Discontinuity Preserving Filtering over Analytic Manifolds

Raghav Subbarao and Peter Meer
Department of Electrical and Computer Engineering, Rutgers University, Piscataway
NJ 08854, USA
rsubbara,meer@caip.rutgers.edu

Abstract

Discontinuity preserving filtering of images is an important low-level vision task. With the development of new imaging techniques like diffusion tensor imaging (DTI), where the data does not lie in a vector space, previous methods like the original mean shift are not applicable. In this paper, we use the nonlinear mean shift algorithm to develop filtering methods for data lying on analytic manifolds. We work out the computational details of using mean shift on Sym_n^+*, the manifold of* $n \times n$ *symmetric positive definite matrices. We apply our algorithm to chromatic noise filtering, which requires mean shift over the Grassmann manifold* $G_{3,1}$*, and obtain better results then standard mean shift filtering. We also use our method for DTI filtering, which requires smoothing over* Sym_3^+*.*

1. Introduction

Discontinuity preserving filtering and regularization of images is a widely researched image-processing task. The ability to smooth image noise while retaining important image structures such as edges has allowed the development of new algorithms for further processing like segmentation.

Filtering and regularization schemes were initially developed for scalar valued images [14, Ch.10]. In this case, the image was considered a real valued map on a 2D lattice which assigned each pixel an intensity value. These methods were later extended to vector valued images. Now, the image was viewed as map which assigned each pixel a vector. Many different algorithms have been proposed for the discontinuity preserving filtering of vector valued images, *e.g.*, *bilateral filtering* [18], *adaptive smoothing* [15], *mean shift* [5, 6] and *partial differential equation* (PDE) based methods [21]. Methods for smoothing vector fields are more elaborate than simply applying the scalar filtering algorithm to each channel. This is due to the coupling between the different components of the vectors. It was shown in [1], that all of the above algorithms are similar with the difference being in the way they control the smoothing process in various directions. For vector-valued image smoothing, mean shift filtering was found to perform the best [1].

Recently, there has been interest in developing filtering and regularization algorithms for non vector-valued images. For applications such as the smoothing of diffusion tensor magnetic resonance images (DT-MRI) [13] and noise chromaticity restoration, it is not possible to use filtering algorithms developed for vector fields [19]. The data values at each pixel (or voxel) satisfy further constraints due to which the space of all data values is *not* a vector space. Since, smoothing algorithms proceed by averaging image values, previous methods are no longer applicable as the concept of a (weighted) average is not clearly defined. The general way of handling this problem has been to use PDE based methods while ensuring that the data points satisfy the required constraints at each step in the evolutions [13, 19].

It has not been possible to use mean shift smoothing for such non-vector values images since the original mean shift algorithm is only applicable to points lying in vector spaces. In this paper we extend the nonlinear mean shift algorithm [17, 22] and use it for the smoothing of tensor fields. Since nonlinear mean shift is applicable points lying on analytic manifolds, we can smooth any lattice where the data values lie on an analytic manifold.

The rest of the paper is organized as follows. In Section 2 we briefly introduce some necessary concepts from the theory of analytic manifolds. This discussion follows the presentation in [16]. The nonlinear mean shift algorithm is introduced in Section 3 and the details of applying the mean shift algorithm to the manifolds of symmetric positive definite matrices are presented in Section 3.2. In Section 4 we propose our smoothing algorithm and in Section 5 we present the results of using our algorithm on real and synthetic data.

2. Analytic Manifolds

A manifold is a (topological) space that is locally similar (homeomorphic) to an Euclidean space. Intuitively, a manifold is a continuous surface lying in some higher dimensional Euclidean space. Analytic manifolds satisfy further conditions of smoothness [3]. From now onwards, we

1-4244-1180-7/07/$25.00 ©2007 IEEE

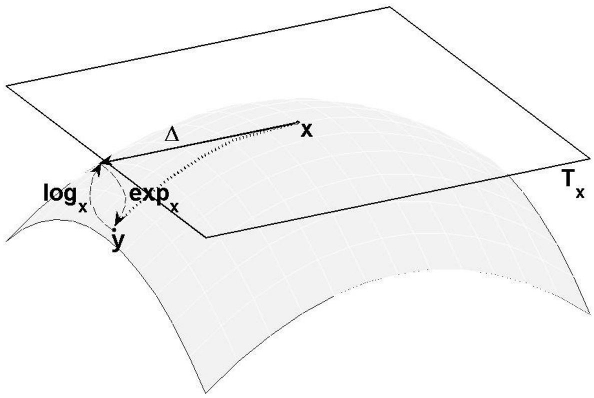


Figure 1. Example of a manifold. The tangent space at the point $\mathbf{x}$ is also shown.

restrict ourselves to analytic manifolds and assume the necessary conditions are satisfied.

The *tangent space*, $\mathbf{T_x}$ at $\mathbf{x}$, is the plane tangent to the surface of the manifold at that point. The tangent space can be considered as the set of allowed instantaneous velocities a point can have while constrained to move on the manifold. For d-dimensional manifolds, the tangent space is a d-dimensional vector space [3]. An example of a two-dimensional manifold lying in $\mathbb{R}^3$, is shown in Figure 1. The solid arrow Δ, is a tangent at $\mathbf{x}$. We can define an inner product $g_{\mathbf{x}}$ on $\mathbf{T_x}$ since it is a vector space. This product induces a norm for tangents $\Delta \in \mathbf{T_x}$ as $\|\Delta\|_{\mathbf{x}}^2 = g_{\mathbf{x}}(\Delta, \Delta)$. It should be noted that the inner product and norm vary with $\mathbf{x}$ and this dependence is indicated by the subscripts.

The length of a curve on the manifold is defined by an integral over the norms of tangents [3]. For a pair of points on the manifold, the curve of minimum length which joins them is known as the *geodesic* and the length of the geodesic is the *intrinsic distance*. Parameter spaces occurring in computer vision usually have well studied geometries and closed form formulae for the intrinsic distance are usually available.

Tangents (on the tangent space) and geodesics (on the manifold) are closely related. For each tangent $\Delta \in \mathbf{T_x}$, there is a unique geodesic starting at $\mathbf{x}$ with initial velocity Δ. The *exponential map*, $exp_{\mathbf{x}}$, maps Δ to the point on the manifold reached by this geodesic. The *logarithm map* is the inverse of the exponential map, $log_{\mathbf{x}} = exp_{\mathbf{x}}^{-1}$. The exponential and logarithm operators vary as the point $\mathbf{x}$ moves. These concepts are illustrated in Figure 1, where $\mathbf{x}$, $\mathbf{y}$ are points on the manifold and $\Delta \in \mathbf{T_x}$. The dotted line shows the geodesic starting at $\mathbf{x}$ and ending at $\mathbf{y}$. This geodesic has an initial velocity Δ and consequently, $\mathbf{y}$ and Δ satisfy $exp_{\mathbf{x}}(\Delta) = \mathbf{y}$ and $log_{\mathbf{x}}(\mathbf{y}) = \Delta$. The specific forms of these operators depend on the manifold. The operator, $exp_{\mathbf{x}}$ is usually onto but not one-to-one. For any $\mathbf{y}$ on the manifold, if there exist many $\Delta \in \mathbf{T_x}$ satisfying $exp_{\mathbf{x}}(\Delta) = \mathbf{y}$, $log_{\mathbf{x}}(\mathbf{y})$ is chosen as the tangent with the smallest norm.

For a smooth, real valued function f defined on the manifold, the *gradient* of f at $\mathbf{x}$, $\nabla f \in \mathbf{T_x}$, is defined to be the *unique* tangent vector satisfying

$$g_{\mathbf{x}}(\nabla f, \Delta) = \partial_{\Delta} f \tag{1}$$

for any $\Delta \in \mathbf{T_x}$, where ∂_{Δ} is the directional derivative along Δ. This gradient has the property of representing the tangent of maximum increase.

We represent the points on manifolds by small bold letters, e.g., $\mathbf{x}, \mathbf{y}$. In some of our examples, the manifold consists of matrices and each point represents a matrix. Although matrices are conventionally represented by capital bold letters, when we consider them to be points on a manifold, we denote them by small letters. This should not be a problem, since any matrix can be represented as a vector by rearranging its elements into a single column.

3. Nonlinear Mean Shift

Mean shift is a nonparametric clustering algorithm which was first proposed in [9, p.535] and then introduced to the vision community by [6]. Since then it has been used for a wide variety of applications.

The original mean shift algorithm was proposed for vector spaces. It works by defining a *kernel density estimate*, which is a data-driven estimate of the original density form which the given data was sampled. The mean shift algorithm is an iterative procedure for finding the modes of the kernel density. At each iteration, the current estimate of the mode is updated to be the weighted mean of all the points lying a local neighbourhood of the present position. The change in the mode location is the *mean shift vector*.

3.1. Nonlinear Mean Shift

While the ideas of weighted means and vectors are well known for vector spaces, they are not well defined for manifolds. In this section we follow the derivation of [16], where the mean shift vector was formulated as the weighted sum of *tangent vectors*. Since tangent spaces are vector spaces, a weighted average of tangents is possible and can be used to update the mode estimate. This method is valid over *any* analytic manifold.

Consider a manifold with a metric d. Given n points on the manifold, $\mathbf{x}_i, i = 1, \ldots, n$, the *kernel density estimate* with *profile* k and *bandwidth* h is

$$\hat{f}_k(\mathbf{x}) = \frac{c_{k,h}}{n} \sum_{i=1}^{n} k\left(\frac{d^2(\mathbf{x}, \mathbf{x}_i)}{h^2}\right). \tag{2}$$

The bandwidth h can be included in the distance as a parameter. However, written in this form, the bandwidth gives a parameter which can be used to tune performance. If the manifold is a Euclidean space with a Euclidean distance metric, (2) reduces to the well known Euclidean kernel density estimate.

Estimating $c_{k,h}$ is requires the integration of the profile function over the manifold. Since a global scaling does not affect the mode, we drop $c_{k,h}$ from now onwards [4].

Taking the gradient of $\hat{f}_k$ at $\mathbf{x}$,

$$\begin{aligned} \nabla \hat{f}_k(\mathbf{x}) &= \frac{1}{n}\sum_{i=1}^{n} \nabla k\left(\frac{d^2(\mathbf{x},\mathbf{x}_i)}{h^2}\right) \\ &= -\frac{1}{n}\sum_{i=1}^{n} g\left(\frac{d^2(\mathbf{x},\mathbf{x}_i)}{h^2}\right)\frac{\nabla d^2(\mathbf{x},\mathbf{x}_i)}{h^2} \end{aligned} \quad (3)$$

where, $g(x) = -k'(x)$. The gradient of $\nabla d^2(\mathbf{x},\mathbf{x}_i)$ is taken with respect to $\mathbf{x}$. Similar to the original mean shift step [6], we define the nonlinear mean shift vector as

$$\mathbf{m}_h(\mathbf{x}) = \frac{-\sum_{i=1}^{n} \nabla d^2(\mathbf{x},\mathbf{x}_i) g\left(\frac{d^2(\mathbf{x},\mathbf{x}_i)}{h^2}\right)}{\sum_{i=1}^{n} g\left(\frac{d^2(\mathbf{x},\mathbf{x}_i)}{h^2}\right)}. \quad (4)$$

The operations in (4) are well defined. The gradient terms, $\nabla d^2(\mathbf{x},\mathbf{x}_i)$ lie in the tangent space $\mathbf{T_x}$, and the kernel terms $g(d^2(\mathbf{x},\mathbf{x}_i)/h^2)$ are scalars. The mean shift vector is a weighted average of tangent vectors, and lies in $\mathbf{T_x}$. The iteration moves the point along the geodesic defined by the mean shift vector. The nonlinear mean shift iteration is

$$\mathbf{x}^{(j+1)} = exp_{\mathbf{x}^{(j)}}\left(\mathbf{m}_h(\mathbf{x}^{(j)})\right). \quad (5)$$

The iteration (5), moves the current mode estimate $\mathbf{x}^{(j)}$ along the geodesic defined by the mean shift vector, to get the next updated estimate, $\mathbf{x}^{(j+1)}$.

3.2. Symmetric Positive Definite Matrices

In practice, nonlinear mean shift requires the computation of a distance metric and its gradient. These functions depend on manifold on which the data lies. The details of nonlinear mean shift over *Grassmann manifolds* [7] and *matrix Lie groups* are in [16, 22]. Here, we state the formulae for applying nonlinear mean shift to the manifold, Sym_n^+ of $n \times n$ symmetric positive definite (SPD) matrices.

Let exp and log be the standard matrix exponential and logarithm operators. These are general matrix operators and no subscript is necessary. They should not be confused with the manifold operators which we define later. For symmetric matrices these operators can be simplified. Let Δ be a symmetric matrix such that $\Delta = \mathbf{udu}^T$, where $\mathbf{u}$ is a matrix of eigenvectors and $\mathbf{d}$ is a diagonal matrix of eigenvalues. The matrix exponential becomes

$$exp(\mathbf{\Delta}) = \mathbf{u}\,\mathbf{diag}(exp(d_i))\,\mathbf{u}^T. \quad (6)$$

This exp operator is defined for all symmetric matrices, not just SPD matrices. However, the eigenvalues of $exp(\mathbf{\Delta})$, are positive and it is always SPD. Therefore, log, which is the inverse of exp, is only defined for SPD matrices. Let $\mathbf{vsv}^T$ be the eigenvalue decomposition of a SPD matrix $\mathbf{x}$, where $\mathbf{v}$ is orthonormal and $\mathbf{s}$ is the diagonal matrix of *positive* eigenvalues. Then

$$log(\mathbf{x}) = \mathbf{v}\,\mathbf{diag}(log(s_i))\,\mathbf{v}^T. \quad (7)$$

We also define the square root of $\mathbf{x}$ as

$$\mathbf{x}^{1/2} = exp(\tfrac{1}{2}log\mathbf{x}) \quad (8)$$

$$\mathbf{x}^{-1/2} = inv(\mathbf{x}^{1/2}). \quad (9)$$

Let $\mathbf{x}$ and $\mathbf{y}$ be two $n \times n$ SPD matrices. The distance between $\mathbf{x}$ and $\mathbf{y}$ is given by

$$d^2(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \ln^2 \lambda_i(\mathbf{x},\mathbf{y}) \quad (10)$$

where, $\lambda_i(\mathbf{x},\mathbf{y})$ is the i-th generalized eigenvalue of $\mathbf{x}$ and $\mathbf{y}$. Computationally, the generalized eigenvalues are the eigenvalues of $\mathbf{x}^{-1/2}\mathbf{y}\mathbf{x}^{-1/2}$. This metric was first proposed in [8] and later it was shown by Pennec *et al.* [13] to be the only affine invariant distance measure on Sym_n^+.

Since, Sym_n^+ is a Riemannian manifold [13], we approximate the gradient by the logarithm operator [16],

$$\nabla d^2(\mathbf{x},\mathbf{y}) = -log_{\mathbf{x}}(\mathbf{y}) \quad (11)$$

where, the manifold logarithm is given by

$$log_{\mathbf{x}}(\mathbf{y}) = \mathbf{x}^{1/2} log(\mathbf{x}^{-1/2}\mathbf{y}\mathbf{x}^{-1/2})\mathbf{x}^{1/2}. \quad (12)$$

The tangent space of Sym_n^+ is the set of symmetric matrices, and $log_{\mathbf{x}}(\mathbf{y})$ lies in this tangent space. The mean shift vector will also lie in this tangent space.

Given a tangent $\mathbf{\Delta}$, the manifold exponential operator is

$$exp_{\mathbf{x}}(\mathbf{\Delta}) = \mathbf{x}^{1/2} exp(\mathbf{x}^{-1/2}\mathbf{y}\mathbf{x}^{-1/2})\mathbf{x}^{1/2}. \quad (13)$$

4. Discontinuity Preserving Filtering

The original mean shift has been used for the discontinuity preserving filtering of color images [5, 6]. Our algorithm works in a similar manner.

We consider the *image* $\mathbf{I}$ as a mapping defined on a n-dimensional lattice which assigns a data value to each point on the lattice. Typically, $n = 2$, *e.g.*, images or $n = 3$, for DTI images. At each location $\mathbf{z}_i$, we require the data values $\mathbf{I}(\mathbf{z}_i)$, to lie on an analytic manifold, $\mathcal{M}$. Each image value $\mathbf{I}(\mathbf{z}_i)$ along with its location $\mathbf{z}_i$ is considered as a single data point $\mathbf{x} = (\mathbf{z}, \mathbf{I}(\mathbf{z}))$, in the *joint domain* $\mathbb{R}^n \times \mathcal{M}$.

For each pixel location $\mathbf{z}_i$, we initialize a mean shift iteration in the joint space at the point $(\mathbf{z}_i, \mathbf{c}_i)$, where

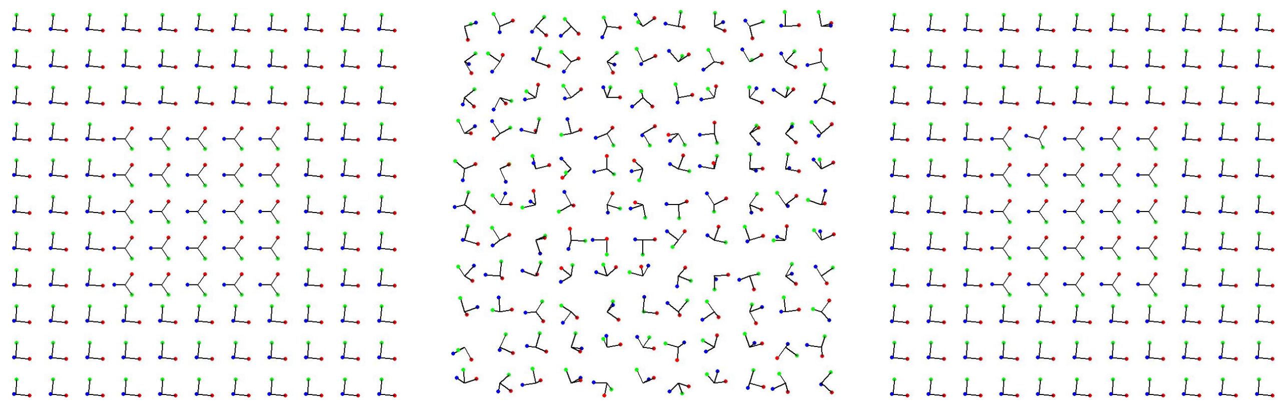


Figure 2. *Orthonormal Field Filtering*. A 2D lattice of rotation matrices. Red, green and blue dots are used to represent the x-, y-, and z-axis, respectively. The original field is on the left, the noisy field is in the middle and the filtered field is on the right.

$\mathbf{c}_i = \mathbf{I}(\mathbf{z}_i)$. Let the point where this iteration converges be $(\hat{\mathbf{z}}_i, \hat{\mathbf{c}}_i)$. In the filtered image $\mathbf{I}_f$, we set $\mathbf{I}_f(\mathbf{z}_i) = \hat{\mathbf{c}}$.

The profile in the joint domain is taken to be the product of a *spatial profile* defined on the Euclidean part of the joint domain and a *parameter profile* defined on the manifold, as

$$k(\mathbf{x}, \mathbf{x}_i) = k_s \left(\frac{\|\mathbf{z} - \mathbf{z}_i\|^2}{h_s^2} \right) k_p \left(\frac{d^2(\mathbf{c}, \mathbf{c}_i)}{h_p^2} \right). \qquad (14)$$

The bandwidth in the joint domain consists of a spatial bandwidth h_s and a parameter bandwidth h_p. In practice, we use a truncated normal kernel and the performance of the algorithm can be controlled by varying h_p and h_s.

Note, when mean shift is run in the joint domain, both the spatial and parameter values are updated in an iteration. Unlike other methods which average parameter values in a fixed spatial window, mean shift accounts for information *beyond* the initial spatial window. In [1], this was found to be the reason for the better performance of mean shift as compared to other color image filtering algorithms.

To optimize performance, we used the heuristic suggested in [5] and used in the EDISON system. The filtering step was not applied to pixels which are on the mean shift trajectory of another (already processed) pixel. These pixels were directly associated with the mode to which the path converged. The approximation does not noticeably change the filtered image but reduces processing time.

5. Results

Post processing of the filtered data is usually necessary. To remove extraneous modes, modes which are close together are fused into one mode. For segmentation, each mode is considered a separate region. Finally, in an (optional) pruning step, modes which have few points are removed by assigning them to the closest mode with sufficient support. Typically, we prune modes with less than 20 points converging to it.

5.1. Synthetic Data Sets

The synthetic data was a two dimensional lattice of 3×3 orthonormal matrices with determinant one. Each lattice point had a value on the manifold (Lie group) of rotation matrices, $SO(3)$. The details of using mean shift over Lie groups are presented in [16, 22]. We perform mean shift over $\mathbb{R}^2 \times SO(3)$, as discussed in the previous section.

The results are shown in Figure 2. The original field is on the left. We show every 10th point in each direction of the 101×101 field. Zero-mean normal noise of variance 0.49 was added along the tangent space to generate the field in the middle. Filtering was done with a spatial bandwidth of $h_s = 7.0$ and a parameter bandwidth of $h_p = 1.5$. The filtered field is shown on the right.

5.2. Chromatic Noise Filtering

Chromatic image noise affects the direction (chromaticity) of the color vector and not its intensity. The direction of a 3D vector can be represented by a unit vector in 3D and these form the *Grassmann manifold*, $\mathbf{G}_{3,1}$. By filtering chromaticity we obtain better results than original mean shift which smooths chromaticity and brightness. The computational details of running nonlinear mean shift on Grassmann manifolds are in [17].

The results for the *baboon* image are shown in Figure 3. Chromatic noise of standard deviation 0.2 was added to the original image. The maximum distance between points on $G_{3,1}$ is 1.0 [16] and a standard deviation of 0.2 represents a 20% level of noise. The original mean shift image filtering algorithm from EDISON, with spatial bandwidth $h_s = 11.0$ and color bandwidth $h_p = 7.0$, was used to get the middle image. Using a larger h_p leads to oversmoothing and using smaller values does not change the image much. Our algorithm was run with $h_s = 11.0$ and $h_p = 0.7$ to get the image on the right. To clearly illustrate the difference in the results, two image regions, outlined in yellow in the input

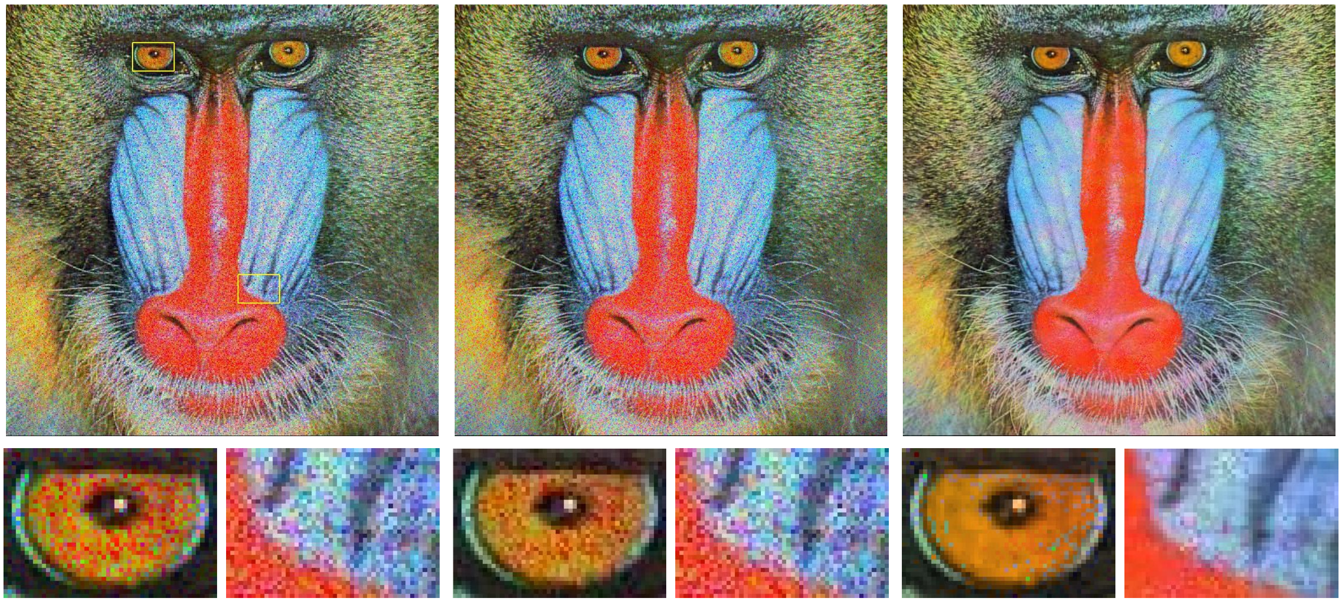


Figure 3. *Chromatic Noise Filtering*. The *baboon* image corrupted with chromatic noise is shown on the left. The results of using standard mean shift filtering with EDISON are in the middle and the results of our method are on the right.

image, are shown in close-up. Our filtering is clearly better than EDISON.

5.3. DT-MRI Filtering

Diffusion tensor imaging (DTI) [2, 11] is a widely used medical imaging method which measures the diffusivity of the water molecules in three dimensional space. Non-invasive reconstruction of connectivity in the brain is usually based on DTI techniques [12, 23]. The diffusivity is encoded as a 3×3 SPD matrix and the image is a 3D grid of 3×3 SPD matrices. Discontinuity preserving regularization of DT-MRI images is necessary to obtain a coherent diffusion map [20]. We smooth the DTI image by performing mean shift over $\mathbb{R}^3 \times Sym_3^+$.

The results for synthetic data are shown in Figure 4. The noisy ellipses are shown on top and the smoothed field is shown below.

Our real data set is a DTI of the human heart obtained from [10]. The lattice size is $128 \times 128 \times 67$ and we ran the smoothing with bandwidth values $h_s = 9.0$ and $h_p = 1.0$. For visualization purposes, each matrix is converted into a single scalar value and planes of the 3D lattice are drawn [21]. We use the *fractional anisotropy* given by

$$\sqrt{\frac{3}{2}\frac{(\lambda_1-\bar{\lambda})^2+(\lambda_2-\bar{\lambda})^2+(\lambda_3-\bar{\lambda})^2}{\lambda_1^2+\lambda_2^2+\lambda_3^2}} \qquad (15)$$

where, λ_1, λ_2 and λ_3 are the eigenvalues and $\bar{\lambda} = (\lambda_1 + \lambda_2 + \lambda_3)/3$. The fractional anisotropy for a particular plane $z = 47$ is shown in Figure 5.

6. Conclusions

We use the nonlinear mean shift algorithm to develop a new discontinuity preserving filtering technique for images where the data values do not lie in a vector space. PDE based methods are the standard way to achieve such smoothing. However, mean shift filtering has been found to be more effective than PDE based methods for vector data [1]. We expect a similar difference for nonlinear data sets and hope to make a rigorous comparison in the future. We also presented the details of using mean shift over Sym_n^+, the manifold of symmetric positive definite matrices.

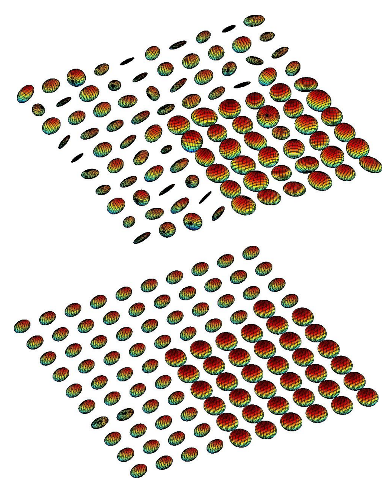
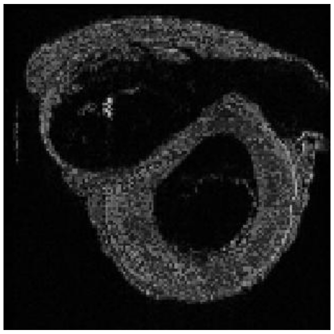


Figure 4. Synthetic DTI data before and after smoothing.



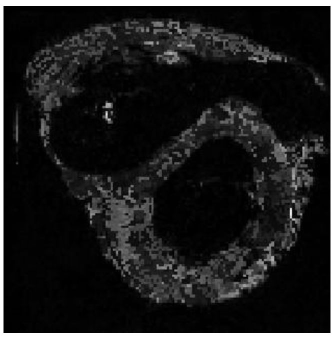


Figure 5. Real DTI data of a human heart before and after smoothing. The jitter in the top image is due to noisy voxels having different anisotropies from their surroundings. These are removed by the smoothing and more continuous regions of uniform anisotropy are visible below.

References

[1] D. Barash and D. Comaniciu, "A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift," *Image and Vision Computing*, vol. 22, no. 1, pp. 73–81, 2004.

[2] P. J. Basser, J. Mattiello, and D. LeBihan, "MR diffusion tensor spectroscopy and imaging," *Biophysical Journal*, vol. 66, pp. 259–267, 1994.

[3] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 2002.

[4] Y. Chikuse, *Statistics on Special Manifolds*. Springer, 2003.

[5] C. M. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Proc. 16th Intl. Conf. on Pattern Recognition,* Quebec, Canada, volume IV, 2002, pp. 150–155.

[6] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, pp. 603–619, May 2002.

[7] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[8] W. Forstner and B. Moonen, "A metric for covariance matrices," Technical report, Dept. of Geodesy and Geoinformatices, Stuttgart University, 1999.

[9] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Academic Press, second edition, 1990.

[10] P. A. Helm, R. L. Winslow, and E. McVeigh, "Center for cardiovascular bioinformatics and modeling," Technical report, Johns Hopkins University.

[11] D. LeBihan, J. F. Mangin, C. Poupon, C. A. Clark, S. Pappata, N. Molko, and H. Chabriat, "Diffusion tensor imaging: Concepts and applications," *Journal of Magnetic Resonance Imaging*, vol. 13, pp. 534–546, 2001.

[12] C. Lenglet, R. Deriche, and O. Faugeras, "Inferring white matter geometry from diffusion tensor MRI: Application to connectivity mapping," in *Proc. European Conf. on Computer Vision,* Prague, Czech Republic, volume IV, 2004, pp. 127–140.

[13] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian framework for tensor computing," *International J. of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.

[14] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, volume 2. Academic Press, 1982.

[15] P. Saint-Marc, J. S. Chen, and G. Medioni, "Adaptive Smoothing: A general tool for early vision," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, no. 6, pp. 514–529, 1991.

[16] R. Subbarao and P. Meer, "Nonlinear mean shift for clustering over analytic manifolds," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* New York, NY, volume I, 2006, pp. 1168–1175.

[17] R. Subbarao and P. Meer, "Subspace estimation using projection based M-estimators over Grassmann manifolds," in *Proc. European Conf. on Computer Vision,* Graz, Austria, volume I, May 2006, pp. 301–312.

[18] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *6th International Conference on Computer Vision*, (Bombay, India), January 1998, pp. 839–846.

[19] D. Tschumperle and R. Deriche, "Orthonormal vector sets regularization with PDE's and applications," *International J. of Computer Vision*, vol. 50, no. 3, pp. 237–252, 2002.

[20] D. Tschumperle and R. Deriche, "DT-MRI images: Estimation, regularization and application," in *Computer Aided Systems Theory - EUROCAST 2003*, 2003, pp. 531–541.

[21] D. Tschumperle and R. Deriche, "Vector-valued image regularization with PDEs: A common framework for different applications," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 4, pp. 506–517, 2005.

[22] O. Tuzel, R. Subbarao, and P. Meer, "Simultaneous multiple 3D motion estimation via mode finding on Lie groups," in *Proc. 10th Intl. Conf. on Computer Vision,* Beijing, China, volume 1, 2005, pp. 18–25.

[23] B. C. Vemuri, Y. Chen, M. Rao, T. McGraw, Z. Wang, and T. Mareci, "Fiber tract mapping from diffusion tensor MRI," in *In Proceedings of the IEEE Workshop on Variational and Level Set Methods*, 2001, pp. 81–88.