# On Improving the Efficiency of Tensor Voting

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

**Abstract**—This paper proposes two alternative formulations to reduce the high computational complexity of tensor voting, a robust perceptual grouping technique used to extract salient information from noisy data. The first scheme consists of numerical approximations of the votes, which have been derived from an in-depth analysis of the *plate* and *ball* voting processes. The second scheme simplifies the formulation while keeping the same perceptual meaning of the original tensor voting: The *stick* tensor voting and the *stick* component of the *plate* tensor voting must reinforce surfaceness, the *plate* components of both the *plate* and *ball* tensor voting must boost curveness, whereas junctionness must be strengthened by the *ball* component of the *ball* tensor voting. Two new parameters have been proposed for the second formulation in order to control the potentially conflictive influence of the *stick* component of the *plate* vote and the *ball* component of the *ball* vote. Results show that the proposed formulations can be used in applications where efficiency is an issue since they have a complexity of order O(1). Moreover, the second proposed formulation has been shown to be more appropriate than the original tensor voting for estimating saliencies by appropriately setting the two new parameters.

**Index Terms**—Perceptual methods, tensor voting, perceptual grouping, nonlinear approximation, curveness and junctionness propagation.

## 1 Introduction

Medioni and colleagues [1], [2], [3] proposed tensor voting as a robust technique for extracting perceptual structures from a cloud of points. This technique has been proven versatile since it has been successfully adapted to problems well beyond the ones to which it was originally applied with excellent results (e.g., [3], [4] and references therein).

Despite its effectiveness, tensor voting cannot be used in applications where efficiency is an issue. This is mainly due to the high computational cost of its classical implementation, especially regarding the *plate* and *ball* tensor voting.

This paper proposes two different ways of implementing tensor voting efficiently. The first one is based on a numerical approximation of the *plate* and *ball* tensor voting, which are mainly responsible for the complexity of the original method. The second one is based on a simplified formulation that fulfills the same perceptual rules followed by tensor voting, although reducing its numerical complexity.

This paper is organized as follows: Section 2 summarizes the original formulation of tensor voting. Section 3 presents the proposed numerical approach for implementing tensor voting efficiently. Section 4 proposes a simplified version of tensor voting based on the perceptual meaning of the *stick*, *plate*, and *ball* tensor voting processes. Section 5 shows an experimental comparison between the original tensor voting and the two proposed schemes. Finally, Section 6 discusses the obtained results and makes some final remarks.

## 2 Tensor Voting

The formulation of tensor voting presented in this section is different from, although equivalent to, the original formulation in [3]. It has been chosen since it simplifies the descriptions in the following sections.

In 3D, tensor voting estimates saliency measurements of how likely it is that a point lies on a surface, a curve, a junction, or is noisy. It is based on the propagation and aggregation of the most likely normal(s) encoded by means of tensors through the so-called *stick*, *plate*, and *ball* tensor voting.

Tensor voting is comprised of three stages. In a first stage, a tensor is initialized at every point of the given cloud of points either with a first estimation of its normal or with a *ball*-shaped tensor if such a priori information is not available. Afterward, every tensor is decomposed into its three components, namely, a *stick*, a *plate*, and a *ball* component. Every component casts votes to the neighboring points by taking into account the information encoded by the voter in that component. Every vote is a tensor that encodes the most likely direction(s) of the normal at a

- R. Moreno is with the Center for Medical Image Science and Visualization (CMIV) and the Department of Medical and Health Sciences (IMH), Linköping University, Campus US, IMT, 13th floor, Linköping 58185, Sweden. E-mail: rodrigo.moreno@liu.se.
- M.A. Garcia is with the Department of Electronic and Communications Technology, Autonomous University of Madrid, Francisco Tomas y Valiente 11, Madrid 28049, Spain. E-mail: miguelangel.garcia@uam.es.
- D. Puig is with the Intelligent Robotics and Computer Vision Group, Rovira i Virgili University, Av. Països Catalans 26, Tarragona 43007, Spain. E-mail: domenec.puig@urv.cat.
- L. Pizarro is with the Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK. He is also with the School of Informatics Engineering, University Diego Portales, Av. Ejército 441, Santiago, Chile. E-mail: l.pizarro@imperial.ac.uk.
- B. Burgeth is with the Faculty of Mathematics and Computer Science, Saarland University, Building E2 4, Saarbrücken 66041, Germany. E-mail: burgeth@math.uni-sb.de.
- J. Weickert is with the Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science, Saarland University, Building E1 1, Saarbrücken 66041, Germany. E-mail: weickert@mia.uni-saarland.de.

*Manuscript received 7 Apr. 2010; revised 6 Oct. 2010; accepted 28 Dec. 2010; published online 28 Jan. 2011.*
*Recommended for acceptance by C.-K. Tang.*
*For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-04-0247.*
*Digital Object Identifier no. 10.1109/TPAMI.2011.23.*

neighboring point. Finally, the votes are summed up and analyzed in order to estimate degrees of surfaceness, curveness and junctionness at every point. Points with low surfaceness, curveness and junctionness are assumed to be noisy observations.

More formally, the tensor voting at $\mathbf{p}$, $\mathrm{TV}(\mathbf{p})$, is given by:

$$\mathrm{TV}(\mathbf{p}) = \sum_{\mathbf{q}\in neigh(\mathbf{p})} (\mathrm{SV}(\mathbf{v}, \mathrm{S}_\mathbf{q}) + \mathrm{PV}(\mathbf{v}, \mathrm{P}_\mathbf{q}) + \mathrm{BV}(\mathbf{v}, \mathrm{B}_\mathbf{q})), \tag{1}$$

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 $\mathrm{S}_\mathbf{q}$, $\mathrm{P}_\mathbf{q}$, and $\mathrm{B}_\mathbf{q}$ are the *stick*, *plate*, and *ball* components of the tensor at $\mathbf{q}$, respectively:

$$\mathrm{S}_\mathbf{q} = (\lambda_1 - \lambda_2)(\mathbf{e}_1\mathbf{e}_1^T), \tag{2}$$

$$\mathrm{P}_\mathbf{q} = (\lambda_2 - \lambda_3)(\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T), \tag{3}$$

$$\mathrm{B}_\mathbf{q} = \lambda_3(\mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_2\mathbf{e}_2^T + \mathbf{e}_3\mathbf{e}_3^T), \tag{4}$$

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

Saliency measurements can be estimated from an analysis of the eigenvalues of the resulting tensors in (1). 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, curveness and junctionness, respectively. Points whose three eigenvalues are small are regarded as noise. 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.

The next sections describe the processes required to calculate *stick*, *plate*, and *ball* tensor votes.

## 2.1 Stick Tensor Voting

*Stick* tensors are used by tensor voting to encode the orientation of the surface normal at a specific 3D 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 a surface change smoothly. This process is illustrated in Fig. 1. Given a known orientation of the normal at a point $\mathbf{q}$, which is encoded by $\mathrm{S}_\mathbf{q}$, 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, $s_{1s}$, is also used to weight the vote as defined below.

For a circumference, it is not difficult to show from Fig. 1 that:

$$\mathrm{SV}(\mathbf{v}, \mathrm{S}_\mathbf{q}) = s_{1s}\left[R_{2\theta} \quad \mathrm{S}_\mathbf{q} \quad R_{2\theta}^T\right], \tag{5}$$

where $\theta$ is the angle shown in Fig. 1 and $R_{2\theta}$ is a rotation with respect to the axis $\mathbf{v} \times (\mathrm{S}_\mathbf{q}\mathbf{v})$, which is perpendicular to the plane that contains both $\mathbf{v}$ and $\mathrm{S}_\mathbf{q}$. Let $\lambda_{\mathrm{S}_\mathbf{q}}$ be the eigenvalue of $\mathrm{S}_\mathbf{q}$ greater than zero. The angle $\theta$ can be calculated as:

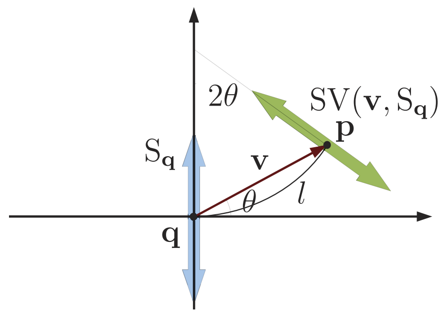

Fig. 1. *Stick* tensor voting. A *stick* $\mathrm{S}_\mathbf{q}$ casts a *stick* vote $\mathrm{SV}(\mathbf{v}, \mathrm{S}_\mathbf{q})$ to $\mathbf{p}$, which corresponds to the most likely tensorized normal at $\mathbf{p}$.

$$\theta = \arcsin\left(\sqrt{\frac{\mathbf{v}^T\mathrm{S}_\mathbf{q}\mathbf{v}}{\lambda_{\mathrm{S}_\mathbf{q}}\mathbf{v}^T\mathbf{v}}}\right). \tag{6}$$

A point $\mathbf{q}$ can only cast *stick* votes for $\theta \leq \pi/4$ since the hypothesis that both points $\mathbf{p}$ and $\mathbf{q}$ belong to the same surface becomes more unlikely for higher values of $\theta$. On the other hand, a weighting function, $s_{1s}$, is used to reduce the strength of the vote with the arc length, $l$, given by:

$$l = \frac{\|\mathbf{v}\|\theta}{\sin(\theta)}, \tag{7}$$

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

$$\kappa = \frac{2\sin(\theta)}{\|\mathbf{v}\|}. \tag{8}$$

Thus, $s_{1s}$ was defined in [3] as:

$$s_{1s}(\mathbf{v}, \mathrm{S}_\mathbf{q}) = \begin{cases} e^{-\frac{l^2+b\kappa^2}{\sigma^2}}, & \text{if } \theta \leq \pi/4, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where $b$ and $\sigma$ are parameters. In practice, $l$ ranges from $\|\mathbf{v}\|$, when $\theta = 0$, to $\frac{\pi}{2\sqrt{2}}\ \|\mathbf{v}\| \approx 1.11\ \|\mathbf{v}\|$, when $\theta = \pi/4$.

## 2.2 Plate Tensor Voting

Tensor voting utilizes *plate* tensors to encode curves in 3D. Ideally, if a point belongs to a curve, the third eigenvector of its tensor must be aligned with the tangent to the curve at that point and $\lambda_3$ must be zero. Tensor voting handles *plate* tensors through the so-called *plate* tensor voting. Unlike *stick* tensor voting, whose formulation derives from perceptual rules to propagate surfaceness, *plate* votes are computed in a constructive way. Thus, *plate* tensor voting uses the fact that any *plate* tensor, P, can be decomposed into all possible *stick* tensors inside the *plate*. Let $\lambda_{i_\mathrm{P}}$ and $\mathbf{e_i}$, respectively, be the $i$th largest eigenvalue of P and its corresponding eigenvector, $R_\beta$ be a rotation with respect to an axis parallel to $\mathbf{e_3}$, which is perpendicular to P, and $\mathrm{S_P}(\beta) = R_\beta\mathbf{e_1}\mathbf{e_1}^T R_\beta^T$ be a *stick* inside the *plate* P derived from $\mathbf{e_1}$. Thus, any *plate* tensor P can be written as:

$$\mathrm{P} = \frac{\lambda_{1_\mathrm{P}} + \lambda_{2_\mathrm{P}}}{2\pi}\int_0^{2\pi} \mathrm{S_P}(\beta)d\beta. \tag{10}$$

Taking into account that $\lambda_{1_{\mathrm{P}}} = \lambda_{2_{\mathrm{P}}}$ and that $\mathrm{S_P}(\beta)$ is a *stick* tensor, the *plate* vote is defined as the aggregation of *stick* votes cast by all the *stick* tensors $\mathrm{S_{P_q}}(\beta)$ that constitute $\mathrm{P}_{\mathbf{q}}$. Thus, the *plate* vote is defined as:

$$\mathrm{PV}(\mathbf{v}, \mathrm{P}_{\mathbf{q}}) = \frac{\lambda_{1_{\mathrm{P_q}}}}{\pi} \int_0^{2\pi} \mathrm{SV}(\mathbf{v}, \mathrm{S_{P_q}}(\beta)) d\beta. \tag{11}$$

## 2.3 Ball Tensor Voting

*Ball* tensors are utilized by tensor voting to encode junctions or noise. *Ball* tensor voting is defined similarly to *plate* tensor voting, that is, in a constructive way. Let $\mathrm{S_B}(\phi, \psi)$ be a unitary *stick* tensor oriented in the direction $(1, \phi, \psi)$ in spherical coordinates. Then, any *ball* tensor B can be written as:

$$\mathrm{B} = \frac{\lambda_{1_{\mathrm{B}}} + \lambda_{2_{\mathrm{B}}} + \lambda_{3_{\mathrm{B}}}}{4\pi} \int_\Gamma \mathrm{S_B}(\phi, \psi) d\Gamma, \tag{12}$$

where $\Gamma$ represents the surface of the unitary sphere and $\lambda_{i_{\mathrm{B}}}$ are the eigenvalues of B. Taking into account that the three eigenvalues $\lambda_{i_{\mathrm{B}}}$ are equal and using the same argument as in the case of *plate* tensor voting, the *ball* vote is defined as:

$$\mathrm{BV}(\mathbf{v}, \mathrm{B}_{\mathbf{q}}) = \frac{3\lambda_{1_{\mathrm{B_q}}}}{4\pi} \int_\Gamma \mathrm{SV}(\mathbf{v}, \mathrm{S_{B_q}}(\phi, \psi)) d\Gamma. \tag{13}$$

## 3 Efficient Formulation for *Plate* and *Ball* Votes

The evaluation of *stick* tensor voting is inexpensive since the rotations involved in that process can be easily avoided by following the geometric constructions of Fig. 1. Actually, the complexity of *stick* tensor voting mainly stems from the computation of an arcsine required to calculate $l$ and the exponential required by (9). In addition, these computations are not necessary for $\theta > \pi/4$.

Additional efforts have also been made to make *stick* tensor voting even more efficient, for example by applying steerable filters in 2D [6] and tensorial harmonics in 3D in order to compute *stick* votes in the frequency domain [7]. Unfortunately, extensions of these methods to calculate *plate* and *ball* votes have not been proposed so far, mainly due to the difficulty in adapting the integrals in (11) and (13) to the frequency domain.

On the other hand, computing *plate* and *ball* votes is highly time consuming since (11) and (13) cannot be analytically simplified. Thus, researchers usually interpolate precomputed tensor fields in order to reduce the complexity of *plate* and *ball* tensor voting. Unfortunately, the amount of precomputed information can grow rapidly if several values of parameter $b$ are used since the voting fields strongly vary with it. In addition, the shape of the voting fields also varies with $\sigma$ since (9) is not scale invariant (cf. Section 3.1). In practice, this fact involves the use of complex systems for data access and memory management which are not always available in many applications.

Following a different strategy, [4], [8], and [9] discard part of the votes for the sake of efficiency. Moreover, [10] proposed an efficient implementation of tensor voting that avoids discarding such information through a parallel implementation on a graphics processing unit (GPU). However, the improvement is determined by the number of available processing units. More recently, [11] and [12] proposed a different weighting factor to be used instead of (9), which aims at avoiding its discontinuity. The introduction of this weighting factor simplifies the computations, but at a cost of yielding very different values from those obtained through the original tensor voting.

The following sections present a numerical approach to implement *plate* and *ball* votes efficiently. Instead of approximating the integrals of (11) and (13), the proposed approach is based on the scale-invariant version of *stick* tensor voting described in the following section.

### 3.1 Scale-Invariant Stick Tensor Voting

Although the formulation of *stick* tensor voting given in Section 2 is inexpensive, it is not scale invariant. Scale invariance, which can be thought of as invariance under change of metric units, is a desirable property since the same results at a particular scale can be obtained for one another by an appropriate scaling of parameters [13]. This property, usually followed by physics laws, has been applied to different fields such as fractal analysis [13], economy [14], and mathematics [15], among many others. Using scale-invariant formulations of tensor voting is advantageous. On the one hand, scale-invariant tensor voting reduces the complexity of the preprocessing step by only precomputing voting fields at a single scale, since votes at a different scale can be interpolated from the precomputed fields by appropriately scaling spatial distances. On the other hand, a scale-invariant version of *stick* tensor voting is essential for analyzing the properties of *plate* and *ball* tensor voting, as shown in the next sections.

Scale invariance can be defined as follows: Let $g$ be a function of a set of variables, $\mathbf{x}$, which directly depend on the spatial length. Function $g$ is scale invariant if [13]:

$$g(\mathbf{x}) = g(h\, \mathbf{x}), \text{ for any } h \in \Re. \tag{14}$$

This definition can be used to check the scale invariance property of (9). Let us consider $s_{1s}$ in (9) as a function of four variables, namely, $l$, $\kappa$, $\sigma$, and $b$. Before checking the scale invariance of $s_{1s}$, it is necessary to determine the dependency of each variable on the spatial length. First, $l$ and $\kappa$ directly and inversely depend on the spatial length, respectively. Second, $\sigma$ directly depends on the spatial length since it is a scale parameter. Finally, $b$ has been chosen in the literature either as a dimensionless constant (e.g., [3], [16]) or as a variable that (mainly) depends on the spatial length (e.g., [17], [4], [8]). It is easy to check that (9) is not scale invariant under these conditions.

One option to make (9) scale invariant is by making $b$ dependent on the fourth power of the spatial length, for instance, with $b$ being proportional to $\sigma^4$, as proposed in [6]. The main problem with this strategy is the difficulty in setting the parameters, since both $b$ and $\sigma$ determine the influence of curvature in the votes. This paper describes an alternative to assure the scale invariance of (9), keeping intuitive parameter tuning.

In particular, the lack of scale invariance of *stick* tensor voting is due to the exponent in (9). From dimensional analysis [18], that exponent must be dimensionless in order to assure scale invariance. Thus, (9) can be converted into a

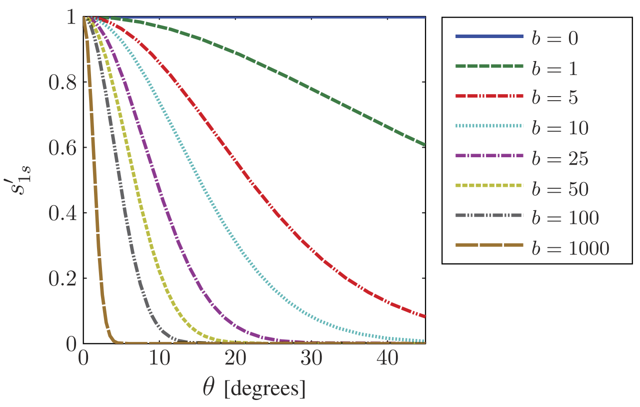

Fig. 2. Evolution of $s'_{1s}$ with respect to $\theta$ for some values of $b$.

scale-invariant equation by using the normalized curvature, $\overline{\kappa}$, instead of the curvature $\kappa$. The normalized curvature is given by [19]:

$$\overline{\kappa} = \kappa\frac{\|\mathbf{v}\|}{2} = \frac{2\sin(\theta)}{\|\mathbf{v}\|}\frac{\|\mathbf{v}\|}{2} = \sin(\theta), \tag{15}$$

where $\theta$ and $\mathbf{v}$ are shown in Fig. 1. Since the normalized curvature is dimensionless, it does not require being weighted by $1/\sigma^2$. Thus, *stick* tensor voting becomes scale-invariant if (9) is replaced by:

$$s_{1s}(\mathbf{v}, \mathrm{S}_{\mathbf{q}}) = \begin{cases} e^{-\frac{l^2}{\sigma^2} - b\overline{\kappa}^2}, & \text{if } \theta \le \pi/4, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

This equation preserves the spirit of (9) in the sense of penalizing *stick* votes by both distance and curvature. Moreover, *plate* and *ball* votes calculated by means of (16) also become scale-invariant thanks to spatial symmetry. Therefore, (16) will be used instead of (9) in the remainder of this work due to its scale invariance.

Fig. 2 shows the effect of parameter $b$ on $s_{1s}$. In this plot, $s'_{1s}$ models the factor of $s_{1s}$ that does not depend on the 3D space, which is given by:

$$s'_{1s} = e^{-b\overline{\kappa}^2}. \tag{17}$$

The figure shows that $b$ can be used to increase the preference for flat surfaces over curved ones. As an example, *stick* votes are negligible when $\theta > 5^\circ$ for $b = 1{,}000$. This means that, in this case, higher values of $\theta$ will not be considered to propagate surfaceness. This behavior could be useful to discriminate between flat surfaces and curved ones.

There are many other alternatives of dimensionless measurements of curvature that can be used instead of the normalized curvature. For example, the degree of curvature, which is given by $2\theta$, is common in engineering (e.g., [20], [21]) and medical sciences (e.g., [22]). Also, the relative eccentricity, which is given by $(1 - \cos(\theta))/(2\sin(\theta))$, has been used in biomechanics [23]. However, the advantage of using the normalized curvature in tensor voting is that its definition is more closely related to the curvature and it is computationally less expensive than the aforementioned measurements.

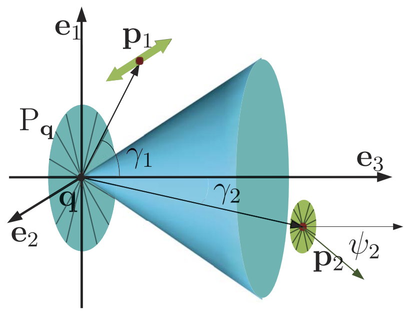

Fig. 3. Examples of the *plate* tensor voting. A tensor $\mathrm{P}_{\mathbf{q}}$ casts votes to its neighbors with a shape that depends on the cone shown in the figure. Votes are close to *sticks* for points outside the cone ($\mathbf{p}_1$) or close to *plates* for points inside the cone ($\mathbf{p}_2$). Note that neither the *stick* component vanishes inside the cone (except for $\gamma = 0$) nor the *plate* component vanishes outside the cone.

### 3.2 Efficient Plate Tensor Voting

Scale invariance and spatial symmetry can be used to analyze *plate* votes. Besides parameters $\sigma$ and $b$, *plate* vote $\mathrm{PV}(\mathbf{v}, \mathrm{P}_{\mathbf{q}})$ depends on two variables: the distance between $\mathbf{p}$ and $\mathbf{q}$, $\|\mathbf{v}\|$, and the angle $\gamma$ between $\mathbf{e_3}$ and $\mathbf{v}$. This angle can be calculated similarly to the angle $\theta$ of the *stick* tensor voting:

$$\gamma = \arcsin\left(\sqrt{\frac{\mathbf{v}^T \mathrm{P}_{\mathbf{q}} \mathbf{v}}{\lambda_{1_{\mathrm{P_q}}} \mathbf{v}^T \mathbf{v}}}\right), \tag{18}$$

with $\lambda_{1_{\mathrm{P_q}}}$ being the largest eigenvalue of $\mathrm{P}_{\mathbf{q}}$.

The shape of *plate* votes is shown in Fig. 3. The first point to remark is that symmetry makes the *ball* component of *plate* votes vanish. This fact has been tested experimentally by checking the coplanarity of the votes cast by every different *stick* inside $\mathrm{P}_{\mathbf{q}}$. Thus, in general, a *plate* vote can be seen as the summation of a *stick* and a *plate* tensor, each of them with a relative strength that depends on the location of the receiver with respect to the cone of Fig. 3.

As shown in that figure, *plate* votes are close to *sticks* for points outside the depicted cone and close to *plates* for points inside the cone ($\gamma \le \pi/4$). This observation stems from the following reasoning: Recall that a *plate* vote is defined as the summation of the *stick* votes cast by all *sticks* inside the voting *plate*. Let $\mathrm{S}_{\mathrm{P_q}}(\beta)$ be the *stick* inside *plate* $\mathrm{P}_{\mathbf{q}}$ that forms an angle $\beta$ with respect to $\mathrm{P}_{\mathbf{q}}\mathbf{v}\mathbf{v}^T\mathrm{P}_{\mathbf{q}}$, which also lies inside $\mathrm{P}_{\mathbf{q}}$. The angle $\theta_\beta$, which is the angle $\theta$ required in (5) to rotate $\mathrm{S}_{\mathrm{P_q}}(\beta)$, can be derived from (6) as:

$$\theta_\beta = \arcsin(|\cos(\beta)|\sin(\gamma)). \tag{19}$$

Thus, $\theta_\beta$ ranges from 0, when $\beta = \pi/2$, to $\gamma$, when $\beta = 0$. Thus, all *sticks* in the *plate* cast nonnull votes for $\gamma \le \pi/4$, while only those whose $\theta_\beta \le \pi/4$ cast nonnull votes for $\gamma > \pi/4$. An extreme case is given for $\gamma = \pi/2$ where only half of the *sticks* in the *plate* cast nonnull votes.

Now, let us consider the case of $b = 0$. In this case, the strength of every *stick* vote is mainly determined by the arc length extended between $\mathbf{p}$ and $\mathbf{q}$ with respect to every voting *stick* $\mathrm{S}_{\mathrm{P_q}}(\beta)$. For points inside the cone depicted in Fig. 3, all $\mathrm{S}_{\mathrm{P_q}}(\beta)$ cast nonnull votes with arc lengths, $l$, varying from $\|\mathbf{v}\|$ and $\|\mathbf{v}\|\gamma/\sin(\gamma)$. Thus, the maximum range of $l$ is attained for $\gamma = \pi/4$ when it ranges between $\|\mathbf{v}\|$ and $1.11\|\mathbf{v}\|$. Thus, for $\gamma \le \pi/4$, the *stick* component of the

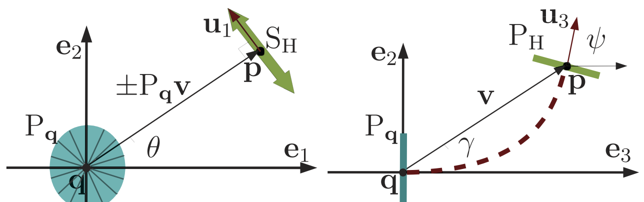

Fig. 4. Components of function $\mathrm{H}$. Left: The *stick* component, $\mathrm{S_H}$, which is always perpendicular to the projection of $\mathbf{v}$ on $\mathrm{P_q}$, given by $\pm\mathrm{P_q}\mathbf{v}$. Right: The *plate* component, $\mathrm{P_H}$, seen from profile. The circumference that joins $\mathbf{q}$ and $\mathbf{p}$ (depicted as a dashed arc) is tangent both to $\mathbf{e}_3$ at $\mathbf{q}$ and to $\mathbf{u}_3$ at $\mathbf{p}$.

*plate* vote is small since the arc length varies in a relatively small range of values. Consequently, the *plate* vote is close to a *plate* inside the cone. For points outside the cone, only a fraction of $\mathrm{S_{P_q}}(\beta)$ cast nonnull votes since $\theta_\beta$ can be higher than $\pi/4$. This makes some orientations more favored than others, leading to an increase in the *stick* component of the *plate* vote. Thus, the *plate* vote becomes closer to a *stick* outside the cone, although the *plate* component does not completely vanish, even for the extreme case of $\gamma = \pi/2$. This general behavior of the *plate* tensor voting can be modified by using higher values of $b$. In that case, the transition between the zone where *mainly-plate* and the zone where *mainly-stick* votes are cast is accelerated.

Thanks to scale invariance, the *plate* vote can be divided into two independent functions: a scalar decaying function $f$, which mainly depends on the spatial distance between the voter and the votee, and a tensorial function H, which does not depend on spatial distance. In practice, $f$ not only depends on $\|\mathbf{v}\|$ and $\sigma$, but also has a slight influence from $\gamma$. Thus, (11) can be rewritten as:

$$\mathrm{PV}(\mathbf{v}, \mathrm{P_q}) = \lambda_{1_{\mathrm{P_q}}} f(\mathbf{v}, \gamma, \sigma)\mathrm{H}(\gamma, b). \tag{20}$$

The scalar function $f$ is given by:

$$f(\mathbf{v}, \gamma, \sigma) = e^{-\frac{t^2\mathbf{v}^T\mathbf{v}}{\sigma^2}}, \tag{21}$$

where $t$ is a factor that takes into account the use of the arc length $l$ instead of the euclidean distance in (16). Although $t$ cannot be derived analytically, good approximations can be obtained as follows: As mentioned above, $l$ ranges between $\|\mathbf{v}\|$ and $\|\mathbf{v}\|\gamma/\sin(\gamma)$ for $\gamma \leq \pi/4$. Thus, $t$ is bound to the range $[1, \gamma/\sin(\gamma)]$. Thanks to the fact that $t$ varies in a small range of values, the mean arc length, which is given by $\|\mathbf{v}\|(1 + \gamma/\sin(\gamma))/2$, can be used to approximate $t$ as $(1 + \gamma/\sin(\gamma))/2$. Note that $t$ varies in a relatively small range since $t \in [1, 1.055]$ in this case. On the other hand, if $\gamma > \pi/4$, only a fraction of $\mathrm{S_{P_q}}(\beta)$ cast nonnull votes, which makes it more difficult to find a close approximation for $t$. The factor $t$ can be experimentally estimated by comparing the trace of *plate* votes computed with arc lengths, as in (9) and (16), to the one computed with euclidean distances instead. Such experiments yielded that $t$ can be approximated by 1.033 for $\gamma > \pi/4$.

On the other hand, H determines the shape of the *plate* vote. H can be decomposed into its *stick* and *plate* components, whose shapes are shown in Fig. 4:

$$\mathrm{H}(\gamma, b) = \mathrm{S_H} + \mathrm{P_H}. \tag{22}$$

These components can be calculated as:

$$\mathrm{S_H} = s'_{1p}\left(\mathbf{u_1}\mathbf{u_1}^T\right), \tag{23}$$

$$\mathrm{P_H} = s'_{2p}\left(\mathbf{u_1}\mathbf{u_1}^T + \mathbf{u_2}\mathbf{u_2}^T\right), \tag{24}$$

with $s'_{1p}$ and $s'_{2p}$ being functions of $\gamma$ and $b$, and $\mathbf{u_i}$ being the eigenvectors of H. Functions $s'_{1p}$ and $s'_{2p}$ capture most of the nonlinearities involved in (11) and cannot be analytically simplified. In turn, $\mathbf{u_i}$ can be calculated as follows: Spatial symmetry makes $\mathbf{u_1}$ perpendicular to $\mathbf{e_3}$ and $\mathbf{v}$. Thus,

$$\mathbf{u_1} = \begin{cases} \dfrac{\mathbf{e_3} \times \mathbf{v}}{\|\mathbf{e_3} \times \mathbf{v}\|}, & \text{if } \mathbf{e_3} \text{ and } \mathbf{v} \text{ are not parallel,} \\ \mathbf{e_1}, & \text{otherwise.} \end{cases} \tag{25}$$

Spatial symmetry also makes $\mathbf{u_3}$ lie on the plane that contains $\mathbf{e_3}$ and $\mathbf{v}$. The angle $\psi$ between $\mathbf{u_3}$ and $\mathbf{e_3}$ is $2\gamma$ for $\gamma < \pi/4$ and $\pi - 2\gamma$ otherwise. Thus, $\mathbf{u_3} = R_\psi\mathbf{e_3}$, where $R$ is a rotation with respect to axis $\mathbf{u_1}$. As in the case of *stick* tensor voting, this rotation can be easily avoided by following the geometry of Fig. 4. Having calculated $\mathbf{u_1}$ and $\mathbf{u_3}$, the remaining eigenvector $\mathbf{u_2}$ can be obtained as $\mathbf{u_2} = \mathbf{u_3} \times \mathbf{u_1}$. As stated before, symmetry makes *plate* votes not to have a *ball* component. Consequently, H does not have a *ball* component either since it models the shape of *plate* votes.

Functions $s'_{1p}$ and $s'_{2p}$ can be estimated from (20) by extracting the eigenvalues of $\mathrm{PV}(\mathbf{v}, \mathrm{P_q})/(\lambda_{1_{\mathrm{P_q}}} f(\mathbf{v}, \gamma, \sigma))$, with $\mathrm{PV}(\mathbf{v}, \mathrm{P_q})$ computed through (11) with a small integration step (e.g., 1 degree). Fig. 5 shows the curves of $s'_{1p}$ and $s'_{2p}$ versus $\gamma$ for different values of $b$. These curves

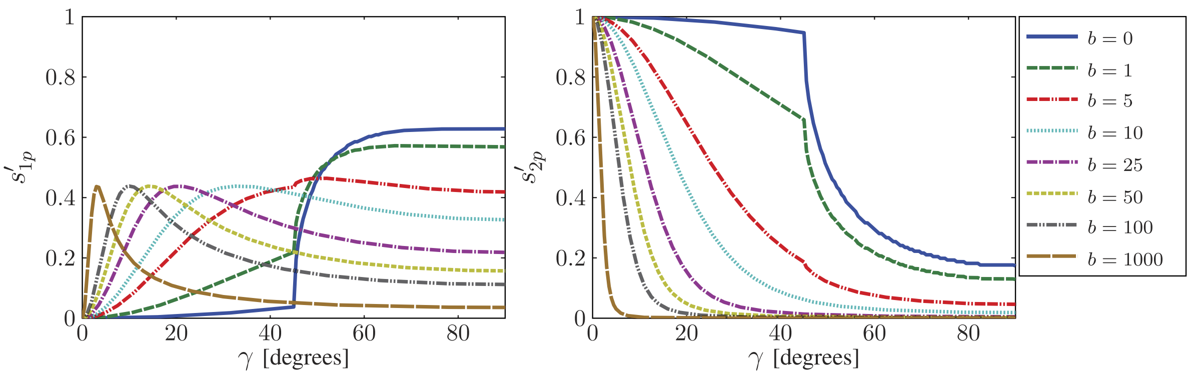

Fig. 5. $s'_{1p}$ and $s'_{2p}$ as functions of $\gamma$ for different values of parameter $b$.

have a discontinuity at $\gamma = \pi/4$. This was expected since the *stick* tensor voting also has a discontinuity at the same angle. The curves corresponding to $s'_{1p}$ and $s'_{2p}$ show that there are mainly two zones in the 3D space, depending on which $s'_{1p}$ or $s'_{2p}$ is dominant, which is congruent with the observations made on Fig. 3. The curves of $s'_{1p}$ and $s'_{2p}$ also show that $b$ can be used to control the spread of the voting cone of Fig. 3 since it becomes narrower as $b$ is increased. These curves can be approximated through any fitting method. As an example, such a fitting can be done by applying two consecutive univariate nonlinear fittings as follows:

1. Selection of univariate nonlinear functions that mimic the shape of $s'_{1p}$ and $s'_{2p}$ for different fixed values of $b$. Some preliminary experiments were conducted with different nonlinear functions in order to determine suitable nonlinear functions to be applied to $\gamma$. These functions contain factors $c_{1i}$ and $c_{2j}$, with $i = 1, \dots, 6$ and $j = 1, \dots, 4$, which can be optimized by nonlinear least squares fitting.
2. Computation of a nonlinear least squares fitting on $\gamma$ for every different value of $b$. This process yields a different value of $c_{1i}$ and $c_{2j}$ for every different value of $b$. At this point, these factors can be stored in order to be queried as look-up tables. These queries are only required once, since $c_{1i}$ and $c_{2j}$ are only functions of $b$. However, univariate nonlinear fitting of $c_{1i}$ and $c_{2j}$ is an advantageous alternative that avoids the use of look-up tables.
3. Selection of univariate nonlinear functions that mimic the evolution of every factor $c_{1i}$ and $c_{2j}$ with $b$. Some preliminary experiments were conducted in order to determine appropriate nonlinear functions to be applied on $b$.
4. Computation of a nonlinear least squares fitting on $b$ for every factor $c_{1i}$ and $c_{2j}$.

The Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.23, shows the fitting yielded by following this methodology. It is important to mention that although more elaborate methodologies can be applied to approximate these curves, the experimental results have shown that their accuracy is good enough to mimic the behavior of the original tensor voting.

The complexity of the proposed implementation of the *plate* tensor voting is mainly due to the computation of an arcsine (which is required to calculate $\gamma$), a logarithm required by the approximation of $s'_{1p}$, and two exponential functions: one for calculating $f(\mathbf{v}, \gamma, \sigma)s'_{1p}$ and another for calculating $f(\mathbf{v}, \gamma, \sigma)s'_{2p}$.

## 3.3 Efficient Ball Tensor Voting

As in the case of *plate* tensor voting, scale invariance and spatial symmetry can also be used to analyze the *ball* votes. Fig. 6 shows some examples of *ball* votes. *Ball* votes are characterized by three properties. The first property is that *ball* votes have an oblate spheroid shape, that is, they are only flattened in one dimension. Thus, $\lambda'_1 = \lambda'_2 > \lambda'_3 > 0$, with $\lambda'_i$ being the eigenvalues of $\mathrm{BV}(\mathbf{v}, \mathrm{B}_\mathbf{q})$ in (13). The second property is that the flattened direction of *ball* votes is always parallel to $\mathbf{v}$. This means that the third eigenvector

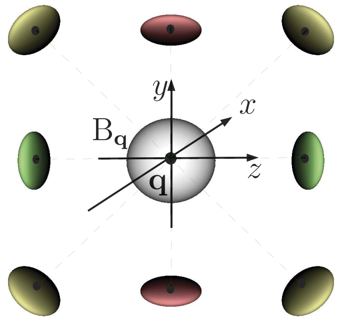

Fig. 6. Examples of *ball* tensor voting. Point $\mathbf{q}$ casts oblate spheroid shaped votes to its neighbors. $\mathrm{BV}(\mathbf{v}, \mathrm{B}_\mathbf{q})$ is shown for different positions of $\mathbf{p}$.

of a *ball* vote, $\mathbf{u_3}$, is parallel to $\mathbf{v}$. The third property is that for some given parameters $\sigma$ and $b$, both the size and flatness of *ball* votes only depend on $\|\mathbf{v}\|$. This condition is given by the isotropic behavior of the *ball* tensor voting. Thus, the *ball* vote can be rewritten as:

$$\mathrm{BV}(\mathbf{v}, \mathrm{B}_\mathbf{q}) = \lambda_{1_{\mathrm{B}_\mathbf{q}}} \left[R_\mathbf{v} S(s_{2b}, s_{2b}, s_{3b}) \mathrm{B}_\mathbf{q} R_\mathbf{v}^T\right], \tag{26}$$

where $R_\mathbf{v}$ is a rotation that makes $\mathbf{u_3}$ and $\mathbf{v}$ parallel, $s_{2b}$ and $s_{3b}$ are functions defined below, and $S$ is a scale transformation that converts the *ball* tensor $\mathrm{B}_\mathbf{q}$ into an oblate spheroid shaped tensor given by:

$$S(s_2, s_2, s_3) = \begin{pmatrix} s_{2b} & 0 & 0 \\ 0 & s_{2b} & 0 \\ 0 & 0 & s_{3b} \end{pmatrix}. \tag{27}$$

The main advantage of (26) is that the expensive integral of (13) is replaced by a rotation. However, this rotation term can also be avoided by constructing the tensor with $\mathbf{v}$. Thus, (26) can be further simplified as:

$$\mathrm{BV}(\mathbf{v}, \mathrm{B}_\mathbf{q}) = \lambda_{1_{\mathrm{B}_\mathbf{q}}} \left[s_{2b}\left(\mathrm{I} - \frac{\mathbf{v}\mathbf{v}^T}{\mathbf{v}^T\mathbf{v}}\right) + s_{3b}\mathrm{I}\right], \tag{28}$$

where I is the identity matrix.

The purpose of $s_{2b}$ is to control the size of the vote, whereas $s_{3b}$ controls how similar the vote is to a *plate* ($s_{3b} = 0$) or to a *ball* ($s_{3b} = s_{2b}$). Unlike *plate* tensor voting, isotropy makes functions $s_{2b}$ and $s_{3b}$ only depend on $\|\mathbf{v}\|$ for specific parameters $\sigma$ and $b$. Thus, thanks to the scale invariance, $s_{2b}$ and $s_{3b}$ can be divided into a Gaussian decaying function on $\|\mathbf{v}\|$ with a standard deviation $\sigma$ and functions $s'_{2b}$, $s'_{3b}$ that only depend on $b$ and cannot be analytically simplified since they capture most of the nonlinearities of (13). Thus, $s_{ib}$ is defined as:

$$s_{ib}(\mathbf{v}, \sigma, b) = s'_{ib} e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2}}, \tag{29}$$

for $i = 2$ and $i = 3$. Factors $s'_{2b}$ and $s'_{3b}$ can be estimated from (13), (28), and (29) by following a similar methodology to the one used for factors $s'_{1p}$ and $s'_{2p}$ in the case of the *plate* tensor voting, that is, by extracting the eigenvalues of $\mathrm{BV}(\mathbf{v}, \mathrm{B}_\mathbf{q})/(\lambda_{1_{\mathrm{B}_\mathbf{q}}} e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2}})$, with $\mathrm{BV}(\mathbf{v}, \mathrm{B}_\mathbf{q})$ computed through (13) with a small integration step (e.g., 1 degree). Fig. 7 shows the evolution of $s'_{2b}$ and $s'_{3b}$ with respect to $b$. It can be seen that $s'_{2b} > s'_{3b}$ for all values of $b$. This means that *ball* votes are more similar to a *plate* than to a *ball* in general. It can also be

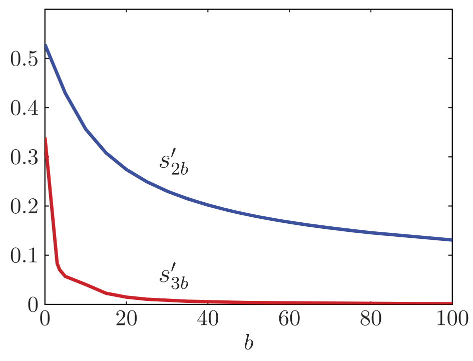

Fig. 7. Evolution of $s'_{2b}$ and $s'_{3b}$ with respect to parameter $b$.

seen that parameter $b$ can be used to reduce the size of the *ball* vote. A similar methodology to the one used to approximate $s'_{1p}$ and $s'_{2p}$ can be applied for approximating $s'_{2b}$ and $s'_{3b}$. In this case, only a single univariate nonlinear fitting is required since these functions only depend on $b$. The Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2011.23, shows the results of least squares fitting for these functions.

The complexity of the proposed implementation of *ball* tensor voting is mainly due to the computation of a single exponential (required in (29)), since the values of $s'_{2b}$ and $s'_{3b}$ are computed only once.

## 4 Simplified Tensor Voting

This section explores an alternative to the numerical approach described in the previous section for calculating tensor voting efficiently. This alternative is based on a simplified formulation that reduces the numerical complexity while keeping the same perceptual rules of tensor voting. The next sections describe the proposed method to calculate *stick*, *plate*, and *ball* tensor votes more efficiently.

### 4.1 Stick Tensor Voting

The original *stick* tensor voting can be further simplified while keeping its perceptual meaning by redesigning the weighting function $s_{1s}$ defined in (16). This function has two parameters: $b$ that penalizes the curvature and $\sigma$ that penalizes both the distance and curvature (the latter through the $\theta/\sin(\theta)$ factor included in the computation of $l$ in (7)). Thus, for example, it is not possible to avoid the influence of curvature on the calculations, even selecting $b = 0$, since $\sigma$ not only affects the distance but also the $\theta/\sin(\theta)$ factor, which is related to curvature. For this reason, every parameter has a single task: $\sigma$ becomes a scale parameter and $b$ a curvature parameter:

$$s_{1s}(\mathbf{v}, \mathrm{S}_\mathbf{q}) = \begin{cases} e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2} - b\,\sin^2(\theta)}, & \text{if } \theta \leq \pi/4, \\ 0, & \text{otherwise.} \end{cases} \qquad (30)$$

This equation has the additional advantage that the arcsine required for calculating *stick* votes is no longer necessary. Note that the only difference between (30) and (16) is the use of the squared euclidean distance ($\mathbf{v}^T\mathbf{v}$) instead of the squared arc length ($l^2$), since $\sin(\theta) = \overline{\kappa}$. This simplification is also based on the fact that the difference between using euclidean distances and arc lengths is relatively small. The use of arc lengths can be seen as spatial stretchings of at most 11 percent (attained at $\theta = \pi/4$) and 5.5 percent (attained at $\gamma = \pi/4$) for *stick* and *plate* votes, respectively. Thus, in general, the effect of the curvature on the votes can be better controlled through $b$.

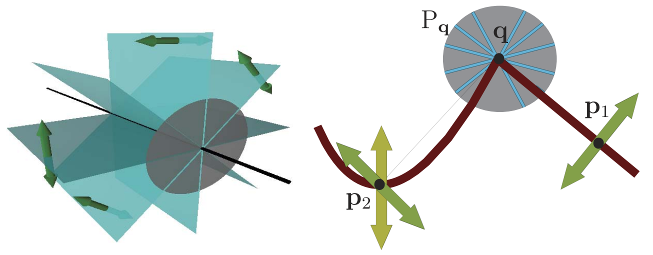

Fig. 8. *Stick* component of the *plate* vote. Left: A curve votes for a plane tangent to the curve. Right: A *plate* tensor in the intersection between a flat and a curved surface (depicted in red). The *stick* component of the *plate* vote can reinforce surfaceness in flat surfaces (see vote at $\mathbf{p}_1$) but also can lead to errors in curved surfaces (see vote at $\mathbf{p}_2$ in green).

### 4.2 Plate Tensor Voting

Proposing simplified equations for the *plate* tensor voting requires understanding the perceptual meaning of *plate* votes. From the analysis carried out in Section 3.2, it can be stated that, from a perceptual point of view, a *plate* vote encodes two different hypotheses, one for every component of the vote.

On the one hand, the hypothesis made by the *stick* component of the *plate* vote is that a neighboring point $\mathbf{p}$ of the voter $\mathbf{q}$ should belong to a surface that abuts the curve that crosses $\mathbf{q}$. However, spatial symmetry makes such a surface a plane since the *stick* component is always tangent to the plate $\mathrm{P}_\mathbf{q}$ (see Fig. 8). Thus, the *stick* component of the *plate* tensor voting can be thought of as a *stick* tensor voting that makes a stronger hypothesis than the *stick* tensor voting itself since curved surfaces are only encouraged in the latter. As seen in Fig. 8, the *stick* component of the plate vote can lead to errors in curved surfaces that must be corrected through *stick* votes cast by other neighbors. This *stick* component mainly appears outside the cone of Fig. 3 since points inside the cone can either belong to the curve that crosses $\mathbf{q}$ or to another surface.

In other words, the *plate* tensor voting has less perceptual information to accurately infer a normal at a neighboring point than the *stick* tensor voting. Thus, it is more likely to estimate normals more accurately with the *stick* tensor voting than with the *stick* component of the *plate* tensor voting. However, if no more information is available, for example, if the receiver only gets votes cast by *plates*, the estimation computed by the *plate* tensor voting can be used as the most likely normal at the receiver.

On the other hand, the hypothesis made by the *plate* component of the *plate* vote is that both points, $\mathbf{p}$ and $\mathbf{q}$, should belong to the same curve. In that sense, $\mathbf{p}$ completes the path of the curve that crosses $\mathbf{q}$. This component mainly appears inside the cone of Fig. 3 since points outside that cone are more unlikely to belong to the same curve. Thus, the *plate* component of the *plate* vote can be thought of as the natural extension of the *stick* tensor voting in which curveness instead of surfaceness is smoothly propagated by following similar rules. Hence, the *plate* component can be considered to be based on the same Gestalt principles as the *stick* tensor voting, namely, proximity, similarity, and good continuation, but adapted to curveness propagation.

Taking into account these arguments, the following equation is proposed to calculate *plate* votes:

$$\mathrm{PV}(\mathbf{v}, \mathrm{P}_{\mathbf{q}}) = s_{2p}\Big[R_{2\gamma} \quad \mathrm{P}_{\mathbf{q}} \quad R_{2\gamma}^T\Big] + \alpha_P \lambda_{1_{\mathrm{P}_{\mathbf{q}}}} s_{1p}\big(\mathbf{u_1}\mathbf{u_1}^T\big), \quad (31)$$

where $\alpha_P \in [0, 1]$ is a new parameter to control the influence of the *stick* component on the *plate* vote, $\lambda_{1_{\mathrm{P}_{\mathbf{q}}}}$ is the largest eigenvalue of $\mathrm{P}_{\mathbf{q}}$, $\mathbf{u_1}$ is calculated through (25), $R_{2\gamma}$ is a rotation with respect to $\mathbf{u_1}$, and $s_{ip}$ are weighting functions given by:

$$s_{2p}(\mathbf{v}, \mathrm{P}_{\mathbf{q}}) = \begin{cases} e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2} - b\,\sin^2(\gamma)}, & \text{if } \gamma \leq \pi/4, \\ 0, & \text{otherwise,} \end{cases} \quad (32)$$

$$s_{1p}(\mathbf{v}, \mathrm{P}_{\mathbf{q}}) = \begin{cases} e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2} - b\,\cos^2(\gamma)}, & \text{if } \gamma > \pi/4, \\ 0, & \text{otherwise.} \end{cases} \quad (33)$$

Factor $s_{2p}$ in (32) has a similar formulation as $s_{1s}$ in (30) since the *plate* component of *plate* votes is the natural extension to *plates* of the perceptual rules of *stick* tensor voting. In this case, $\gamma$ is used instead of $\theta$ since curvature is related to the former in *plate* votes. In turn, the *stick* component of *plate* votes has a mirroring evolution with $\gamma$ when compared to the *plate* component. This inverse relation is modeled in $s_{1p}$ by making it dependent on $\pi/2 - \gamma$ instead of on $\gamma$. This is achieved by using $\cos(\gamma)$ instead of $\sin(\gamma)$ in (33). As stated in the previous section, the rotation term can be avoided by following the geometry of Fig. 4. Thus, the complexity of this alternative mainly stems from an exponential function for $s_{1p}$ or $s_{2p}$, depending on the angle $\gamma$.

The selection of $\alpha_P$ implies a trade-off that depends on the type, density of the data, as well as the level of noise. Thus, by setting $\alpha_P = 0$, the responsibility for estimating normals at surfaces is mainly endorsed to the *stick* tensor voting. This setting should not be used when the data is too sparse to get enough *stick* votes at points in surfaces. In turn, by setting $\alpha_P = 1$, the responsibility for estimating normals at surfaces is shared between the *stick* and the *plate* tensor voting. At flat surfaces, this setting is beneficial since it can help to improve the estimation of normals as more votes are collected, especially in very noisy scenarios. However, at points in curved surfaces, the *stick* component of a *plate* vote can introduce errors whose relevance inversely depends on the number and strength of the *stick* votes cast by other neighbors. For data sets with both flat and curved surfaces, $\alpha_P$ should be set to zero in order to avoid the introduction of errors in the curved surfaces, unless the density of points allowed to cast *stick* votes is large enough to make such an error negligible.

### 4.3 Ball Tensor Voting

A perceptual interpretation of *ball* votes, necessary for proposing a simplified *ball* tensor voting, can be obtained from the analysis performed in Section 3.3. As stated before, a *ball* vote only consists of a *plate* and a *ball* component. On the one hand, the meaning of the *plate* component is that both points, $\mathbf{p}$ and $\mathbf{q}$, should belong to a straight edge in the direction that joins both points. Although, a *ball* tensor at $\mathbf{q}$ represents a complete uncertainty about the normal direction at that point, this uncertainty is reduced in direction $\mathbf{v}$ because both points could belong to a straight edge that is likely joining both points.

On the other hand, the meaning of the *ball* component is that points near a junction should have a junctionness saliency different from zero. From a different point of view, normal uncertainty at a point infers some normal uncertainty at its neighborhood. Unlike the *plate* component, it is difficult to justify from the perceptual point of view the existence of the *ball* component of the *ball* vote since junctions are not usually close to each other. However, it could be useful in iterative schemes, e.g., [16], [24], in order to induce uncertainty for those cases in which the tensors are initialized with not too accurate values.

Hence, the same (28) is proposed to calculate *ball* votes, but with the following weighting functions:

$$s_{2b}(\mathbf{v}, \mathrm{B}_{\mathbf{q}}) = e^{-\frac{\mathbf{v}^T\mathbf{v}}{\sigma^2}}, \quad (34)$$

$$s_{3b}(\mathbf{v}, \mathrm{B}_{\mathbf{q}}) = \alpha_B s_{2b}(\mathbf{v}, \mathrm{B}_{\mathbf{q}}), \quad (35)$$

where parameter $\alpha_B \geq 0$ can be used to control the influence of the *ball* component on the *ball* vote. Thus, the high complexity of the *ball* tensor voting is reduced to the computation of a single exponential function. It is important to remark that isotropy makes these functions to be independent from curvature.

A similar reasoning as the one described for the *stick* component of the *plate* tensor voting can be used for the *plate* component of the *ball* tensor voting. The *ball* tensor voting has less information to accurately infer a curve continuation at a neighboring point than the *plate* tensor voting; hence the latter is preferred if available. This would give place to a new parameter to control the influence of the *plate* component of the *ball* vote on the estimation of surface intersections. However, in practice, tensor voting is usually run as proposed in [3], that is, tensors are initialized with unitary *balls* and two rounds of tensor voting are applied: The first one only considers the *ball* tensor voting, whereas the second round only considers both the *stick* and *plate* tensor voting. Since *plate* and *ball* tensor voting are not usually run at the same time, it is safe to avoid this new parameter.

## 5 Experimental Results

The formulations of the original (OTV), efficient (ETV), and simplified tensor voting (STV) presented above were coded in MATLAB on an Intel Core 2 Quad Q6600 with a 4 GB RAM in order to compare the new proposed schemes with the original tensor voting. In addition, the approximation scheme described in [4] and [8], referred to as MM, has also been coded to compare its performance with the proposed methods. Equation (16) has been applied instead of (9) in order to make the results of all tested methods comparable.

### 5.1 Efficiency

Table 1 shows the mean execution times of the tested methods. This table shows that OTV is impractical for many

TABLE 1
Speed Measurements of the Original Tensor Voting

| Integration step (degrees) | *Plate* votes | | *Ball* votes | |
|---|---|---|---|---|
| | Time (ms) | Votes per second | Time (s) | Votes per second |
| 0.5 | 87.1 | 11.48 | 124.48 | $8.03 \times 10^{-3}$ |
| 1 | 79.3 | 12.61 | 41.10 | $2.43 \times 10^{-2}$ |
| 2 | 32.5 | 30.77 | 10.90 | $9.17 \times 10^{-2}$ |
| 5 | 19.0 | 52.63 | 1.75 | 0.57 |
| 10 | 12.0 | 83.33 | 0.45 | 2.22 |
| 20 | 6.6 | 152.88 | 0.12 | 8.01 |
| 30 | 5.6 | 178.25 | 0.05 | 19.19 |
| 45 | 5.3 | 189.04 | 0.03 | 28.49 |

applications. As an example, assume that a small cloud of points consists of 1,000 points, and that the propagation of votes is restricted to the 25 nearest points. Thus, the computation of 25,000 *stick*, *plate,* and *ball* votes is required. With OTV, they can be calculated in between 16.85 minutes and 36.04 days depending on the desired precision (controlled by the integration step). For this reason, precomputing and storing the votes in voting fields by means of look-up tables is the only practical solution to apply OTV. Unfortunately, minimal or negligible loss of accuracy can only be attained through an expensive preprocessing stage to compute the voting fields using a small integration step. On the other hand, ETV does not require preprocessing and only takes 0.05, 0.18, and 0.05 milliseconds for every *stick*, *plate,* and *ball* vote, respectively, with an affordable loss of accuracy. Thus, in the aforementioned example, the proposed formulation only takes 7.05 seconds. In addition, this time can be further improved by implementing the method in a noninterpreted programming language, such as C/C++. The efficiency of STV is slightly better than ETV. In this case, the *stick*, *plate,* and *ball* votes can be processed in 0.05, 0.15, and 0.04 milliseconds, respectively, on average. MM also has an efficient performance, since *plate* and *ball* votes can be processed in 0.20 and 0.04 milliseconds, respectively, on average. In these experiments, it is clear that the efficiency of OTV is affected by the use of loops in MATLAB. Although, the relative improvement in speed from ETV, STV, and MM is expected to decrease with a C/C++ implementation, such a reduction is rather limited since OTV is more computationally complex than ETV, STV, and MM.

### 5.2 Comparisons with OTV

In order to assess the differences between OTV and the other methods, tensors computed with ETV, STV, and MM have been compared to those obtained through OTV with a small integration step at a number of random points in the space. Two data sets of 1,000 and 100 normally distributed random points with $\sigma = 1$ have been generated to assess *plate* and *ball* votes, respectively. Fewer points have been tested for comparing *ball* votes in order to obtain results in a reasonable amount of time. For this experiment, OTV has been run with an integration step of 1 degree for $\sigma = 1$ and 5. Eight independent experiments have been run for $b = 0$, 1, 5, and 10 with $\sigma = 1$ and 5, respectively. Table 2 shows the differences between OTV and MM, ETV, and STV. These differences have been computed through the mean angular error of $\mathbf{e_1}$ and $\mathbf{e_3}$ for *plate* votes cast by a *plate* at the origin ($\lambda_{1_P} = \lambda_{2_P} = 1$), and $\mathbf{e_3}$ for *ball* votes cast by a *ball* at the origin ($\lambda_{1_B} = \lambda_{2_B} = \lambda_{3_B} = 1$), in addition to the root mean square error of the eigenvalues $\lambda_1$ and $\lambda_2$ for *plate* votes and $\lambda_2$ and $\lambda_3$ for *ball* votes. As shown in the table, ETV makes a better approximation of OTV than MM and STV. In particular, MM and STV introduce relevant diferences in eigenvalues for both *plate* and *ball* votes. This result was expected for STV since it does not aim at mimicking the behavior of OTV (cf. Section 4). It can also be seen that *ball* votes are equivalent for MM and STV with $\alpha_B = 0$, which was also expected. In summary, ETV is approximately equivalent to OTV, while MM and STV are different than OTV.

### 5.3 Accuracy

Accuracy has been measured by comparing the ground-truth with the results of applying the tested methods to some synthetic data sets. Fig. 9 shows the point-sampled surfaces used in these experiments and their noisy counterparts. As suggested in [3], tensors were initialized with unitary *balls* and two rounds of tensor voting were applied: The first one only considered the *ball* tensor voting, whereas

TABLE 2
Mean Angular Errors of Eigenvectors (in Degrees) and Root Mean Square Errors of the Eigenvalues

| | | *Plate* votes ($\sigma = 1$) | | | | *Plate* votes ($\sigma = 5$) | | | | *Ball* votes ($\sigma = 1$) | | | *Ball* votes ($\sigma = 5$) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $b$ | $\mathbf{e}_1$ | $\mathbf{e}_3$ | $\lambda_1$ | $\lambda_2$ | $\mathbf{e}_1$ | $\mathbf{e}_3$ | $\lambda_1$ | $\lambda_2$ | $\mathbf{e}_3$ | $\lambda_2$ | $\lambda_3$ | $\mathbf{e}_3$ | $\lambda_2$ | $\lambda_3$ |
| | 0 | 0.00 | 0.00 | 0.068 | 0.315 | 0.00 | 0.00 | 0.107 | 0.577 | 0.00 | 0.074 | 0.184 | 0.00 | 0.113 | 0.336 |
| MM | 1 | 0.00 | 0.00 | 0.116 | 0.311 | 0.00 | 0.00 | 0.195 | 0.513 | 0.00 | 0.120 | 0.125 | 0.00 | 0.216 | 0.251 |
| | 5 | 0.00 | 0.00 | 0.234 | 0.288 | 0.00 | 0.00 | 0.412 | 0.402 | 0.00 | 0.236 | 0.047 | 0.00 | 0.450 | 0.094 |
| | 10 | 0.00 | 0.00 | 0.300 | 0.246 | 0.00 | 0.00 | 0.534 | 0.322 | 0.00 | 0.299 | 0.020 | 0.00 | 0.575 | 0.039 |
| | 0 | 0.00 | 0.00 | 0.040 | 0.015 | 0.00 | 0.00 | 0.072 | 0.030 | 0.00 | 0.009 | 0.008 | 0.00 | 0.013 | 0.013 |
| ETV | 1 | 0.00 | 0.00 | 0.035 | 0.012 | 0.00 | 0.00 | 0.050 | 0.025 | 0.00 | 0.007 | 0.006 | 0.00 | 0.021 | 0.017 |
| | 5 | 0.00 | 0.00 | 0.019 | 0.012 | 0.00 | 0.00 | 0.028 | 0.022 | 0.00 | 0.013 | 0.013 | 0.00 | 0.031 | 0.029 |
| | 10 | 0.00 | 0.00 | 0.015 | 0.018 | 0.00 | 0.00 | 0.021 | 0.033 | 0.00 | 0.012 | 0.013 | 0.00 | 0.025 | 0.026 |
| | 0 | 0.00 | 0.00 | 0.382 | 0.140 | 0.00 | 0.00 | 0.702 | 0.264 | 0.00 | 0.074 | 0.184 | 0.00 | 0.113 | 0.336 |
| STV | 1 | 0.00 | 0.00 | 0.339 | 0.102 | 0.00 | 0.00 | 0.620 | 0.194 | 0.00 | 0.120 | 0.125 | 0.00 | 0.216 | 0.251 |
| $\alpha_P = 0$ | 5 | 0.00 | 0.00 | 0.265 | 0.046 | 0.00 | 0.00 | 0.479 | 0.087 | 0.00 | 0.236 | 0.047 | 0.00 | 0.450 | 0.094 |
| $\alpha_B = 0$ | 10 | 0.00 | 0.00 | 0.223 | 0.030 | 0.00 | 0.00 | 0.403 | 0.057 | 0.00 | 0.299 | 0.020 | 0.00 | 0.575 | 0.039 |
| | 0 | 0.00 | 0.00 | 0.068 | 0.140 | 0.00 | 0.00 | 0.107 | 0.264 | 0.00 | 0.564 | 0.330 | 0.00 | 1.069 | 0.620 |
| STV | 1 | 0.00 | 0.00 | 0.107 | 0.102 | 0.00 | 0.00 | 0.187 | 0.194 | 0.00 | 0.615 | 0.372 | 0.00 | 1.173 | 0.705 |
| $\alpha_P = 1$ | 5 | 0.00 | 0.00 | 0.213 | 0.046 | 0.00 | 0.00 | 0.378 | 0.087 | 0.00 | 0.732 | 0.449 | 0.00 | 1.406 | 0.862 |
| $\alpha_B = 1$ | 10 | 0.00 | 0.00 | 0.227 | 0.030 | 0.00 | 0.00 | 0.405 | 0.057 | 0.00 | 0.795 | 0.476 | 0.00 | 1.531 | 0.917 |

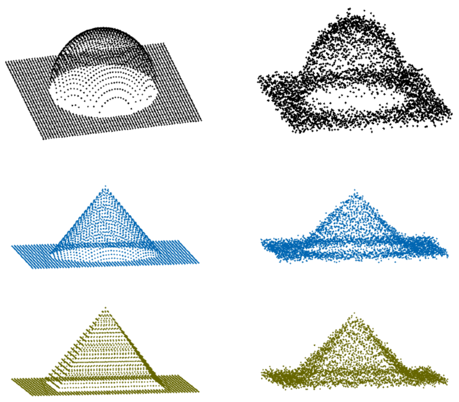

Fig. 9. Clouds of points used in the experiments. Left: Point-sampled surfaces, each constituted by 3,721 points. Right: A noisy version of the same surfaces (Gaussian noise with standard deviation of 0.2).

the second round only considered both the *stick* and *plate* tensor voting. Parameter $\alpha_B$ was set to zero and $\sigma$ was set to five. Independent experiments were run for $b = 0$ and $b = 10$. STV has been run with $\alpha_P = 0$ and $\alpha_P = 1$ in order to assess the effect of this parameter. For this and the following experiments, OTV has been computed by the interpolation of precomputed voting fields since the application of OTV with a small integration step is impractical in this case.

The mean angular error between $\mathbf{e}_1$ and ideal normals on surfaces, and of $\mathbf{e}_3$ and ideal edge orientations at edges have been used to measure the accuracy of the algorithms for estimating normals and curve orientations, respectively. In addition, the following measurement of discriminability of saliencies has been used to assess the saliency estimation:

$$d_1 = \frac{1}{\|S\|} \sum_{\mathbf{p} \in S} \frac{s_1(\mathbf{p})}{\lambda_1(\mathbf{p})} - \frac{1}{n - \|S\|} \sum_{\mathbf{p} \notin S} \frac{s_1(\mathbf{p})}{\lambda_1(\mathbf{p})}, \tag{36}$$

$$d_2 = \frac{1}{\|C\|} \sum_{\mathbf{p} \in C} \frac{s_2(\mathbf{p})}{\lambda_1(\mathbf{p})} - \frac{1}{n - \|C\|} \sum_{\mathbf{p} \notin C} \frac{s_2(\mathbf{p})}{\lambda_1(\mathbf{p})}, \tag{37}$$

$$d_3 = \frac{1}{\|J\|} \sum_{\mathbf{p} \in J} \frac{s_3(\mathbf{p})}{\lambda_1(\mathbf{p})} - \frac{1}{n - \|J\|} \sum_{\mathbf{p} \notin J} \frac{s_3(\mathbf{p})}{\lambda_1(\mathbf{p})}, \tag{38}$$

where $n$ is the total number of points in the data set, $S$, $C$, and $J$ are the set of points that belong to surfaces, curves, and junctions, respectively, and $\|\cdot\|$ is the cardinality of a set. In addition, $d_3$ has independently been computed for the junction at the top of the pyramid, $d_{3t}$, and for the junctions at the base, $d_{3b}$, since they represent two different types of junctions. As pointed out in [3], the classification of points into surfaces, curves, and junctions cannot be performed by selecting the points where one saliency is larger than the others. Instead, the classification is performed by extracting local maxima of $s_3$ for junctions and through marching schemes for surfaces and curves, which also search for local maxima of $s_1$ and $s_2$, respectively. Thus, the proposed discriminability measurements estimate the degree of difficulty of deciding whether or not a point belongs to a surface, a curve or a junction, respectively, from the saliency measurements estimated through every method.

Tables 3 and 4 summarize the results for the noiseless and noisy data sets, respectively. Table 3 shows that all methods yield similar angular errors in all data sets. The reason for this behavior is that the number of points that belong to surfaces where *stick* votes are more important is

TABLE 3
Mean Angular Error of $\mathbf{e}_1$ and $\mathbf{e}_3$ in Degrees and Discriminability of Saliencies for the Noiseless Data Sets of Fig. 9

| Method | | Semisphere | | | | Cone | | | | Pyramid | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Surf. | | Curv. | | Surf. | | Curv. | | Surf. | | Curv. | | Junctions | | |
| | b | $\mathbf{e}_1$ | $d_1$ | $\mathbf{e}_3$ | $d_2$ | $\mathbf{e}_1$ | $d_1$ | $\mathbf{e}_3$ | $d_2$ | $\mathbf{e}_1$ | $d_1$ | $\mathbf{e}_3$ | $d_2$ | $d_3$ | $d_{3b}$ | $d_{3t}$ |
| OTV | 0 | **0.33** | **0.08** | 0.45 | **0.05** | **0.90** | **0.09** | 0.48 | **0.09** | 1.06 | **0.16** | 0.64 | **0.15** | 0.19 | 0.11 | 0.51 |
| MM | 0 | **0.33** | **0.08** | 0.56 | 0.04 | **0.90** | 0.08 | 0.58 | 0.08 | 1.12 | 0.15 | 0.74 | 0.14 | 0.19 | **0.12** | 0.46 |
| ETV | 0 | **0.33** | **0.08** | 0.45 | **0.05** | **0.90** | **0.09** | 0.48 | **0.09** | 1.08 | **0.16** | 0.65 | **0.15** | 0.19 | 0.11 | 0.50 |
| STV ($\alpha_P = 0$) | 0 | **0.33** | **0.08** | **0.42** | **0.05** | **0.90** | **0.09** | **0.45** | **0.09** | 1.09 | **0.16** | 0.68 | **0.15** | 0.18 | **0.12** | 0.41 |
| STV ($\alpha_P = 1$) | 0 | **0.33** | **0.08** | 0.44 | **0.05** | 0.91 | **0.09** | 0.51 | **0.09** | **1.05** | **0.16** | **0.59** | **0.15** | **0.20** | 0.11 | **0.55** |
| OTV | 10 | **0.32** | **0.07** | 0.32 | **0.05** | **0.91** | 0.08 | 0.35 | **0.08** | 1.17 | **0.15** | 0.89 | 0.13 | 0.19 | **0.07** | 0.68 |
| MM | 10 | **0.32** | **0.07** | 0.54 | 0.04 | 0.93 | 0.08 | 0.58 | **0.08** | 1.16 | 0.14 | 0.94 | **0.14** | 0.18 | 0.06 | 0.67 |
| ETV | 10 | **0.32** | **0.07** | 0.32 | **0.05** | **0.91** | 0.08 | 0.33 | **0.08** | 1.17 | **0.15** | 0.88 | 0.13 | 0.19 | **0.07** | 0.68 |
| STV ($\alpha_P = 0$) | 10 | **0.32** | **0.07** | **0.31** | **0.05** | 0.92 | **0.08** | **0.30** | **0.08** | 1.17 | **0.15** | 0.92 | 0.13 | 0.19 | 0.06 | 0.69 |
| STV ($\alpha_P = 1$) | 10 | **0.32** | **0.07** | **0.31** | **0.05** | 0.92 | 0.08 | 0.31 | **0.08** | **1.15** | **0.15** | **0.83** | 0.13 | **0.21** | 0.06 | **0.80** |

TABLE 4
Mean Angular Error of $\mathbf{e}_1$ and $\mathbf{e}_3$ in Degrees and Discriminability of Saliencies for the Noisy Data Sets of Fig. 9

| Method | | Semisphere | | | | Cone | | | | Pyramid | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Surf. | | Curv. | | Surf. | | Curv. | | Surf. | | Curv. | | Junctions | | |
| | b | $\mathbf{e}_1$ | $d_1$ | $\mathbf{e}_3$ | $d_2$ | $\mathbf{e}_1$ | $d_1$ | $\mathbf{e}_3$ | $d_2$ | $\mathbf{e}_1$ | $d_1$ | $\mathbf{e}_3$ | $d_2$ | $d_3$ | $d_{3b}$ | $d_{3t}$ |
| OTV | 0 | 5.62 | 0.07 | 3.90 | **0.04** | 5.62 | **0.09** | 5.06 | **0.08** | 5.21 | 0.15 | 4.30 | **0.14** | **0.18** | 0.11 | 0.44 |
| MM | 0 | 5.63 | 0.07 | 4.09 | 0.03 | 5.64 | 0.08 | 5.57 | 0.07 | 5.23 | 0.14 | 4.44 | 0.13 | 0.17 | 0.11 | 0.39 |
| ETV | 0 | 5.62 | 0.07 | 3.96 | **0.04** | 5.63 | **0.09** | 5.05 | **0.08** | 5.23 | 0.15 | 4.31 | **0.14** | **0.18** | 0.11 | **0.45** |
| STV ($\alpha_P = 0$) | 0 | **5.61** | **0.08** | **3.68** | **0.04** | **5.59** | **0.09** | **4.85** | **0.08** | 5.32 | 0.15 | 4.41 | **0.14** | **0.18** | **0.12** | 0.40 |
| STV ($\alpha_P = 1$) | 0 | 5.66 | 0.07 | 4.02 | **0.04** | 5.69 | **0.09** | 4.90 | **0.08** | **5.18** | **0.16** | **4.28** | **0.14** | **0.18** | **0.12** | 0.43 |
| OTV | 10 | 5.02 | 0.06 | 2.54 | **0.06** | 5.09 | 0.08 | 3.48 | **0.08** | 4.78 | 0.15 | 3.38 | **0.13** | **0.15** | **0.08** | 0.43 |
| MM | 10 | 5.09 | 0.06 | 3.20 | 0.04 | 5.12 | 0.08 | 4.24 | **0.08** | 4.78 | 0.14 | 3.70 | **0.13** | **0.15** | **0.08** | 0.43 |
| ETV | 10 | 5.01 | 0.06 | 2.53 | **0.06** | 5.09 | 0.08 | 3.46 | **0.08** | 4.79 | 0.15 | 3.37 | **0.13** | **0.15** | **0.08** | 0.44 |
| STV ($\alpha_P = 0$) | 10 | **4.96** | **0.07** | **2.51** | **0.06** | **5.05** | **0.09** | **3.45** | **0.08** | 4.81 | 0.14 | 3.50 | **0.13** | **0.15** | 0.07 | 0.46 |
| STV ($\alpha_P = 1$) | 10 | 4.98 | 0.06 | 2.60 | **0.06** | 5.09 | 0.09 | 3.68 | **0.08** | **4.75** | **0.16** | **3.35** | **0.13** | **0.15** | 0.07 | **0.47** |

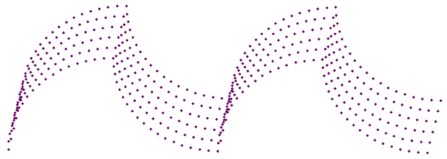

Fig. 10. Cloud of points used to assess the effect of the *stick* component of *plate* votes.

much higher than those that belong to curves or junctions. Thus, the total vote is much more influenced by the *stick* votes cast by neighboring points at the same surface than by *plate* votes cast by points located at neighboring curves. However, STV with $\alpha_P = 0$ has the better performance in curved data sets since it yields smaller angular errors for $\mathbf{e_3}$. In turn, STV with $\alpha_P = 1$ has the best performance for the pyramid according to the mean angular errors. That means that points in curves are actually affected by *plate* votes cast from points located at neighboring surfaces. Errors related to *plate* votes are mitigated at points belonging to surfaces by the fact that they receive more *stick* votes from other points in the same surface. It is also important to note that parameter $b$ barely influences angular errors and it tends to reduce the discriminability measurements in noiseless scenarios. Another observation with respect to this table is that discriminability measurements are relatively small. This does not suppose a problem for detecting curves and junctions since they are located at points where saliencies $s_2$ and $s_3$ attain local maxima values, respectively. Alternatively, iterative schemes can also be used to increase these discriminability measurements.

In turn, Table 4 shows that, although the angular errors are higher, the observations made for noiseless scenarios are also valid in noisy ones. That is, STV yields the best results for curved scenarios by setting $\alpha_P = 0$ and for the pyramid by setting $\alpha_P = 1$. An interesting observation is that discriminability measurements $d_1$ and $d_2$ are similar to those reported in Table 3. That means that the detection of curves is almost not influenced by noise. On the other hand, although the discriminability measurement $d_{3t}$ is more affected by noise, the values are still high. In addition, the discriminability $d_{3b}$ is less affected by noise. This means that junctions at the base of the pyramid can also be extracted from a noisy scenario.

Regarding MM, this experiment confirms the observation made in Table 2 in the sense that it is different from OTV since both yield different results. In addition, the approximation made by MM usually yields worse results than OTV. However, it remains a good alternative to the methods proposed in this paper. As for ETV, these experiments show that it succeeds in mimicking the behavior of OTV since it yields almost the result in all measurements.

### 5.4 Effect of Parameter $\alpha_P$ of STV

The effect of the *stick* component of *plate* votes has been assessed by measuring the distortion introduced by the methods when the tensors are initialized with the ground truth for the cloud of points shown in Fig. 10. The same measurements used in the previous experiment have been applied to this experiment and $\sigma$ has been set to five. Table 5 shows that STV with $\alpha_P = 0$ is the method that less angular distortion introduces both for $b = 0$ and $b = 10$. However,

TABLE 5
Mean Angular Error of $\mathbf{e_1}$ and $\mathbf{e_3}$ in Degrees and Discriminability of Saliencies for the Cloud of Points of Fig. 10 for Two Types of Initialization

| Method | | Ground-truth | | | | Unitary *balls* | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $b$ | $\mathbf{e_1}$ | $d_1$ | $\mathbf{e_3}$ | $d_2$ | $\mathbf{e_1}$ | $d_1$ | $\mathbf{e_3}$ | $d_2$ |
| OTV | 0 | 2.32 | 0.88 | 0.00 | 0.92 | 4.98 | **0.72** | 0.00 | **0.59** |
| MM | 0 | 2.07 | 0.82 | 0.00 | 0.92 | 5.02 | 0.66 | 0.00 | 0.57 |
| ETV | 0 | 2.30 | 0.88 | 0.00 | 0.92 | 4.96 | 0.71 | 0.00 | **0.59** |
| STV ($\alpha_P = 0$) | 0 | **1.69** | 0.87 | 0.00 | 0.90 | **4.83** | 0.70 | 0.00 | 0.57 |
| STV ($\alpha_P = 1$) | 0 | 2.84 | **0.91** | 0.00 | **0.93** | 4.98 | **0.72** | 0.00 | 0.58 |
| OTV | 10 | 1.87 | 0.96 | 0.00 | **0.97** | 4.70 | **0.49** | 0.00 | 0.39 |
| MM | 10 | 2.24 | 0.84 | 0.00 | 0.93 | 4.75 | **0.49** | 0.00 | **0.40** |
| ETV | 10 | 1.86 | **0.97** | 0.00 | **0.97** | 4.71 | **0.49** | 0.00 | 0.38 |
| STV ($\alpha_P = 0$) | 10 | **1.56** | 0.96 | 0.00 | 0.96 | **4.59** | 0.47 | 0.00 | **0.40** |
| STV ($\alpha_P = 1$) | 10 | 2.22 | **0.97** | 0.00 | **0.97** | 4.63 | 0.47 | 0.00 | **0.40** |

the differences between STV and the other methods are reduced for $b = 10$. The reason for this behavior is that fewer points at curves are allowed to cast *stick* components, so their influence in the total vote is reduced in such a case. Although STV with $\alpha_P = 1$ is the method that induces less reductions in discriminability of saliencies, it has bad performance in this data set since it introduces higher angular errors. That means that setting $\alpha_P = 1$ is not appropriate for curved data sets. An expected result was a reduction in the discriminability of saliencies, which are one in the ground truth, for all tested methods. Since these reductions are relatively small, especially for $b = 10$, they can be thought of as the small price that tensor voting has to pay for yielding robust results.

In addition, Table 5 shows the results on this data set for tensors initialized with unitary *balls* and two applied rounds of tensor voting: one for *ball* votes and the other for *stick* and *plate* votes. For this data set, angular errors and discriminabilities are larger than the values reported in Tables 3 and 4 for other data sets. This is mainly due to two factors. First, the surfaces are intersected at an angle of 90 degrees, which maximizes the saliency $s_2$ in curves, leading to an increase in $d_1$ and $d_2$. Second, the data set is rather sparse, so fewer votes are received at every point. Thus, the effect of an erroneous vote cannot be effectively compensated for with enough correct ones, leading to an increase in the angular errors.

### 5.5 Effect of Parameter $\alpha_B$ of STV

Values of $\alpha_B$ greater than zero are only useful in iterative schemes where tensors have been initialized with bad estimations of the normals. In order to test the effect of this parameter, 15 iterations of the *stick*, *plate*, and *ball* tensor voting have been run for a sampled flat surface. Tensors have been initialized with one of the worst possible scenarios, that is, with tensors that are perpendicular to the normals in the surface. In particular, tensors have been initialized with *plate* tensors that are tangent to the surface. Thus, the angular error of $\mathbf{e}_1$ is 90 degrees at the beginning of the process. In addition, a small *ball* component has also been added to the tensors in order to force the application of the *ball* tensor voting in the first iteration. Parameter $\alpha_B$ has been set to 10 for the first iteration and to 0 for the subsequent iterations since better estimations of the tensors are then available. Parameter $\sigma$ has been set to two, while $\alpha_P$ has been set to 0. The arc cosine of the normalized tensor

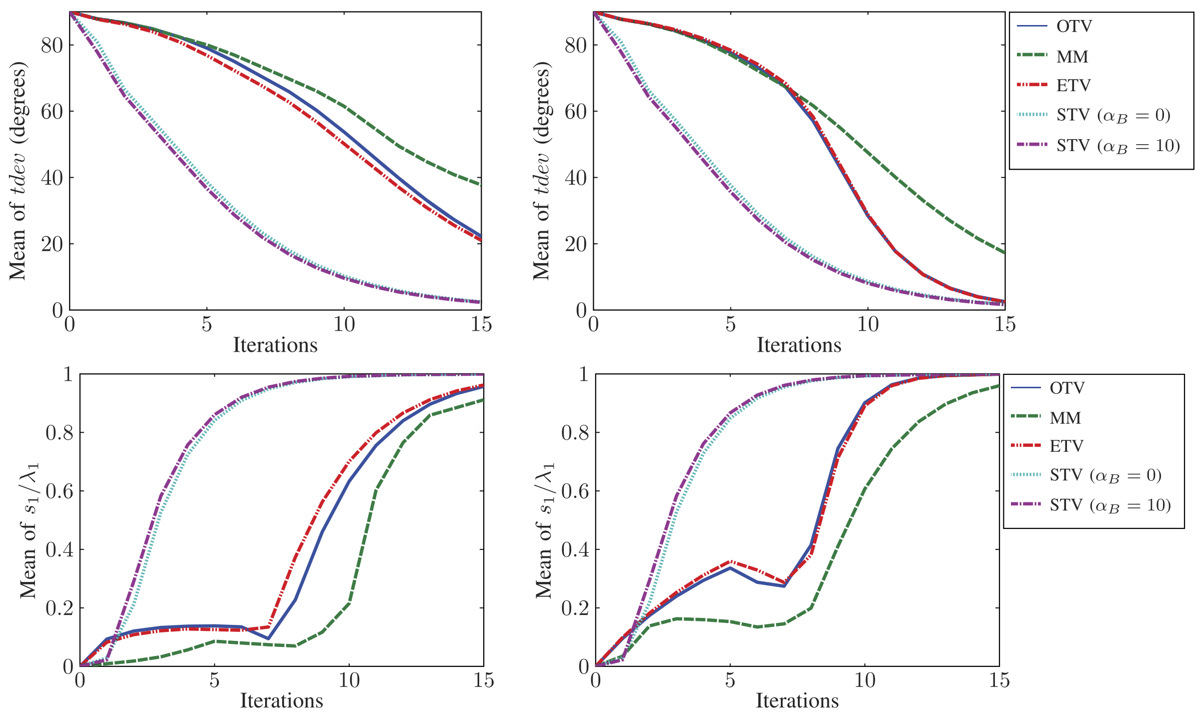

Fig. 11. Experiments on $\alpha_B$ for a flat surface with wrong initialization tensors. Top: Evolution of the mean of $tdev$ with the iterations for $b = 0$ (left) and $b = 10$ (right). Bottom: Evolution of $s_1/\lambda_1$ with the iterations for $b = 0$ (left) and $b = 10$ (right).

scalar product has been used to measure the tensor deviation, $tdev$, of the yielded tensor $\mathrm{T}(\mathbf{p})$ with respect to the ground truth $\mathrm{T_g}(\mathbf{p})$ at every point of the data set. This measure is given by [25], [26]:

$$tdev(\mathbf{p}) = \arccos\left(\frac{\langle \mathrm{T}(\mathbf{p}), \mathrm{T_g}(\mathbf{p})\rangle}{trace(\mathrm{T}(\mathbf{p}))trace(\mathrm{T_g}(\mathbf{p}))}\right), \quad (39)$$

where $\langle \mathrm{A}, \mathrm{B}\rangle = trace(\mathrm{AB}^T)$ is the scalar product between tensors A and B. This measurement has the advantage that it is able to assess differences in orientation and anisotropy of the tensors at the same time.

Fig. 11 shows the evolution with the iterations of the mean of $tdev$ and the mean of the saliency $s_1$ normalized by the largest eigenvalue $\lambda_1$. This figure shows that STV has the best performance among the tested methods. In addition, $\alpha_B$ can be used to accelerate the convergence of STV. If fairly good initialization tensors are available, as is the case from the second iteration onward, $\alpha_B$ should be set to zero in order to avoid the introduction of unnecessary uncertainty. This experiment also shows the power of tensor voting in iterative schemes. Despite the poor initialization, all of the implementations converge to the solution in a few iterations. In addition, it can also be seen in this experiment that the estimation of saliency $s_1$ tends to be improved with the number of iterations. Moreover, values of $b$ greater than zero appear advantageous since all methods perform better in such a condition, especially for OTV and ETV. Furthermore, the experiment also shows that the performances of OTV and ETV are very close, while MM and STV perform differently.

## 6 Concluding Remarks

This paper has proposed two alternative formulations in order to significantly reduce the high computational complexity of the *plate* and *ball* tensor voting. The first formulation makes numerical approximations of the votes which have been derived from an in-depth analysis of the *plate* and *ball* voting processes. The second one proposes simplified equations to calculate votes that are based on the perceptual meaning of the original tensor voting. Both formulations have a complexity of order O(1). This can help broaden the use of tensor voting in more applications.

The numerical approach mimics the original formulation of tensor voting efficiently with a small error. On the other hand, the analytical approach has been found more appropriate for estimating saliencies at a cost of setting two new parameters. In both noisy scenarios and clouds of points with curved surfaces, the simplified tensor voting yields better results by setting the new parameter $\alpha_P$ to 0. In addition, higher values of $\alpha_P$ improve its performance for data sets with flat surfaces. Furthermore, parameter $\alpha_B$ can be used in iterative schemes where tensors are initialized with not too accurate values in order to artificially introduce uncertainty.

Moreover, perceptual interpretations for *stick*, *plate*, and *ball* tensor voting have been established. The *stick* tensor voting and the *stick* component of the *plate* tensor voting are used to reinforce surfaceness, whereas the *plate* components of both the *plate* and *ball* tensor voting are used to boost curveness. Junctionness is only intentionally strengthened by the *ball* component of the *ball* tensor voting.

Future work includes efficient implementations of the *plate* and *ball* votes in the frequency domain and extensions to higher dimensions.

## Acknowledgments

This research has been partially supported by the Spanish Ministry of Science and Technology (project DPI2007-66556-C03-03).

## REFERENCES

[1] G. Guy and G. Medioni, “Inferring Global Perceptual Contours from Local Features,” *Int'l J. Computer Vision,* vol. 20, nos. 1/2, pp. 113-133, 1996.

[2] G. Guy and G. Medioni, “Inference of Surfaces, 3D Curves and Junctions from Sparse, Noisy 3D Data,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 11, pp. 1265-1277, Nov. 1997.

[3] G. Medioni, M.S. Lee, and C.K. Tang, *A Computational Framework for Feature Extraction and Segmentation.* Elsevier Science, 2000.

[4] P. Mordohai and G. Medioni, *Tensor Voting: A Perceptual Organization Approach to Computer Vision and Machine Learning.* Morgan and Claypool Publishers, 2006.

[5] V. Bruce, P.R. Green, and M.A. Georgeson, *Visual Perception: Physiology, Psychology and Ecology,* fourth ed. Psychology Press, 2003.

[6] E. Franken, M. van Almsick, P. Rongen, L. Florack, and B. ter Haar Romeny, “An Efficient Method for Tensor Voting Using Steerable Filters,” *Proc. European Conf. Computer Vision,* vol. IV, pp. 228-240, 2006.

[7] M. Reisert and H. Burkhardt, “Efficient Tensor Voting with 3D Tensorial Harmonics,” *Proc. IEEECS Conf. Computer Vision and Pattern Recognition Workshops,* pp. 1-7, 2008.

[8] P. Mordohai and G. Medioni, “Dimensionality Estimation, Manifold Learning and Function Approximation Using Tensor Voting,” *J. Machine Learning Research,* vol. 11, pp. 411-450, 2010.

[9] D. Lu, H. Zhao, M. Jiang, S. Zhou, and T. Zhou, “A Surface Reconstruction Method for Highly Noisy Point Clouds,” *Proc. Variational, Geometric, and Level Set Methods in Computer Vision,* pp. 283-294, 2005.

[10] C. Min and G. Medioni, “Tensor Voting Accelerated by Graphics Processing Units (GPU),” *Proc. 18th Int'l Conf. Pattern Recognition,* pp. III:1103-1106, 2006.

[11] T.-P. Wu, J. Jia, and C.-K. Tang, “A Closed-Form Solution to Tensor Voting for Robust Parameter Estimation via Expectation-Maximization,” technical report, Hong Kong Univ. of Science Technology, 2009.

[12] T.-P. Wu, S.-K. Yeung, J. Jia, and C.-K. Tang, “Quasi-Dense 3D Reconstruction Using Tensor-Based Multiview Stereo,” *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1482-1489, 2010.

[13] D. Sornette, *Critical Phenomena in Natural Sciences. Chaos, Fractals, Selforganization and Disorder: Concepts and Tools,* second ed. Springer, 2006.

[14] T. Marchant, “Scale Invariance and Similar Invariance Conditions for Bankruptcy Problems,” *Social Choice and Welfare,* vol. 31, no. 4, pp. 693-707, 2008.

[15] K. Belbahri, “Scale Invariant Operators and Combinatorial Expansions,” *Advances in Applied Math.,* vol. 45, no. 4, pp. 548-563, 2010.

[16] L. Loss, G. Bebis, M. Nicolescu, and A. Skurikhin, “An Iterative Multi-Scale Tensor Voting Scheme for Perceptual Grouping of Natural Shapes in Cluttered Backgrounds,” *Computer Vision and Image Understanding,* vol. 113, no. 1, pp. 126-149, 2009.

[17] W.S. Tong, C.K. Tang, P. Mordohai, and G. Medioni, “First Order Augmentation to Tensor Voting for Boundary Inference and Multiscale Analysis in 3D,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 594-611, May 2004.

[18] P.W. Bridgman, *Dimensional Analysis.* Yale Univ. Press, 1922.

[19] E.W. Andrews and L.J. Gibson, “The Role of Cellular Structure in Creep of Two-Dimensional Cellular Solids,” *Materials Science and Eng.: A,* vol. 303, nos. 1/2, pp. 120-126, 2001.

[20] J.T. Ricketts, M.K. Loftin, and F.S. Merritt, *Standard Handbook for Civil Engineers,* fifth ed. McGraw-Hill Professional, 2003.

[21] *The CRC Handbook of Mechanical Engineering*. F. Kreith, and D.Y. Goswami eds., second ed. CRC Press, 2005.

[22] D.J. Magee, *Orthopedic Physical Assessment,* fifth ed. Saunders, 2008.

[23] Z. Miller and M.B. Fuchs, “Effect of Trabecular Curvature on the Stiffness of Trabecular Bone,” *J. Biomechanics,* vol. 38, no. 9, pp. 1855-1864, 2005.

[24] S. Fischer, P. Bayerl, H. Neumann, G. Cristobal, and R. Redondo, “Iterated Tensor Voting and Curvature Improvement,” *J. Signal Processing,* vol. 87, no. 11, pp. 2503-2515, 2007.

[25] T. Peeters, P. Rodrigues, A. Vilanova, and B. ter Haar Romeny, “Analysis of distance/similarity measures for Diffusion Tensor Imaging,” *Visualization and Processing of Tensor Fields: Advances and Perspectives,* Springer, pp. 113-138, 2009.

[26] L. Jonasson, X. Bresson, P. Hagmann, O. Cuisenaire, R. Meuli, and J. Thiran, “White Matter Fiber Tract Segmentation in DT-MRI Using Geometric Flows,” *Medical Image Analysis,* vol. 9, pp. 223-236, 2005.

**Rodrigo Moreno** received the BS and MS degrees in computer science from the University of Los Andes, Bogotá, Colombia, in 1995 and 1997, respectively, the BS degree in electrical engineering from the National University of Colombia, Bogotá, in 1995, the MS degree in organizations management from the University of Quebec, Chicoutimi, Canada, in 2006, and the diploma in advanced studies and the PhD degree from the Polytechnic University of Catalonia, Barcelona, Spain, in 2007 and 2010, respectively. Between 1997 and 1999, he was involved in engineering projects in Colombian companies. In 1999, he joined St. Martin University, Bogotá, Colombia, where he was the head of the Department of Computer Science from 2001 to 2006. In 2006, he joined the Intelligent Robotics and Computer Vision Group at Rovira I Virgili University, Tarragona, Spain. He is currently a postdoctoral researcher at the Center for Medical Image Science and Visualization (CMIV) and the Department of Medical and Health Sciences (IMH) at Linköping University, Sweden. His research interests include image analysis, computer vision, machine learning, and biomedical imaging. He is a member of the IEEE.

**Miguel Angel Garcia** received the BS, MS, and PhD degrees in computer science from the Polytechnic University of Catalonia, Barcelona, Spain, in 1989, 1991, and 1996, respectively. He joined the Department of Software at the Polytechnic University of Catalonia in 1996 as an assistant professor. From 1997 to 2006, he was with the Department of Computer Science and Mathematics at Rovira i Virgili University, Tarragona, Spain, where he was the head of Intelligent Robotics and Computer Vision Group. In 2006, he joined the Department of Informatics Engineering at Autonomous University of Madrid, Spain, and in 2010, he joined the Department of Electronic and Commuications Technology at the Autonomous University of Madrid, Spain, where he is currently an associate professor and a member of the Video Processing and Understanding Lab. His research interests include image processing, computer vision, 3D modeling, and mobile robotics. He is a member of the IEEE.

**Domenec Puig** received the MS and PhD degrees in computer science from the Polytechnic University of Catalonia, Barcelona, Spain, in 1992 and 2004, respectively. In 1992, he joined the Department of Computer Science and Mathematics at Rovira i Virgili University, Tarragona, Spain, where he is currently an associate professor. Since July 2006, he has been the head of the Intelligent Robotics and Computer Vision Group at the same university. His research interests include image processing, texture analysis, perceptual models for image analysis, scene analysis, and mobile robotics.

**Luis Pizarro** received the BE and MS degrees in informatics engineering from the Technical University Federico Santa Maria, Chile, in 2003 and the PhD degree from Saarland University, Saarbrücken, Germany, in 2011. From 2001 to 2003, he worked as a research assistant at the Technical University Federico Santa Maria, Chile, and was a consulting engineer at Anglo American Chile. In 2004, he spent five months as a research trainee at INRIA Sophia-Antipolis, France. From 2004 to 2005, he worked as a researcher at the Mining and Metallurgy Innovation Institute, IM2, Chile. From 2006 to 2010, he was working toward the PhD degree in computer science at Saarland University, Saarbrücken, Germany. Since October 2009, he has been with the Department of Computing and the National Heart and Lung Institute at Imperial College, London, United Kingdom, as a postdoctoral researcher. His research interests include image analysis, computer vision, machine learning, and biomedical imaging.

**Bernhard Burgeth** received the diploma and doctoral degree in mathematics from the University of Erlangen-Nürnberg, Germany, in 1991 and 1996, respectively. He worked as a research assistant at the University of Erlangen-Nürnberg, and as a researcher (DFG research grant) at McGill University, Montreal, Canada, at the Technical University Eindhoven, The Netherlands, at the Karlsruhe Research Center, Karlsruhe, Germany, and as an assistant professor in the Mathematical Image Analysis Groups at Saarland University, Saarbrücken, Germany. After receiving the habilitation in mathematics in 2009, he joined the Faculty of Mathematics at Saarland University, Saarbrücken, Germany, as an associate professor. His research interests include partial differential equations and their applications to the processing of tensor fields and medical image analysis in general.

**Joachim Weickert** received the graduation and PhD degrees from the University of Kaiserslautern, Germany, in 1991 and 1996. He is a professor of mathematics and computer science at Saarland University, Saarbrücken, Germany, where he heads the Mathematical Image Analysis Group. He worked as a postdoctoral researcher at the University Hospital of Utrecht, The Netherlands and the University of Copenhagen, Denmark, and as an assistant professor at the University of Mannheim, Germany. He has developed many models and efficient algorithms for image processing and computer vision using partial differential equations and optimization principles. In particular, he has contributed to diffusion filtering, optical flow computation, processing of tensor fields, and image compression. His scientific work covers approximately 230 refereed publications. He has served on the editorial boards of nine international journals and given more than 130 invited talks. In 2010, he has received the Gottfried Wilhelm Leibniz Prize, which is the highest German science award.