Diffusion Tensor Imaging Denoising

Xiang Hao

School of Computing, University of Utah, USA, hao@cs.utah.edu.

Abstract. This is an assignment report of Mathematics of Imaging course. The topic is diffusion tensor imaging denoising.

1 Introduction

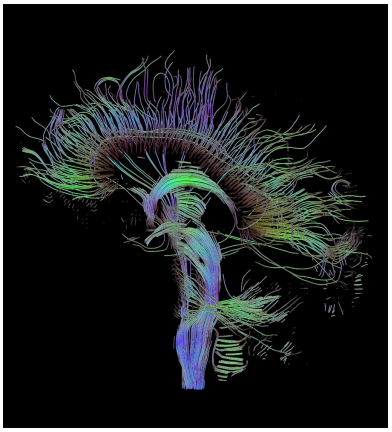


Fig. 1. Illustration of fiber tracking.

Diffusion tensor imaging (DTI) is a medical imaging modality that can reveal the underlying brain anatomy of the white matter in our brain. It also provides certain information about the brain connectivity. Thus, many algorithms have been proposed to find the brain connectivity based on DTI, such as deterministic fiber tracking (as shown in 1, probabilistic fiber tracking and geodesic tracking. DTI is estimated from diffusion weighted images (DWI), which has Rician distributed noises. Since DWI is noisy, the estimated DTI also has noises, so denoising is an important question. There are basically two types of denoising algorithms. The first type focuses on how to estimate better DTI from noisy DWI [3, 5, 1]. The second type denoise the DTI directly. One important work is done by Pennec et al. [4], who propose a Riemannian framework for tensor denoising, but they did not mention how to do the total variation based denoising. In this paper, we will focus on removing noises in DTI based on H1 regularization and total variation.

2 Method

In each grid of a DTI, there is a 3 by 3 positive definite symmetric matrix. So, a DTI can be thought as a function f from R^3 to $PD(3)$, where $PD(3)$ represents the space of 3 by 3 positive definite symmetric matrix. The partial derivative of F with respect to x at point $p(x, y, z)$ in R^3 is defined by

$$\frac{\partial f}{\partial x} = \frac{d(f(p + \epsilon(1, 0, 0)))}{d\epsilon}.$$

In finite difference, we usually set $d\epsilon = 1$ and approximate the above partial derivative with

$$\frac{\partial f}{\partial x} \approx f(p+(1,0,0)) - f(p) = f(x+1, y, z) - f(x, y, z) = \overrightarrow{f(x, y, z)f(x+1, y, z)},$$

Both $f(x + 1, y, z)$ and $f(x, y, z)$ are in $PD(3)$, so the difference between $f(x + 1, y, z)$ and $f(x, y, z)$ are not the Euclidean difference of each component of the matrices, and it is the Riemannian log map as shown in [2]. The Riemannian log map between from points p to q in $PD(3)$ is represented as $\overrightarrow{pq}$. The partial derivative of R with respect to y and z is defined in the similar way. Having the approximation of the partial derivatives, we can approximate other operator similarly, such as Laplace operator.

2.1 H1 Regularization

As shown in the previous image denoising project. In H1 regularization, the functional we want to minimize is

$$\min_u \quad \int_\Omega |\nabla_x u|^2 dx + \lambda \|f - u\|_2^2,$$

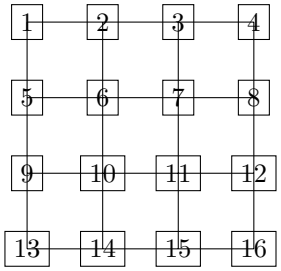
where, f is the original image and u is the restored image.

Using calculus of variation, we arrive at the Euler-Lagrange equations

$$\begin{aligned} &-\Delta u + \lambda(f - u) = 0 \quad \text{on}\Omega \\ &\langle \nabla u, v \rangle = 0 \quad on \quad \partial\Omega \end{aligned} \tag{1}$$

In (1), $f - u$ is just $\overrightarrow{uf}$. The Laplace operator can be approximated similarly.

Suppose we want to compute the Laplace on the following regular-square domain.



For example, for grid 6, the Laplace operator can be approximated by the second order finite difference scheme

$$\overrightarrow{u_6u_7} - \overrightarrow{u_5u_6} + \overrightarrow{u_6u_2} - \overrightarrow{u_{10}u_6} \tag{2}$$

The difficulty of the above approximation (3) is that the log maps are not in the same tangent space, so technically we can not directly compute the addition and the subtraction. We can move the log maps to the same tangent space, do the computation and then move the results back.

So, for grid 6, the first equation in (1) can be approximated by

$$-(\overrightarrow{u_6u_7} - (\overrightarrow{u_5u_6})_{u_6} + \overrightarrow{u_6u_2} - (\overrightarrow{u_{10}u_6})_{u_6}) + \lambda\overrightarrow{u_6f_6} \tag{3}$$

$(\overrightarrow{u_5u_6})_{u_6}$ represents the corresponding tangent vector in the tangent space of u_6.

There is another approximation. Pennec [4] show that the approximation (2) can also be approximated by

$$\overrightarrow{u_6u_7} + \overrightarrow{u_6u_5} + \overrightarrow{u_6u_2} + \overrightarrow{u_6u_{10}},$$

where all log maps are in the same tangent space.

In the end, for grid 6, the first equation in (1) can also be approximated by

$$-(\overrightarrow{u_6u_7} + \overrightarrow{u_6u_5} + \overrightarrow{u_6u_2} + \overrightarrow{u_6u_{10}}) + \lambda\overrightarrow{u_6f_6}$$

2.2 Total Variation

In Total Variation Primal form, the functional we want to minimize is

$$\min_u \quad \int_\Omega |\nabla_x u| dx + \lambda \|f - u\|_2^2,$$

We arrive at the Euler-Lagrange equations

$$\begin{aligned} &-\operatorname{div}(\frac{\nabla u}{|\nabla u|}) + 2\lambda(f-u) = 0 \quad \text{on}\Omega \\ &\langle \nabla u, v\rangle = 0 \quad on \quad \partial\Omega \end{aligned} \tag{4}$$

The approximation of $\operatorname{div}(\frac{\nabla u}{|\nabla u|})$ is similar to the approximation of Laplace operator. We just need to normalize ∇u before taking the divergence.

We also need to approximate the other components of the gradient in order to do the normalization. For example, have the partial derivative with respect to x, $\overrightarrow{u_6u_7}$, we need to approximate the partial derivative with respect to y. I approximate it with the average of $\overrightarrow{u_6u_2}$ and $(\overrightarrow{u_{10}u_6})_{u_6}$.

The approximation of the first equation in (4) is

$$-\left(\frac{\overrightarrow{u_6u_7}}{\|(\overrightarrow{u_6u_7},\frac{1}{2}(\overrightarrow{u_6u_2}+(\overrightarrow{u_{10}u_6})_{u_6}))\|}-\frac{(\overrightarrow{u_5u_6})_{u_6}}{\|((\overrightarrow{u_5u_6})_{u_6},\frac{1}{2}(\overrightarrow{u_6u_2}+(\overrightarrow{u_{10}u_6})_{u_6}))\|}\right.$$
$$\left.+\frac{\overrightarrow{u_6u_2}}{\|(\frac{1}{2}(\overrightarrow{u_6u_7}+(\overrightarrow{u_5u_6})_{u_6}),\overrightarrow{u_6u_2})\|}-\frac{(\overrightarrow{u_{10}u_6})_{u_6}}{\|\frac{1}{2}(\overrightarrow{u_6u_7}+(\overrightarrow{u_5u_6})_{u_6}),(\overrightarrow{u_{10}u_6})_{u_6}\|}\right)+2\lambda\overrightarrow{u_6f_6}$$

The norm of a log map X, $\|X\|$, is defined by the inner product on the $PD(3)$. In $PD(3)$, the inner product [2] of X and Y at point u in $PD(3)$ is defined by

$$\langle X,Y\rangle = tr(g^{-1}Xu^{-1}Y(g^{-1})^T),$$

where $gg^T = u$. Thus, $\|X\| = \sqrt{\langle X,X\rangle}$.

I define the norm of (X,Y) as

$$\|(X,Y)\| = \sqrt{\langle X,X\rangle + \langle Y,Y\rangle}$$

For both H1 regularization and total variation, the optimization is done by steepest gradient descent algorithm, which is the Riemannian exponential map in $PD(3)$.

3 Results

The H1 and Total Variation denoising algorithm is tested on real data as shown in figure 2.

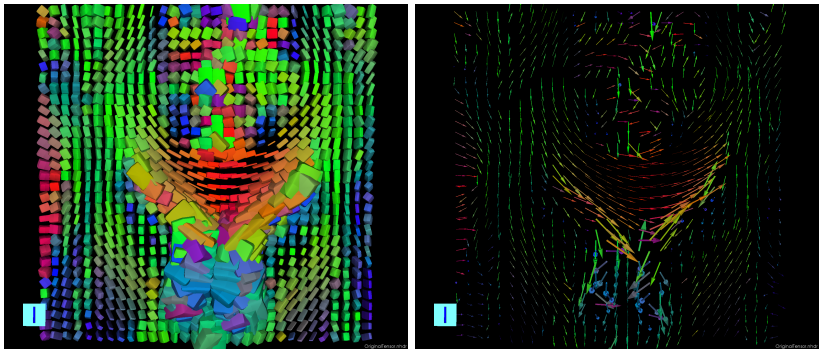


Fig. 2. Illustration of of the real data. The left picture is the DTI visualized by superquadrics. The right picture is the DTI visualized by the color coded principal eigenvector.

3.1 H1 Regularization

In figure 3, both the tensor fields and the vector fields look much more smoothing than the original data.This is not surprising since H1 denoising tends to blur the image.

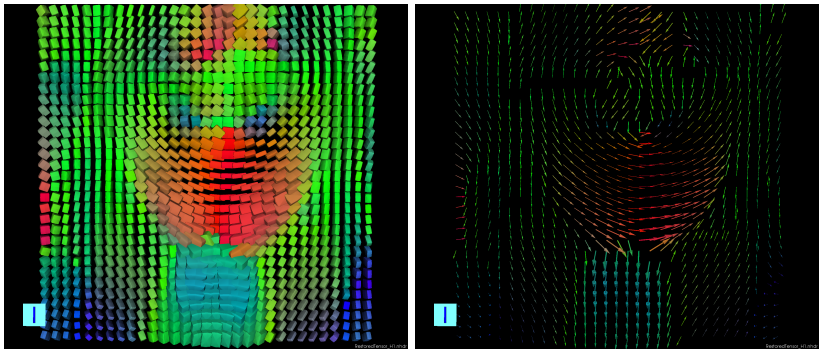


Fig. 3. Illustration of of the H1 denoising result. The left picture is the DTI visualized by superquadrics. The right picture is the DTI visualized by the color coded principal eigenvector.

3.2 Total Variation

The total variation based denoising results are shown in figure 4. We can see that the results are less blurring, which are expected.

Another slice of the results is shown in figure 5.

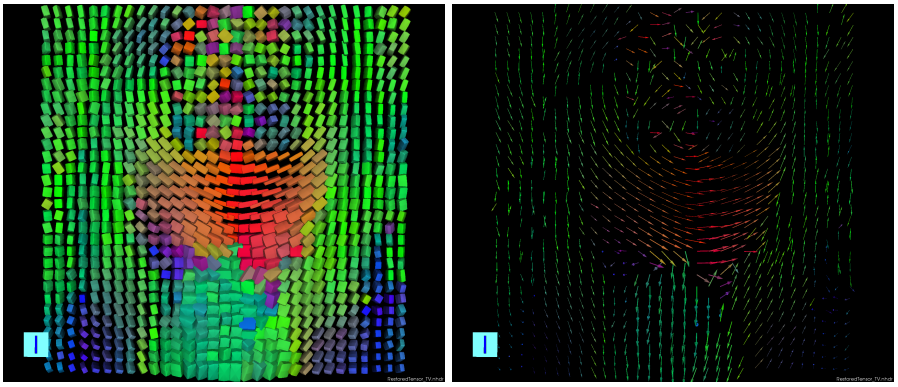


Fig. 4. Illustration of of the total variation denoising result. The left picture is the DTI visualized by superquadrics. The right picture is the DTI visualized by the color coded principal eigenvector.

4 Discussion

In H1 denoising, since the gradient has a large norm (could be infinite), the H1 denoising tends to blur the image. While in total variation, the gradient has small norm, and it can preserve the boundary. In the total variation results, we can find some artifacts. I am not clear what's happening, and I think there is a better way to approximate the total variation. It is also interesting to compare thest results with component-wise denoising.

References

1. S. Basu, T. Fletcher, and R. Whitaker. Rician noise removal in diffusion tensor mri. 9(pt. 1):117–125, 10 2006.
2. P. Thomas Fletcher and Sarang Joshi. Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Processing*, 87:250–262, 2007.
3. Geoffrey J.M. Parker, Julia A. Schnabel, Mark R. Symms, David J. Werring, and Gareth J. Barker. Nonlinear smoothing for reduction of systematic and random errors in diffusion tensor imaging. *Journal of Magnetic Resonance Imaging*, 11(6):702–710, 2000.
4. Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *Int. J. Comput. Vision*, 66:41–66, January 2006.
5. Zhizhou Wang, B.C. Vemuri, Y. Chen, and T.H. Mareci. A constrained variational principle for direct estimation and smoothing of the diffusion tensor field from complex dwi. *Medical Imaging, IEEE Transactions on*, 23(8):930 –939, aug. 2004.

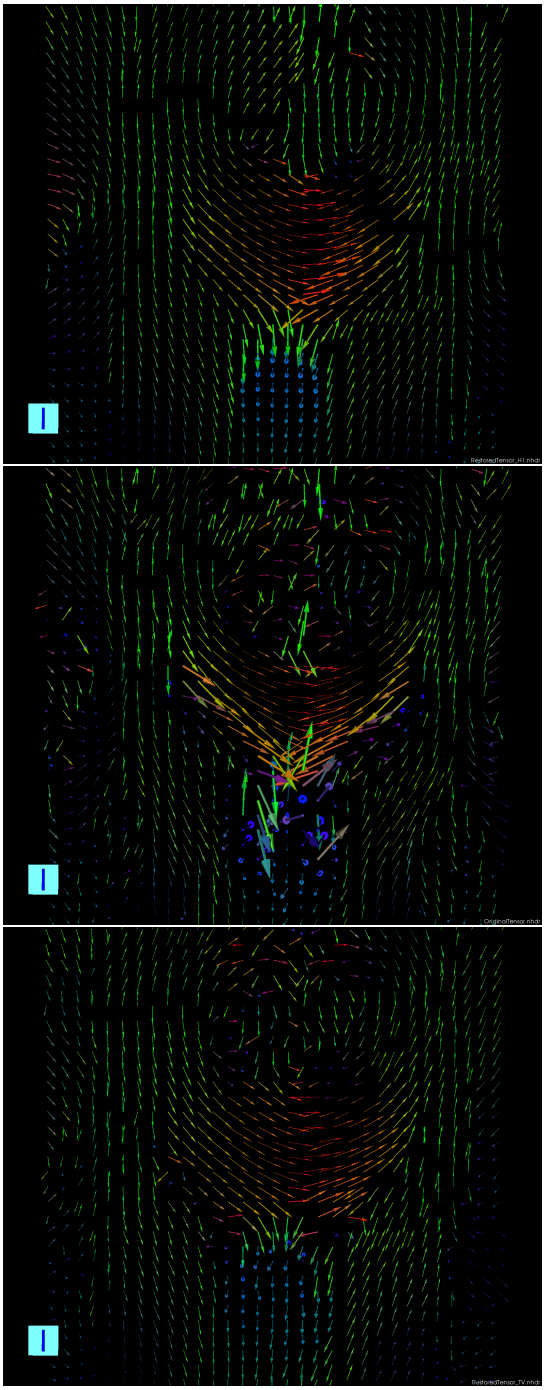


Fig. 5. Compare the original data (middle), the H1 denoising result (top), and the total variation denoising result (bottom). All images are visualized by the color coded principal eigenvector.