A Viewer-dependent Tensor Field Visualization Using Multiresolution and Particle Tracing

José Luiz Ribeiro de Souza Filho, Marcelo Caniato Renhe, Marcelo Bernardes Vieira, and Gildo de Almeida Leonel

Universidade Federal de Juiz de Fora, DCC/ICE,
Cidade Universitária, CEP: 36036-330, Juiz de Fora, MG, Brazil
{jsouzaf,marcelo.caniato,marcelo.bernardes,gildo.leonel}@ice.ufjf.br
http://www.gcg.ufjf.br

Abstract. This paper presents an adaptive method for visualization of tensor fields using multiresolution and viewer position and orientation. A particle tracing method is used in order to explore the benefits of motion to the human perceptual system. The particles are inserted and advected through the field based on a priority list which ranks tensors according to anisotropy measures and viewer parameters. Tensor fields representing colinear and coplanar structures are suitable for multiresolution analysis. Using multiple scales, we propose the use of anisotropic information in multiresolution, yielding an effective and simple method to compute priority values for particle creation. We also propose a new deterministic criterion for particle insertion in the field that balances their distribution in the tensor field domain. Our results show that our method enhances the visualization and reduces artifacts encountered in previous approaches.

Keywords: Tensor Field, Particle Tracing, Multiresolution, Scientific Visualization.

1 Introduction

Tensor field properties, such as curvatures and continuities, are sometimes hard to visualize. Particle-tracing methods using tensorlines provide a good way to observe these features. But the tensorlines could represent some inharmonious or even discontinuous paths present in tensor fields. Smoothness is an important factor to be analyzed. Being able to enhance this feature without changing the peculiarities of the field provides an opportunity to further explore these characteristics of tensor fields.

In this work, we propose an improvement of method presented in [1] which used particle tracing to generate a viewer-dependent visualization. This method used a particle creation criterion based on a priority list which sorted the tensors according to their importance. However, the choice of the tensor in the list was done through a normal distribution function, which sometimes resulted in creation of particles in less interesting sites. The previous approach also generated a flickering effect, due to particles reaching isotropic regions shortly after being created. In this paper, we propose a new approach using multiresolution, and we present a new criterion for the creation of particles. We decompose the original field in order to get smaller subsamples. With this approach, we were able to significantly reduce the amount of parameters necessary in the calculation of the scalar used in the priority list sorting.

One common problem in tensor field visualization is ambiguity. In glyph-based visualization, tensors with different forms may appear similar from a particular point of view. Tensors with linear anisotropy may be identified as an isotropic if the main eigenvector is aligned to the observer. To solve this problem, we follow previous works [2, 1] in adopting a metric to evaluate the tensor orientation in regard to the observer. This strategy can be efficient not only to treat the degeneration problem, but also to improve other visualization methods. Aiding to that, we use the multiple scales of the field to enhance tensors based on their distance to the observer.

2 Related work

Research in tensor field visualization is generally concerned with the problem of achieving a more intuitive visualization of the field. The large amount of information present in a field usually makes its analysis difficult for the observer. Thus, different approaches have been tried in past works. An overview about some of them is presented in this section.

In cases where punctual data is used to obtain information from the field, the discrete approach plays an important role. Shaw *et al*, in [3] and later in [4], proposed a glyph-based visualization of general multi-dimensional data using superquadrics, seeking to explore human perceptual system characteristics in order to obtain a meaninful display of the data. Kindlmann [5] later used superquadrics to specifically describe a tensor glyph that encodes the shape of the tensor and displays it in a consistent orientation. He used measures defined by Westin *et al* [6] to better adapt the geometry of the tensor, avoiding symmetry problems and ambiguity in the identification of its shape. These measures allow classification of diffusion tensors by its shape. They are useful in DT-MRI, since diffusion can be anisotropic or isotropic depending on the tissue characteristics.

Delmarcelle *et al* [7] used another approach, in which they produced a continuous representation of the data contained in the tensor field. They introduced the concept of hyperstreamline to define continuous paths along which the tensor field can be visualized. This method is, however, subject to degeneration [8] and more suitable to symmetric tensor fields. Thus, Weinstein *et al* [9] introduced the tensorlines method, in an attempt of stabilizing the propagation in regions of non-linear diffusion, where the hyperstreamlines method encountered difficulties. They proposed a combination of diffusion with advection vectors applied to DT-MRI. More information on the use of tensor glyphs and continuous methods, as well as a number of other DTI visualization techniques, can be found in [10].

Another approach has also been proposed by Kondratieva *et al* [2]. They provided a dynamic visualization, which aims at taking more advantage of the human perceptual system. A GPU particle tracing was used to produce motion in order to enhance the user perception. They advected particles along the directions of a generated vector field, while allowing the user to interactively visualize the tensor field. Leonel *et al* [1] used this same approach, but also taking into account the position and orientation of the observer. An adaptive visualization of the tensor field was provided, enhancing the features that are more likely to interest the viewer.

Some more recent works following [2] focused on improving fiber tracking algorithms. A stochastic method to determine connectivity in a fiber path was presented in [11]. A GPU implementation of the method is also presented. Köhn *et al* [12] and Evert *et al* [13] also made use of graphics hardware to achieve a better and faster fiber tracking, allowing for interactive visualization. Mittmann *et al* [14] presented a real-time interactive fiber tracking method, in which the user defined volumes of interest in the tensor field, and the algorithm calculated new fiber paths automatically based on the user choices. Finally, in [15] an interpolation method was introduced in order to avoid low-anisotropy regions in the trajectory calculation. When the algorithm reaches such a region, it interpolates the tensors in some neighborhood and continues the path along the main eigenvector of the interpolated tensor.

This work is focused on improving visualization of diffusion tensor images. Thus, we still used the tensorlines method [9] as the tracking algorithm. Our method was implemented in CPU, yielding good results and allowing a fast and real-time interactive visualization, even for a large amount of particles, as will be shown in the paper. We also adopted a multiresolution approach associated to the dynamic visualization employed by [2] and [1]. Multiresolution analysis of diffusion tensor images can be found in the literature. Rodrigues et al [16], for example, proposed a scale-space representation of a DTI image, using a multiresolution watershed segmentation method to separate coarse from fine data. They generated a hierarchical representation afterwards, through a cross scale linking of the segmented regions. In this paper, we present a different and much simpler multiresolution scheme, based on wavelet theory.

3 Fundamentals

3.1 Tensors

Second-order tensors can be defined as linear transformations between vector spaces. They are represented by 3x3 matrices. In this work, a tensor of particular interest is the one presented by Westin [6]. It is called a local orientation tensor, and it is a special case of a non-negative symmetric rank 2 tensor. This tensor can be used to estimate orientations in a field. Mathematically, it can be defined as following:

$$\mathbf{T} = \sum_{i=1}^{n} \lambda_i e_i {e_i}^T$$

where λ_i represent the eigenvalues and e_i the associated eigenvectors.

In $\mathbb{R}^3$, the equation above can be decomposed in such a way that $\mathbf{T}$ can be expressed in terms of its linear, planar and spherical intrinsic features [1]. So, the tensor definition becomes:

$$\mathbf{T} = (\lambda_1 - \lambda_2)\mathbf{T}_l + (\lambda_2 - \lambda_3)\mathbf{T}_p + \lambda_3\mathbf{T}_s$$

This decomposition reveals an important geometric interpretation about the tensor. Assuming that $\lambda_1 \geq \lambda_2 \geq \lambda_3$, we can analyze the eigenvalues to identify the shape of the tensor, which is of much more use than its magnitude, for example. If, for instance, we have $\lambda_1 >> \lambda_2 \approx \lambda_3$, the tensor is approximately linear. If $\lambda_1 \approx \lambda_2 >> \lambda_3$, then the shape of the tensor is approximately planar. Finally, if all eigenvalues are almost equal, then the tensor is approximately isotropic. In this case, there is no main orientation present in the tensor.

Coefficients of anisotropy The tensor eigenvalues can be used to calculate coefficients of anisotropy. The eigenvalues are obtained by solving $\det(\lambda\mathbf{I} - \mathbf{D}) = 0$. We can define three of these coefficients: linear (c_l), planar (c_p) and spherical (c_s). These three coefficients must sum to 1.

$$\begin{aligned} c_l &= \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \\ c_p &= \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \\ c_s &= \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \end{aligned} \tag{1}$$

It is also possible to calculate a number of coefficients which are insensitive to basis changing. These coefficients are called algebraic invariants. Among them, the ones presented below are helpful in the definition of a series of parameters that can be used to analyse the characteristics of the field.

$$\begin{aligned} J_1 &= \mathrm{tr}(\mathbf{D}) \\ J_2 &= \frac{\mathrm{tr}(\mathbf{D})^2 - \mathrm{tr}(\mathbf{D}^2)}{2} \\ J_3 &= \det(\mathbf{D}) \\ J_4 &= ||\mathbf{D}||^2 \end{aligned}$$

where $\mathrm{tr}(\mathbf{D})$ and $\det(\mathbf{D})$ are the trace and the determinant of $\mathbf{D}$, respectively [1].

Kindlmann [17] presents three more algebraic invariants, which are used not only to describe what is called the eigenvalue wheel, but also to define the central moments of the tensor. These invariants are defined as follows:

$$Q = \frac{3J_4 - 3J_1^2}{18}$$

$$R = \frac{-5J_1J_2 + 27J_3 + 2J_1J_4}{54}$$
$$\Theta = \frac{1}{3}\cos^{-1}\left(\frac{R}{\sqrt{Q^3}}\right)$$

The central moments are related to the geometric parameters of the wheel. The definition of the wheel, along with a detailed explanation of it, can be found in the work by Kindlmann [17]. Using the Kindlmann invariants, we can define the central moments as shown below:

$$\mu_1 = \frac{J_1}{3}$$
$$\mu_2 = 2Q$$
$$\mu_3 = 2R$$

The second central moment μ_2 represents eigenvalues variance. Taking its square root, we can obtain the standard deviation σ. This allows us to define an important parameter in this work, called the asymmetry of the eigenvalues. The asymmetry parameter varies from negative to positive as the tensor changes from planar to linear. It is calculated as follows [18]:

$$A_3 = \frac{\mu_3}{\sigma^3} = \frac{R}{\sqrt{2Q^3}} \tag{2}$$

A more complete description of several anisotropy coefficients are found in [1]. In this paper, we use only the A_3 and c_l coefficients since they indicate linear and planar continuities suitable for particle tracing.

3.2 Tensorlines

Previous works intended for path tracing in tensor fields lacked stability in certain scenarios. The hyperstreamlines method [19] used just the main tensor eigenvector in order to obtain a smooth tracing, but it was subject to degeneration. Seeking to work around the inherent problems of this method, Weinstein *et al* [9] proposed an extension called tensorlines. Instead of only using the main eigenvector to determine the path, it applies the tensor to a vector corresponding to the propagation direction in the previous step.

$$\mathbf{v}_{out} = \mathbf{T}\mathbf{v}_{in} \tag{3}$$

The new vector $\mathbf{v}_{out}$ produced by the transformation above is linearly combined with $\mathbf{v}_{in}$ and the main eigenvector e_1. Thus, we obtain the new propagation vector, which is dependent on the shape of the tensor. It is calculated as follows:

$$\mathbf{v}_{prop} = c_l e_1 + (1 - c_l)((1 - w_{punct})\mathbf{v}_{in} + w_{punct}\mathbf{v}_{out}) \tag{4}$$

The parameter w_{punct} lies in the range $[0, 1]$ and defines how much the propagation should penetrate planar tensors. This parameter is controlled by the user. The coefficient c_l is the linear anisotropy coefficient defined in the previous subsection.

The vector field produced by the tensorlines method can be applied to a particle tracing procedure to visualize the tensor field. The particles introduced in the field have no mass. At each time step, we update the position of each particle over time t, using a vector v from the generated field as the velocity.

3.3 Multiresolution

In this work, we used a multiresolution scheme based on the Daubechies analysing filters. The Daubechies low pass filter is applied to the tensor field, separately for each tensor component, with the purpose of generating lower resolution fields with half spectrum of the previous scale. Since the tensor fields we worked on have the maximum dimensions of $148 \times 190 \times 160$, two decimated scales seemed to be enough for our purposes.

A decimated tensor captures the anisotropy of a group of local tensors. It can be used, for example, to filter the particle paths during tensorlines computation. The anisotropic features of the scaled tensors are linear combinations of the underlying tensors shape in full resolution. As such, its anisotropic coefficents bring new information to form an improved priority list. Details about signal multiresolution can be found on [20].

In the previous work of Leonel [1], an extensive list of parameters were used to calculate the importance of a single tensor to the observer. This importance was determined by a scalar parameterized by the user. Here, we present a new formulation for calculating this scalar with a reduced number of parameters, taking into account the lower resolution fields obtained. The next section presents the equations for this calculation and other contributions of this work.

4 Proposed method

The previous approach [1] was conceived to induce the human perceptual system to detect continuity using particle motion. Particles in motion represent the features of the tensor field. One critical point in visualization using particle tracing is to define the particle starting point. The easier approach to insert particles into the domain is to compute new positions randomly. A fixed distribution function, however, generally does not insert new particles in most interesting sites. Using tensorlines to indicate suitable particle paths, the idea was to carefully select the position where a new particle should start. It was based on tensor field anisotropic features and viewer-dependent relationships. The maximum number of particles at a time was fixed. A priority list determined which particle should be chosen. Several coefficients for particle sorting were presented.

In this work, we propose major modifications for the priority list calculation and new particles selection. Our approach is based on the use of multiresolution of the tensor field. Each scale of a tensor field in multiresolution combines the tensor of the previous, higher resolution, scale. We exploit the anisotropic features of the resulting tensors to provide a new scalar value for the priority list (Eq. 5). Viewer dependent and independent coefficients in multiresolution are computed (Fig. 1).

4.1 Priority Features

Let $\mathbb{T}_{x \times y \times z}$ be a discrete and finite tensor field with lattice given by $x, y, z \in \mathbb{N}$, so that $\mathbb{T}^s = \{\mathbf{t}_1^s, \mathbf{t}_2^s, \mathbf{t}_3^s ... \mathbf{t}_n^s\}$ is composed by $|\mathbb{T}^s| = n$ tensors, where s is the scale index. For a given voxel (a, b, c), where $a, b, c \in \mathbb{N}$ and such that $a \leq x$, $b \leq y$ and $c \leq z$, we have the correspondent tensor $\mathbf{t}_i^s \in \mathbb{T}$. As explained in Section 3.3, the tensor fields are decomposed two times in this work, resulting in three scales: $s = 0$ is the original tensor field, $s = 1$ is the tensor field with half spectrum, $s = 2$ is the tensor field with a quarter of original spectrum.

The eigensystem of a tensor $\mathbf{t}_l^s$, $1 \leq l \leq n$ is represented by the eigenvectors $\vec{e}_1^s \perp \vec{e}_2^s \perp \vec{e}_3^s$ and the eigenvalues $\lambda_1^s \geq \lambda_2^s \geq \lambda_3^s \geq 0$.

The goal of the priority list is to define in which lattice location a new particle should be inserted. In this work we propose a new scalar $\Upsilon \in \mathbb{R}$ which defines the priority of a voxel having a particle created in it. This priority is calculated using multiresolution tensors characteristics and geometric features of the scene (Fig. 1).

In [1], the position and orientation of a tensor in relation to the observer are evaluated by three scalars k_1, k_2 and k_3. Using multiresolution with three scales $s = \{1, 2, 3\}$, the coefficients can be evaluated for each scaled tensor of a location. We propose the following scalars to capture the viewer-dependent orientation of the l-th multiresolution tensors $\mathbf{t}_l^1$, $\mathbf{t}_l^2$ and $\mathbf{t}_l^3$, all centered in the domain at position $\vec{x}_l$:

$$
\begin{aligned}
k_1^s &= 1 - |\vec{e}_1^s \cdot \vec{obs}| \\
k_2^s &= 1 - |\vec{e}_2^s \cdot \vec{obs}| \\
k_3^s &= |\vec{e}_3^s \cdot \vec{obs}|,
\end{aligned}
$$

where $\vec{e}_1^s$, $\vec{e}_2^s$ and $\vec{e}_3^s$ are the eigenvectors of the tensor and $\vec{obs}$ corresponds to the camera view vector (Fig. 1). Thus, we propose nine scalars to capture the orientation of the local tensor in relation to the observer, which means three values for each of the three scales. These nine values quantify the relative position of the observer with respect to the tensors eigensystems, so that we can prioritize tensors representing colinear or coplanar structures which are perpendicular to the observer.

We need to define the weight of each scale in the calculation of the priority value. A simple but effective approach is to fix the weights in 2.0 for the original tensor (scale 1), 1.0 for the intermediate tensor (scale 2), and 0.5 for the tensor of the maximum scale 3. The distance of the tensor to the observer d_{obs}:

$$
d_{obs} = \frac{|\vec{x}_l - \vec{x}_{obs}|}{MAX(d_{obs})},
$$

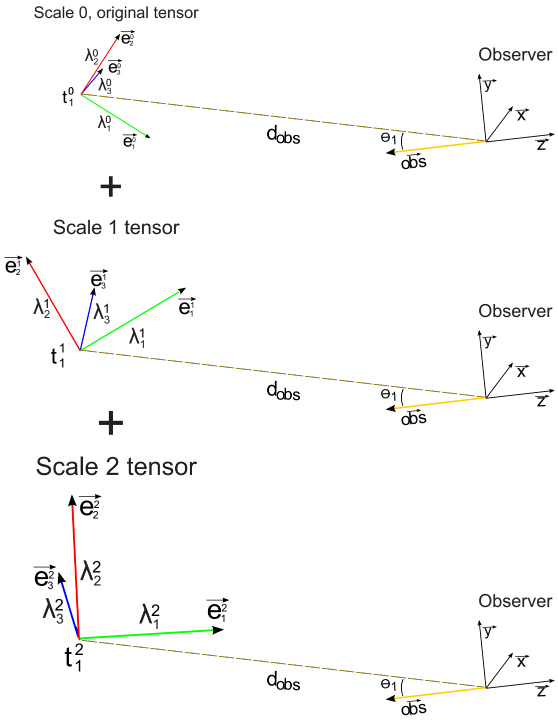


Fig. 1. Combination of tensors in multiple scales, related to the observer.

which is normalized by the greatest distance in the field $MAX(d_{obs})$, gives the the viewer-dependent weight:

$$w = 1 - d_{obs}.$$

The scalar Υ, that indicates the priority of a voxel to receive a particle, is defined as:

$$\begin{aligned} \Upsilon_{\mathbf{t}} =& 2.0 \cdot w \cdot (A_3^1 + c_l^1 + k_1^1 + k_2^1 + k_3^1) + \\ & 1.0 \cdot w \cdot (A_3^2 + c_l^2 + k_1^2 + k_2^2 + k_3^2) + \\ & 0.5 \cdot w \cdot (A_3^3 + c_l^3 + k_1^3 + k_2^3 + k_3^3), \end{aligned} \tag{5}$$

which is a linear combination of the following terms:

– coefficient of linear anisotropy of the scaled tensor (c_l^s) (Eq. 1);
– asymmetry of the tensor eigenvalues (A_3^s): changes from negative to positive as the scaled tensor vary from planar to linear (Eq. 2);
– coefficients of orthogonality between the observer and the first eigenvector of each scaled tensor (k_1^s): bigger if the main direction of the tensor is perpendicular to the view vector;

– coefficients of orthogonality between the observer and the second eigenvector of each scaled tensor (k_2^s): bigger if the second main direction of the tensor is perpendicular to the view vector;
– coefficients of parallelism between the observer and the third eigenvector of each scaled tensor (k_3^s): bigger if the third eigenvector is aligned with the view vector, which implies that the other eigenvectors are perpendicular to the observer.

4.2 Particle Insertion

A maximum number of particles $N_p \in \mathbb{N}$ is fixed by the user. This value is generally small compared to the size of the tensor field. At most N_p particles exist and walk through the field at a given time. The priority list is used to achieve better visualization results by inserting particles in the more interesting sites.

When the simulation begins or the user changes its position or orientation, all tensors $\mathbf{t} \in \mathbb{T}^0$ have their priority value computed (Eq. 5). They are sorted in a list where the highest priorities are positioned on the top. Using the N_p topmost tensors, the total priority is computed:

$$m = \sum_{l=1}^{N_p} |\Upsilon_l|.$$

The topmost tensors $\mathbf{t}_l \in \mathbb{T}^0$, $1 \le l \le N_p$, are allowed to have

$$n_l = N_p \cdot \frac{|\Upsilon_l|}{m}$$

particles, which represents the proportion of new particles that can be assigned to the position of the tensor $\mathbf{t}_l$ along an insertion round. Note that some of the tensors will not have enough priority to receive a particle. Particles are thus created in less than N_p tensor positions.

Initially, there are N_p particles to be inserted into the domain at the beggining of an insertion round. If we insert n_l particles for the topmost tensor, there may be several particles walking together or very close to each other. This is not desired because multiple particles together are not visually salient. Thus, we propose to assign only one particle to each tensor of the list (with non-zero n_l) at a time, decrementing its n_l value upon insertion. If there are still particles left for insertion after visiting the position N_p of the list we return to its topmost tensor, running through the list in a circular way. When all particles are inserted, all n_l are zero, indicating the end of an insertion round. We then reestablish n_l and a new insertion round begins. This round-robin policy for particle insertion guarantees all sites with non-zero n_l have at least one particle inserted before any previously assigned tensor is visited again.

Note that only N_p particles are viewed in the domain. As the simulation runs, some particles are removed. Their reinsertion obeys the round-robin policy and the visual result are well distributed particle clouds. The topmost tensors are guaranteed to have more particles inserted during simulation.

The Υ_l scalar has viewer-dependent terms, so, it is necessary to reorder the priority list when the camera position or orientation changes. A merge sort algorithm is enough for having good response times with 100.000 particles.

The simulation and the particle removal steps are explained in [1]. Given the tensorlines, a simple advection step determines the next position of a particle. Due to isotropic regions in the tensor field, particles can get stuck. To reduce the creation of particles in an isotropic region, some tensors are flagged as bad places when particles inserted on them are removed after few advection iterations. Those tensors periodically receive particles that disappear rapidly, generating flickering regions. Their elimination from the particle insertion process resulted in a much better visualization.

5 Results

Here we present the results for the application of our method to three different tensor fields: the 3-point field, the helical flow and a diffusion tensor field of a brain. In all of the experiments, we used the color palette shown in Figure 2 to represent the importance of a given tensor to the observer. Each particle was represented as a pointer glyph, just as shown in [1].

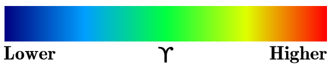


Fig. 2. Color palette used for Υ [1].

For the helical field with 38x39x40 grid, we used 7000 particles spread through the sites according to the generated priority list. In this process, 3214 sites in isotropic regions were eliminated from the 12073 possible ones. Figure 3 shows the helical field. Figure 4 shows the visualization of the helical tensor field from different points of view. Notice that tensors nearer and perpendicular to the observer tend to have higher priority, thus their color being closer to red. As the camera orientation is changed, the priority list is recalculated and the new best ranked tensors in the list are then displayed with proper colors. This can be seen by looking at the density of particles. The amount of particles decreases as it gets far from the observer, since the nearest sites have higher priority.

Next, we present the results obtained for a diffusion tensor field of a brain [21] (Fig. 5). These tensor fields are usually generated by magnetic resonance imaging. They are very useful in detecting fibers, which are represented by regions of high linear and planar anisotropy. Similarly, the crossings of white matter tracts in the brain are also identified with higher planar anisotropy. Thus, we can use this knowledge to enhance the visualization of these regions of the brain DT image.

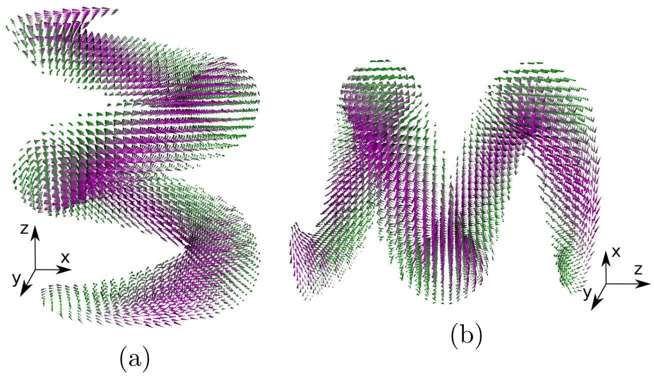


Fig. 3. The helical tensor field visualization from two different angles.

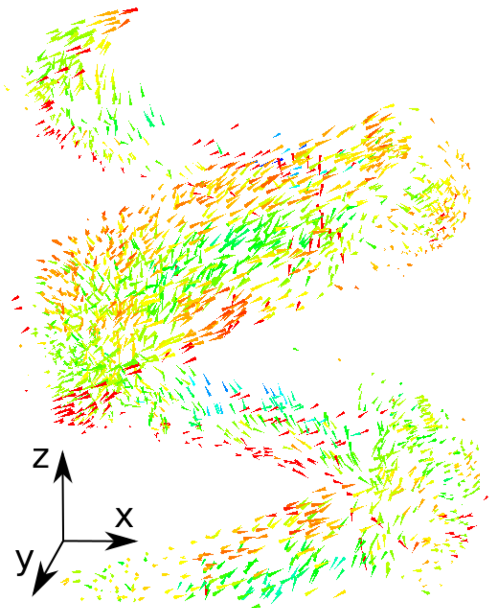
Figures 6 and 7 show examples of the field visualization under different camera orientations. In this simulation, the grid dimension was 74x95x80 and the number of particles created was 15000. Plus, 12366 sites were flagged for elimination from the 24582 initially available. In this field it was possible to see one of the main advantages of excluding sites from creation: significantly reduction of flickering effect. The removal criterion destroys particles which they could cause flickering by reaching isotropic regions. But particles that have a short time between their creation and destruction, like 1 or just 2 simulation steps, also result in flickering sensation. As mentioned in Section 4, we exclude sites in which created particles are soon destroyed, and that really presented a better view for the simulation.

Finally, we simulated a 3-point field (Fig. 8). It represents a 38x39x40 grid where there are three spherical charges at positions (0,0,0), (38,0,40) and (38,39,40). The tensor field is calculated as the geometric influence of all three charges at every position of the grid.

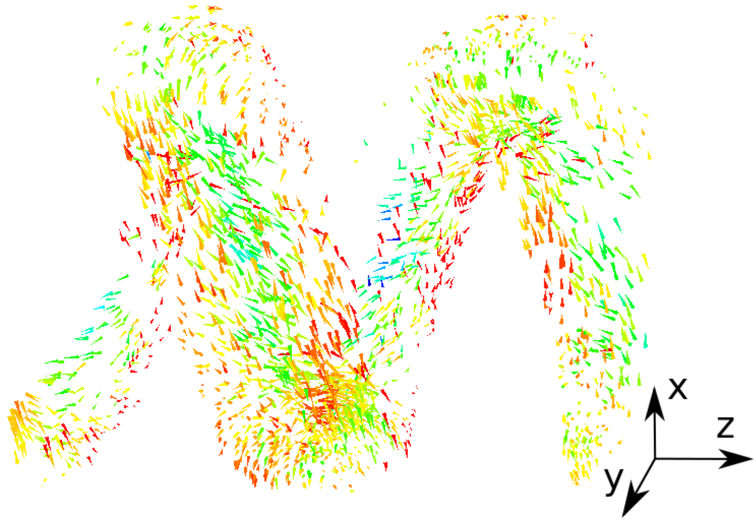
With this example it was possible to analyse some features of tensor fields that are of interest for visualization: continuities and curvatures. We could see the importance of the anisotropy factor on choosing where to create the particles (Fig. 9). This visualization used 30000 particles. This factor combined with the relative position of the observer creates a huge flow near the observer, allowing to follow the particles and to notice the smoothness of most of the field. There were 9115 eliminated sites from a total of 56495 initially possible.

6 Conclusion

Choosing where to create particles in a tensor field for a good visualization is not an easy task. We have combined multiresolution coefficients of the field with viewer-dependent terms in order to evaluate the importance of each site of the grid at the current observer's position. The multiresolution coefficients allowed



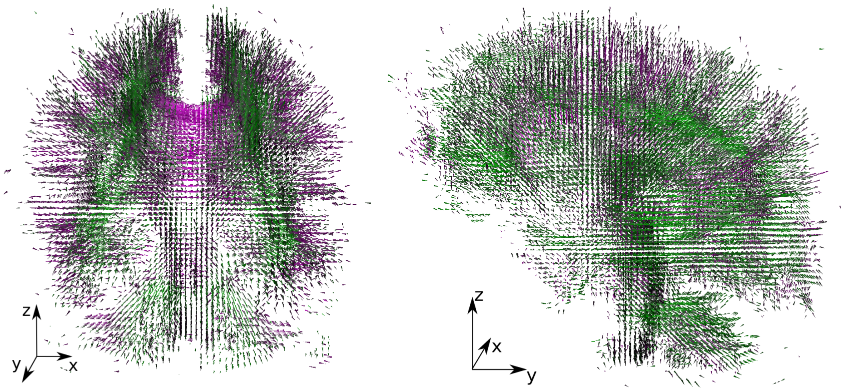
(a)



(b)

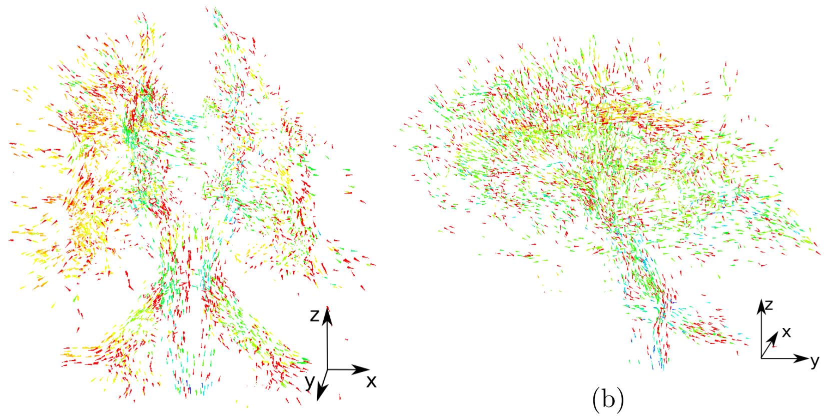
Fig. 4. Visualization of the helical tensor field with 7000 particles.

us to check the anisotropy of the field at different scales. With this information, it was possible to reduce the high amount of terms used on our last approach [1] for ranking each possible creation site. The priority list using multiresolution



(a) (b)

Fig. 5. The original brain tensor field visualization from two different angles.



(a) (b)

Fig. 6. Visualization of the brain field simulation associated to the viewing angles in Figure 5

information and a deterministic algorithm for balanced particle insertion are the main contributions of this paper.

We have shown our results on three different tensor fields. Increasing the capacity for creating particles at higher priority sites did concentrate a large number of particles on the spots of most interest, near the observer. We have also determined rules to permanently remove sites from the priority list. By eliminating these sites from the list we could reduce the flickering problem we had to almost none.

The multiresolution terms of the priority value (Eq. 5) represent smoothed tensor structures. Our results show that these local and filtered anisotropy esti-

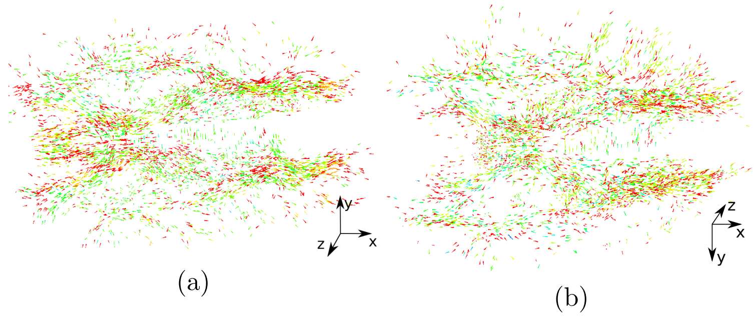


Fig. 7. Additional viewing angles from the brain simulation

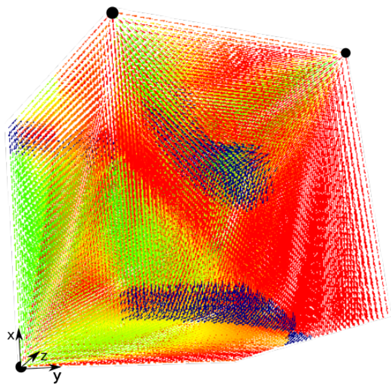


Fig. 8. Original 3-point field. The black circles represents charge positions.

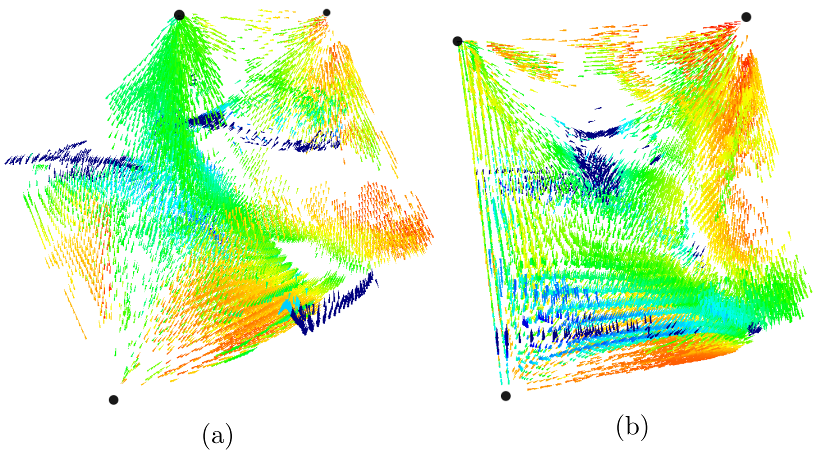


Fig. 9. Visualization of the 3-point field simulation at two different simulation steps.

mations have improved the particle tracing proposed in [1] for tensor field visualization, since tensors representing colinear and coplanar structures (anisotropic in several scales) tend to have more particles during simulation.

References

1. Leonel, G.A., Peçanha, J.P., Vieira, M.B.: A viewer-dependent tensor field visualization using particle tracing. In: Proceedings of the 2011 international conference on Computational science and its applications - Volume Part I. ICCSA'11, Springer-Verlag (2011) 690–705
2. Kondratieva, P., Krüger, J., Westermann, R.: The application of gpu particle tracing to diffusion tensor field visualization. In: Visualization, 2005. VIS 05. IEEE. (2005) 73–78
3. Shaw, C.D., Ebert, D.S., Kukla, J.M., Zwa, A., Soboroff, I., Roberts, D.A.: Data visualization using automatic, perceptually-motivated shapes. In: Proceeding of Visual Data Exploration and Analysis, SPIE. (1998)
4. Shaw, C.D., Hall, J.A., Blahut, C., Ebert, D.S., Roberts, D.A.: Using shape to visualize multivariate data. In: NPIVM '99: Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM internation conference on Information and knowledge management, New York, NY, USA, ACM (1999) 17–20
5. Kindlmann, G.: Superquadric tensor glyphs. In: Proceedings of IEEE TVCG/EG Symposium on Visualization 2004. (May 2004) 147–154
6. Westin, C.F.: A Tensor Framework for Multidimensional Signal Processing. PhD thesis, Linköping University, Sweden, S-581 83 Linköping, Sweden (1994) Dissertation No 348, ISBN 91-7871-421-4.
7. Delmarcelle, T., Hesselink, L.: Visualization of second order tensor fields and matrix data. In: VIS '92: Proceedings of the 3rd conference on Visualization '92, Los Alamitos, CA, USA, IEEE Computer Society Press (1992) 316–323
8. Delmarcelle, T., Hesselink, L.: Visualizing second-order tensor fields with hyper streamlines. In: IEEE Computer Graphics and Applications, Volume 13, Issue 4, Los Alamitos, CA, USA, IEEE Computer Society Press (1993) 25–33
9. Weinstein, D., Kindlmann, G., Lundberg, E.: Tensorlines: advection-diffusion based propagation through diffusion tensor fields. In: VIS '99: Proceedings of the conference on Visualization '99, Los Alamitos, CA, USA, IEEE Computer Society Press (1999) 249–253
10. Vilanova, A., Zhang, S., Kindlmann, G., Laidlaw, D.: An introduction to visualization of diffusion tensor imaging and its applications. Visualization and Processing of Tensor Fields (2006) 121–153
11. McGraw, T., Nadar, M.: Stochastic dt-mri connectivity mapping on the gpu. Visualization and Computer Graphics, IEEE Transactions on **13**(6) (2007) 1504–1511
12. Köhn, A., Klein, J., Weiler, F., Peitgen, H.: A gpu-based fiber tracking framework using geometry shaders. In: Proceedings of SPIE Medical Imaging. Volume 7261. (2009) 72611J
13. Evert, A., Neda, S., Andrei, J.: Cuda-accelerated geodesic ray-tracing for fiber tracking. International Journal of Biomedical Imaging **2011** (2011)
14. Mittmann, A., Nobrega, T., Comunello, E., Pinto, J., Dellani, P., Stoeter, P., von Wangenheim, A.: Performing real-time interactive fiber tracking. Journal of Digital Imaging **24**(2) (2011) 339–351

15. Crippa, A., Jalba, A., Roerdink, J.: Enhanced dti tracking with adaptive tensor interpolation. Visualization in Medicine and Life Sciences II (2012) 175–192
16. Rodrigues, P., Jalba, A., Fillard, P., Vilanova, A., ter Haar, B.: A multi-resolution watershed-based approach for the segmentation of diffusion tensor images. In: MICCAI Workshop on Diffusion Modelling. (2009) 161–172
17. Kindlmann, G.: Visualization and Analysis of Diffusion Tensor Fields. PhD thesis (September 2004)
18. Bahn, M.: Invariant and Orthonormal Scalar Measures Derived from Magnetic Resonance Diffusion Tensor Imaging. Journal of Magnetic Resonance **141**(1) (November 1999) 68–77
19. Delmarcelle, T., Hesselink, L.: Visualization of second order tensor fields and matrix data. In: Visualization, 1992. Visualization'92, Proceedings., IEEE Conference on, IEEE (1992) 316–323
20. Mallat, S.: A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way. 3rd edn. Academic Press (2008)
21. Kindlmann, G.: Diffusion tensor mri datasets. http://www.sci.utah.edu/~gk/DTI-data/