Similar NoE Tensor Workshop, Las Palmas, November 2006

MEDINRIA : DT-MRI PROCESSING AND VISUALIZATION SOFTWARE

Pierre Fillard, Nicolas Toussaint and Xavier Pennec

Asclepios Research Team, INRIA Sophia Antipolis, France.
Pierre.Fillard@sophia.inria.fr

ABSTRACT

Major advances in medical imaging raised the need for adapted methods and softwares. The recent emergence of diffusion tensor MRI (DT-MRI) was challenging since data produced by this modality are not simple grey-value images but complex diffusion tensor fields. Tensors are symmetric, positive definite matrices and suffer from a lack of adapted theoretical tools to manipulate them. In this paper, we propose two solutions to tensor computing. First, we endow the tensor space with an affine-invariant Riemannian metric and show how some well-known numerical schemes for scalar- or vector-valued images can be adapted to tensors. Second, we present the Log-Euclidean (LE) metrics as a more judicious choice as they are less computationally-intensive than the previous one. We show how LE metrics are implemented into a software called MedINRIA, and more especially in a module named DTI Track dedicated to DT-MRI processing. In this tool targeting the clinicians, we offer the full pipeline for DTI analysis, including diffusion tensor estimation, anisotropic tensor smoothing, and fiber tracking. A complementary module called TensorViewer is presented as a nice way to visually inspect the quality of tensor fields produced by DTI Track. This set of softwares, freely available on-line, offers extra features for further DTI analysis, like volumetric image visualization and fiber bundling.

1. INTRODUCTION

The increase and diversification of sources of information in medical imaging has raised the need for adapted tools for visualization and processing of these data. For instance, the progress made in magnetic resonance imaging (MRI) for the past two decades has prodigiously improved the quality and resolution of 3D images. Moreover, these images can take various form: anatomical MRI, functional MRI (fMRI) and diffusion tensor MRI (DT-MRI or DTI) are only a few examples. Among them, DT-MRI [4] is a relatively new imaging modality that has the ability to measure in vivo the anisotropy of water diffusion within tissues, and in the case of brain imaging, may eventually lead to white matter connectivity reconstruction. This new source of information requires adapted tools: we are not working only with 3D grey value images anymore, but with diffusion tensor fields.

Tensors are 3×3 symmetric, positive definite matrices. Working with those is arduous since the tensor space endowed with the classical matrix operations $(+, .)$ is not a vector space. Convex operations are stable (i.e., the mean of N tensors is a tensor), but one can quickly go out the tensor space with non-convex operations (like a simple subtraction), which is problematic in most applications.

However, in DTI, diffusion tensors are corrupted by noise during acquisition (MRI noise). One would like, for instance, to denoise the tensor image as a preprocess step. Moreover, one may want to resample it,

FREE DOWNLOAD: HTTP://WWW-SOP.INRIA.FR/ASCLEPIOS/SOFTWARE/MEDINRIA

segment it, etc. How can we simply adapt our favorite image processing algorithms working perfectly on scalar images to tensors?

In this paper, we propose two solutions to this problem, both related to Riemannian geometry. In Sec. 2, we describe how to endow the tensor space with an affine-invariant Riemannian metric, which has the advantage to turn the tensor space into a regular manifold where matrices with null and negative eigenvalues are at an infinite distance of any tensor. Then, we present a second family of metrics, called Log-Euclidean (LE), that have (almost) the same properties with a lower computational cost. In Sec. 3, we detail the implementation of the LE metrics that has been done in the software MedINRIA: We describe the features, the libraries used for programming as well as some implementation strategies. We conclude in Sec. 4 and discuss about future development.

2. METHODOLOGY

2.1. An Affine-Invariant Riemannian Metric for Tensors

The affine-invariant metric for tensors was originally proposed by several groups in about the same time: [12, 11, 10, 7]. It provides a framework to overcome the limitations of Euclidean calculus: it endows the tensor space with a highly regular structure, in which matrices with null or negative eigenvalues are at an infinite distance from any positive definite matrix. Moreover, the geodesic path between any two tensors is uniquely defined, leading to interesting properties such as the existence and uniqueness of the mean [12].

On Riemannian manifolds, geodesics realize a local diffeomorphism, called the exponential map, from the tangent space at a given point to the manifold itself. This allows us to (locally) identify points of the manifold with tangent vectors (see Fig. 1). With the invariant metric on tensors, the geodesic starting at Σ and with tangent vector W can be expressed simply with the classical matrix exponential and the (Riemannian) exponential map realizes a global diffeomorphism [12]. The inverse mapping, the logarithmic map, turns geodesics into straight lines (tangent vectors) in the tangent space. The expression of the exponential and logarithmic maps are:

$$\exp_\Sigma(W) = \Sigma^{1/2} \exp\Big(\Sigma^{-1/2} W \Sigma^{-1/2}\Big) \Sigma^{1/2} \quad \text{and} \quad \log_\Sigma(\Lambda) = \Sigma^{1/2} \log\Big(\Sigma^{-1/2} \Lambda \Sigma^{-1/2}\Big) \Sigma^{1/2}.$$

These two diffeomorphisms are the key to the numerical implementation and generalization to manifolds of numerous algorithms that work on a vector space. For example, the Euclidean gradient descent scheme $\Sigma_{t+1} = \Sigma_t - \varepsilon \nabla C(\Sigma_t)$ where C is a criterion to minimize, could easily lead to a non-positive matrix. We advantageously replace it by the *geodesic marching scheme* $\Sigma_{t+1} = \exp_{\Sigma_t}(-\varepsilon\, \nabla C(\Sigma_t))$.

For completeness, we give the expression of the affine-invariant distance between two tensors:

$$\text{dist}^2(\Sigma_1, \Sigma_2) = \text{trace}\left(\left[\log\left(\Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2}\right)\right]^2\right). \tag{1}$$

The affine-invariant metric overcomes the defects of the Euclidean calculus, but with a high computational cost: It makes an intensive use of the matrix exponential, logarithm, inverse and square root. The Log-Euclidean metrics are a very attractive alternative, as they have almost the same invariant properties with a lower computational cost.

2.2. Log-Euclidean Metrics for Tensors

Log-Euclidean (LE) metrics [3] are a novel family of Riemannian metrics, which combines the benefits of the affine-invariant family with the lower computational cost of the Euclidean calculus. The price to pay

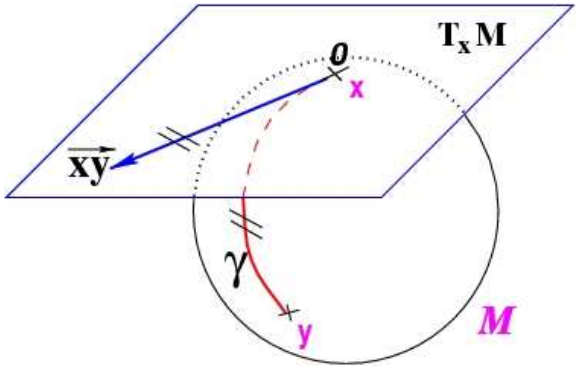

Fig. 1. **Geodesic and tangent space in a Riemannian manifold.** The geodesic joining x and y is a straight line in the tangent space T_xM. The distance is conserved, i.e. the length of the geodesic is equal to the norm of the vector xy. The operation that turns the geodesic into a straight line is called the **logarithmic map**. The inverse mapping is called the **exponential map**.

for that is rather cheap: LE metrics are only similitude-invariant (i.e. invariant under rigid body transforms, including translations and rotations).

The basic idea is to take the matrix logarithm L of a tensor D: $L = \log(D)$, and to run computations on L. The new processed value $\tilde{L}$ obtained is turned back into a tensor by taking the matrix exponential: $\tilde{D} = \exp(\tilde{L})$. The expression of the L_2 LE metric is simply:

$$\text{dist}^2\left(\Sigma_1, \Sigma_2\right) = \text{Trace}\left(\left(\log\left(\Sigma_1\right) - \log\left(\Sigma_2\right)\right)^2\right).$$

We proved that it yields to excellent theoretical properties, such as the monotone interpolation of the determinants, and the prevention of the swelling effect (more details can be found in [3]).

It is now straightforward to extend any vector-valued image processing algorithm to tensors: one simply needs to perform algorithms on the tensor logarithms, and to exponentiate the result to obtain a tensor. Surprisingly, results with this metric are very similar to the affine-invariant family, as shown in [3]. The results are so close that it is almost impossible to quantify the difference.

In the next section, we exemplify the difference between the affine-invariant and LE metrics with the computation of the mean value.

2.3. Example with the Mean Value

Many image processing algorithms can be handled by a simple mean (or weighted mean) computation: linear interpolation, radial basis functions, Gaussian filtering can all be performed using weighted means. The key idea for an extension of these algorithms to tensors is to generalize the mean value computation.

Let $\Sigma_1 \ldots \Sigma_N$ be a set of measures of the same tensor. For manifolds, the Fréchet mean is the set of tensors Σ (it may not be unique) minimizing the sum of squared distance: $C(\Sigma) = \sum_{i=1}^{N} \text{dist}^2(\Sigma, \Sigma_i)$. We investigate the result in the affine-invariant and Log-Euclidean cases.

2.3.1. The Affine-Invariant Case

In the case of the affine-invariant metric, there is not cut locus, so that there is one and only one mean value $\bar{\Sigma}$ [12]. Moreover, a necessary and sufficient condition for an optimum is a null gradient of the criterion. Differentiating one step further, we obtain a constant Hessian matrix. Thus, the intrinsic second order Newton gradient descent algorithm gives the following mean value at estimation step $t+1$:

$$\bar{\Sigma}_{t+1} = \exp_{\bar{\Sigma}_t}\left(\frac{1}{N}\sum_{i=1}^{N}\log_{\bar{\Sigma}_t}(\Sigma_i)\right) = \bar{\Sigma}_t^{\frac{1}{2}}\exp\left(\frac{1}{N}\sum_{i=1}^{N}\log\left(\bar{\Sigma}_t^{-\frac{1}{2}}\Sigma_i\bar{\Sigma}_t^{-\frac{1}{2}}\right)\right)\bar{\Sigma}_t^{\frac{1}{2}} \tag{2}$$

Notice that we cannot easily simplify more this expression as in general the data Σ_i and the mean value $\bar{\Sigma}_t$ cannot be diagonalized in a common basis. However, this gradient descent algorithm usually converges very fast (about 10 iterations).

2.3.2. The Log-Euclidean Case

With Log-Euclidean metrics, the Fréchet mean takes a unique closed-form. As we are working on the tensor logarithm, one simply writes:

$$\bar{\Sigma} = \exp\left(\frac{1}{N}\sum_{i=1}^{N}\log\left(\Sigma_i\right)\right).$$

Note that in both cases, the mean is in fact the generalization of the geometric mean, and not the arithmetic mean. We showed in [12, 3] that it leads to excellent theoretical properties in case of diffusion tensors, and more generally covariance matrices: linear interpolation of the tensor volumes (see Fig. 2), smooth interpolation of the eigenvectors, symmetry w.r.t the identity (a tensor and its inverse are at equal distance to the identity). However, it may not be the correct choice when dealing for instance with structure tensors as shown in [5], as the null eigenvalue is possible and desirable (it means a perfect edge in one direction). This raises the question of which metric for which application.

We now describe the implementation of Log-Euclidean metrics in the software MedINRIA.

3. IMPLEMENTATION OF MEDINRIA

The initial goal of MedINRIA is to provide the Asclepios team with a front end software for all algorithms developed internally. Each teammate willing to contribute can develop a small application and include it in MedINRIA. Each application is called a module, and is loaded at the execution. Up to now, two modules are available: one for Log-Euclidean DT-MRI processing (DTI Track), and one for visualizing tensors (TensorViewer).

The development of MedINRIA is made with four libraries: MIPS (Medical Image Processing and Simulation), which is our homemade image processing library (no public release yet), the Insight ToolKit (ITK [8]), the Visualization ToolKit (VTK [1]), and wxWidgets [2] for the user interface. The motivation behind the choice of the user interface library (wxWidgets) was to have a C++ and L-GPL library, and have native widgets on any platform, which eliminates others like GTK, FLTK and Qt.

Fig. 2. **Linear interpolation of two tensors.** Left: interpolated tensors. Right: graphs of the determinants (volumes) of the interpolated tensors. Top: linear interpolation on coefficients (Euclidean calculus). Middle: affine-invariant interpolation. Bottom: Log-Euclidean interpolation. Note the characteristic swelling effect observed in the Euclidean case due to a parabolic interpolation of determinants. This effect is not present in both Riemannian frameworks since determinants are monotonically interpolated. Note also that Log-Euclidean means are more anisotropic than their affine-invariant counterparts.

3.1. DTI Track: Log-Euclidean DT-MRI Processing

Originally, DTI Track was an application dedicated to DT-MRI processing already available on-line[1]. Once at INRIA, I decided to completely re-write to include new features, like Log-Euclidean processing.

The first issue was to have a tool flexible to any DTI sequence: it should be possible to define an arbitrary number of encoding gradients, as well as arbitrary gradient directions. The patient orientation should be taken into account too. For these reasons, we developed a file format, the DTI Study (.dts), which is nothing more than a text file summarizing all necessary information of a DTI dataset: files, encoding gradients and possible flips in image orientation.

We provide a wizard to guide users importing their data into DTI Track. Once the importation is done, the DTI processing algorithms can be launched. They consist in: diffusion tensor field estimation, non-positive tensors (NPT) removal, Log-Euclidean anisotropic smoothing, tensor-derived coefficient calculation (FA, color FA, ADC), and fiber tracking. Each of them is briefly described below.

3.1.1. Algorithms

The tensor estimation is done with a least-squares on the linearized version of the Stejskal & Tanner diffusion equation [4]. The only parameter is a threshold on the baseline image (or b_0 image: diffusion weighted mage without an encoding gradient) to avoid tensor estimation outside the brain.

[1] http://www-sop.inria.fr/asclepios/personnel/Pierre.Fillard/sofwares

Right after the estimation, the NPT removal algorithm is run: if a NPT is found, it is replaced by the LE mean of its direct positive neighbors. The NPT removal step is important, as we need to feed the smoothing algorithm with tensors that are all positive definite.

The Log-Euclidean smoothing is optional. However, we showed in [3, 6] that it can extremely enhance the quality of the tensor field, and largely improve fiber tracking. Our implementation is an iterative process based on Perona-Malik diffusion scheme: a tensor at a given position will receive contributions from its direct neighbors depending on the norm of the spatial gradient at the neighbor position. For high gradients, the contribution will be small (one does not want to diffuse through edges). On the contrary, low gradient values mean low variations, and neighbor tensors will have a high impact on the result. Figure 3 illustrates the tensor smoothing with MedINRIA.

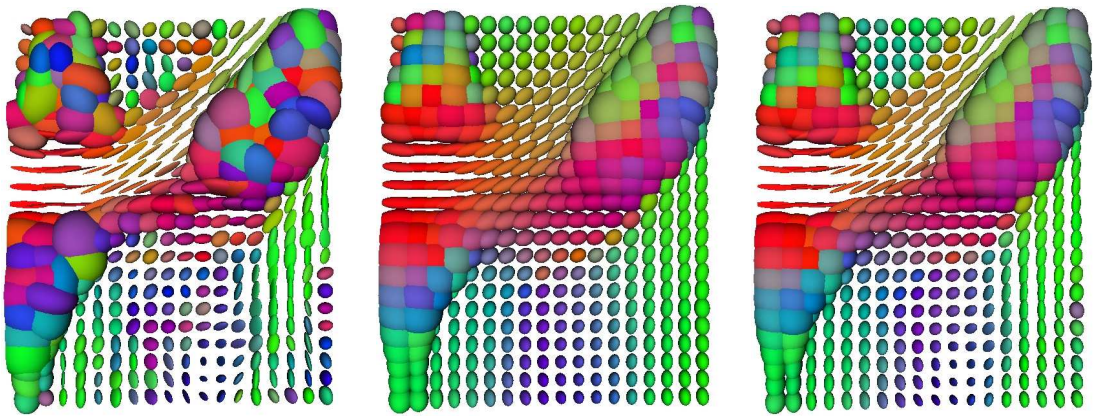

Fig. 3. **Anisotropic Log-Euclidean smoothing of a tensor field implemented in MedINRIA.** Left: close-up on a slice containing part of the left ventricle and nearby. Middle: Euclidean regularization. Right: Log-Euclidean regularization. Note that there is no tensor swelling in the LE case. On the contrary, in the Euclidean case, a swelling effect occurs almost everywhere (except maybe in the ventricles), in particular in regions of high anisotropy.

Fiber tracking is also available, using algorithm in [13]. We use a Log-Euclidean linear interpolation of the tensor field to have access to a tensor value at any continuous position of the grid. This allows to reconstruct smoother tracts, more especially when the out-plane resolution of the 3D images is low. See figures 4 for an example of fiber reconstruction with MedINRIA before and after tensor smoothing.

These algorithms are coded using ITK. It means that each of them is an ITK filter, creating an ITK pipeline when plugged together. If one parameter is changed, the pipeline will be re-computed starting by the modified filter. On the contrary, if a user launches several times the same processing without changing any parameter, nothing will be done. For optimization reasons, we wrote a very efficient tensor class, which can be interfaced for eigen decomposition with the vnl library (ITK numerical library), the MKL (Mathematic Kernel Library - Blas/Lapack routines optimized for Intel processors), or the ACML (same for AMD processors). MKL and ACML implementation gives eigen decompositions up to 4 times faster than the vnl. Last but not least, each filter is multithreaded, making full use of new intel and AMD processors.

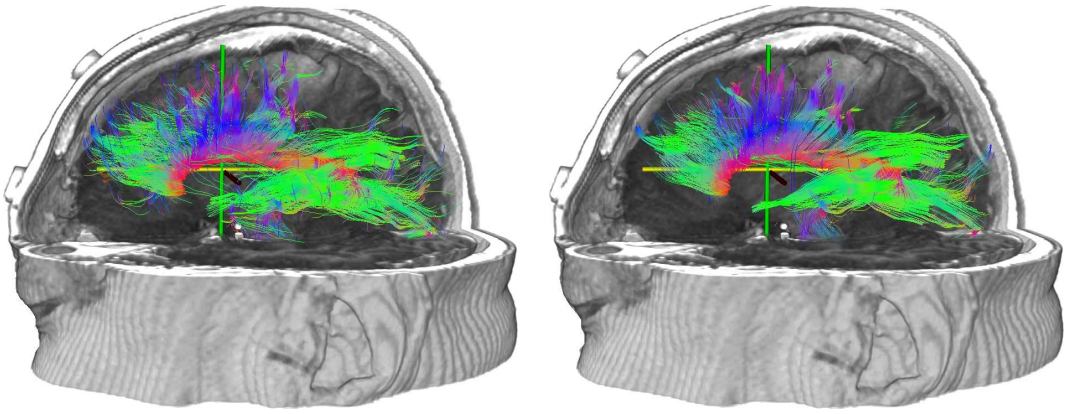

Fig. 4. **Effect of the LE smoothing on fiber tracking results.** Left: Fiber tracking without prior regularization of the tensor field. Right: Same reconstruction after anisotropic LE smoothing. Fibers look smoother and show less dispersion.

3.1.2. Interactive Fiber Bundling

Lots of effort was also made to give users the maximum possible interaction to extract a fiber bundle of interest (BOI). Two methods are available: the user can either define 2D regions of interest (ROIs), where he wants fibers to pass through, or he can interactively translate and resize a 3D cropping box (CP) to limit the fibers to those that go through it (Fig. 5). The two methods are linked, i.e. one can define a BOI using ROI drawings, and refine the result with the CP. Once the user is satisfied with the BOI, he can label it, color it, name it and save it into the DTI study.

3.1.3. Visualization

VTK offers several possibilities to visualize 3D items:

- 3D images are rendered using either multi-planar reconstruction (MPR) or volume rendering (VR) techniques (Fig. 6).
- Fibers are displayed using lines, ribbons or tubes (Fig. 7).
- ROIs are visualized as isosurfaces (Fig. 5 right).

3.2. TensorViewer

Visualizing tensor fields produced by DTI Track is important for quality-control. A common issue is when the image orientation does not match the patient orientation: it may result in a wrong geometry of the tensors, i.e., they may be flipped left-right, up-down or antero-posterior. The TensorViewer module allows to control and correct for disoriented tensors (Fig. 8).

The 3D tensor field is considered as an image: it can be visualized slice by slice, in three orientations (namely axial, coronal and sagittal) and downsampled for faster rendering. It also gives the user the possibility to flip tensors along the three axes, enabling geometry correction. User can save the flipped tensor field, and re-open it into DTI Track for further processing.

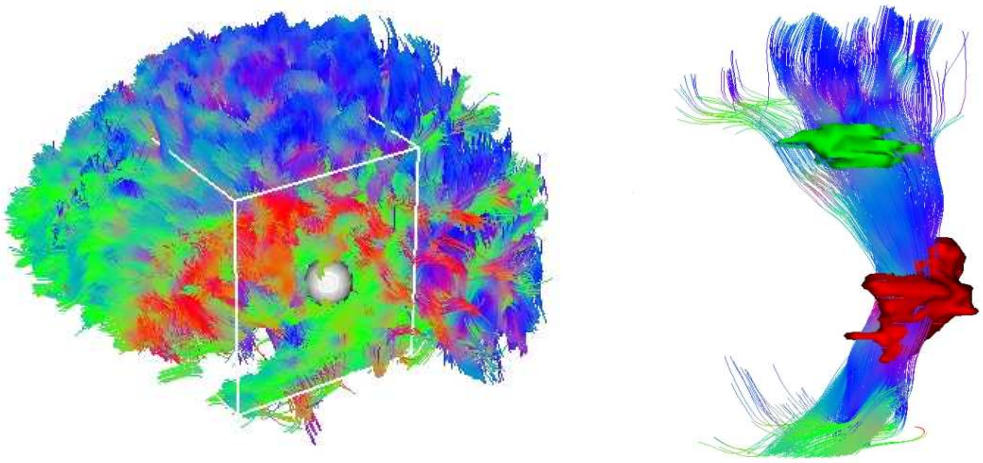

Fig. 5. **Example of fiber bundling with MedINRIA.** Left: The complete set of reconstructed fibers. The white box displayed is called the cropping box. Only fibers that go through that box are shown. Right: Two ROIs were defined (in red and green): Only fibers going through the two ROIs are displayed.

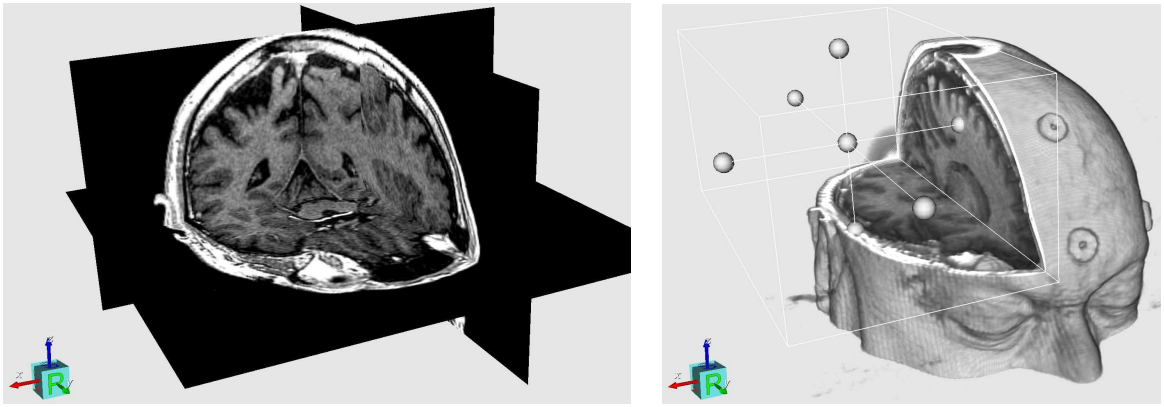

Fig. 6. **Image visualization techniques in MedINRIA.** Left: Multi planar reconstruction (MPR). Right: Volume rendering technique. The white box is a cropping box to manually extract a volume of interest by resizing and translating it.

The tensor shape is rendered using VTK and the vtkTensorGlyph filter. This filter takes as input a tensor image and a geometric primitive. This primitive is then copied at each position of the grid, and transformed according to the tensor at this location. Thus, it is aligned and scaled with the tensor eigenvectors and eigenvalues. The primitives used are various: lines (corresponding to the major tensor eigenvector), arrows, disks, cubes, cylinders, spheres and super quadrics are coded in TensorViewer (Fig. 9). Each tensor is color coded by its major eigenvector: absolute values of the coordinates are considered as RGB. This gives an

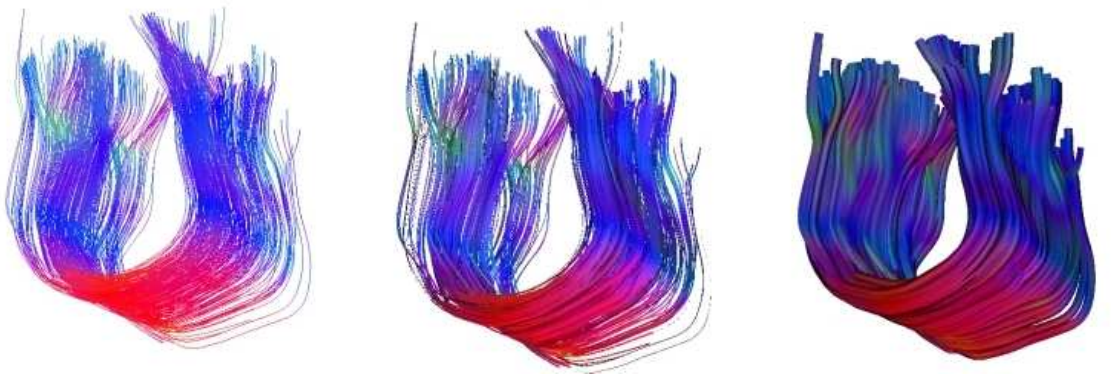

Fig. 7. **Fiber rendering techniques of MedINRIA.** Left: Streamlines. Middle: Ribbons (flat surfaces). Right: Streamtubes.

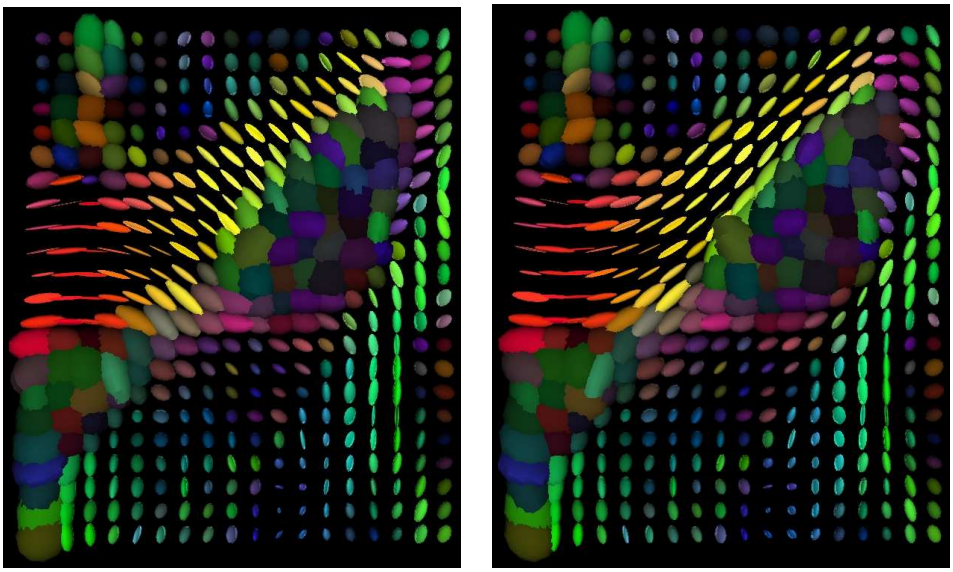

Fig. 8. **Tensor geometry correction with MedINRIA.** Left: The tensor field, estimated by DTI Track, looks flipped left-right. Right: We applied a flipping to correct the geometry. The tensor field can be saved once flipped.

overview of the main direction of tensors: red is for a left-right oriented tensor, blue for up-down and green for antero-posterior.

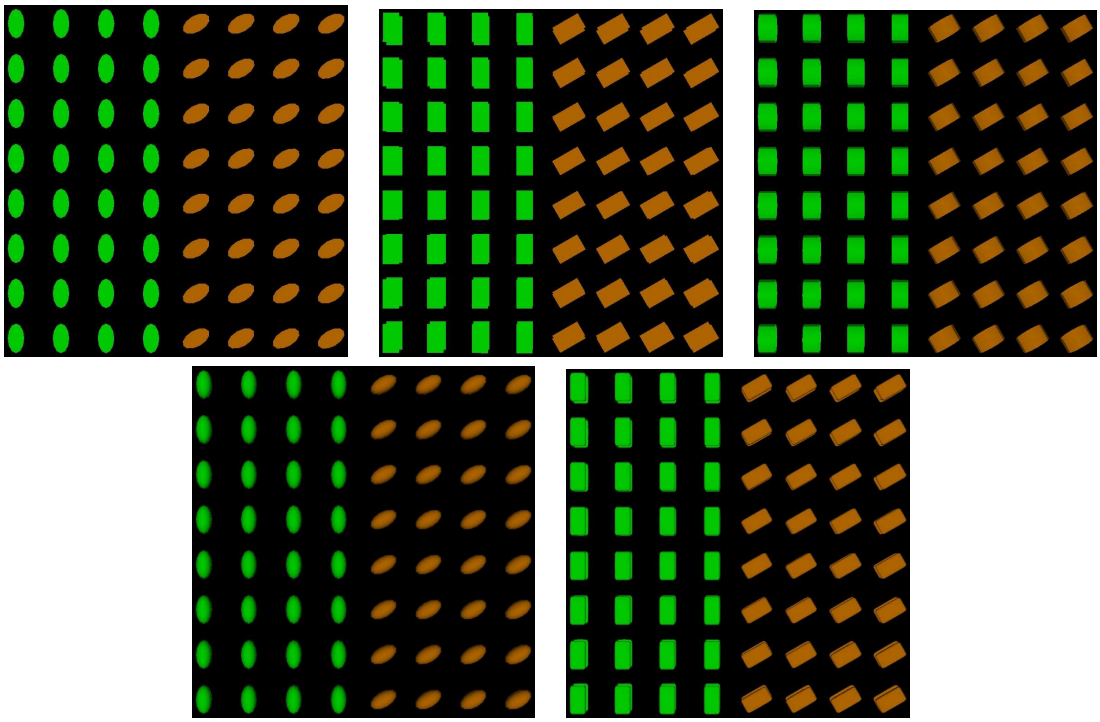

Fig. 9. **Tensor representations available in MedINRIA.** From top to bottom, left to right: disks, cubes, cylinders, spheres and super quadrics.

4. CONCLUSION

This paper presents a brief overview of two Riemannian frameworks for tensor computing. In order to overcome the defect of the standard Euclidean calculus (possible negative eigenvalues, swelling effect), we chose to use the Riemannian geometry and consider the tensor space as a manifold and not a simple vector space. First, we propose to endow the tensor space with an affine-invariant metric, which gives it a regular structure, rejecting any matrix with null or negative eigenvalues at an infinite distance from any tensor. The exponential and logarithmic maps are presented as the keys for a practical extension of image processing algorithms to tensors. However, the computational cost of this metric may be high, due to the use of the matrix exponential, logarithm, square root and inverse. Another family of Riemannian metrics, called Log-Euclidean, are much easier to use: they simply consist in running computations on the tensor logarithm and exponentiating the result. This straightforward principle is the key for a direct generalization of many image processing algorithms to diffusion tensors. Even if any type of tensors is covered by these frameworks, one needs to question if the metric is the correct choice for the application. For example, the proposed metrics may not be the best choice to process structure tensors, as the null eigenvalue needs to be reachable.

The software MedINRIA presented in this paper is a collection of applications dedicated to medical image computing. The DTI Track module is a concrete implementation of DT-MRI processing using LE metrics, providing all necessary tools for diffusion tensor estimation, LE anisotropic filtering, tensor-derived coefficients computation and fiber tracking. Using ITK coding standards allows a pipelined architecture, as

well as a multithreaded execution. Last but not least, MedINRIA can handle any DTI sequence, thanks to a homemade format, and provides intuitive interactions to extract a fiber bundle of interest. The TensorViewer module is complementary to DTI Track. Tensor fields are displayed using various primitives, slice by slice, as it is done for regular images. Tensors can be flipped among the three basis axes, enabling to correct for defects in tensor geometry due to patient orientation issues.

Future work includes a better flexibility in the DTI sequence to handle acquisitions with non-constant b-values. Other types of diffusion tensor estimations assuming other noise models [6] could be incorporated to improve tensor fields quality. Tensor visualization should be improved, for instance by using super quadrics as in [9].

5. REFERENCES

[1] The Visualization ToolKit: http://www.vtk.org/.

[2] wxWidgets: http://www.wxwidgets.org/.

[3] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421, August 2006.

[4] P. Basser, J. Mattiello, and D. Le Bihan. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, 66:259–267, 1994.

[5] Pierre Fillard, Vincent Arsigny, Nicholas Ayache, and Xavier Pennec. A Riemannian framework for the processing of tensor-valued images. In Ole Fogh Olsen, Luc Florak, and Arjan Kuijper, editors, *Deep Structure, Singularities, and Computer Vision (DSSCV)*, number 3753 in LNCS, pages 112–123. Springer Verlag, June 2005.

[6] Pierre Fillard, Vincent Arsigny, Xavier Pennec, and Nicholas Ayache. Clinical DT-MRI estimation, smoothing and fiber tracking with log-Euclidean metrics. In *Proceedings of the Third IEEE International Symposium on Biomedical Imaging (ISBI 2006)*, pages 786–789, Crystal Gateway Marriott, Arlington, Virginia, USA, April 2006.

[7] P.T. Fletcher and S.C. Joshi. Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors. In *Proc. of CVAMIA and MMBIA Workshops, Prague, Czech Republic, May 15, 2004*, LNCS 3117, pages 87–98. Springer, 2004.

[8] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-10-6, http://www.itk.org/ItkSoftwareGuide.pdf, first edition, 2003.

[9] G. Kindlmann. Superquadric tensor glyphs. In *IEEE TCVG Symposium on Visualization*, pages 147–154, May 2004.

[10] C. Lenglet, M. Rousson, and R. Deriche. A statistical framework for DTI segmentation. In *Proc. IEEE Intl. Symposium on Biomedical Imaging*, pages 794–797, April 2006.

[11] M. Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appli.*, 2005. Accepted for publication.

[12] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, January 2006.

[13] David M. Weinstein, Gordon L. Kindlmann, and Eric C. Lundberg. Tensorlines: Advection-diffusion based propagation through diffusion tensor fields. *vis*, 00:40, 1999.