Flexible Segmentation and Smoothing of DT-MRI Fields Through a Customizable Structure Tensor

Thomas Schultz[1], Bernhard Burgeth[2], and Joachim Weickert[2]

[1] MPI Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
schultz@mpi-inf.mpg.de
[2] Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science, Saarland University, 66041 Saarbrücken, Germany
{burgeth, weickert}@mia.uni-saarland.de

Abstract. We present a novel structure tensor for matrix-valued images. It allows for user defined parameters that add flexibility to a number of image processing algorithms for the segmentation and smoothing of tensor fields. We provide a thorough theoretical derivation of the new structure tensor, including a proof of the equivalence of its unweighted version to the existing structure tensor from the literature. Finally, we demonstrate its advantages for segmentation and smoothing, both on synthetic tensor fields and on real DT-MRI data.

1 Introduction

In recent years, second-order tensor fields have received increasing attention. This is partly due to the now widely-used diffusion tensor magnetic resonance imaging (DT-MRI) modality that uses a diffusion tensor in each voxel to describe the self-diffusion of water molecules [1]. The methods presented in this paper have been developed with an eye on the processing of DT-MRI data, but can be used wherever real-valued, symmetric 3×3 matrix fields arise.

To approach the smoothing and segmentation of such fields, Feddern et al. [2,3] have proposed an extension of some well-known curvature-based partial differential equations (PDEs), like mean curvature motion and active contour models, to the tensor case. Other methods for tensor image regularization [4,5,6,7] and segmentation [8,9,10,11] have been suggested. However, all of them use a fixed distance measure on the tensors. Therefore, none of the existing approaches allow the user to emphasize the relevance of particular properties of the diffusion tensor (i.e., overall diffusivity, anisotropy, and orientation) for a given application.

Feddern et al. derive a structure tensor for tensor-valued images. Subsequently, they define generalized level lines as integral lines of its minor eigenvector field and generalized gradient magnitude as the structure tensor trace.

Our present work uses a decomposition of the tensor field gradient which has been suggested by Kindlmann [12] to replace this structure tensor with a new formulation that has user defined parameters. When they are all set to

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4291, pp. 455–464, 2006.

one, the new formulation is equivalent to the previous one. However, we will show that the option to weight its individual terms is crucial for some specific applications.

The paper is organized as follows: Section 2 revises the previous work by Kindlmann which serves as the basis of our re-formulation, presented in Section 3. A proof of the equivalence to the previous structure tensor is given in Section 4. In Section 5, example applications on synthetic and real DT-MRI data follow, before Section 6 concludes the paper.

2 Projected Tensor Gradients

Let Sym_3 denote the six-dimensional vector space of symmetric, real-valued 3×3 matrices and let $\mathbf{D}$ be a field of Sym_3 matrices over $\mathbb{R}^3$. Then, the gradient $\nabla\mathbf{D}$ of the field is a $3 \times 3 \times 3$ third-order tensor which we will index such that $(\nabla\mathbf{D})_{ijk} = \frac{\partial D_{jk}}{\partial x_i}$. Thus, $\nabla\mathbf{D}$ can be thought of as a three-vector of second-order tensors, expressing the partial derivatives of $\mathbf{D}$ in each image direction.

Kindlmann's contribution in [12] is to decompose $\nabla\mathbf{D}$ into parts that correspond to changes in three tensor invariants which cover changes in shape, as well as the parts of $\nabla\mathbf{D}$ that correspond to rotations around each eigenvector.

The invariants he uses to describe tensor shape are derived from the moments of the eigenvalues λ_1, λ_2, and λ_3. In the context of DT-MRI, $\mu_1 := \frac{1}{3}\sum_i \lambda_i$, the eigenvalue mean, is a measure of *bulk diffusivity.* The eigenvalue variance $\mu_2 := \frac{1}{3}\sum_i (\lambda_i - \mu_1)^2$ measures *diffusion anisotropy.* The eigenvalue skewness $\alpha_3 := \mu_3 / \sqrt{\mu_2^3}$ (with $\mu_3 := \frac{1}{3}\sum_i (\lambda_i - \mu_1)^3$) reflects the *type of anisotropy.* It is a dimensionless quantity with range $\left[-1/\sqrt{2}, 1/\sqrt{2}\right]$, where $\alpha_3 = -1/\sqrt{2}$ indicates a perfectly planar tensor and $\alpha_3 = 1/\sqrt{2}$ a perfectly linear one.

If we consider these invariants as scalar-valued functions over Sym_3, their gradient is a map from Sym_3 to Sym_3. We will denote this gradient of an invariant J as the *invariant gradient* $\boldsymbol{\nabla} J$, marked by a boldface $\boldsymbol{\nabla}$.

Kindlmann decomposes the tensor field gradient $\nabla\mathbf{D}$ by projecting it onto the invariant gradients. In order to avoid undesired scaling in this step, he first normalizes $\boldsymbol{\nabla} J$ with respect to the tensor scalar product $\mathbf{A} : \mathbf{B} := \sum_{i,j} a_{ij} b_{ij}$. Let $\|\mathbf{D}\| := \sqrt{\mathbf{D} : \mathbf{D}}$ denote the associated Frobenius norm. If $\|\boldsymbol{\nabla} J\| > 0$, its normalized version is $\hat{\boldsymbol{\nabla}} J := \boldsymbol{\nabla} J / \|\boldsymbol{\nabla} J\|$. However, $\boldsymbol{\nabla}\mu_2$ vanishes when $\mu_2 = 0$, and $\boldsymbol{\nabla}\alpha_3$ vanishes when α_3 reaches one of its extrema as well as in the isotropic case ($\mu_2 = 0$), for which α_3 is undefined.

Kindlmann derives expressions for the invariant gradients[1] and suggests auxillary constructs to ensure that $\hat{\boldsymbol{\nabla}}\mu_1$, $\hat{\boldsymbol{\nabla}}\mu_2$ and $\hat{\boldsymbol{\nabla}}\alpha_3$ are always orthonormal and span the space of changes in tensor shape, even if some of the underlying invariant gradients are undefined. As a consequence, the directions of $\hat{\boldsymbol{\nabla}}\mu_2$ and $\hat{\boldsymbol{\nabla}}\alpha_3$ are arbitrary and exchangeable when $\mu_2 = 0$.

[1] The exact formulas are not required to understand this paper. The interested reader can find them in [12, p. 73].

The *projected gradient* ∇J is obtained by taking the tensor scalar product of $\hat{\mathbf{\nabla}} J$ with each of the three Sym_3 tensors within $\nabla \mathbf{D}$:

$$(\nabla J)_i := \sum_{j,k} (\hat{\mathbf{\nabla}} J)_{jk} (\nabla \mathbf{D})_{ijk} \tag{1}$$

∇J is a vector in $\mathbb{R}^3$, expressing for each of the three spatial directions the amount of tensor change that corresponds to changes in the invariant J.

To cover changes in orientation, Kindlmann calculates *rotation tangents* ($\mathbf{\Phi}_1$, $\mathbf{\Phi}_2, \mathbf{\Phi}_3 \in Sym_3$) as the changes in $\mathbf{D}$ caused by infinitesimal rotations around its eigenvectors. Subsequently, the effect of finite rotations is approximated by adding some scalar multiple of the normalized tangents $\hat{\mathbf{\Phi}}_i$ [12, p. 87].

Kindlmann shows that the rotation tangents are orthogonal both to each other and to the invariant gradients defined above. The rotation tangents are used in the same manner as before to define projected tensor field gradients $\nabla \phi_i$.

3 The New Structure Tensor

The structure tensor used by Feddern et al. [3] is defined as follows:

$$\mathbf{J}_{\text{orig}}(\nabla \mathbf{D}_\sigma) := \sum_{i=1}^{3} \sum_{j=1}^{3} \nabla (\mathbf{D}_\sigma)_{i,j} \; \nabla^T (\mathbf{D}_\sigma)_{i,j} \tag{2}$$

The indexing with σ indicates that the gradient is calculated from a Gaussian-smoothed version of the original tensor field, to allow noise-scale pre-processing. This definition can be considered an extension of Di Zenzo's approach for vector-valued images [13]. Its advantages are that it makes use of the full tensor information, it is rotationally invariant, and it has proven to work well in practice.

Equation (2) is based on the gradients of the nine tensor channels. Our alternative approach uses the six projected gradients from Section 2 instead. As these are physically meaningful, their individual influence can reasonably be weighted via user coefficients ${w_*}^2$.

However, we now need to handle cases in which the invariant gradients become ill-defined. We will tackle this problem with functions ψ_* that calculate effective weights from the user-controlled parameters w_*.

For isotropic tensors ($\mu_2 = 0$), $\hat{\mathbf{\nabla}} \mu_2$ and $\hat{\mathbf{\nabla}} \alpha_3$ are arbitrary and exchangeable. We reason that in the case of perfect isotropy, it does not make sense to speak of changes in the *type* of anisotropy, so all tensor change that gets projected on the span of $\hat{\mathbf{\nabla}} \mu_2$ and $\hat{\mathbf{\nabla}} \alpha_3$ should be attributed to changes in variance.

To make a smooth transition towards this case, we use the fractional anisotropy (FA), a common anisotropy measure in the context of DT-MRI [14]:

$$\text{FA} := \sqrt{\frac{3}{2}} \frac{\|\mathbf{D} - \mu_1 \mathbf{I}\|}{\|\mathbf{D}\|} = \frac{3}{\sqrt{2}} \frac{\sqrt{\mu_2}}{\|\mathbf{D}\|} \tag{3}$$

[2] With the asterisk $*$, we refer to all possible indices of a variable.

where $\mathbf{I}$ is the unit matrix. It is clear from Equation (3) that $\mathrm{FA} = 0$ if $\mu_2 = 0$, so we define ψ_{α_3} such that it tends to w_{μ_2} as $\mathrm{FA} \to 0$:

$$\psi_{\alpha_3}(\mathrm{FA}; w_{\mu_2}, w_{\alpha_3}) := \begin{cases} w_{\alpha_3} & \text{if } \mathrm{FA} > \epsilon \\ \left(\frac{\mathrm{FA}}{\epsilon} - 1\right)^2 w_{\mu_2} + \left[1 - \left(\frac{\mathrm{FA}}{\epsilon} - 1\right)^2\right] w_{\alpha_3} & \text{else} \end{cases}$$

Making a smooth transition requires to introduce a parameter ϵ that defines the threshold starting from which we fully rely on the gradient direction $\hat{\boldsymbol{\nabla}}\alpha_3$. In our experiments, a value of $\epsilon = 0.1$ worked well.

Further singularities occur when dealing with changes in orientation. Since the two ill-defined eigenvectors of a perfectly linear ($\alpha_3 = 1/\sqrt{2}$) or planar ($\alpha_3 = -1/\sqrt{2}$) tensor are exchangeable, the effect of rotating the tensor around any of them may just as well be attributed to the other one. Thus, the ψ_{ϕ_i} are designed to let them share the total amount of rotation around any of them by averaging their weights in the limit case:

$$\psi_{\phi_1}(\mathrm{FA}, \alpha_3; w_{\phi_1}, w_{\phi_2}) = \begin{cases} w_{\phi_1} & \text{if } \mathrm{FA} = 0 \text{ or } \alpha_3 > 0 \\ \left(1 - \alpha_3^2\right) w_{\phi_1} + \alpha_3^2 w_{\phi_2} & \text{else} \end{cases}$$

$$\psi_{\phi_2}(\mathrm{FA}, \alpha_3; w_{\phi_1}, w_{\phi_2}, w_{\phi_3}) = \begin{cases} w_{\phi_2} & \text{if } \mathrm{FA} = 0 \\ \left(1 - \alpha_3^2\right) w_{\phi_2} + \alpha_3^2 w_{\phi_1} & \text{if } \mathrm{FA} > 0 \text{ and } \alpha_3 < 0 \\ \left(1 - \alpha_3^2\right) w_{\phi_2} + \alpha_3^2 w_{\phi_3} & \text{else} \end{cases}$$

$$\psi_{\phi_3}(\mathrm{FA}, \alpha_3; w_{\phi_2}, w_{\phi_3}) = \begin{cases} w_{\phi_3} & \text{if } \mathrm{FA} = 0 \text{ or } \alpha_3 < 0 \\ \left(1 - \alpha_3^2\right) w_{\phi_3} + \alpha_3^2 w_{\phi_2} & \text{else} \end{cases}$$

With these definitions, the new structure tensor reads:

$$\begin{aligned} \mathbf{J}_{\mathrm{new}}(\nabla\mu_1, \nabla\mu_2, \nabla\alpha_3, \nabla\phi_1, \nabla\phi_2, \nabla\phi_3; \mathrm{FA}, \alpha_3) := \\ w_{\mu_1} \nabla\mu_{1,\sigma} \nabla\mu_{1,\sigma}^T + w_{\mu_2} \nabla\mu_{2,\sigma} \nabla\mu_{2,\sigma}^T + \\ \psi_{\alpha_3}(\mathrm{FA}; w_{\mu_2}, w_{\alpha_3}) \nabla\alpha_{3,\sigma} \nabla\alpha_{3,\sigma}^T + \\ \sum_{i=1}^{3} \psi_{\phi_i}(\mathrm{FA}, \alpha_3; w_{\phi_1}, w_{\phi_2}, w_{\phi_3}) \nabla\phi_{i,\sigma} \nabla\phi_{i,\sigma}^T \end{aligned} \tag{4}$$

It is rotationally invariant for arbitrary sets of weights and uses the full tensor information when all weights are non-zero.

4 Equivalence to the Previous Structure Tensor

In its unweighted form ($w_* = 1$), the new structure tensor $\mathbf{J}_{\mathrm{new}}$ is equivalent to the one used by Feddern et al., $\mathbf{J}_{\mathrm{orig}}$. This fact ensures that our new structure tensor has all the desirable properties of the established one.

Proof. We can write element (i, j) of the original structure tensor $\mathbf{J}_{\text{orig}}$ (2) in terms of the tensor scalar product:

$$(\mathbf{J}_{\text{orig}})_{ij} = \nabla \mathbf{D}_i : \nabla \mathbf{D}_j \tag{5}$$

Note that the normalized invariant gradients and rotation tangents defined by Kindlmann form an orthonormal basis of Sym_3. If we denote the kth of these basis vectors by $\mathbf{B}^k$, we can write element (i, j) of the new structure tensor $\mathbf{J}_{\text{new}}$ (4) correspondingly as

$$(\mathbf{J}_{\text{new}})_{ij} = \sum_k \left(\nabla \mathbf{D}_i : \mathbf{B}^k\right) \left(\nabla \mathbf{D}_j : \mathbf{B}^k\right) \tag{6}$$

For the sake of simplicity, we are going to embed the tensors isometrically in $\mathbb{R}^6$. This can be done by using the six non-redundant tensor channels as components of the vector, where non-diagonal elements are multiplied by $\sqrt{2}$.

Let $\nabla \mathbf{d}_i$ and $\mathbf{b}^k$ be the embedded versions of $\nabla \mathbf{D}_i$ and $\mathbf{B}^k$, respectively. Then, we can re-write Equation (6) and reorder the terms as follows:

$$\begin{aligned}(\mathbf{J}_{\text{new}})_{ij} &= \sum_k \left(\nabla \mathbf{d}_i \cdot \mathbf{b}^k\right) \left(\nabla \mathbf{d}_j \cdot \mathbf{b}^k\right) \\ &= \sum_l \left(\sum_k (b_l^k)^2\right) \nabla d_{il} \nabla d_{jl} + \sum_{\substack{l,m \\ l \neq m}} \left(\sum_k b_l^k b_m^k\right) \nabla d_{il} \nabla d_{jm}\end{aligned} \tag{7}$$

If we arrange the $\mathbf{b}^k$ as rows of a matrix, the resulting matrix is orthogonal, because the $\mathbf{B}^k$ were orthonormal and our mapping preserved the scalar product. Thus, the column vectors of the matrix will also be orthonormal: $\sum_k b_l^k b_m^k = \delta_{lm}$.

With this result, Equation (7) reduces to $\nabla \mathbf{d}_i \cdot \nabla \mathbf{d}_j$, which is by our definition equivalent to Equation (5). □

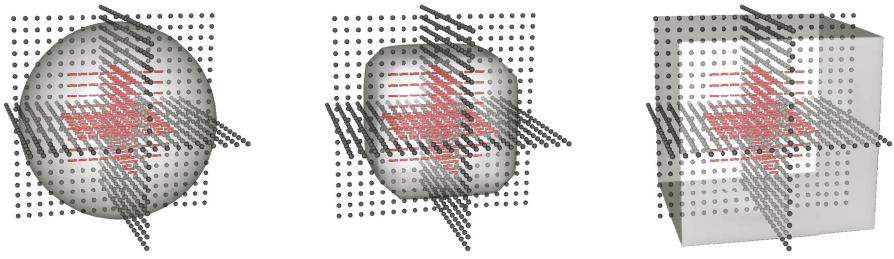
5 Applications

5.1 Application to Segmentation

We will now demonstrate the advantages of the weightable structure tensor in the context of a geodesic active contour model [15,16] for interactive segmentation. This model allows the user to provide the approximate position and shape of the object that is to be segmented. Consequently, the contour moves to the exact boundary of the object, based on edge information from the image.

For tensor-valued images $\mathbf{f}$, Feddern et al. [3] suggest to use the structure tensor trace as an edge detector. A simple implementation of the segmentation model is then given by embedding the initial contour as a zero level set into a function u_0 (via a distance transformation) and evolving it under the PDE

$$\partial_t u = |\nabla u| \operatorname{div} \left(g(\operatorname{tr} \mathbf{J}(\nabla \mathbf{f}_\sigma)) \frac{\nabla u}{|\nabla u|} \right) \tag{8}$$



(a) Initialization of the active surface, between two edges

(b) Without weighting, the inner box gets segmented

(c) Setting $w_{\mu_1} := 10$ leads to a segmentation of the outer box

Fig. 1. Active surface segmentation of a synthetic data set ($t = 500$). Varying the weights draws the segmentation towards specific types of edges in the data.

which simultaneously regularizes the surface and draws it towards edges in the image $\mathbf{f}$. g is a non-increasing stopping function. In our experiments, we used the following diffusivity [17, p. 114]:

$$g(s^2) := \begin{cases} 1 & \text{if } s^2 = 0 \\ 1 - \exp\left(\frac{-3.31488}{(s^2/\lambda^2)^4}\right) & \text{if } s^2 > 0 \end{cases} \tag{9}$$

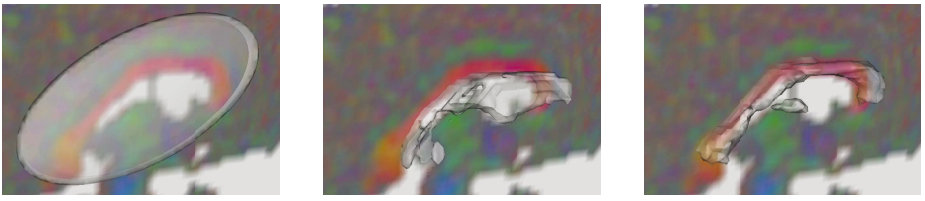
The evolution of Equation (8) is stopped at a time t, when u no longer changes significantly. The segmentation result is extracted as the zero level set of u. Unlike the version presented in [3], our implementation of (8) works in 3D, so we will refer to it as an *active surface model*.

We first applied this method to the synthetic dataset shown in Figure 1. It consists of two nested boxes of different materials: The tensors in the outer box are isotropic, but have the same trace as the linear ones in the inner box. Thus, two types of edges arise: A change of tensor trace between air and the outer box and a change of anisotropy between the outer and the inner box.

Figure 1 shows the setup using superquadric glyphs [18] on three orthogonal slices. When we initialize the active surface to a sphere that lies between the two boundaries (Figure 1(a)), the unweighted structure tensor leads to a segmentation of the inner box (Figure 1(b)). By increasing the weight of changes in trace ($w_{\mu_1} := 10$), we can guide the segmentation to the outer box.

We also segmented the corpus callosum (CC), a major white-matter structure, in a real DT-MRI dataset. Figure 2(a) shows the initialization of the active surface to an ellipsoid, superimposed on a sagittal slice of the data in the standard xyz-RGB eigenvector color coding. It encompasses the structure of interest, which appears as red in the color image, or as dark in halftones.

The original, unweighted structure tensor basically leads to a segmentation of the ventricle (Figure 2(b), the ventricle is shown in white). The CC is distinguished from its neighborhood by its anisotropy ($w_{\mu_2} := 1$) and major



(a) Initialization of the active surface

(b) The unweighted structure tensor segments the ventricle

(c) The weights allow to capture the corpus callosum

Fig. 2. Active surface segmentation of a DT-MRI data set ($t = 1200$). Weighting the new structure tensor allows to specify the structure of interest.

eigenvector orientation ($w_{\phi_2} := w_{\phi_3} := 0.2$). Setting all other weights zero allows for a plausible segmentation of this structure, shown in Figure 2(c).

A small structure inferior to the CC also got segmented. Although it differs from the CC by its global orientation, we cannot differentiate it in our model, which only considers local variations in tensor value. In practice, a simple connected component analysis would be sufficient to remove the undesired island.

For the numerical implementation, we have used the additive operator splitting (AOS) scheme as described by Weickert and Kühne [19]. It allowed us to solve Equation (8) on the real DT-MRI dataset (grid size $74 \times 95 \times 80$) for $t = 1200$ in 40 seconds on a 2 GHz dual-core Athlon 64 processor; the synthetic examples took two seconds each.

5.2 Application to Smoothing

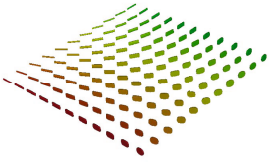
Feddern et al. [3] also generalize the *self-snakes* smoothing process initially proposed by Sapiro [20] to the tensor-valued case. In 3D, it can be written as a system of six coupled PDEs for the individual tensor channels,

$$\partial_t u_{i,j} = g(\operatorname{tr} \mathbf{J}(\nabla \mathbf{u}_\sigma)) \left(\partial_{\mathbf{vv}} u_{i,j} + \partial_{\mathbf{ww}} u_{i,j}\right) + \nabla^T (g(\operatorname{tr} \mathbf{J}(\nabla \mathbf{u}_\sigma))) \nabla u_{i,j} \qquad (10)$$

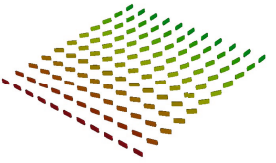
with $u_{i,j}(x, y, z, 0) = f_{i,j}(x, y, z)$ as the initial condition. The vectors $\mathbf{v}$ and $\mathbf{w}$ denote the minor and medium eigenvectors of the structure tensor $\mathbf{J}$, respectively. Equation (10) leads to a smoothing along the plane spanned by $\mathbf{v}$ and $\mathbf{w}$; this is analogous to the smoothing along level surfaces in scalar-valued mean curvature motion (MCM). The diffusivity function g stops the smoothing process at important image features.

Our new structure tensor allows to steer this process. We first demonstrate this with the synthetic dataset from Figure 3(a), in which the tensors vary continuously in shape (from linear to planar) and orientation (rotation by 90°). In this example, we set $g := 1$ to obtain purely MCM-like smoothing.

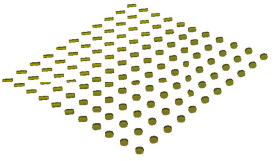
Setting the weights of $\mathbf{J}_{\text{new}}$ allows to specify which features we would like to preserve. If we set $w_{\phi_*} := 1$ (all other weights zero), the result in Figure 3(b) shows that the shapes are completely averaged at $t = 250$, while the orientation



(a) A tensor field that varies in shape and orientation



(b) Result of orientation-preserving smoothing



(c) Result of shape-preserving smoothing

Fig. 3. Mean Curvature Motion-like smoothing of a synthetic data set ($t = 250$). Customizing the structure tensor allows to select which features should be preserved.

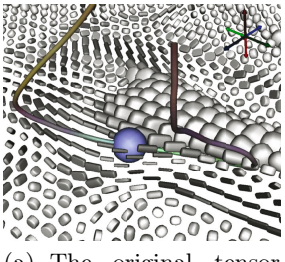
has been preserved. On the other hand, we can set only $w_{\mu_2} := w_{\alpha_3} := 1$ to preserve shape and average the orientation, as shown in Figure 3(c).

In our experiments on real DT-MRI data, we found that for many smoothing tasks, $\mathbf{J}_{\text{orig}}$ already works well in its default configuration. Figure 4 presents a situation in which its behaviour can still be improved by changing the weights.

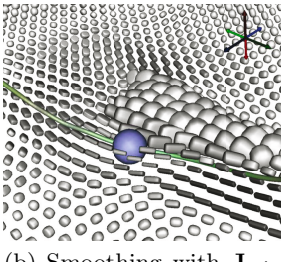
Figure 4(a) shows two touching fiber tracts. One of them lies in the depicted sagittal plane, the other one (shown by an integral curve of major eigenvectors, started at the position marked by the ball) touches the plane almost tangentially. It is a delicate task to smooth the dataset without mixing these two tracts.

Figure 4(b) shows that after applying self-snakes with $\mathbf{J}_{\text{orig}}$, the tangentially touching tract has been lost at $t = 5$. To preserve the directions of both tracts, we configured $\mathbf{J}_{\text{new}}$ to concentrate on orientation ($w_{\phi_*} := 1$) and to prevent influence of the nearby ventricle ($w_{\mu_1} := 0.1$, all other weights zero). Figure 4(c) shows that this configuration makes it possible to preserve both tracts.

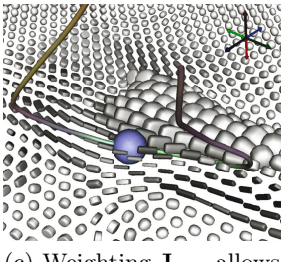
In this experiment, we set g to the diffusivity function (9). As the trace of different structure tensors does not in general lie on the same order of magnitude, we select the contrast parameter λ^2 as 0.01 times the maximum structure tensor trace in the first iteration and then keep it constant.



(a) The original tensor field with a tracked fiber



(b) Smoothing with $\mathbf{J}_{\text{orig}}$ changes the fiber significantly



(c) Weighting $\mathbf{J}_{\text{new}}$ allows to preserve the fiber

Fig. 4. Emphasizing orientation using the new structure tensor can help to preserve the direction of fibers. Here, the glyphs are shaded to indicate the degree of linearity.

Solving Equation (10) with a simple explicit scheme took 30 seconds on the real dataset ($t = 5$) and four seconds in the synthetic case ($t = 250$).

6 Conclusion

We have presented methods for flexible smoothing and segmentation of matrix-valued images. User-controllable options to concentrate on specific features in the high-dimensional data are introduced by integrating Kindlmann's decomposition of the tensor field gradient [12] into the PDE framework of Feddern et al. [3]. We found solutions for cases in which Kindlmann's invariant gradients become ill-defined and proved that the new structure tensor in its unweighted form is equivalent to the previous one. Finally, we successfully demonstrated the advantages of our re-formulation, both on synthetic and on real data.

Future work may use diffusion processes to preprocess DT-MRI data for fiber tracking.

Acknowledgements

We would like to thank Holger Theisel for discussions about the present work.

The human brain dataset[3] is courtesy of Gordon Kindlmann at the Scientific Computing and Imaging Institute, University of Utah, and Andrew Alexander, W. M. Keck Laboratory for Functional Brain Imaging and Behavior, University of Wisconsin, Madison.

Visualization has been done using a modified version of the BioTensor program [21], developed at the University of Utah. Handling the dataset in this framework made it necessary to downsample it by factor two in each dimension.

References

1. Pierpaoli, C., Jezzard, P., Basser, P., Barnett, A., Di Chiro, G.: Diffusion tensor MR imaging of the human brain. Radiology **201** (1996) 637–648
2. Feddern, C., Weickert, J., Burgeth, B.: Level-set methods for tensor-valued images. In Faugeras, O.D., Paragios, N., eds.: Proc. Second IEEE Workshop on Geometric and Level Set Methods in Computer Vision, Nice, France, INRIA (2003) 65–72
3. Feddern, C., Weickert, J., Burgeth, B., Welk, M.: Curvature-driven PDE methods for matrix-valued images. International Journal of Computer Vision **69** (2006) 93–107
4. Chefd'hotel, C., Tschumperlé, D., Deriche, R., Faugeras, O.: Constrained flows of matrix-valued functions: Application to diffusion tensor regularization. In Heyden, A., Sparr, G., Nielsen, M., eds.: Computer vision - ECCV 2002. Volume 2350 of LNCS., Springer (2002) 251–265
5. Welk, M., Feddern, C., Burgeth, B., Weickert, J.: Median filtering of tensor-valued images. In Michaelis, B., Krell, G., eds.: Pattern Recognition. Volume 2781 of LNCS. (2003) 17–24

[3] `http://www.sci.utah.edu/~gk/DTI-data/`

6. Westin, C.F., Knutsson, H.: Tensor field regularization using normalized convolution. In Moreno-Diaz, R., Pichler, F., eds.: Computer Aided Systems Theory - EUROCAST 2003. Volume 2809 of LNCS., Springer (2004) 564–572
7. Rodríguez-Florido, M.A., Ruiz-Alzola, J., Westin, C.F.: DT-MRI regularization using anisotropic tensor field filtering. In: Proc. 2004 IEEE International Symposium on Biomedical Imaging, Arlington, VA, USA (2004) 336–339
8. Wiegell, M.R., Tuch, D., Larsson, H.B., Wedeen, V.J.: Automatic segmentation of thalamic nuclei from diffusion tensor magnetic resonance imaging. NeuroImage **19** (2003) 391–401
9. Rousson, M., Lenglet, C., Deriche, R.: Level set and region based surface propagation for diffusion tensor MRI segmentation. In Sonak, M., Kakadiaris, I.A., Kybic, J., eds.: Computer vision and mathematical methods in medical and biomedical image analysis. Volume 3117 of LNCS., Springer (2004) 123–134
10. Wang, Z., Vemuri, B.C.: An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation. In: Proc. 2004 IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2004) 228–233
11. Jonasson, L., Bresson, X., Hagmann, P., Cuisenaire, O., Meuli, R., Thiran, J.P.: White matter fiber tract segmentation in DT-MRI using geometric flows. Medical Image Analysis **9** (2005) 223–236
12. Kindlmann, G.L.: Visualization and Analysis of Diffusion Tensor Fields. PhD thesis, School of Computing, University of Utah (2004) `http://www.cs.utah.edu/research/techreports/2004/pdf/UUCS-04-014.pdf`.
13. Di Zenzo, S.: A note on the gradient of a multi-image. Computer Vision, Graphics, and Image Processing **33** (1986) 116–125
14. Basser, P.J., Pierpaoli, C.: Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. Journal of Magnetic Resonance **B** (1996) 209–219
15. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. Fifth International Conference on Computer Vision. (1995) 694–699
16. Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, A.: Gradient flows and geometric active contour models. In: Proc. Fifth International Conference on Computer Vision. (1995) 810–815
17. Weickert, J.: Anisotropic Diffusion in Image Processing. Teubner, Stuttgart (1998)
18. Kindlmann, G.L.: Superquadric tensor glyphs. In: Joint Eurographics/IEEE Symposium on Visualization. (2004) 147–154
19. Weickert, J., Kühne, G.: Fast methods for implicit active contour models. In Osher, S., Paragios, N., eds.: Geometric Level Set Methods in Imaging, Vision, and Graphics. Springer, New York (2003) 43–58
20. Sapiro, G.: Vector (self) snakes: a geometric framework for color, texture and multiscale image segmentation. In: Proc. 1996 International Conference on Image Processing. Volume 1. (1996) 817–820
21. BioPSE: Problem Solving Environment for modeling, simulation, image processing, and visualization for biomedical computing applications. Scientific Computing and Imaging Institute (SCI), `http://software.sci.utah.edu/biopse.html`, 2002.