

Segmenting Images on the Tensor Manifold

Yogesh Rathi, Allen Tannenbaum
Georgia Institute of Technology, Atlanta, GA
yogesh.rathi@gatech.edu, tannenba@ece.gatech.edu

Oleg Michailovich
University of Waterloo, Canada
olegm@uwaterloo.ca

Abstract

In this note, we propose a method to perform segmentation on the tensor manifold, that is, the space of positive definite matrices of given dimension. In this work, we explicitly use the Riemannian structure of the tensor space in designing our algorithm. This structure has already been utilized in several approaches based on active contour models which separate the mean and/or variance inside and outside the evolving contour. We generalize these methods by proposing a new technique for performing segmentation by separating the entire probability distributions of the regions inside and outside the contour using the Bhattacharyya metric. In particular, this allows for segmenting objects with multimodal probability distributions (on the space of tensors). We demonstrate the effectiveness of our algorithm by segmenting various textured images using the structure tensor. A level set based scheme is proposed to implement the curve flow evolution equation.

1. Introduction

Segmentation is an important aspect in the understanding and analysis of images. It involves separating an image into N disjoint regions such that each region corresponds to an object of interest. In many practical situations, one is required to separate an image into only 2 distinct regions, i.e., object and background. This has many applications such as tracking a given object in a sequence of images or extracting a medical structure from a 3D volume. Extensive research has been done (see [30, 17, 18, 21, 23] and the references therein) to automate the process of segmentation using various features of the image such as intensity, color and texture.

Recent advances in medical imaging technology (e.g., DT-MRI) has led to the availability of multi-valued image data. This data is used to compute a tensor that represents the diffusion of water molecules at each point (pixel) of the image. Figure 1 shows a color-coded fractional anisotropy (FA) image. The FA image is computed from the eigenvector/eigenvalues of the tensor at each pixel of the image. The

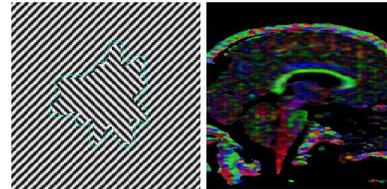


Figure 1. Left: Synthetic texture image with same intensity distribution of the object and background. Right: A color coded fractional anisotropy image showing diffusion of water molecules.

need to perform segmentation using such tensor data has led to a number of segmentation algorithms; see [22, 13, 20] and the references therein.

Figure 1 shows an image with a “butterfly like” object whose intensity distribution is the same as that of the background. For such images, it is not possible to segment the object based on intensity information alone. Many methods [25, 10, 7] have been proposed to address this problem. The authors in [25, 10] have proposed to use the response of Gabor filters to distinguish between textures, since their response may be different for different types of textures. The method however suffers from the drawback of having to process a high-dimensional feature vector (i.e., the response of the filter) making it computationally expensive. A similar problem exists with wavelet based segmentation algorithms [7].

The structure tensor [3, 8] on the other hand is a low dimensional feature computed from spatial derivatives of the image. It forms the basis of many segmentation algorithms, most notably, in the geometric active contour framework. The authors in [27] assume that the image is composed of two piecewise constant regions. They propose a variational framework to segment the image by minimizing an energy functional that separates the mean tensor inside and outside the evolving contour. To compute the tensor statistics (mean), the Frobenius norm is used assuming that the tensors lie in a vector space (i.e., the positive-definiteness is not taken into account). Another method proposed in [22] assumes a Gaussian distribution of the tensors for the object and the background. The tensor space is again assumed to be Euclidean for the purpose of computing the mean and

variance of this distribution. A different method by [13] uses the symmetrized KL-divergence to compute the distance between two tensors.

As has been pointed out in [20, 9], the space of positive definite matrices (tensors) is not a vector space but instead forms a Riemannian manifold. Hence the methods proposed in [27, 22, 13] do not utilize the true variation of the tensor field. This drawback has been addressed by the authors in [14] where they propose a segmentation algorithm that computes the mean and variance of the tensor field using the *intrinsic* Riemannian distance on the manifold. The method however assumes that the distribution of tensors within the object and background each follow a normal distribution with different mean and variance.

In this work, we propose to extend the method in [14] to any arbitrary distribution. In particular, the separation between object and background is obtained using the level set framework for contour evolution by minimizing the Bhattacharyya distance between the probability distribution inside and outside the evolving contour taking into account explicitly the Riemannian geometry of the tensor space. We also demonstrate the utility of our proposed method on several textured images.

The rest of the paper is organized as follows. First, in Section 2 we provide a brief overview of basic Riemannian geometry as applied to the tensor manifold. We then describe the variational framework for separating two distributions using the Bhattacharyya distance in Section 3. Next in Section 4, we present some experimental results and provide some of the numerical implementation details. We conclude with future research directions in Section 5.

2. The Tensor Manifold

Positive definite symmetric matrices (tensors) are widely used in image processing. These are all $n \times n$ symmetric matrices A such that $x^T A x > 0$, for all nonzero $x \in \mathbb{R}^n$. They can either reveal structural information of an image (structure tensor) [3] or characterize the diffusion of water molecules as in diffusion tensor imaging (DTI) [2]. A structure tensor is used to extract important features (e.g., edges, corners, texture informational, etc.) from an image. The structure tensor is classically obtained by Gaussian smoothing of the tensor product of the image gradient, i.e.,

$$\mathbf{T} = K_\rho * (\nabla I \nabla I^T) = \begin{pmatrix} K_\rho * I_x^2 & K_\rho * I_x I_y \\ K_\rho * I_x I_y & K_\rho * I_y^2 \end{pmatrix} \quad (1)$$

where I is a scalar valued image, K_ρ is a Gaussian kernel with standard deviation ρ , and subscripts denote partial derivatives. For vector valued images (e.g., color images) the structure tensor is given by

$$\mathbf{T} = K_\rho * \left(\sum_{i=1}^N \nabla I_i \nabla I_i^T \right). \quad (2)$$

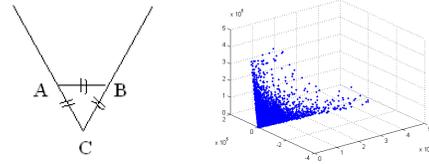


Figure 2. Left: A 2D cone, Right: Distribution of structure tensor in \mathbb{R}^3 for a typical image.

Recently, a nonlinear structure tensor was proposed in [5], where nonlinear smoothing is performed (instead of Gaussian smoothing) so that edges are better preserved and do not get dislocated. One can use any such method to obtain a structure tensor that can be used for the purpose of segmentation as described in the remainder of this paper.

It is well known that the space of tensors is not a vector space, but instead forms a Riemannian manifold \mathcal{M} . More specifically, \mathcal{M} forms a convex cone, endowed with a Riemannian metric g . Figure 2 shows the manifold (cone) of structure tensors for a typical image. Using this metric, it is possible to define the geodesic distance between two points (tensors) on the manifold. Many past works such as [27, 22] have used the Euclidean metric or Frobenius norm to compute distances between two tensors. This can produce erroneous results as can be seen from Figure 2.

For the sake of simplicity, let us consider a 2D cone with points A , B and C (see Figure 2). Furthermore, assume that the Euclidean distance is given by $d(A, B) = d(A, C) = d(B, C) = d_1$. Then, the Riemannian distance between A and B is given by $d(A, C) + d(B, C) = 2d_1$. Thus, under the Euclidean space structure, points B and C are equidistant from A , but not quite so if one considers the geodesic distance on the manifold. Hence, using the Euclidean metric can produce erroneous estimates of the tensor statistics (e.g., mean, variance, probability distribution, etc.) which form the basis of many active contour based segmentation algorithms. In this work, we propose to account for the Riemannian geometry of the tensor manifold for computing probability distributions which is then used in the segmentation algorithm.

The theory of symmetric spaces has been extensively studied since the seminal work of Nomizu [15]. A comprehensive work on tensor manifolds can be found in [20, 9]. In this work, we briefly describe the concepts of geodesic distance, the exponential map and the logarithmic map which will be required in the sequel. For the sake of brevity, our treatment will not be completely rigorous or complete, but we want to at least outline some of the key ideas. A Riemannian metric on a manifold \mathcal{M} smoothly assigns to each point $x \in \mathcal{M}$ an inner product on $T_x \mathcal{M}$, the tangent space to \mathcal{M} at x . For our case, the tangent space $T_x \mathcal{M}$ at x can be identified with the space of $n \times n$ symmetric matrices, $Sym(n)$, which is isomorphic to the vector space

$\mathbb{R}^{n(n+1)/2}$.

The function that maps to each vector $\overrightarrow{xy} \in T_x\mathcal{M}$ the point y of the manifold that is reached after a unit time by the geodesic starting at x with this tangent vector is called the *exponential map*. This map, $\exp_x : T_x\mathcal{M} \rightarrow \mathcal{M}$ is defined on the whole tangent space $T_x\mathcal{M}$ and for the space of tensors, this map is also one-to-one. There also exists a unique inverse map called the *Riemannian log map* $\text{Log}_p : \mathcal{M} \rightarrow T_p\mathcal{M}$ that maps a point $x \in \mathcal{M}$ to the unique tangent vector at p that is the initial velocity of the unique geodesic γ with $\gamma(0) = p$ and $\gamma(1) = x$.

Given a point $x \in \mathcal{M}$ and a tangent vector $W \in \text{Sym}(n) = T_x\mathcal{M}$, the exponential map is given by the following expression:

$$\exp_x(W) = x^{\frac{1}{2}} \exp\left(x^{-\frac{1}{2}} W x^{-\frac{1}{2}}\right) x^{\frac{1}{2}}. \quad (3)$$

Similarly, given points $p, \Lambda \in \mathcal{M}$, the log map at p can be computed as follows:

$$\text{Log}_p(\Lambda) = p^{\frac{1}{2}} \log(\Sigma) p^{\frac{1}{2}}, \quad \Sigma = p^{-\frac{1}{2}} \Lambda p^{-\frac{1}{2}}. \quad (4)$$

The geodesic distance between points $p, \Lambda \in \mathcal{M}$ can be computed using the following expression:

$$d^2(p, \Lambda) = \sum_{i=1}^N (\log(\sigma_i))^2, \quad (5)$$

where σ_i are the eigenvalues of Σ defined in (4) above.

The intrinsic mean of a random variable in an arbitrary metric space is the point that minimizes the expected value of the sum-of-squared distance function. As has been pointed out in [20, 9], there is no closed form expression for computing the mean of n points lying on the manifold \mathcal{M} . There, however, exists a unique intrinsic mean μ of a set of points $\{p_1, p_2, \dots, p_n\}$ and it can be computed using a gradient descent algorithm as follows:

$$\mu_{t+1} = \mu_t^{\frac{1}{2}} \exp\left(\frac{-dt}{n} \sum_{i=1}^n \log\left(\mu_t^{-\frac{1}{2}} p_i \mu_t^{-\frac{1}{2}}\right)\right) \mu_t^{\frac{1}{2}}. \quad (6)$$

Details about computing the variance (or principal geodesic analysis) can be obtained from [20, 9].

3. Segmentation Algorithm

Region-based geometric active contours have been successfully used for segmentation of images (see [6, 29, 19] and the references therein). In these methods, a contour is evolved so as to minimize an image based energy functional resulting in the required segmentation. Most algorithms employ an energy functional that depends only on the first two moments of a feature vector. In this work, we perform segmentation by separating regions with different probability density functions (pdf's). As such, we propose to minimize the Bhattacharyya distance, which gives a measure of similarity between two pdf's.

3.1. The Bhattacharyya Flow

An object can be represented by a closed curve enclosing its boundary. Many possible parameterizations of planar shapes described as closed contours have been proposed (see [26, 4] and the references therein). Recently, level set methods, which use an implicit representation of contours, have become very popular [16, 24]. The curve C is represented as the zero level set of a higher dimensional function, typically a signed distance function $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}$, such that $\phi < 0$ inside C and $\phi > 0$ outside C . This representation allows for natural breaking and merging of curve topologies, hence we have decided to use it in the present work.

The Bhattacharyya distance [12] gives a measure of similarity between two pdf's, i.e.,

$$B = \int_{\mathcal{Z}} \sqrt{P_{in}(z) P_{out}(z)} dz, \quad (7)$$

where $z \in \mathcal{Z}$ is a photometric variable such as intensity, a color vector or a texture vector, and lives in the space \mathcal{Z} , while P_{in} and P_{out} are pdf's defined on the variable z for the region inside and outside C respectively. This measure varies between 0 and 1, where 0 indicates a complete mismatch and 1 indicates complete agreement between the pdf's. Note that, B can also be thought of as the direction cosine between two functionals P_{in} and P_{out} . Hence, minimizing B is equivalent to separating the distributions P_{in} and P_{out} .

Let $x \in \mathbb{R}^2$ specify the coordinates in the image plane, and let $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathcal{Z}$ be a mapping from the image plane to the space of the photometric variable (in our case, the space of structure tensor). The pdf P_{in} (or P_{out}) is assumed to be defined by

$$P_{in}(z) = \frac{\int_{\omega} K(z - I(x)) dx}{\int_{\omega} dx} \quad (8)$$

which is the nonparametric kernel density estimate of the pdf of z for a given kernel K . Typical choices for K are the Dirac delta function $\delta(\cdot)$ and the multi-dimensional Gaussian kernel given by

$$K(y) = \frac{1}{(2\pi |C|)^{n/2}} \exp\left(-\frac{y^T C^{-1} y}{2}\right), \quad (9)$$

where $|C|$ is the determinant of the covariance matrix (bandwidth) of the kernel and n is dimension of y . Extensive research has been done on choosing the optimal C . A detailed analysis can be found in [11]. The rest of the derivation is independent of the choice of the kernel K . For the case of curve evolution, P_{in} is the density of the region inside the curve C . Thus, ω is the region enclosed by C with area denoted by A_{in} . Writing (8) in terms of the level set function ϕ , we get

$$P_{in}(z) = \frac{\int_{\Omega} K(z - I(x)) H(-\phi(x)) dx}{\int_{\Omega} H(-\phi(x)) dx}, \quad (10)$$

where H is the Heaviside step function given by:

$$H(\phi) = \begin{cases} 1 & \phi > \epsilon, \\ 0 & \phi < -\epsilon, \\ \frac{1}{2}\left\{1 + \frac{\phi}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\phi}{\epsilon}\right)\right\} & \text{else,} \end{cases}$$

and Ω is the whole image domain. Similarly, $P_{out}(z)$ can be written as

$$P_{out}(z) = \frac{\int_{\Omega} K(z - I(x)) H(\phi(x)) dx}{\int_{\Omega} H(\phi(x)) dx}. \quad (11)$$

Denoting by A_{out} the area outside the curve C , and computing the first variation of (7), we get the following:

$$\frac{\partial P_{in}(z)}{\partial \phi} = \frac{\delta_{\epsilon}(\phi)}{A_{in}} (P_{in}(z) - K(z - I(x))),$$

$$\frac{\partial P_{out}(z)}{\partial \phi} = \frac{\delta_{\epsilon}(\phi)}{A_{out}} (K(z - I(x)) - P_{out}(z)),$$

$$\nabla_{\phi} B = \frac{1}{2} \int_{\mathcal{Z}} (P_{in}(z) P_{out}(z))^{-1/2} \times \left(\frac{\partial P_{in}(z)}{\partial \phi} P_{out}(z) + P_{in}(z) \frac{\partial P_{out}(z)}{\partial \phi} \right) dz.$$

Combining all of the equations above, we obtain the following PDE:

$$\frac{\partial \phi(x, t)}{\partial t} = -\frac{B \delta_{\epsilon}(\phi)}{2} \left(\frac{1}{A_{in}} - \frac{1}{A_{out}} \right) - \frac{\delta_{\epsilon}(\phi)}{2} \times \int_{\mathcal{Z}} K(z - I(x)) \left(\frac{1}{A_{out}} \sqrt{\frac{P_{in}(z)}{P_{out}(z)}} - \frac{1}{A_{in}} \sqrt{\frac{P_{out}(z)}{P_{in}(z)}} \right) dz. \quad (12)$$

The first term in the latter equation determines the ‘‘global’’ direction in which the entire curve moves, whereas the second term determines the ‘‘local’’ evolution direction. Thus, the initial motion of the curve is influenced by the ‘‘global’’ term, while its contribution is minimal when B is close to zero indicating convergence of the curve evolution. In numerical experiments, a regularizing term is added to the above flow equation penalizing the curve length so that the contour is smooth, and the final expression for the level set evolution becomes

$$\frac{\partial \phi}{\partial t} = \delta_{\epsilon}(\phi) (V + \alpha \kappa) \quad (13)$$

where κ is the curvature, α is a user defined weight and V is the speed term from equation (12).

3.2. Application to Tensors

In this work, the space of photometric variable \mathcal{Z} is obtained by mapping each point (tensor) of the image lying on the manifold \mathcal{M} onto the tangent space $T_x \mathcal{M}$ at point $x \in \mathcal{M}$ using the Riemannian log map Log_x defined in (4). As stated earlier, $T_x \mathcal{M}$ is isomorphic to $\mathbb{R}^{n(n+1)/2}$. Thus, for structure tensors of dimension 2×2 , $\mathcal{Z} = \mathbb{R}^3$, whereas for diffusion tensors $\mathcal{Z} = \mathbb{R}^6$. The log map allows us to go from the tensor cone to Euclidean space where one can use standard techniques to compute the tensor statistics like mean, variance and probability distributions.

One can choose the tangent space corresponding at any given point $x \in \mathcal{M}$ of the manifold. A natural choice for x is the intrinsic mean tensor μ of the image, which can be computed using (6). The probability distribution P_{in} and P_{out} can be approximated using kernel density estimate [28] on a set of bins or ‘‘target points’’ as follows:

$$P(z) = \frac{1}{n} \sum_{i=1}^n K(z - x_i), \quad (14)$$

where n is the number of points, x_i are points in the tangent space $T_{\mu} \mathcal{M}$ and K is a multivariate Gaussian kernel defined in (9). Note that, for computing P_{in} (P_{out}), x_i are all points lying inside (outside) the curve C . The bins should be chosen in such a manner that each point ‘‘belongs’’ to at least one bin. The contour evolution can now be performed using (13).

The entire algorithm can be summarized as follows :

Algorithm 1 : Segmentation Algorithm

- 1: Compute structure tensor at each point of the image using (1).
 - 2: Use (6) to compute the mean tensor μ of the image.
 - 3: Use the Riemannian log map Log_{μ} (4) to map each point (tensor) of the image to the tangent plane $T_{\mu} \mathcal{M}$.
 - 4: Construct bins so that each point falls in atleast one bin.
 - 5: Use a multi-variate Gaussian kernel (or any other kernel) to compute the initial pdf's P_{in} and P_{out} based on the initialization of the starting contour.
 - 6: Evolve the curve using (13) until convergence.
-

Notice that steps (1) to (5) above are preprocessing steps and can be computed off-line before executing the contour evolution of step (6).

4. Experiments

We tested the proposed method to perform segmentation of several textured images. We certainly do not claim that the method proposed in this work is optimal in all cases, but it did give good results on the set of images on which it was tested. In this section, we show comparative results with the

method in [14] where the authors assume a Gaussian distribution for the object and background. We also compare the results obtained if one assumes a Euclidean metric instead of the Riemannian metric. The latter comparison was performed using the Bhattacharyya flow without assuming any prior knowledge about the distribution of the object or background.

To the best of our knowledge, this is the first time a method has been proposed that performs segmentation taking into account the Riemannian structure of the tensor manifold and without any prior assumption on the distribution of tensor field of the object and background. Thus, the proposed method generalizes the algorithm of [14] to any arbitrary and non-analytic distribution.

4.1. Numerical Implementation

In this work, the structure tensors were computed using (1) for gray level images and (2) for color images. Computing the intrinsic mean μ of the image is quite fast and the gradient descent algorithm (6) usually converges in a few iterations [20]. The tangent plane at the mean is a good choice for mapping the points from the tensor manifold. This fact was also used in [9] for performing principal geodesic analysis (PCA on manifolds). We have tested our segmentation algorithm by mapping points from the manifold onto several different tangent planes and each time we obtained similar results. A particular case to note is that if one uses the tangent plane at the identity tensor, the Riemannian metric is equivalent to using the Log-Euclidean metric proposed in [1].

Bins (“target points”) can be computed by creating a multi-dimensional grid (3D in the case of structure tensor) that spans the whole data set. Alternatively, one can use fast algorithms like the improved fast Gauss transform [28] to find a reasonable set of target points.

Once the initial pdf’s P_{in} and P_{out} are computed, one can recursively update them as the contour evolves. This can be achieved as follows:

Algorithm 2 : Updating Probability distributions

- 1: Compute the histograms p_{in} and p_{out} .
 - 2: Find the points in the image that have moved from “outside C ” to “inside C ” or vice-versa after one iteration of contour evolution.
 - 3: Compute the contribution of these points at each of the target points.
 - 4: Add/subtract this contribution appropriately from p_{in} and p_{out} .
 - 5: Divide this histogram with the new area to obtain the updated pdf’s P_{in} and P_{out} .
-

Note that this process is quite fast. However, a further speed improvement can be achieved by pre-computing the

contribution of each point at each of the target points. This is in contrast to the method in [14] where the mean and variance (for the inside and outside) have to be computed after every iteration. This whole process is computationally very expensive (*all the points in the image have to be mapped from the manifold to the tangent plane at each iteration*) compared to the simple probability update methodology proposed in this work. Also note that in our proposed framework, all the points are mapped from the manifold to the tangent plane only once, during the preprocessing step (see Algorithm 1). This is another advantage of the proposed method. We now discuss the results obtained on 4 test images.

4.2. Lizard example

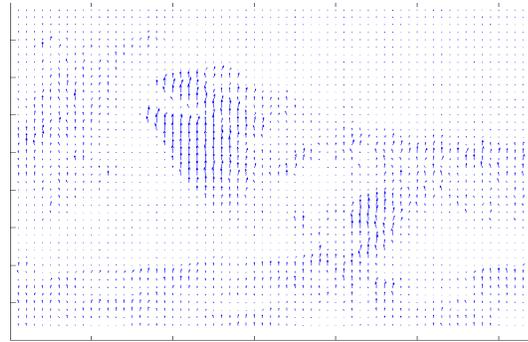
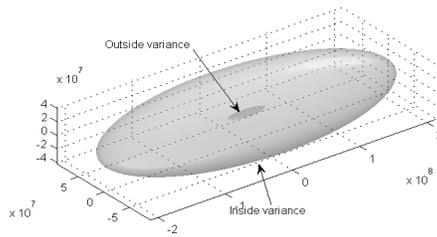
Figure 5 shows the starting contour on the lizard image. The objective is to separate the lizard from the background. We ran the algorithm in [14] on this image. Figure 3 shows the variance (Gaussian assumption) of the distribution inside and outside at initialization. Clearly, the variance of the lizard (in the tensor space) is about an order of magnitude larger than the background. Figure 3 also shows the distribution of the tensor field in the image, with each arrow representing the major eigenvector of the tensor. As can be seen, the tensor distribution of lizard is multi-modal. Thus, the Gaussian assumption of object and background as in [14], is not valid and hence the method fails to segment the lizard (see Figure 4). Figure 5 shows evolution of the contour and final segmentation using the proposed method with the same initialization as shown in Figure 4. This example demonstrates the usefulness of the proposed method when the object to be segmented has a multi-modal distribution.

4.3. Tiger example

In this image, the tiger and the background have very similar textural features. Figure 6 shows the segmentation result assuming the tensors lie in a vector space, and by using the Bhattacharyya flow for contour evolution. As can be seen, the method fails to capture the neck region of the tiger. Figure 7 shows the result of the segmentation using the proposed method. The contour in this case properly encloses the tiger. This example clearly demonstrates that using the Riemannian geometry of the manifold (instead of Euclidean metric) in the segmentation framework can lead to better results.

4.4. Duck example

In this example, the duck is difficult to segment based on intensity information alone due to the texture of water in the background. Figure 8 shows the segmentation assuming a Euclidean metric for the tensors. The contour evolution was



(a)

(b)

Figure 3. Lizard example: (a) variance of the Gaussian distribution inside and outside the contour at initialization, (b) Tensor field distribution using major eigenvectors.

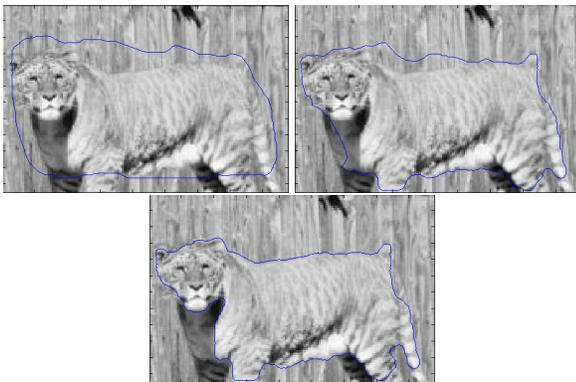


Figure 6. Tiger example: Segmentation using Euclidean metric

done using the Bhattacharyya flow described earlier in this paper. Figure 9 shows the result of the segmentation using the Riemannian metric as proposed in this work. Once again it becomes clear that the Euclidean metric captures incorrect information in computing the probability distribution.

4.5. Butterfly example

In this case, we demonstrate segmentation using color texture information. The color structure tensor was computed as described earlier in Section 2. Figure 10 shows segmentation assuming a Euclidean metric while Figure 11 shows segmentation using the Riemannian metric. Both segmentation results were obtained using the Bhattacharyya flow. Once again, the proposed method gives better results. Notice, however, that the final contour in Figure 11 does not converge at the actual edges. This is due to the dislocation of edges that occurs when computing the classical structure tensor using (1). Better results can be obtained if one uses the nonlinear structure tensor of [5].

5. Discussion

In this note, we proposed a novel method to perform segmentation of tensor valued images taking into account the natural Riemannian structure of the tensor manifold. Segmentation was performed in a variational setting by minimizing the Bhattacharyya distance which gives a measure of similarity between two pdf's. This region-based statistical segmentation approach on the space of tensors seemed to give encouraging results for several nontrivial examples. In particular, we believe that by exploiting the intrinsic metric on the space of tensors in the Bhattacharyya flow context, we were able to employ more of the available textural information in deriving our segmentations as compared to using a strictly linear vector space structure in which the positive-definiteness of the symmetric matrices is essentially ignored.

Some of the future work will entail performing segmentation on medical tensor data, especially diffusion tensor brain imagery for studying the white matter tracts. We would also like to use other optimization techniques like graph cuts which give a global minimum of the energy that is being minimized.

References

- [1] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Fast and simple calculus on tensors in the log-euclidean framework. In *MICCAI*, pages 115–122, 2005. 5
- [2] P. Basser, J. Mattiello, and D. LeBihan. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66:259–267, 1994. 2
- [3] J. Bigun, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE-PAMI*, 13(8):775–790, 1991. 1, 2
- [4] A. Blake and M. Isard, editors. *Active Contours*. Springer, 1998. 3
- [5] T. Brox and J. Weickert. Nonlinear matrix diffusion for optic flow estimation. *Pattern Recognition*, 2449:446–453, 2002. 2, 6

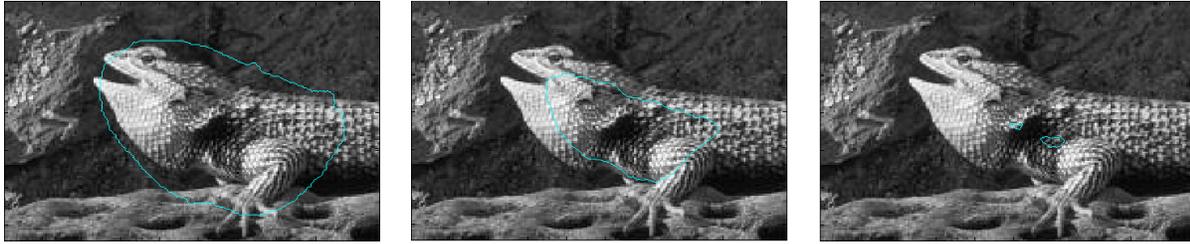


Figure 4. Lizard example: Segmentation using Gaussian assumption of method in [14].

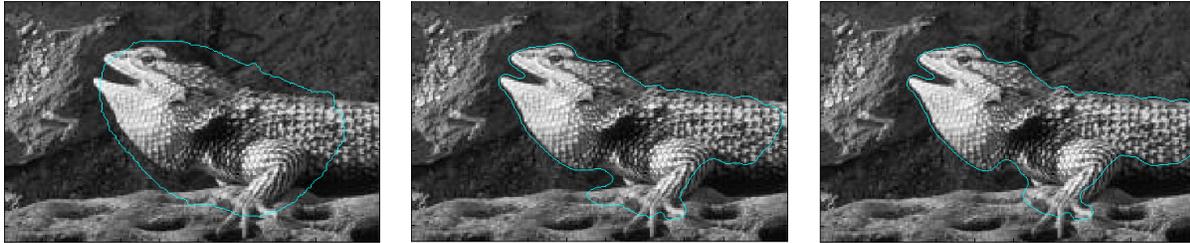


Figure 5. Lizard example: Segmentation with the proposed method (intermediate and final result)

- [6] T. Chan and L. Vese. Active contours without edges. *IEEE Trans. on Image Processing*, 10(2):266–277, 2001. **3**
- [7] T. Chang and C. Kuo. Texture analysis and classification with tree-structured wavelet transform. *Image Processing*, 2(4):429–441, October 1993. **1**
- [8] R. de Luis-Garcia, M. Rousson, R. Deriche, and C. Alberola-Lopez. Tensor processing for texture and color segmentation. In *SCIA*, pages 1117–1127, 2005. **1**
- [9] P. T. Fletcher and S. C. Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *ECCV Workshops CVAMIA and MMBIA*, pages 87–98, 2004. **2, 3, 5**
- [10] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *BioCyber*, 61:102–113, 1989. **1**
- [11] M. C. Jones, J. S. Marron, and S. J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of American Statistical Assoc.*, 91:401–407, 1996. **3**
- [12] T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Tran. Communication Technology*, 15(1):52–60, 1967. **3**
- [13] C. Lenglet, M. Rousson, and R. Deriche. Segmentation of 3d probability density fields by surface evolution: Application to diffusion mri. In *MICCAI*, pages 18–25, 2004. **1, 2**
- [14] C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras. Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor mri processing. *Journal of Mathematical Imaging and Vision*, 2006. **2, 5, 7**
- [15] K. Nomizu. Invariant affine connections on homogeneous spaces. *American Journal of Math*, 1954. **2**
- [16] S. Osher and R. P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics*, 169:463–502, 2001. **3**
- [17] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer Verlag, 2003. **1**
- [18] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *Transactions on Pattern analysis and Machine Intelligence*, 22(3):266–280, 2000. **1**
- [19] N. Paragios and R. Deriche. Geodesic active regions and level set methods for motion estimation and tracking. In *CVIU*, 2005. **3**
- [20] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *IJCV*, 66. **1, 2, 3, 5**
- [21] M. Rousson and D. Cremers. Efficient Kernel Density Estimation of Shape and Intensity Priors for Level Set Segmentation. In *Medical Image Computing and Computer Assisted Intervention*, volume 1, pages 757–764, 2005. **1**
- [22] M. Rousson, C. Lenglet, and R. Deriche. Level set and region based propagation for diffusion tensor mri segmentation. In *CVAMIA-MMBIA*, pages 123–134, 2004. **1, 2**
- [23] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2000. **1**
- [24] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999. **3**
- [25] S. Shah and J. Aggarwal. A bayesian segmentation framework for textured visual images. In *CVPR97*, pages 1014–1020, 1997. **1**
- [26] D. Terzopoulos and R. Szeliski. *Active Vision*, chapter Tracking with Kalman Snakes, pages 3–20. MIT Press, 1992. **3**
- [27] Z. Wang and B. Vemuri. Tensor field segmentation using region based active contour model. In *ECCV*, volume 3024, 2004. **1, 2**
- [28] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *ICCV03*, pages 464–471, 2003. **4, 5**
- [29] A. Yezzi, A. Tsai, and A. Willsky. A statistical approach to curve evolution for image segmentation. Technical report, MIT, 1999. **3**
- [30] T. Zhang and D. Freedman. Tracking objects using density matching and shape priors. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 1950–1954, 2003. **1**

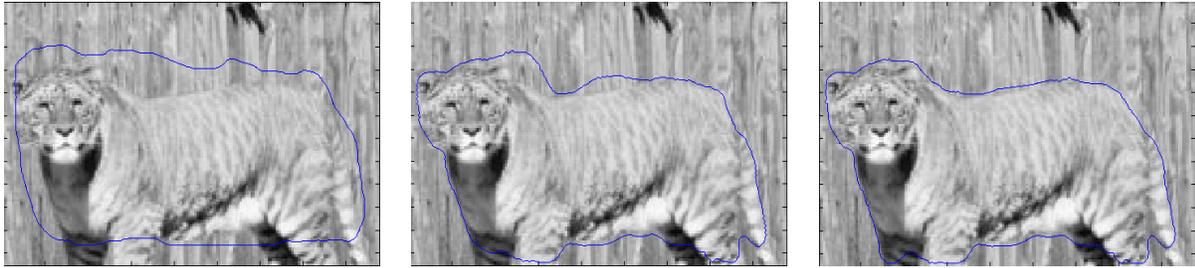


Figure 7. Tiger example: Segmentation with the proposed method (intermediate and final result)

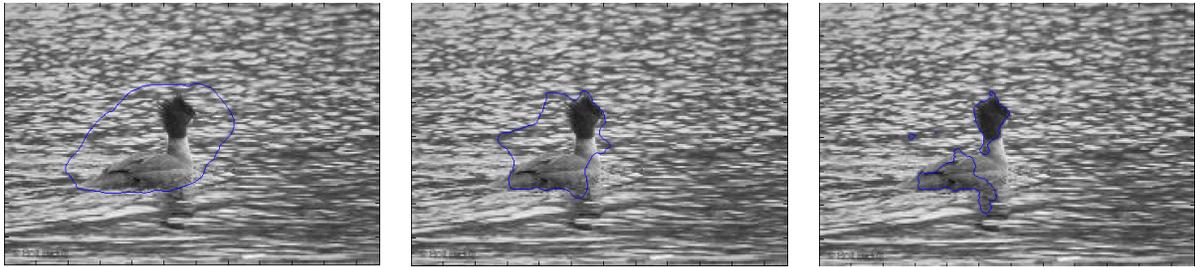


Figure 8. Duck example: Segmentation assuming Euclidean metric

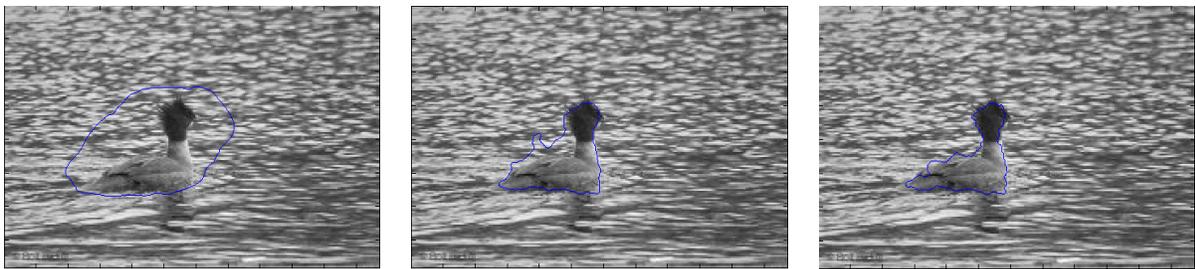


Figure 9. Duck example: Segmentation with the proposed method (intermediate and final result)



Figure 10. Butterfly example: Segmentation assuming Euclidean metric

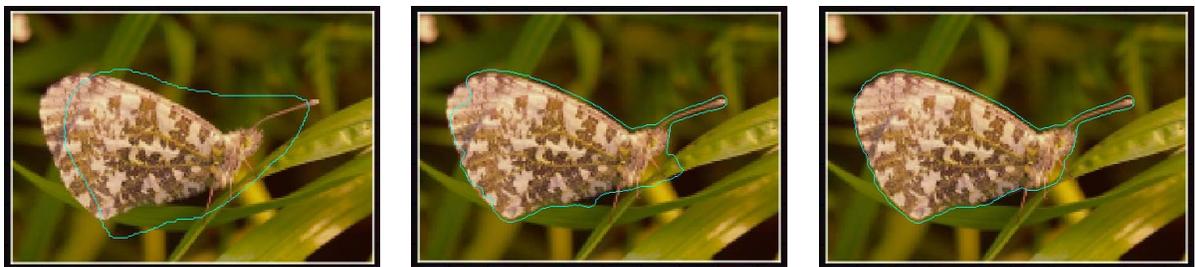


Figure 11. Butterfly example: Segmentation with the proposed method (intermediate and final result)