

Efficient Tracking of MR Tensor Fields Using a Multilayer Neural Network

L.M. San-José-Revuelta, M. Martín-Fernández and C. Alberola-López *

Abstract—In this paper, a recently developed fiber tracking algorithm to be used with diffusion tensor (DT) fields acquired via magnetic resonance imaging (MRI) is improved and applied to real brain DT-MR images. The method performs satisfactorily in regions where branching and crossing fibers exist and offers the capability of reporting a probability value for the computed tracts. This certainty figure takes into account both the anisotropy and the information provided by *all* the eigenvectors and eigenvalues of the diffusion matrix at each voxel. In previous papers of the authors, a simpler algorithm was applied only to elementary synthetic DT-MR images. As now presented, this algorithm is now adequately used with more intricate synthetic images and is applied to real white matter DT-MR images with successful results. Besides, the paper presents a novel neural network that is used to estimate the crucial parameters of the algorithm. Numerical experiments show a performance gain over previous approaches, specially with respect to convergence and computational load. The tracking of white matter fibers in the human brain will improve the diagnosis and treatment of many neuronal diseases.

Keywords: diffusion tensor magnetic resonance imaging, fiber tracking, neural network

1 Introduction

The technique of Diffusion Tensor Magnetic Resonance Imaging (DT-MRI) measures the diffusion of hydrogen atoms within water molecules in 3D space. Since in cerebral white matter most random motion of water molecules are restricted by axonal membranes and myelin sheets, diffusion anisotropy allows depiction of directional anisotropy within neural fiber structures [1, 5].

The DT-MRI technique has raised great interest in the neuro-science community for a better understanding of the fiber tract anatomy of the human brain and several methods have already been reported to use DT-MRI data for tracking nerve fibers and derive connectivity between different parts of the brain. Though this field of search is still in its early stages, its development is growing very

fast during the last decade.

There exist many important applications for white matter tractography and more will appear in the next future as DT-MRI and fiber tracking are becoming standard clinical procedures. Among their most important applications we can find: brain surgery (knowing the extension of the fiber bundles could minimize the functional damage to the patient), white matter visualization using fiber traces (for a better understanding of brain anatomy) and inference of connectivity between different parts of the brain (useful for functional and morphological research of the brain).

Apart from a very few approaches for direct volume rendering [9], the great majority of DT-MRI visualization techniques focuses on the integration of sample points along fiber trajectories and their three-dimensional representation [11]. These streamline-based approaches usually make only use the principal eigenvector of the diffusion ellipsoid as an estimate of the predominant direction of water diffusion in a voxel [5].

However, due to both some deficiencies in these tracking methods and several shortcomings inherent in datasets (such as noise, artifacts or partial voluming), these algorithms may depict fiber tracts which do not exist in reality or miss to visualize important connectivity features, e.g. branching structures. In order to avoid misinterpretations, the viewer must be provided with some information on the uncertainty of every depicted fiber and of its presence in a certain location. In [15, 16] we proposed an estimation algorithm that takes into account the whole information provided by the diffusion matrix, i.e., it does not only consider the principal eigenvector direction but the complete 3D information about the certainty of continuing the path through every possible future direction. Since in [15, 16] we presented an initial version of our work, numerical simulations were only performed on simple DT-MR synthetic images.

The proposed algorithm also includes a procedure that on-line adapts the number of *offspring paths* emerging from the actual voxel, to the degree of anisotropy observed in its proximity. This strategy has been proved to enhance the estimation robustness in areas where multiple fibers cross while it keeps complexity to a moderate

*Dept. Teoría de la Señal y Comunicaciones e Ingeniería Telemática. E.T.S.I. Telecomunicación, University of Valladolid, 47010 Valladolid, Spain. Tel: +34-983-423660, Fax: +34-983-423667. Email: lsanjose@tel.uva.es.

level. However, the main drawback of this approach is that the parameters of the algorithm must be heuristically adjusted every time that: (i) a new patient is scanned, (ii) a different volume is processed, or (iii) the scanning conditions vary. For this reason, a neural Network (NN) is here proposed to adjust the parameters of the algorithm in a user-directed training stage. This issue has been traditionally undertaken under an heuristically approach and, to the best of our knowledge, no previous work has dealt with the initial tuning of any kind of DT-MRI tracking algorithm. On the other hand, the detailed study of convergence depending on the strategy used to create the pool of “future seeds” is new in this paper, as well.

The rest of the paper is organized as follows: first, Section 2 briefly summarizes the review of related work. Next, Section 3 describes the tracking algorithm including the main equations as well as the physical interpretation of the parameters and the strategy proposed to deal with regions where multiple fiber bundles cross or divide themselves (by making use of the local anisotropy). Section 4 is devoted to describe the NN scheme proposed for estimating the principal parameters of the tracking algorithm, including a detailed graphical description of this neural system. The paper finishes with the numerical results, the conclusions and the outline of the future lines of research.

2 Review of previous related approaches

DT-MRI fiber-tracking algorithms commonly use the principal eigenvectors as estimates of the fiber orientation and the linear diffusion anisotropy values at each voxel in order to guide the tracking process. Several seed points are selected within a region of interest at voxels where a predetermined anisotropy threshold is reached. For each seed point a fiber is tracked in both directions by following the local vector orientation. If we step from voxel to voxel using the discrete vector at each voxel, the tracked line will increasingly differ from the ideal line with each further step. Therefore, a more continuous approach is required [5]. The general solution for the vector field integration is the Runge-Kutta approach [17], which allows estimation of the next sample on the line by computing a weighted average of vector orientations. In [10], Mori et al. proposed a method which traces the line on a subvoxel level, linearly interpolating the principal eigenvectors as well as the anisotropy values from the neighboring voxels. In order to avoid non-realistic connections, the angle between the vector orientation of the sample and a predefined angle are compared. However, even the selection of high thresholds cannot ensure the production of absolutely correct results [13]. Coulon et al. [3, 4] proposed a field regularization technique for principal eigenvector diffusion fields.

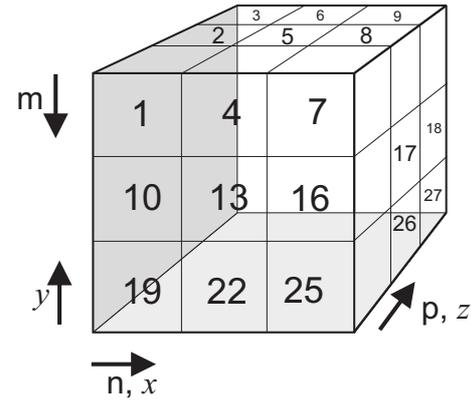


Figure 1: Cubic structure and reference criteria for the axes indexes (x, y, z). Central point is V_c (No. 14).

On the other hand, Poupon et al. [14] base their regularization technique on a low curvature hypothesis for fascicles. More recently, Pasternak et al. [12] proposed to use a multiple diffusion tensor approach, where the regularization algorithm results in multiple orientation describing each voxel. Other approaches include the tensorline approach [19] and the exhaustive search approach [2, 18]. The former uses the incoming vector from the last tracking step as an estimate and reflects it according to the tensor shape in the current voxel. The exhaustive search approach by Conturo et al. [2], initiates tracking using all brain voxels as seed candidates, thus generating a great number of fiber lines. Afterwards, only those fibers which penetrate the branching region of interest are kept. In 2004, a similar approach with seeds that are evenly distributed over the volume was proposed by Vilanova et al. [18].

3 Tracking Algorithm Description

3.1 Problem description and basic concepts

The basic version of the algorithm used in this paper was first described in [15]. Thus, this section just presents a summary of the method, with special emphasis on the new aspects. The algorithm uses probabilistic criteria and iterates over several points in the analyzed volume (the points given by the highest probabilities in the previous iteration). The process starts in a user-selected seed voxel, V_0 , and, at every iteration, it evaluates a set of parameters related to the central voxel of a cubic structure consisting of $3 \times 3 \times 3 = 27$ voxels, similar to that shown in Figure 1.

The central point, V_c , (No. 14 in the figure) represents the last point of the tract being analyzed. In the first iteration, $V_c = V_0$. Obviously, there exist 26 possible directions to take for the next iteration in order to select the next point of the tract. Once V_c is selected, the previous point and all those points exceeding the limits of

	mnp	$\pi_m\pi_n\pi_p$		mnp	$\pi_m\pi_n\pi_p$		mnp	$\pi_m\pi_n\pi_p$
	↓ ↓ ↓	↓ ↓ ↓		↓ ↓ ↓	↓ ↓ ↓		↓ ↓ ↓	↓ ↓ ↓
1°)	111	(---)	10°)	211	(0--)	19°)	311	(+--)
2°)	112	(--0)	11°)	212	(0-0)	20°)	312	(+-0)
3°)	113	(--+))	12°)	213	(0-+)	21°)	313	(+++))
4°)	121	(-0-)	13°)	221	(00-)	22°)	321	(+0-)
5°)	122	(-00)	14°)	222	(000)	23°)	322	(+00)
6°)	123	(-0+)	15°)	223	(00+)	24°)	323	(+0+)
7°)	131	(-+-)	16°)	231	(0+-)	25°)	331	(++-)
8°)	132	(-+0)	17°)	232	(0+0)	26°)	332	(++0)
9°)	133	(-++)	18°)	233	(0++)	27°)	333	(+++)

Figure 2: Modifications of indices (m, n, p) when moving from V_c to the neighboring voxel V_i , $1 \leq i \leq 27$, $i \neq 14$.

the MR volume are also removed from the list of possible destination points (*valid points*).

The algorithm works as follows. First, a measure P_i , $i \in \{\text{valid points}\}$, is evaluated based on the probability of going from voxel V_c to voxel V_i . This probability takes into account the eigenvalues and eigenvectors available at point V_c from the DT-MR image diffusion matrix. In order to calculate this probability, the information shown in Fig. 2 is used.

The table in Fig. 2 shows, for every voxel shown in Fig. 1, the changes that must occur in indices (m, n, p) , when a tract experiments a transition from voxel V_c to voxel V_i . For instance: when going from point No. 14 to point No. 6, coordinate m reduces by 1, n remains the same, and p increases by 1. This is represented in the table with “ $\pi_m\pi_n\pi_p = (-0+)$ ”. With this information, the probability of each possible destination V_i can be calculated taking into account the projection of each of the eigenvectors to each of the directions defined in the triplet $\pi_m\pi_n\pi_p$. Besides, each projection is weighted by the corresponding eigenvalue λ . Thus, in the previous example, P_i should be calculated as $P_i = V_{1y}\lambda_1 + V_{2y}\lambda_2 + V_{3y}\lambda_3 + V_{1z}\lambda_1 + V_{2z}\lambda_2 + V_{3z}\lambda_3$, where $V_{j,\alpha}$ represents the α -component of eigenvector¹ j , $1 \leq j \leq 3$, $\alpha \in \{x, y, z\}$.

In the general case we have,

$$P_i = \sum_{\alpha \in \{x, y, z\}} \chi_\alpha \sum_{j=1}^3 V_{j,\alpha} \lambda_j \quad (1)$$

with χ_x, χ_y, χ_z being zero if π_m, π_n, π_p are zero, respectively, and equal to 1 otherwise.

The axes reference criterion for the (x, y, z) vector components is also shown in Fig. 1. Note that, for this calculus, the sign “-” in triplets $\pi_m\pi_n\pi_p$ is equivalent to sign “+”. Finally, in order to properly calculate P_i , it must be weighed by 0.33 if there are no zeros in triplet i ,

¹Assuming that $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

and by 0.5 if there is one zero.

3.2 Anisotropy and local probability

The following anisotropy index is used in the algorithm:

$$fa = \sqrt{\frac{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_1 - \lambda_3)^2}{2(\lambda_1^2 + \lambda_2^2 + \lambda_3^2)}}, \quad (2)$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3$. When both $fa(V_c)$ and $fa(V_i)$ do not exceed a certain threshold, point V_i is eliminated as a possible destination point.

Taking into account both P_i and the anisotropy given by Eq. (2), the local probability of voxel i is defined as

$$P'_i = a \cdot \mu_1 \cdot fa(V_i) + (1 - a) \cdot \mu_2 \cdot P_i, \quad 0 < a < 1 \quad (3)$$

where parameter a allows the user to give a higher relative weight to either the anisotropy or the local probability, and μ_1 and μ_2 are scaling factors (normally, 1 and 1000, respectively). The set of values P'_i is properly normalized so that they can be interpreted as probabilities.

3.3 Eigenvectors and direction considerations

Besides these considerations, the final probability of voxel i makes also use of the so-called *smoothness parameters* (described in [8]) which judge the coherence of fiber directions among the trajectories passing through voxel V_c . The mathematical expressions of these four parameters are:

$$sp_1 = \hat{\mathbf{v}}(V_c(k)) \cdot \hat{\mathbf{v}}(V_c(k-1)) \quad (4)$$

$$sp_2 = |\hat{\mathbf{v}}(V_c(k)) \cdot \mathbf{v}_{max}(V_c(k))| \quad (5)$$

$$sp_3 = |\hat{\mathbf{v}}(V_c(k)) \cdot \mathbf{v}_{max}(V_i)| \quad (6)$$

$$sp_4 = |\mathbf{v}_{max}(V_c(k)) \cdot \mathbf{v}_{max}(V_i)| \quad (7)$$

where $\hat{\mathbf{v}}(V_c(k))$ and $\hat{\mathbf{v}}(V_c(k-1))$ are normalized vectors, $\hat{\mathbf{v}}(V_c(k)) = \mathbf{v}(V_c(k)) / \|\mathbf{v}(V_c(k))\|$, $\hat{\mathbf{v}}(V_c(k-1)) = \mathbf{v}(V_c(k-1)) / \|\mathbf{v}(V_c(k-1))\|$. Besides, $V_c(k-1)$ denotes

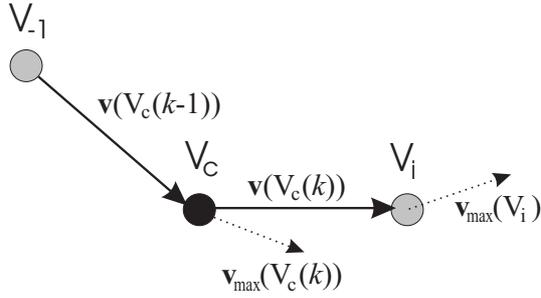


Figure 3: Voxels and vectors involved in the calculus of the smoothness parameters.

the preceding voxel of V_c , and $\mathbf{v}(V_c(k-1))$ is the vector pointing from $V_c(k-1)$ towards V_c (see Figure 3).

$\mathbf{v}_{max}(V_c(k))$ and $\mathbf{v}_{max}(V_i)$ are the principal eigenvectors (corresponding to the largest eigenvalue of the diffusion tensor) at the voxels V_c and V_i , respectively. Note that parameters sp_1, \dots, sp_4 measure the angles between the orientations of these vectors. As shown in Figure 3, Equations (4)-(7) imply that $sp_1 \in [-1, 1]$, and $sp_2, sp_3, sp_4 \in [0, 1]$. The thresholds for sp_2, sp_3 and sp_4 are chosen to be the same value. These three parameters are used to maintain the local directional coherence of the estimated tract and avoid the trajectory to follow unlikely pathways [8]. The threshold for sp_1 is set to a value between 0.6 and 0.9 such that the tracking direction could be moved forward consistently and smoothly, preventing the computed path from sharp transitions.

Next, the following parameter is calculated for every valid point whose smoothness parameters satisfy the four corresponding threshold conditions,

$$P_i'' = b \sum_{i=1}^4 (\xi_i \cdot sp_i) + (1-b)P_i' \quad (8)$$

where, $\xi_i, i = 1, 2, 3, 4$ are the corresponding weights of the smoothness parameters (normally, 0.25), and b stands for a weighting factor.

3.4 Path probabilities

Probabilities P_i'' can be recursively accumulated, yielding the probability of the path generated by the successive values of V_c ,

$$P_p(k) = P_i''' \cdot P_p(k-1) \quad (9)$$

with k being the iteration number, and $P_i''' = P_i'' / \sum_i P_i''$.

At the end of the visualization stage, every estimated path is plotted with a color that depends on its probability P_p .

3.5 Final criterion and pool of “future seeds”

A pool of voxels is formed by selecting, at the end of each iteration, the s best voxels according to Eq. (8). The first voxel of the pool becomes the central voxel V_c at next iteration (afterwards, it is removed from the pool) expanding, this way, the current pathway.

As proposed in [16], the value of s is adjusted depending on the degree of anisotropy found in current voxel V_c and its surroundings. When this anisotropy is high, it means that a high directivity exists in that zone, and the probability that V_c belongs to a region where fibers cross is really low. Consequently, s takes a small value (1, 2 or 3). On the other hand, if V_c is found to be situated in a region of low anisotropy, the probabilities of having fibers crossing or branching is higher. In this case, it is interesting to explore various paths starting in V_c . This can be easily achieved by setting parameter s to a higher value.

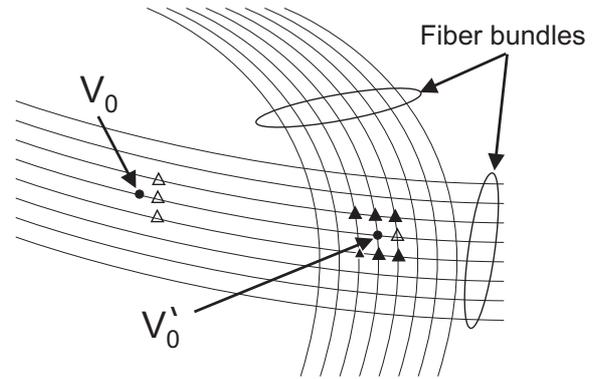


Figure 4: Selection of seeds around V_c for different anisotropies.

This idea is illustrated in Fig. 4, where point V_0 belongs to a very anisotropic region. In this case, $\mathbf{v}(V_c(k-1))$ and $\mathbf{v}(V_c(k))$ are nearly parallel, and parameter sp_1 is close to 1. The pool of surviving voxels should be augmented with those points marked with a white filled triangle. On the other hand, points like V_0' , which are situated in a region where two orthogonal fiber bundles cross, have a small anisotropy. In this case, all the points marked with black triangles could be added to the pool in order to be considered as seeds in future iterations. Numerical results show that the two main consequences of the on-line adjustment of s are: (i) a better use of computational resources, and (ii) a performance gain in regions where crossing or branching fibers exist.

Notice that parameters $(a, b, \mu_1, \mu_2, \xi_1, \xi_2, \xi_3, \xi_4)$ must be adjusted in order to get satisfactory results when estimating the tracts of the volume being analyzed. This is a tedious task that has always been heuristically approached. In this paper, we propose a neural network-based scheme

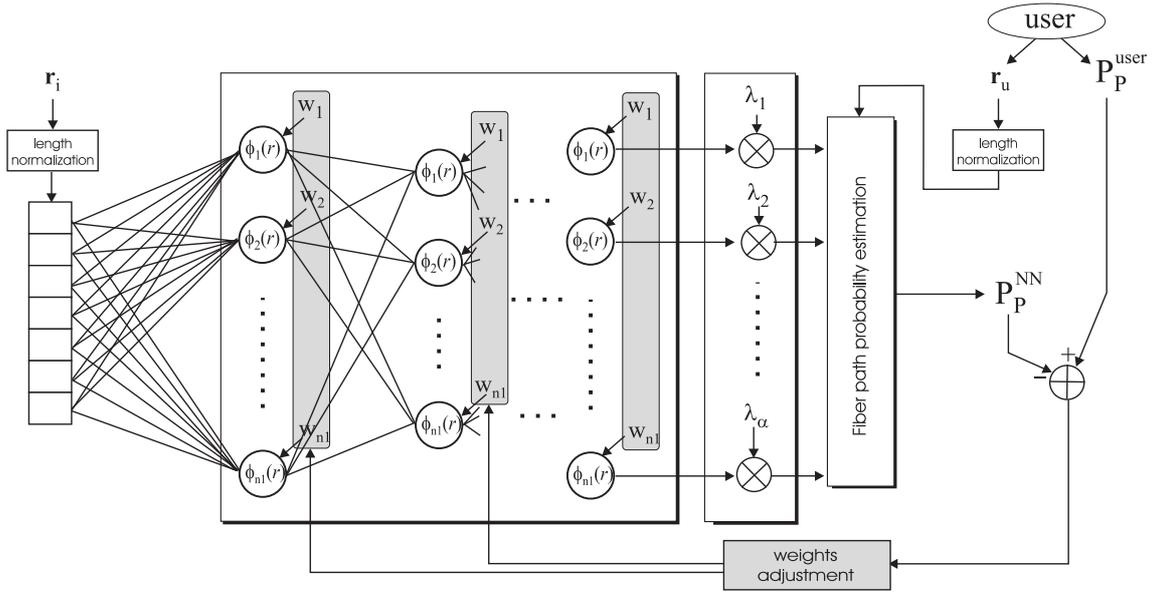


Figure 5: Representation of the proposed NN-based scheme for the estimation of the n_p tracking algorithm's parameters.

to estimate these parameters.

4 Neural network for parameter estimation

The fiber tracking algorithm outlined in section 3 was first developed in [15] and later improved in [16]. Numerical simulations showed that it makes an efficient use of computational resources while it attains a good performance in regions with crossing or branching fibers. However, as described in [15, 16], the algorithm's parameters and thresholds had to be heuristically determined.

In this section we propose a neural network with a variable number of hidden layers and the back-propagation algorithm for learning that can be used to adjust the n_p main parameters of the algorithm $(a, b, \mu_1, \mu_2, \xi_1, \xi_2, \xi_3, \xi_4)$.

When this strategy is used, the user is requested to manually draw a sample fiber path as well as to assign a certainty value to the paths automatically generated in an initial training user-directed stage. This parameter adjustment is useful when the algorithm is applied to a different part of the brain (fiber bundles) or even to the same portion but having been scanned under different conditions. In these cases, the volume of interest will have a different smoothness and anisotropy characterization. To our knowledge, no previous work has proposed any mechanism to estimate the parameters of any existing tracking algorithm, and these parameters are always heuristically adjusted in a very time-consuming task.

The proposed scheme works as follows (see Fig. 5): a set of N different paths \mathbf{r}_i automatically estimated with the

algorithm is used as the input of the NN. A processing block consisting of α layers –with its last one consisting of as many units as parameters need to be estimated, n_p – is next placed. Processing unit n of layer α evaluates a function $\phi_{\alpha,n} = \phi(\mathbf{w}_{\alpha,n} \mathbf{r}_i - \mathbf{r}_u)$, where \mathbf{r}_u denotes the user-drawn tract, i.e. the reference tract. Since every path in \mathbf{r}_i can have a different length, and \mathbf{r}_u may have a different number of samples, as well, a length-normalization procedure is initially required. In order to have a tolerable complexity, this length is normally reduced to 25 samples (this number can be increased in case of complex tracts).

Once the α hidden processing layers have completed their evaluation, the n_p outputs are weighed with $\{\lambda_i\}_{i=1}^{n_p}$ and a path probability P_p^{NN} is calculated using the n_p available NN outputs. Finally, P_p^{NN} is compared to the user-given certainty value for current training path \mathbf{r}_i . This measure is proportional to the error associated to consider path \mathbf{r}_i with the current set of n_p parameters. This information is then used to adjust the weights of the net using a learning algorithm.

In section 5.3, the performance of this neural system will be evaluated as a function of different parameters such as the number of hidden layers, the kernel function ϕ , the number of parameters to be estimated, the kind of image being processed (synthetic or real) and the quality of the original DT-MR image.

5 Numerical Results

In order to evaluate the tracking properties of the proposed algorithm, we have used both synthetic and real DT-MR images.

5.1 Synthetic images

First, four different synthetic DT-MRI data in a $50 \times 50 \times 50$ grid have been generated (see Fig. 6). The first three images (“cross”, “earth” and “log”) were used for testing in [15, 16], while the most complex one –Fig. 6, bottom-right– is new. In the *earth* volume fibers look like geographical parallels of different spheres with the same center and different radius situated on plane $x - y$, and with a totally isotropic cylindrical volume in the center, parallel to the z axis. On the other hand, the *log* testing volume consists of logarithmic curves symmetrically distributed around the y axis.

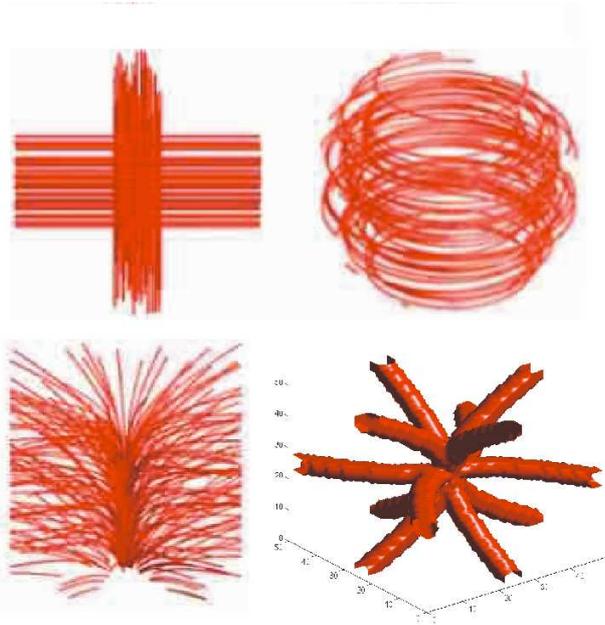


Figure 6: Synthetic DT-MR images used for testing the proposed algorithm: “cross” (left), “earth”, “log” and “star” (right).

To make the simulated field more realistic, Rician noise [7] was added in the diffusion weighted images which were calculated from the Stejskal-Tanner equation using the gradient sequence in [20] and a b -value of 1000.

The desired noisy synthetic diffusion tensor data was obtained using an analytic solution to the Stejskal-Tanner equation. The eigenvectors in the isotropic areas were $\lambda_1 = \lambda_2 = \lambda_3$, while in the remaining voxels of the volume: $\lambda_1 = 7$, $\lambda_2 = 2$ and $\lambda_3 = 1$. In the section devoted to study the robustness, the SNR varies from 0 to 30 dB. As an example, Fig. 7 shows the cross test volume with SNR= 10 dB (top), and with SNR= 30 dB (bottom).

5.1.1 Tracking capability

Satisfactory tracing results for the first three cases can be found in [15, 16], where a much simpler algorithm

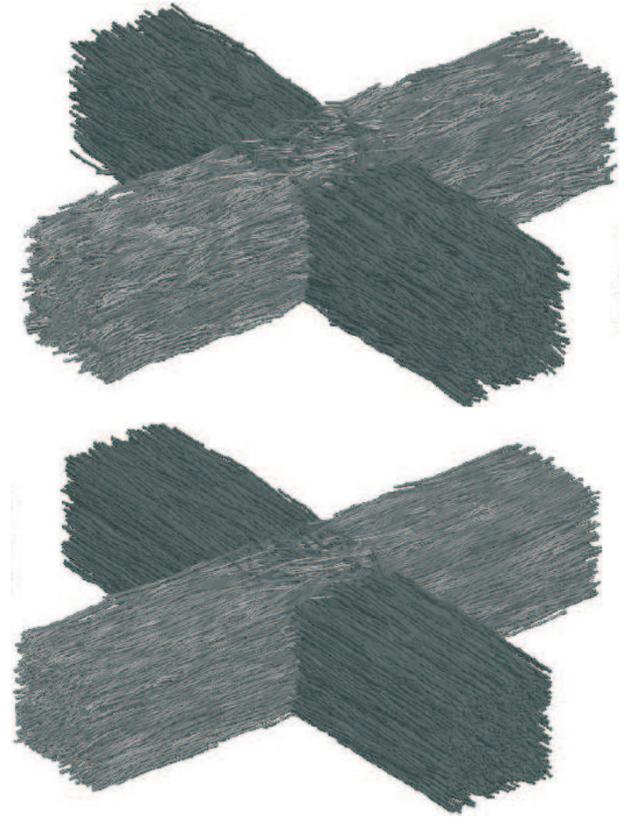


Figure 7: Representation of the cross synthetic image with SNR= 10 dB (top), and SNR= 30 dB (bottom).

was used. For the sake of brevity, in this paper we have just included the new and most complex case, the star. This image consists of six orthogonal sine half-waves, each of them with an arbitrary radius². Under this scenario the diffusion field experiments variations with the three coordinate axes and there exists a crossing region. Three different tracking results are shown in Fig. 8 each of them for a different seed V_0

It can be seen how the algorithm can be designed in order to pass through isotropic zones where different fiber bundles cross. It is also appreciated how the algorithm differentiates between the totally isotropic zones extrinsic to the tracts and the fiber bundles.

The differentiation between voxels belonging to a fiber or to a very isotropic area, respectively, is attained by mapping the path probabilities given by Eq. (9) into a color scale and classifying them according to some fixed thresholds. Three different seeds (S_1 , S_2 and S_3) are shown. S_1 and S_2 belong to the intrinsic volume (voxels with a very high anisotropy) and the algorithm moves through the most probable direction following the main direction of the cross in each situation. On the other hand, when an extrinsic point such as S_3 is selected as seed, the algo-

²For the sake of clarity, Fig. has been depicted with the six sine waves having the same radius.

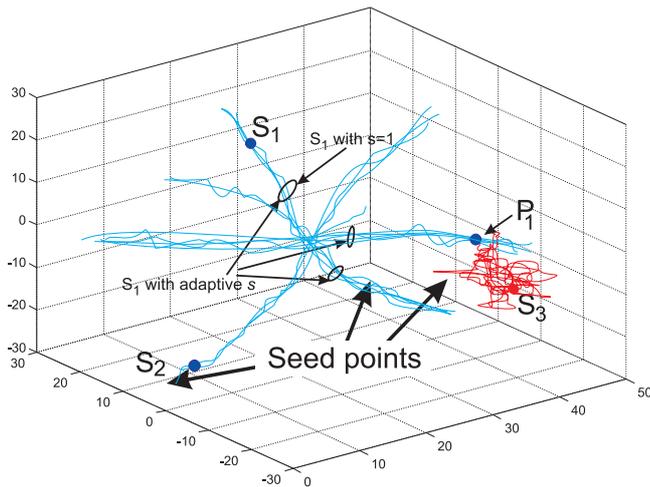


Figure 8: Tracking results for the “star” synthetic DT-MR image. Green: seed points. Blue: fiber path voxels. Red: extrinsic voxels. Initial seeds $V_0 = \{S_1, S_2, S_3\}$.

gorithm explores in the neighboring voxels until it finds a voxel with a high anisotropy value (point P_1). Once P_1 is found, the tracking algorithm proceeds as in the case of S_1 and S_2 . Fig. 8 shows how the algorithm finds the proper fiber path whatever (extrinsic or intrinsic) seed voxel is chosen.

Notice that, the extrinsic seeds S_3 are located far away from the fiber bundles region, thus making the algorithm explore a wider range of points before reaching the points P_1 that belong to an existing fiber path.

5.1.2 Robustness

Next, the robustness of the tracking algorithm is now studied for: (i) parameter s is fixed during the whole estimation of the path, and (ii) parameter s is dynamically changed depending on the anisotropy.

- **Parameter s fixed.** These simulations were carried out with $s = 2$. The plots depicted in Fig. 9 correspond to this situation. Notice that, in the case of the cross-like image the algorithm starts in S_1 , S_2 or S_3 (in this later case it first reaches point P_1) and it follows a path which passes straight ahead through the crossing volume ($20 \leq x, y \leq 30$) without considering a branching situation. The results would be almost the same considering $s = 1$.

The convergence performance for different SNRs is shown in Table 1. It can be seen how the algorithm converges properly within a wide range of SNRs. The percentages obtained for the “cross” and the “earth” test images are very close, while for the “log” case the algorithm exhibits a slightly lower convergence. Notice that this table also shows the results for a

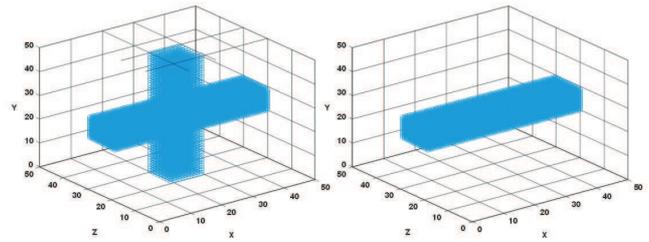


Figure 10: Effect of varying the value of parameter s .

third method used for comparison³ [6]. Comparing both methods, we see that the proposed algorithm performs slightly better when the SNR is low, while both methods tend to similar results with high SNRs.

- **Parameter s depends on the anisotropy.** We illustrate this subsection with two examples. First, Fig. 10, left, shows the results for the “cross” image when $s = 6$ for high anisotropy and $s = 2$, otherwise. The bundle is estimated and branching is properly managed. Setting the smoothness parameters more restrictively, a single fiber bundle can be estimated, as shown in Fig. 10, right.

On the other hand, Fig. 8 shows the results for the “star” image when: (i) $s = 1$ and (ii) $s = f(\text{anisotropy})$, as explained in section 3.5.

Analyzing the simulations of the four synthetic images considered, it is seen that convergence results improve whenever the MR image contains branching or crossing areas –as it is the case in real DT-MR images. This is the case of our “cross” image. For this image, the convergence results are improved $\sim 5\%$ when parameter s is modified according to the anisotropy. Besides, for these studied cases, we see that the influence of the procedure that adapts s is higher for low SNRs. In case the SNR of the image is large, this procedure scarcely affects the results.

Consequently, the algorithm converges properly within a very wide range of SNRs. The percentages obtained for the “cross” and the “earth” test images are very close, while for the “log” case the algorithm exhibits a slightly lower convergence. Besides, when parameter s is on-line tuned-up the robustness of the algorithm in branching and crossing situations becomes more flexibly controlled and the computational load can be maintained to its lowest value for the kind of desired estimation.

³The Bayesian algorithm implemented for comparison is a slightly modified version of the method proposed in [6]. This stands for the results given in Tables 1 and 2.

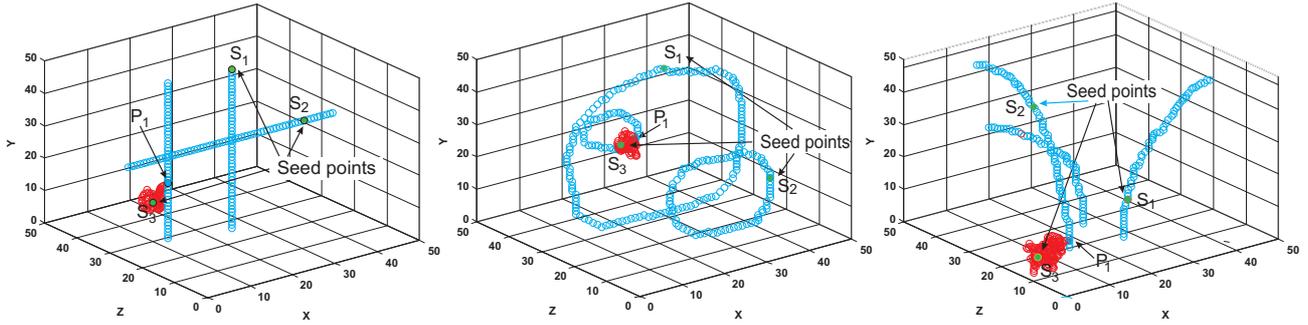


Figure 9: Tracing results for: (a) the cross-like image, (b) the earth-like test images (curves are in the $x - y$ plane), and (c) for the logarithmic test image. Green: seed points. Blue: fiber path points. Red: extrinsic voxels.

		SNR (dB)					
		5	10	15	20	25	30
Image	Cross	78.3/ 82.8	89.7/ 93.6	92.1/ 94.3	98.3/ 98.7	99.0/ 99.0	100/ 100
		76.8	89.0	90.7	97.0	100	100
	Earth	77.7/ 76.2	88.6/ 87.5	89.9/ 89.0	98.2/ 98.2	99.0/ 99.0	100/ 100
	74.4	83.2	85.0	97.3	99.2	100	
	Log	71.0/ 69.7	82.1/ 81.0	86.1/ 85.5	96.0/ 95.8	98.0/ 97.8	100/ 100
		68.8	78.3	85.2	96.0	98.0	100

Table 1: Convergence performance for different SNRs values. Cell values represent percentage of right convergence for two configurations of the algorithm: $s = 1/s = 4$, as well as the results obtained with [6].

	Proposed algorithm				Method [6]
	$s = 1$	$s = 6$	$s = 9$	Adaptive s $1 \leq s \leq 12$	
Cross	1	2.1	3.2	4.9	18.9
Earth	1	2.1	3.2	4.9	20.1
Log	1.1	2.1	3.3	5.0	19.9
Star	1.1	2.2	3.3	5.1	20.0

Table 2: Comparison of computational load. Execution times required to estimate 20 fibers of 200 points. Values normalized to the first case shown (the proposed method with $s = 1$ and “cross” image).

5.1.3 Computational load

Table 2 shows a computational load comparison with respect to [6]. Values are normalized to the time required by our method with $s = 1$ considering the “cross” image.

The Bayesian approach of [6] requires much more computational time and resources, while it does not get better convergence results (see Table 1) than the proposed algorithm with fixed or adaptive s . Furthermore, it can be seen that the computational load of the proposed procedure does not increase linearly with the size of the surviving seeds’ pool, s .

5.2 Real images

Next, we have applied the proposed tracking algorithm to real DT-MR images. Specifically, we have selected the *corpus callosum* of the brain (Fig. 11) and the fiber pathways in the optic radiations (Fig. 12).

Simulation results are shown on the right side of both images. It can be appreciated how the algorithm is able to follow the main fiber bundle direction without getting out of the area of interest. These figures show some bundles of properly estimated tracts. Red/green color indicates high/low certainty.

5.3 Neural Network properties

5.3.1 Complexity of the NN vs BER

In this first simulation, the complexity (assessed as the number of hidden layers of the net) required to attain good fiber tract estimation for different images (synthetic and real) under different conditions (SNRs) is evaluated. Fig. 13 shows the results. The number of hidden layers varies between 1 and 7, and the the quality of the DT-MR image is $0 \leq \text{SNR} \leq 30$ dB.

It can be seen how the number of required hidden layers decreases as the quality of the image improves (higher SNR). When both the SNR and the number of hidden layers are low, the number of samples required for training increases considerably. On the other hand, with a

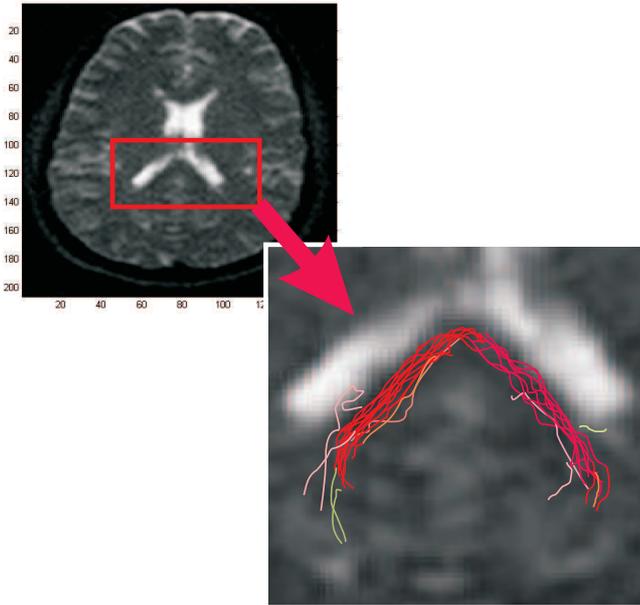


Figure 11: Tracking results for the *corpus callosum* area of the human brain.

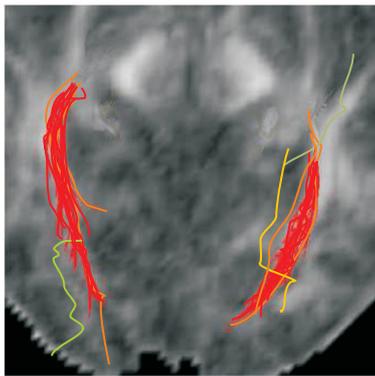


Figure 12: Tracking results for the optic radiation. Seeds are selected lateral to each of the lateral geniculate nuclei.

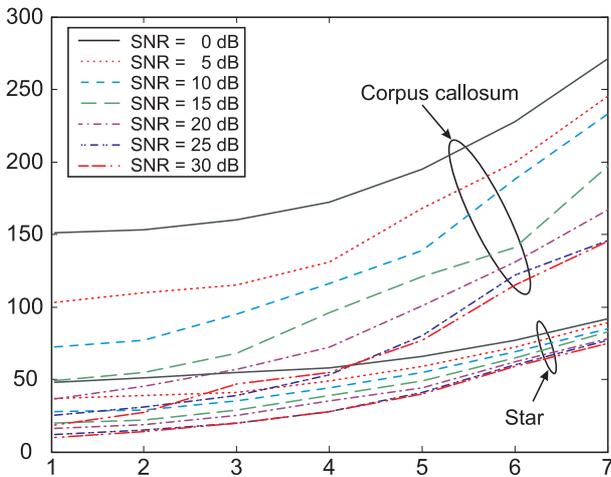


Figure 13: Number of training samples required to attain good estimation for different NN configurations.

sufficiently complex net, the number of samples required scarcely increases when the image quality worsens.

The figure also shows that the real image requires twice or three times the number of layers required in the equivalent (in terms of SNR and number of layers) synthetic image. More complex DT-MR images, with very complex tracts, would require much more samples.

5.3.2 Estimation of the smoothness parameters

When the algorithm outlined in section 3 is used, the four weights $\{\xi_1, \xi_2, \xi_3, \xi_4\}$ of the smoothness parameters shown in Eq. (8) must be estimated, as well. Normally, these parameters are not difficult to adjust heuristically since they remain nearly unchanged in different images. For this reason, they are almost never considered in the estimation process. However, in this section the additional complexity required for their estimation is studied. Table 3 shows the training samples required for their proper estimation, as a function of the image SNR and the number of NN layers.

		No. of layers			No. of layers		
		1	3	5	1	3	5
SNR (star)	30	10	20	40	25	47	80
	20	16	25	44	41	58	83
	10	28	35	55	-	84	103
	0	48	55	66	-	-	131
SNR (corpus callosum)	30	18	47	77	57	126	191
	20	36	57	101	113	155	240
	10	72	95	139	-	-	320
	0	151	160	195	-	-	-
		w/o $\{sp_i\}$ estimation			With $\{sp_i\}$ estimation		

Table 3: Number of training samples required to attain good estimation for different NN configurations. Left: without estimating the smoothness parameters' weights, right: with estimation.

It is evident that the number of training samples required for a successful estimation of the algorithm's parameters greatly increases when the smoothness' thresholds are considered as unknown parameters. The table also shows that certain complex situations with the *corpus callosum* can not be effectively solved.

Considering the results in Fig. 13 it could be concluded that the simplest configuration is always valid and the others should never be used. However, this is not a general correct conclusion since when a larger set of test images is used, it can be seen that more complex images can not be successfully processed with simple configurations. This is shown in the unresolved cases shown in Table 3. These situations would require different both NN architecture and estimation process, and they will be matter of study in future work.

5.3.3 RBF basis function effect

The algorithm proposed in section 3 has been tested using different kernel functions in the NN configuration. It is interesting to note that, the choice of the Mahalanobis distance in the RBF basis function allows some advantages over the Euclidean one due to the non-spherical shape of the multidimensional clusters (highly non-linear problem). Though, for the sake of brevity, no table is included here, simulations showed that simple functions were not able to properly estimate the parameters. A good trade-off between complexity of the kernel functions and efficiency was found with the Mahalanobis function.

6 Conclusions and Future Work

In this paper, a recently developed fiber tracking algorithm has been improved and tested with several synthetic and real DT-MRI data. This algorithm combines both the probability of advancing in a specific direction based on the projection of *all* the eigenvectors components into the corresponding directions (making use of more information than merely the principal eigenvector) and four smoothness criteria based on the relation between the potential future path directions and the eigenvectors' orientations. The number of possible paths emerging from the actual point is on-line adjusted based on the local anisotropy. Numerical simulations show that the two main consequences of this are: (i) a better use of computational resources, and (ii) a better performance in regions with crossing or branching fibers. The method was tested with synthetic and real DT-MR images with notably satisfactory results, showing better computational and convergence properties than already existing traditional and Bayesian methods. Finally, a brand new NN-based scheme has been proposed for the estimation of the parameters of the algorithm. To our knowledge, no previous attempts exist to automate this task. The performance of this NN has been studied for different network configurations (number of hidden layers, kernel function) and image qualities for both real and synthetic images.

Future work will focus on: semiautomatic selection of seeds, improvement of visualization characteristics, development of a user-friendly interface, and study of different techniques and criteria to create and maintain the pool of "future seeds" (directly related to the behavior in crossing and branching regions).

7 Acknowledgements

The authors acknowledge the Spanish CICYT for research grant TEC2004-06647-C03-01, as well as the European Commission for the funding associated to the Network of Excellence SIMILAR (FP6-507609).

References

- [1] Bjornemo, M., Brun, A., "White Matter Fiber Tracking Diffusion Tensor MRI," *Master's Thesis*, Linköping University, 2002.
- [2] Conturo, T.E., Lori, N.F., Cull, S.T. *et al.*, "Tracking Neuronal Fiber Pathways in the Living Human Brain," *Proc. of the National Academy of Sciences of the United States of America*, pp. 10422-10427, 1999.
- [3] Coulon, O., Alexander, O.C., Arridge, S.R., "A Regularization Scheme for Diffusion Tensor Magnetic Resonance Images", *Lecture Notes in Computer Science*, pp. 92-105, 2082, 2001.
- [4] Coulon, O., Alexander, D.C., Arridge, S.R., "Diffusion Tensor Magnetic Resonance Image Regularization," *Medical Image Analysis*, pp. 47-67, 8/04.
- [5] Ehrlicke, H.H., Klose, U., Grodd, U., "Visualizing MR Diffusion Tensor Fields by Dynamic Fiber Tracking and Uncertainty Mapping," *Computers & Graphics*, pp. 255-264, 30/06.
- [6] Friman, O., Westin, C.-F., "Uncertainty in White Matter Fiber Tractography," *Proceedings of the MICCAI 2005, LNCS 3749*, pp. 107-114, 2005.
- [7] Gudbjartsson, H., Patz, S., "The Rician Distribution of Noisy MRI Data," *Magnetic Resonance in Medicine*, pp. 910-914, 34/95.
- [8] Kang, N., Zhang, J., Carlson, E.S., Gembris, D., "White Matter Fiber Tractography Via Anisotropic Diffusion Simulation in the Human Brain," *IEEE Transactions on Medical Imaging*, pp. 1127-1137, 24/05.
- [9] Kindlmann, G., Weinstein, D., Hart, D., "Strategies for Direct Volume Rendering of Diffusion Tensor Fields," *IEEE Transactions on Visualization and Computer Graphics*, pp. 124-138, 6/00.
- [10] Mori, S., Crain, B.J., Chacko, V.P., van Zijl, P.C.M., "Three-Dimensional Tracking of Axonal Projections in the Brain by Magnetic Resonance Imaging", *Annals of Neurology*, pp. 265-269, 45, 1999.
- [11] Mori, S., van Zijl, P.C.M., "Fiber Tracking: Principles and Strategies – A Technical Review," *Nuclear Magnetic Resonance in Biomedicine*, pp. 468-480, 15/02.
- [12] Pasternak, O., Sochen, N., Assaf, Y., "Separation of White Matter Fascicles from Diffusion MRI Using Φ -Functional Regularization", *Proc. of the International Society on Magnetic Resonance in Medicine*, p. 1227, 2004.

- [13] Pierpapoli, C., Barnett, A.S., Pakevic, S., Virta, A., Basser, P.J. "Validation of DT-MRI Tractography in the Descending Motor Pathways of Human Subjects", *Proc. of the International Society of Magnetic Resonance in Medicine*, p. 501, 9, 2001.
- [14] Poupon, C., Clark, C.A., Frouin, V., Regis, J., Bloch, I., Le Bihan, D., Mangin, J.-F., "Regularization of Diffusion-Based Direction Maps for the Tracking of Brain White Matter Fascicles", *NeuroImage*, pp. 184-195, 12, 2000.
- [15] San-José-Revuelta, L.M., Martín-Fernández, M., Alberola-López, C., "A New Method for Fiber Tractography in Diffusion Tensor Magnetic Resonance Images," *Proc. of the IEEE Int'l Conf. on Signal and Image Processing, ICSIP'06*, Karnataka, India, pp. 380-385, I/06.
- [16] San-José-Revuelta, L.M., Martín-Fernández, M., Alberola-López, C., "A New Proposal for 3D Fiber Tracking in Synthetic Diffusion Tensor Magnetic Resonance Images," *Proceedings of the IEEE Int'l Symp. on Signal Processing and its Applications, ISSPA '07*, Sharjah, United Arab Emirates, 2007.
- [17] Stalling, D., Hege, H.C. "Fast and Resolution Independent Line Integral Convolution", *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive techniques*, pp. 249-256, 1995.
- [18] Vilanova, A., Berenschot, G., van Pul, C., "DTI Visualization with Streamsurfaces and Evenly Spaced Volume Seeding", *Proc. of the IEEE Visualization*, pp. 173-182, 2004.
- [19] Weinstein, D., Kindlmann, G., Lundberg, E.C., "Tensorlines: Advection Diffusion Based Propagation Through Diffusion Tensor Fields", *Proc. of the IEEE Visualization*, pp. 249-253, 99, 1999.
- [20] Westin, C.-F., Mainer, S.E., Mamata, H., Nabavi, A., Jolesz, F.A., Kikinis, R., "Processing and Visualization for Diffusion Tensor MRI," *Medical Image Analysis*, pp. 93-108, 6/02.