2 \geq 0$  and  $\lambda_3 = 0$ . *Plate* tensors are processed through the so-called *plate* tensor voting. The *plate* tensor voting uses the fact that any *plate* tensor,  $P$ , can be decomposed into all possible *stick* tensors inside the *plate*. Let  $S_P(\beta) = R_\beta \mathbf{e}_1 \mathbf{e}_1^T R_\beta^T$  be a *stick* inside the *plate*  $P$ , with  $\mathbf{e}_1$  being its principal eigenvector, and  $R_\beta$  being a rotation with respect to an axis perpendicular to  $\mathbf{e}_1$  and  $\mathbf{e}_2$ . Thus,  $P$  can be written as:

$$P = \frac{\lambda_1 + \lambda_2}{2\pi} \int_0^{2\pi} S_P(\beta) d\beta. \quad (11)$$

Taking into account that  $S_P(\beta)$  is a *stick* tensor, the *plate* vote is defined as the aggregation of *stick* votes cast by all the *stick* tensors  $S_{P_q}(\beta)$  that constitute  $P_q$ . Thus, the *plate* vote is defined as:

$$PV(\mathbf{v}, P_q) = \frac{\lambda_1}{\pi} \int_0^{2\pi} SV(\mathbf{v}, S_{P_q}(\beta)) d\beta. \quad (12)$$

Although this integral cannot be simplified, *plate* votes can be computed efficiently by using the method proposed in [33].

## 2.3 The Ball Tensor Voting

A *ball* tensor is a tensor with  $\lambda_1 = \lambda_2 = \lambda_3 \geq 0$ . The *ball* tensor voting is defined in a similar way as the *plate* tensor voting. Let  $S_B(\phi, \psi)$  be a unitary *stick* tensor oriented in the direction  $(1, \phi, \psi)$  in spherical coordinates. Then, any *ball* tensor  $B$  can be written as:

$$B = \frac{\lambda_1 + \lambda_2 + \lambda_3}{4\pi} \int_{\Gamma} S_B(\phi, \psi) d\Gamma, \quad (13)$$

where  $\Gamma$  represents the surface of the unitary sphere. Using the same argument as in the case of the *plate* tensor voting, the *ball* vote is defined as:

$$BV(\mathbf{v}, B_q) = \frac{3\lambda_1}{4\pi} \int_{\Gamma} SV(\mathbf{v}, S_{B_q}(\phi, \psi)) d\Gamma. \quad (14)$$

Similarly to the *plate* tensor voting, this integral cannot be simplified. However, *ball* votes can be computed efficiently by using the method proposed in [33].

### 3 Relationships Between the Structure Tensor and Tensor Voting

Although the structure tensor and tensor voting are usually applied to two different scopes, images and clouds of points, both aim at estimating structure, as it will be shown in this section. This section describes the relationships between the structure tensor and tensor voting.

#### 3.1 Similarities

With the exception of the rotation term and the restriction of  $\theta_{\mathbf{pq}} \leq \pi/4$  in (10), the formulation of the *stick* tensor voting in (6) has a structure similar to that of the structure tensor in (1). In particular, the term  $\nabla u \nabla u^T$  in (1) plays a similar role as the term  $S_{\mathbf{q}}$  in (6), while function  $w_s$  of the *stick* tensor voting is closely related to the Gaussian kernel used by the structure tensor. In addition to these structural similarities, both methods have functional connections, since they can be adapted to be applied to the same contexts. Especially, the structure tensor can be adapted to estimation of structures in 3D, and tensor voting can be adapted to estimation of structures in gray-scale images.

On the one hand, the structure tensor can be adapted to estimation of structures in 3D with the help of a norm estimator. For example, the local norm can be estimated by computing the equation of the most likely tangent plane at every point. The norms obtained with such an estimator can be tensorized and convolved with a Gaussian in order to estimate structure in 3D. The resulting tensors yielded by both methods can be analyzed in the same manner. For example,  $\lambda_1 - \lambda_2$  can be used as a measure of surfaceness,  $\lambda_2 - \lambda_3$  as a measure of curviness, and  $\lambda_3$  as a measure of junctionness, as in the case of tensor voting [26].

In turn, tensor voting can be adapted to image structure estimation by designing an appropriate encoding step. Taking into account that the normal,  $\mathbf{n}_{\mathbf{q}}$  in a gray-scale image corresponds to the normalized gradient,  $\nabla u_{\mathbf{q}} / \|\nabla u_{\mathbf{q}}\|$ , the *stick* component  $S_{\mathbf{q}}$  in (3) can be written as:

$$S_{\mathbf{q}} = (\lambda_1 - \lambda_2) \left( \frac{\nabla u_{\mathbf{q}} \nabla u_{\mathbf{q}}^T}{\|\nabla u_{\mathbf{q}}\|^2} \right), \quad (15)$$

which can be further simplified by choosing  $(\lambda_1 - \lambda_2) = \|\nabla u_{\mathbf{q}}\|^2$ . Thus:

$$S_{\mathbf{q}} = \nabla u_{\mathbf{q}} \nabla u_{\mathbf{q}}^T. \quad (16)$$

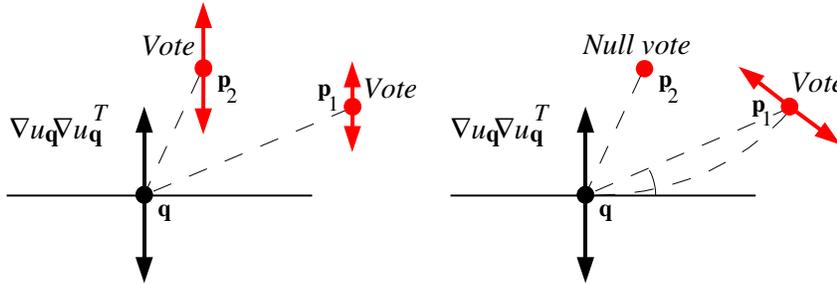
In addition, if the components  $P_{\mathbf{q}}$  and  $B_{\mathbf{q}}$  are set to zero, the input of both, the structure tensor and tensor voting, becomes equivalent for gray-scale images. As in the 3D case, the output of both methods can be analyzed in a similar way, since, in 2D, the shape of the tensors at edges is closer to a *stick*, while the shape tends to a *ball* at corners in both cases (in 2D, the *plate* component is undefined). However,

the tensors obtained by means of tensor voting are in a different scale. Hence, it is necessary to apply a rescaling function in order to have comparable results.

### 3.2 Differences

As already mentioned, both methods have two essential differences: the rotation term in (6) and the restriction of  $\theta_{\mathbf{p}\mathbf{q}} \leq \pi/4$  in (10). These differences are given by the different assumptions made by both methods. On the one hand, the hypothesis of tensor voting is that  $\mathbf{p}$  and  $\mathbf{q}$  belong to the same smooth curve and the voting processes are adjusted according to this hypothesis. On the other hand, the hypothesis made by the structure tensor is that the orientation of the normal at neighboring points should be similar, by taking into account that the orientation of the normal in a smooth curve usually changes slowly.

These differences can be seen in Figure 2. The structure tensor can be modeled as a voting process in which every point votes for its own orientation with a strength given by a Gaussian function. Thus, the structure tensor propagates its own orientation isotropically. This approach can be seen as a displacement to  $\mathbf{p}$  of the surface at  $\mathbf{q}$ . In turn, tensor voting propagates a rotated version of the original orientation when  $\theta_{\mathbf{p}\mathbf{q}} \leq \pi/4$ . It is expected that tensor voting performs better than the structure tensor as it makes stronger assumptions.



**Fig. 2** Left: the structure tensor seen as a voting process. Right: the *stick* tensor voting. The main differences between both are the rotation term (see the difference of votes at  $\mathbf{p}_1$ ) and the anisotropic behavior of tensor voting (tensor voting does not cast votes to  $\mathbf{p}_2$ ).

## 4 Tensor Voting for Structure Estimation

The structural relationships shown in Section 3 lead to a general methodology to extend tensor voting to different types of images. These extensions can be used to

improve the image structure estimation obtained by means of the structure tensor. The methodology comprises three steps. First, tensors are initialized in the same way as for the structure tensor in every different type of images. Second, the Gaussian convolution used by the structure tensor is replaced by tensor voting. Finally, the resulting tensors are rescaled in order to renormalize the total energy stored in the tensors. The following subsections show how this general methodology can be applied to different types of images.

### 4.1 Gray-Scale Images

From Section 3, tensor voting can be directly applied to image structure estimation in gray-scale images by following the next three steps. First, the tensorized gradient,  $\nabla u \nabla u^T$ , is used to initialize a tensor at every pixel. It is important to remark that other types of tensor can be used in the initialization step, for example, *ball* tensors as proposed in [23]. However, the advantage of initializing the tensors with the tensorized gradient is that the input of the structure tensor and tensor voting is the same, easing the comparison between both methods. Second, the *stick* voting process is applied in order to propagate the information encoded in the tensors. That is:

$$\text{TV}(\mathbf{p}) = \sum_{\mathbf{q} \in \text{neigh}(\mathbf{p})} \text{SV}(\mathbf{v}, \nabla u_{\mathbf{q}} \nabla u_{\mathbf{q}}^T). \quad (17)$$

Notice that it is not necessary to apply the *plate* and *ball* voting processes since the *plate* and *ball* components are zero at every pixel due to the initialization step. Finally, the resulting tensors are rescaled by the factor:

$$\xi = \frac{\sum_{\mathbf{p} \in \Omega} \text{trace}(\nabla u_{\mathbf{p}} \nabla u_{\mathbf{p}}^T)}{\sum_{\mathbf{p} \in \Omega} \text{trace}(\text{TV}(\mathbf{p}))}, \quad (18)$$

in order to renormalize the total energy of the tensorized gradient, where  $\Omega$  refers to the given image. This scaling is applied in order to get comparable results to those obtained with the structure tensor.

### 4.2 Color and Vector-Valued Images

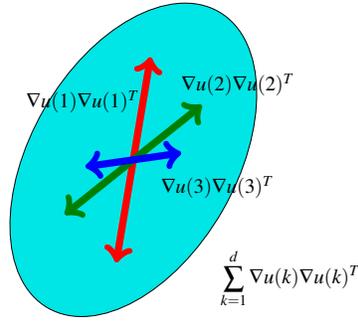
The structure tensor has already been extended to multivalued images in [7] and in a more general way in [45]:

$$J = \sum_{k=1}^d G_\rho * w_k \nabla u(k) \nabla u(k)^T = G_\rho * \sum_{k=1}^d w_k \nabla u(k) \nabla u(k)^T, \quad (19)$$

where  $d$  is the number of channels,  $\nabla u(k)$  is the gradient at channel  $k$ , and  $w_k$  are weights used to give different relevance to every channel. From (19), the structure tensor can be equivalently estimated either by adding  $d$  structure tensors, one for every channel, or by applying a Gaussian kernel on the (weighted) summation of the tensorized gradients  $\nabla u(k) \nabla u(k)^T$ . The reason why both alternatives are equivalent for computing the structure tensor is that Gaussian convolution is linear. However, this equivalence does not hold for non-linear averaging methods, including tensor voting. Thus, there are two options to extend tensor voting for this kind of images, considering that tensor voting must replace the Gaussian convolution used in the structure tensor. The first option is to apply the *stick* tensor voting independently to every channel and then adding up the individual results:

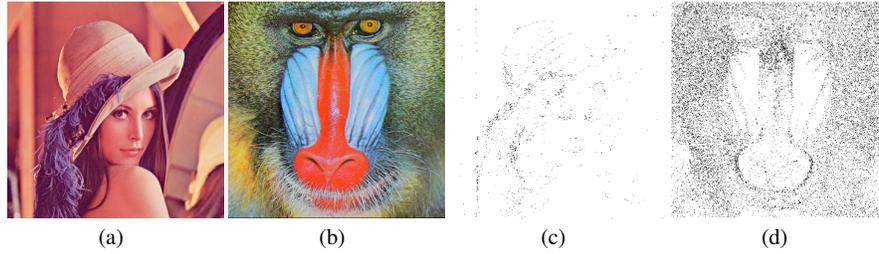
$$\text{TV}(\mathbf{p}) = \sum_{k=1}^d \sum_{\mathbf{q} \in \text{neigh}(\mathbf{p})} w_k \text{SV}(\mathbf{v}, \nabla u_{\mathbf{q}}(k) \nabla u_{\mathbf{q}}(k)^T). \quad (20)$$

The second option is to apply (2) to the sum of tensorized gradients with  $S_{\mathbf{q}}$ ,  $P_{\mathbf{q}}$  and  $B_{\mathbf{q}}$  being the *stick*, *plate* and *ball* components of  $T_{\mathbf{q}} = \sum_{k=1}^d w_k \nabla u_{\mathbf{q}}(k) \nabla u_{\mathbf{q}}(k)^T$ . For two-dimensional images,  $P_{\mathbf{q}} = 0$ . In both options, rescaling the calculated tensors is performed in a similar way as described for the gray-scale images. Figure 3 shows the options described above.



**Fig. 3** Tensor voting can be applied to the channels independently (the red, green and blue *sticks*) or to the sum of the tensorized gradients (the ellipse).

The first option has the advantage that only the application of the *stick* tensor voting is necessary, whereas for the second option, the *stick*, *plate* (for 3D color images) and *ball* tensor voting are required. On the other hand, the second option tends to be more robust since it is less sensitive to bad initial estimations of the gradient. However, in practice,  $T_{\mathbf{q}} \approx S_{\mathbf{q}}$  in most pixels of natural images. As an

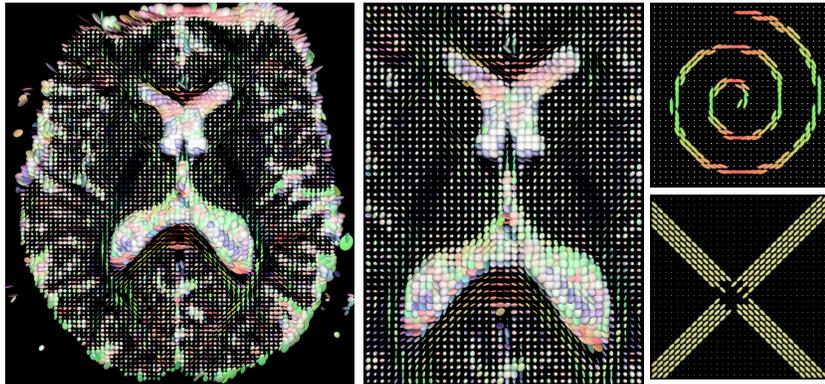


**Fig. 4** (a) Lenna. (b) Mandrill. (c-d) Pixels (in black) with  $\lambda_2 \geq 0.1 \lambda_1$  of  $T_q$  for both images. Processing channels independently is appropriate in most pixels of natural images.

example, in Figure 4 the number of pixels with  $\lambda_2$  greater than the 10% of  $\lambda_1$  of  $T_q$  corresponds to only 0.8% of the total for Lenna and 12.2% for the more textured Mandrill. Thus, the first option can be used in most of the pixels and the second one only in those pixels in which the approximation is not valid.

### 4.3 Tensor-Valued Images

A tensor-valued image is an image in which a tensor is associated with every pixel or voxel. As an example, images acquired through diffusion tensor magnetic resonance imaging (DT-MRI) are tensor-valued. Figure 5 shows examples of this kind of images.



**Fig. 5** Left: 2D slice extracted from a 3D DT-MRI data set ( $128 \times 128$  voxels). Middle: magnified slice around the lateral ventricles ( $40 \times 55$  voxels). Right: two synthetic data sets ( $32 \times 32$  voxels each) of a spiral (top) and a cross (bottom). Ellipsoids are used to represent the tensors associated with every voxel.

Unlike gray-valued and color images, there are several ways to extend the structure tensor concept to tensor-valued images. One of them was proposed by Weickert and Brox [46] in which the structure tensor is calculated through (19), with the channels corresponding to the entries in the tensors. Thus, the same methodology presented in the previous subsection can be used for adapting tensor voting to tensor-valued images by using the entries in the tensors as the channels of a vector-valued image. Moreover, the factors  $w_k$  can be set for tensor-valued images by using the fact that any symmetric matrix,  $M$ , can be modeled by means of a vector,  $m$ , which is given in an orthonormal tensorial basis with respect to the internal product  $\langle A, B \rangle = \text{trace}(AB^T)$  [37, 18]:

$$M = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \iff m = \begin{bmatrix} m_{11} \\ \sqrt{2}m_{12} \\ \sqrt{2}m_{13} \\ m_{22} \\ \sqrt{2}m_{23} \\ m_{33} \end{bmatrix}. \quad (21)$$

This modeling makes equivalent the Frobenius norm  $|M|_F = \sqrt{\text{trace}(MM^T)}$  and the norm of  $m$ . Thus, tensor voting can be applied to vectors  $m$  instead of to tensors  $M$  by using the methodology presented in the previous subsection, with  $w_k = 1$  for the diagonal entries and  $w_k = \sqrt{2}$  for the other entries.

Similarly to the case of color images, there are two options for applying tensor voting: to compute six *stick* votes, one for every channel, or to compute the complete tensor voting framework on the summation of six *stick* tensors, one for every channel (cf. Figure 3). As in the case of color images, the more expensive second option is only necessary at pixels (voxels) where the gradient computed for one channel is very different from the one computed for one another channel.

An alternative extension of tensor voting for diffusion images can be proposed by taking into account the differential nature of this type of images. In DT-MRI, the eigenvectors of the acquired tensors are tangent to the main diffusivity orientations of the movement of water molecules at every voxel. Instead, normal orientations are required to compute structure. Following the approach in [2], such orientations can be extracted from tensors  $R$  computed as:

$$R = \text{trace}(T) I - T, \quad (22)$$

where  $T$  is the acquired tensor and  $I$  the identity matrix. This transformation only modifies the eigenvalues, since tensors  $R$  and  $T$  share the same eigenvectors. The eigendecomposition of these tensors is given by:

$$R = \sum_{i=1}^3 \lambda_i \mathbf{e}_i \mathbf{e}_i^T. \quad (23)$$

Factor  $\mathbf{e}_i \mathbf{e}_i^T$  can be interpreted as tensorized gradients in the image. Thus, by using this analogy, a structure tensor can be defined as:

$$\mathbf{J} = G_\rho * \sum_{i=1}^3 \lambda_i \mathbf{e}_i \mathbf{e}_i^T = \sum_{i=1}^3 G_\rho * \lambda_i \mathbf{e}_i \mathbf{e}_i^T, \quad (24)$$

where the equivalence is given by the linearity of the Gaussian convolution.

Similarly to the case of Figure 3, there are two approaches to extend tensor voting to this type of images: to apply the *stick* tensor voting to every  $\lambda_i \mathbf{e}_i \mathbf{e}_i^T$  or to directly apply the *stick*, *plate* and *ball* tensor voting to the tensors  $R$ .

In Chapter 9, the *stick* votes are represented and accumulated as higher-order tensors, whose weight and orientation are derived from second-order tensors. The analysis of these higher-order tensors is then performed through a low-rank approximation, as proposed in [41].

More sophisticated methods have already been proposed for extending the concept of the structure tensor to tensor-valued images. For example, Burgeth *et al.* [6] use an algebraic approach to deal with the intrinsic third order nature of the gradient of tensor-valued images. Nevertheless, an adaptation of tensor voting based on these methods requires the extension of the voting processes for higher-dimensional tensor-valued images, which is out of the scope of this chapter.

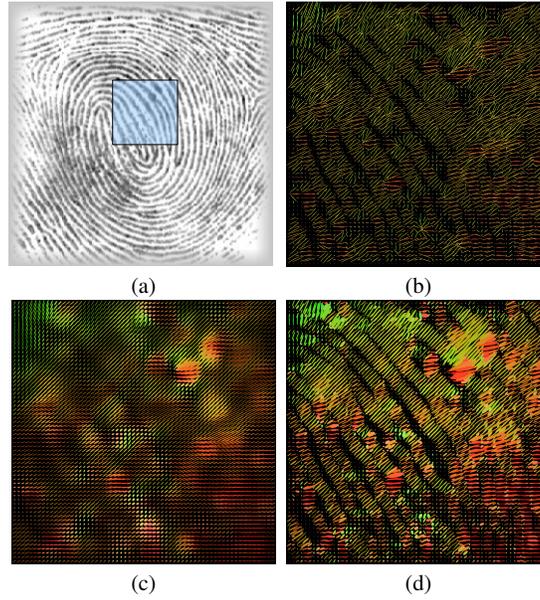
## 5 Experimental Results

Figures 6 to 8 present the structure estimation in a fingerprint by means of both the structure tensor and tensor voting. Figure 6 shows that tensor voting is able to preserve the gaps in the image, while the structure tensor is not. This means that tensor voting avoids estimating structure in unstructured regions, which is one of the known problems of the structure tensor.

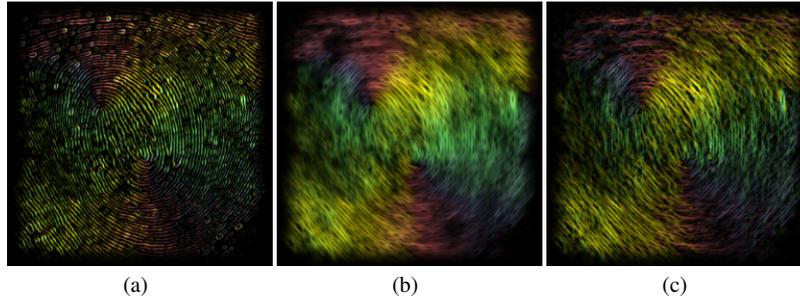
Figure 7 shows that the orientation of the gradient is smoothed by both the structure tensor and tensor voting. This is a good property of a structure estimator, since orientation usually changes slowly in an image and is noisy in  $\nabla u \nabla u^T$ .

Figure 8 shows the map of  $\lambda_1 - \lambda_2$ , which can be used to extract edges. It can be seen that the structure tensor is more sensitive to the selection of the parameter  $\rho$ , while tensor voting yields similar results for a greater range of values. Thus, it is more difficult to tune the parameter of the structure tensor than the scale parameter of tensor voting.

Figure 9 shows an example for edge detection. Since ideal edges are characterized by *stick* tensors, edges can be obtained by applying non-maximum suppression and hysteresis to the map of  $\lambda_1 - \lambda_2$ , which measures how far every pixel is from that condition. It can be seen that the structure tensor blurs that map. This can lead to misplacements of the binary edges extracted from these maps and to loss of small edges. For example, edges inside faces are completely lost, and the eyebrow of the



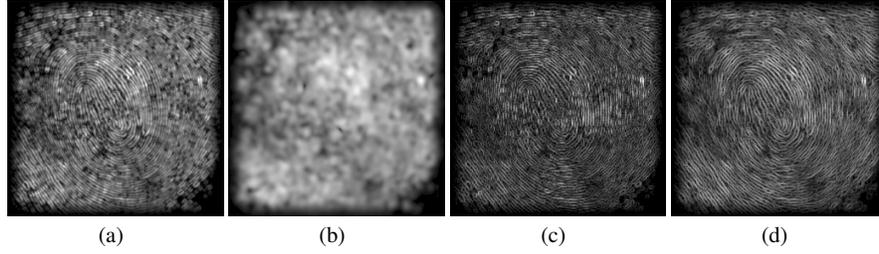
**Fig. 6** (a) A fingerprint with a region of interest (ROI). (b)  $\nabla u \nabla u^T$  in the ROI. (c-d) The structure tensor and tensor voting in the ROI respectively ( $\rho = 2/\sqrt{2}$ ). Tensor voting preserves gaps.



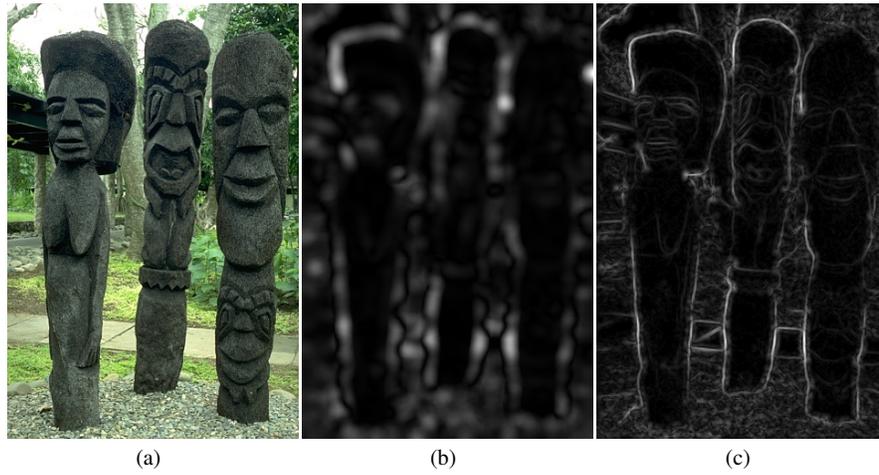
**Fig. 7** (a-c) Color coded orientation (green=0, yellow= $\pi/4$ , red= $\pi/2$ , blue= $3\pi/4$ ) of  $\nabla u_q \nabla u_q^T$ , the structure tensor and tensor voting respectively ( $\rho = 3/\sqrt{2}$ ) for the fingerprint of Figure 6a. Both methods smooth the orientation of the gradient.

totem at the left-hand side is misplaced. Tensor voting is able to keep edges thinner, reducing in that way the problems of the structure tensor.

Most corner detectors apply a function on the eigenvalues of the structure tensor [16]. Hence, accuracy and robustness in the estimation of eigenvalues are requirements for this application. Figures 10 shows plots of  $\lambda_1$  and  $\lambda_2$  from tensors estimated by means of both the structure tensor and tensor voting for a noiseless and a noisy synthetic image. Figure 10 shows that tensor voting is more robust and more accurate than the structure tensor in the estimation of  $\lambda_1$ . In addition, the structure



**Fig. 8** (a-b) Map of  $\lambda_1 - \lambda_2$  obtained with the structure tensor for  $\rho = 1/\sqrt{2}$  and  $\rho = 2/\sqrt{2}$  respectively. (c-d) Map of  $\lambda_1 - \lambda_2$  obtained with tensor voting for the same values of  $\rho$ . The structure tensor is more sensitive to  $\rho$ .

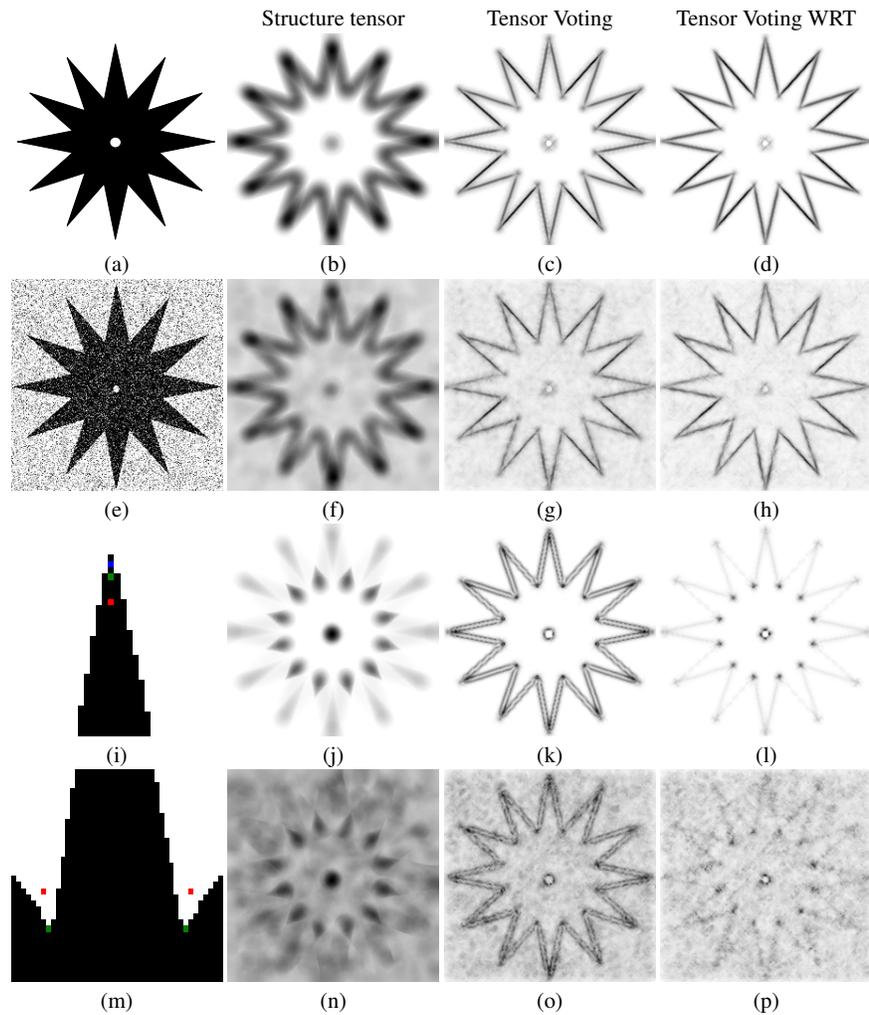


**Fig. 9** (a) Original image. (b-c) Map of  $\lambda_1 - \lambda_2$  for the structure tensor and tensor voting respectively ( $\rho = 3/\sqrt{2}$ ). The structure tensor blurs the edges.

tensor mistakenly introduces a maximum in  $\lambda_1$  in the middle of the small hole inside the star, while tensor voting does not.

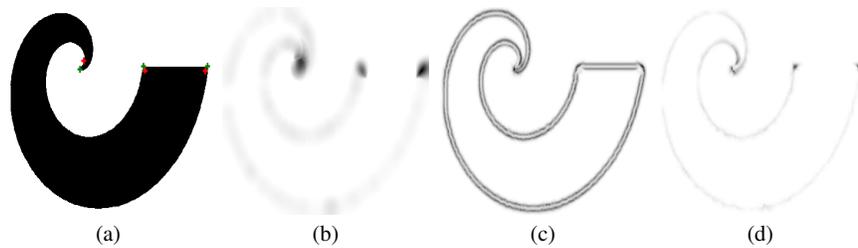
In addition, Figure 10 shows that the structure tensor has a bad performance for both noiseless and noisy images. Actually, it blurs  $\lambda_2$  in such a way that the corners are displaced. In addition, it is very sensitive to noise and generates a false maximum in the hole at the middle of the star. On the other hand, tensor voting has a more consistent performance in estimating  $\lambda_2$  in both noiseless and noisy images. Although tensor voting generates a halo near edges, it can be filtered out by taking into account that it only appears near edges and has smaller values of  $\lambda_2$  than in the corners.

Figure 10 also shows the effect of the rotation term in (6). This figure shows plots of  $\lambda_1$  and  $\lambda_2$  from tensors estimated by means of tensor voting without the rotation term for the image of the star. It can be seen that the effect of the rotation term in



**Fig. 10** (a) Original image. (e) Noisy image (truncated Gaussian noise with  $\sigma = 100$ ). (b-d) Maps of  $\lambda_1$  obtained with the structure tensor, and tensor voting with and without rotation term (WRT) in (6) respectively for the original image ( $\rho = 3/\sqrt{2}$ ). (f-h) Maps of  $\lambda_1$  obtained with the three methods for the noisy image. (j-l) Maps of  $\lambda_2$  obtained with the three methods for the original image. (n-p) Maps of  $\lambda_2$  obtained with the three methods for the noisy image. (i) Detected corner at a peak of the star by the structure tensor (red), tensor voting (green) and tensor voting without the rotation term (blue). (m) Detected corners at two valleys of the star by the structure tensor (red) and both versions of tensor voting (green).

(6) on  $\lambda_1$  is almost negligible, since the results are similar for both the noiseless and noisy images (see Figure 10c vs. 10d, and Figure 10g vs. 10h). Regarding  $\lambda_2$ , tensor voting without the rotation term has a better performance in the noiseless image, since it does not insert halos (see Figure 10l). However, its performance is not robust, since it is difficult to extract maxima from its estimation for the noisy image (see Figure 10p). Thus, tensor voting with the rotation term is more robust in the estimation of  $\lambda_2$ . This effect also appears in curved edges, as shown in Figure 11. In conclusion, the rotation term of (6) robustifies the estimation of  $\lambda_2$  at a cost of introducing halos that should be filtered out a posteriori. It is noteworthy to remark that the method proposed by Köthe [20] is closely related to tensor voting without the rotation term. The only difference between both methods is the use of a different, but still closely related, weighting function.



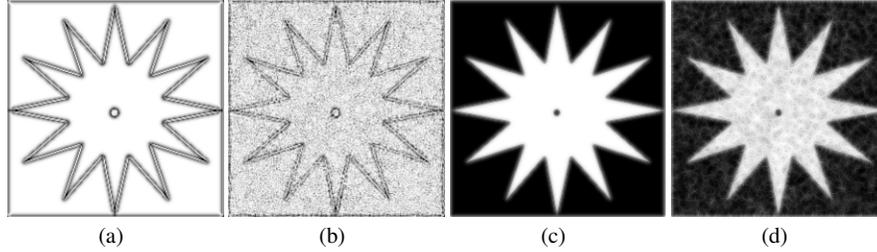
**Fig. 11** (a) Original image with the detected corners with the structure tensor (in red) and tensor voting with and without rotation term in (6) (in green). (b-d) Maps of  $\lambda_2$  obtained with the three methods respectively for the original image ( $\rho = 3/\sqrt{2}$ ).

Regarding precision, tensor voting both with and without the rotation term is able to detect corners with a smaller error. Corners have been detected by looking at local maxima in the map of  $\lambda_2$  (see Figures 10i, 10m and 11a). Table 1 shows the mean errors yielded by both the structure tensor and tensor voting. The strategies based on tensor voting yield better results than the structure tensor in all cases. Notice that corners at the peaks of the star are more difficult to detect, since the angles between the edges that abut at the corner are smaller. In turn, binary edges extracted from the star and the spiral through non-maximum suppression coincide with the ground-truth for both versions of tensor voting. The accuracy of the edges extracted from the structure tensor is also good in regions far away from corners, but it is largely degraded in regions close to corners.

**Table 1** Mean error in corner detection (in pixels) for the synthetic images of Figures 10 and 11.

	Structure tensor	Tensor Voting	Tensor voting (WRT)
Peaks of the star	6.4	3.1	0.9
Valleys of the star	6.5	0.2	0.2
Center of the spiral	9.5	0.0	0.0
Ends of the spiral	4.1	0.0	0.0

Moreover, the method proposed by Loss et al. [23] has been implemented in order to compare two different approaches for extending tensor voting to gray-scale images. Figure 12 shows the results of applying this method to the images of Figures 10a and 10e. As can be seen,  $\lambda_1 - \lambda_2$  generates similar responses at both edges and flat regions, making it difficult to detect edges in noisy images. In turn,  $\lambda_2$  gives no additional information, since it yields a smoothed version of the original image.

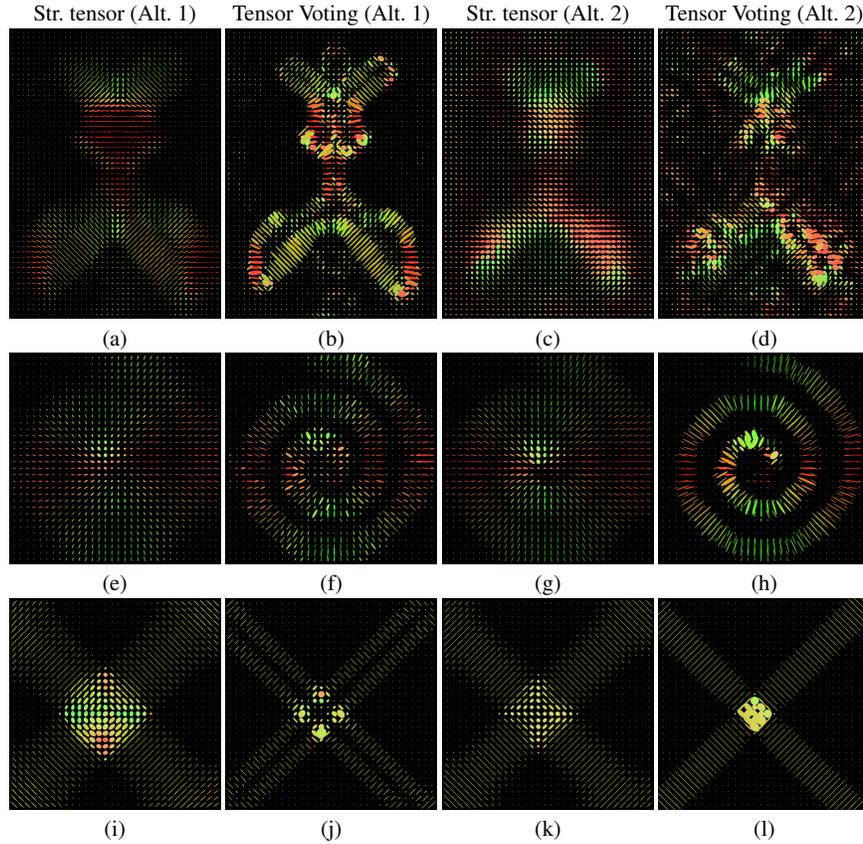


**Fig. 12** (a-b) Maps of  $\lambda_1 - \lambda_2$  calculated with the method by Loss et al. [23] for the images of Figure 10a and 10e respectively ( $\rho = 3/\sqrt{2}$ ). (c-d) Maps of  $\lambda_2$  calculated with the method by Loss et al. [23] for the images of Figure 10a and 10e respectively ( $\rho = 3/\sqrt{2}$ ). Values have been inverted for a better visualization.

Finally, Figure 13 shows that tensor voting is also a better option to be used instead of the structure estimation for tensor-valued images. This figure shows the results yielded for the images of Figure 5 by both the structure tensor and tensor voting computed through the two alternatives described in Subsection 4.3, that is, by modeling tensors as vectors, and by applying (24) and its extension to tensor voting. Notice that both alternatives are not comparable since the former estimates structure in the input tensorial image, while the latter estimates structure in an image related to the inverse gradient [8] of the input image. This fact explains, for example, why tensor voting detects two edges in Figure 13j for every leg of the cross, while it detects only one edge in Figure 13i. As appreciated in these images, the structure tensor blurs the resulting tensors in such a way that it is difficult to extract edges and corners from them. On the contrary, tensor voting is able to estimate structure in a better way.

## 6 Concluding Remarks

This chapter proposes a general methodology to adapt tensor voting for estimating image structure based on the fact that the *stick* tensor voting and the structure tensor are structurally similar, as shown in Section 3. Section 4 has shown how this methodology can be applied to different types of images. Experimental results show that tensor voting can estimate structure more appropriately than the structure tensor. In addition, tensor voting yields more robust estimations of structure than the



**Fig. 13** Resulting tensor fields after applying the structure tensor and the two alternative extensions of tensor voting described in Subsection 4.3 respectively ( $\rho = 5/\sqrt{2}$ ) for the images of Figure 5. Alt. 1 models tensors as vectors, and Alt. 2 is based on (24).

structure tensor. The rotation term in the stick tensor voting leads to more robust estimations of  $\lambda_2$  but also generates halos that should be filtered out a posteriori.

It is interesting to remark that the close relationship between the structure tensor and tensor voting has advantages and shortcomings. On the one hand, this relationship can be used to extend tensor voting to different types of images, as proposed in this paper. On the other hand, this relationship also limits the scope of use of tensor voting to structure estimation. Thus, there are three options to extend tensor voting to other applications. The first one is to use tensor voting in the processing step where structure estimation is required, as many previous works have done. The second one is to model the problem in terms of structure estimation, for example, by using different encoding steps. The third one is to adapt the voting process by encoding new application-dependent perceptual rules. Given its recent success

[31, 32, 30], the third option appears to be the most promising approach for the majority of applications.

Future work includes comparing different ways to perform tensor voting on tensor-valued images and extending the proposed methodology to higher-order tensors. In addition, the inclusion of new perceptual rules in the voting process will be explored in order to eliminate the halos generated by tensor voting in the estimation of  $\lambda_2$  without a post-processing step. Furthermore, comparisons with other approaches in order to combine tensors locally, e.g. [48], are planned.

**Acknowledgements** This research has been partially supported by the Spanish Ministry of Science and Technology under project DPI2007-66556-C03-03, by the Commissioner for Universities and Research of the Department of Innovation, Universities and Companies of the Catalanian Government and by the European Social Fund.

## References

1. Arseneau, S., Cooperstock, J.R.: An improved representation of junctions through asymmetric tensor diffusion. In: Proc. Int. Symp. Visual Computing (ISVC), Lect. Notes Comput. Sci. 4291, pp. 1:363–372 (2006)
2. Bigün, J., Bigun, T., Nilsson, K.: Recognition by symmetry derivatives and the generalized structure tensor. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(12), 1590–1605 (2004)
3. Bigün, J., Granlund, G., Wiklund, J.: Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(8), 775–790 (1991)
4. Brox, T., Weickert, J., Burgeth, B., Mrázek, P.: Nonlinear structure tensors. *Image Vis. Comput.* **24**(1), 41–55 (2006)
5. Bruce, V., Green, P.R., Georgeson, M.A.: *Visual Perception: physiology, psychology and ecology*, fourth edn. Psychology Press (2003)
6. Burgeth, B., Didas, S., Weickert, J.: A general structure tensor concept and coherence-enhancing diffusion filtering for matrix fields. In: D. Laidlaw, J. Weikert (eds.) *Visualization and Processing of Tensor Fields: Advances and Perspectives*, pp. 305–323. Springer (2009)
7. Di Zenzo, S.: A note on the gradient of a multi-image. *Comput. Vis., Graphics, and Image Process.* **33**(1), 116–125 (1986)
8. Farnéback, G., Rydell, J., Ebberts, T., Andersson, M., Knutsson, H.: Efficient computation of the inverse gradient on irregular domains. In: *Int. Conf. Comput. Vis. (ICCV)*, pp. 1–8 (2007)
9. Felsberg, M., Jonsson, E.: Energy tensors: Quadratic, phase invariant image operators. In: Proc. Symp. Ger. Assoc. Pattern Recognit. (DAGM), *Lecture Notes in Computer Science*, vol. 3663, pp. 493–500 (2005)
10. Förstner, W.: A feature based correspondence algorithm for image matching. In: *Int. Arch. of Photogramm. and Remote Sens.*, vol. 26, pp. 150–166 (1986)
11. Förstner, W.: A framework for low-level feature extraction. In: Proc. Eur. Conf. Comput. Vis. (ECCV), Lect. Notes Comput. Sci. 801, pp. 383–394 (1994)
12. Granlund, G., Knutsson, H.: *Signal Processing for Computer Vision*. Kluwer Academic Press (1995)
13. Hahn, J., Lee, C.O.: A nonlinear structure tensor with the diffusivity matrix composed of the image gradient. *J. Math. Imaging Vis.* **34**, 137–151 (2009)
14. Hwang, C., Zhuang, S., Lai, S.H.: Efficient intra mode selection using image structure tensor for H.264/AVC. In: Proc. Int. Conf. on Image Process. (ICIP), pp. V:289–292 (2007)
15. Jia, J., Tang, C.K.: Inference of segmented color and texture description by tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(6), 771–786 (2004)

16. Kenney, C., Zuliani, M., Manjunath, B.: An axiomatic approach to corner detection. In: Proc. Comput. Vis. Pattern Recognit. (CVPR), pp. I:191–197 (2005)
17. Kim, H.S., Choi, H.K., Lee, K.H.: Feature detection of triangular meshes based on tensor voting theory. *Comput.-Aided Des.* **41**(1), 47–58 (2009)
18. Kindlmann, G., Ennis, D.B., Whitaker, R., Westin, C.F.: Diffusion tensor analysis with invariant gradients and rotation tangents. *IEEE Trans. Med. Imag.* **26**(11), 1483–1499 (2007)
19. Knutsson, H.: A tensor representation of 3-D structures. In: Proc. Workshop on Multidimens. Signal Process. (1987)
20. Köthe, U.: Edge and junction detection with an improved structure tensor. In: Proc. Symp. Ger. Assoc. Pattern Recognit. (DAGM), Lect. Notes Comput. Sci. 2781, pp. 25–32 (2003)
21. Köthe, U., Felsberg, M.: Riesz-transforms versus derivatives: On the relationship between the boundary tensor and the energy tensor. In: Scale Space and PDE Methods in Computer Vision, *Lecture Notes in Computer Science*, vol. 3459, pp. 179–191 (2005)
22. Lim, J., Park, J., Medioni, G.: Text segmentation in color images using tensor voting. *Image Vis. Comput.* **25**(5), 671–685 (2007)
23. Loss, L.A., Bebis, G., Parvin, B.: Iterative tensor voting for perceptual grouping of ill-defined curvilinear structures: Application to adherens junctions. *IEEE Trans. Med. Imag.* (2011). In press
24. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. Imaging Underst. Workshop, pp. 121–130 (1981)
25. Massad, A., M., B., Mertsching, B.: Application of the tensor voting technique for perceptual grouping to grey-level images. In: Proc. Symp. Ger. Assoc. Pattern Recognit. (DAGM), Lect. Notes Comput. Sci. 2449, pp. 306–313 (2002)
26. Medioni, G., Lee, M.S., Tang, C.K.: A Computational Framework for Feature Extraction and Segmentation. Elsevier Science (2000)
27. Min, C., Medioni, G.: Inferring segmented dense motion layers using 5D tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(9), 1589–1602 (2008)
28. Mordohai, P., Medioni, G.: Dimensionality estimation, manifold learning and function approximation using tensor voting. *J. of Mach. Learn.* **11**, 411–450 (2010)
29. Moreno, R., Garcia, M.A., Puig, D.: Graph-based perceptual segmentation of stereo vision 3D images at multiple abstraction levels. In: Proc. Workshop on Graph-based Represent. in Pattern Recognit. (GbrPR), Lect. Notes Comput. Sci. 4538, pp. 148–157 (2007)
30. Moreno, R., Garcia, M.A., Puig, D.: Robust color image segmentation through tensor voting. In: Proc. Int. Conf. Pattern Recognit. (ICPR), pp. 3372–3375 (2010)
31. Moreno, R., Garcia, M.A., Puig, D., Julià, C.: On adapting the tensor voting framework to robust color image denoising. In: Proc. Comput. Anal. Images and Patterns (CAIP), Lect. Notes Comput. Sci. 5702, vol. 5702, pp. 492–500 (2009)
32. Moreno, R., Garcia, M.A., Puig, D., Julià, C.: Robust color edge detection through tensor voting. In: Proc. Int. Conf. Image Process. (ICIP), pp. 2153–2156 (2009)
33. Moreno, R., Garcia, M.A., Puig, D., Pizarro, L., Burgeth, B., Weickert, J.: On improving the efficiency of tensor voting. *IEEE Trans. Pattern Anal. Mach. Intell.* (2011). In press
34. Nagel, H.H., Gehrke, A.: Spatiotemporally adaptive estimation and segmentation of OF-fields. In: Proc. Eur. Conf. Comput. Vis. (ECCV), Lect. Notes Comput. Sci. 1407, pp. 86–102 (1998)
35. Nath, S., Palaniappan, K.: Adaptive robust structure tensors for orientation estimation and image segmentation. In: Proc. Int. Symp. Vis. Comput. (ISVC), Lect. Notes Comput. Sci. 3804, pp. 445–453 (2005)
36. Nicolescu, M., Medioni, G.: A voting-based computational framework for visual motion analysis and interpretation. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(5), 739–752 (2005)
37. Pajevic, S., Aldroubi, A., Basser, P.J.: A continuous tensor field approximation of discrete DT-MRI data for extracting microstructural and architectural features of tissue. *J. of Magn. Reson.* **154**, 85–100 (2002)
38. Rao, A.R., Schunck, B.G.: Computing oriented texture fields. *CVGIP: Graph. Models Image Process.* **53**, 157–185 (1991)
39. Rohr, K.: Localization properties of direct corner detectors. *J. Math. Imaging and Vis.* **4**, 139–150 (1994)

40. Rousson, M., Brox, T., Deriche, R.: Active unsupervised texture segmentation on a diffusion based feature space. In: Proc. Comput. Vis. Pattern Recognit. (CVPR), pp. II-699-704 (2003)
41. Schultz, T., Seidel, H.P.: Estimating crossing fibers: A tensor decomposition approach. IEEE Trans. Vis. Comput. Graphics **14**(6), 1635-1642 (2008)
42. Tai, Y.W., Tong, W.S., Tang, C.K.: Perceptually-inspired and edge-directed color image super-resolution. In: Proc. Comput. Vis. and Pattern Recognit. (CVPR), pp. II:1948-1955 (2006)
43. Tang, C.K., Medioni, G., Lee, M.S.: N-Dimensional tensor voting and application to epipolar geometry estimation. IEEE Trans. Pattern Anal. Mach. Intell. **23**(8), 829-844 (2001)
44. Weickert, J.: Coherence-enhancing diffusion filtering. Int. J. Comput. Vis. **31**(2-3), 111-127 (1999)
45. Weickert, J.: Coherence-enhancing diffusion of colour images. Image Vis. Comput. **17**, 199-212 (1999)
46. Weickert, J., Brox, T.: Diffusion and regularization of vector- and matrix-valued images. In: M.Z. Nashed, O. Scherzer (eds.) Inverse Problems, Image Analysis, and Medical Imaging, pp. 251-268. AMS, Providence (2002)
47. van de Weijer, J., van den Boomgaard, R.: Least squares and robust estimation of local image structure. Int. J. Comput. Vis. **64**(2/3), 143-155 (2005)
48. Westin, C.F., Knutsson, H.: Tensor field regularization using normalized convolution. In: Proc. Int. Conf. Comput. Aided Syst. Theory (EUROCAST), Lect. Notes Comput. Sci. 2809, pp. 564-572 (2003)
49. Wu, T.P., Yeung, S.K., Jia, J., Tang, C.K.: Quasi-dense 3D reconstruction using tensor-based multiview stereo. In: Proc. Comput. Vis. Pattern Recognit. (CVPR), pp. 1482-1489 (2010)