

A Viewer-dependent Tensor Field Visualization Using Particle Tracing

Gildo de Almeida Leonel, João Paulo Peçanha, and Marcelo Bernardes Vieira

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

Abstract. *Tensor field visualization is a hard task due to the multivariate data contained in each local tensor. In this paper, we propose a particle-tracing strategy to let the observer understand the field singularities. Our method is a viewer-dependent approach that induces the human perceptual system to notice underlying structures of the tensor field. Particles move throughout the field in function of anisotropic features of local tensors. We propose a easy to compute, viewer-dependent, priority list representing the best locations in tensor field for creating new particles. Our results show that our method is suitable for positive semi-definite tensor fields representing distinct objects.*

Key words: Tensor Field, Particle Tracing, Dynamic Visualization, Scientific Visualization.

1 Introduction

Arbitrary tensor fields are very useful in a large number of knowledge areas like physics, medicine, engineering and biology. The main goal of the study of tensors in these areas is to investigate and seek for collinear and coplanar objects represented by tensors. These objects or artifacts are formed by subsets of arranged and structured tensors which capture some geometric continuity like, for example, fibers.

The best visualization methods must offer different features to allow the observer to see as many aspects of tensor multivariate data as possible. Therefore, it is very hard to combine in a single method all the expected functionalities. In this paper we introduce a visualization process suitable for many different positive semi-definite tensor fields. Our goal is to highlight most continuity information in a simple and adaptive fashion. An interesting approach may take into account not only the static data given by an ordinary tensor field. It can also use other information like the object's surrounding space and the observer (i.e. camera model) to generate and modify the visual data.

In this paper we present a dynamic method to visualize tensor fields. It takes into account the observer point of view and other attributes aiming to highlight collinear and coplanar information. The particle motion incites the human perceptual system to fuse and perceive salient features. The work of [1]

also use particle tracing to extract visual information of a tensor field. However, the criterion to create particles is purely random, which may generate some confusing results. In this paper we defined a priority list to choose the best places in the space where particles should be born to produce a superior viewing result. The priorities are computed by a linear combination of anisotropic measures of tensors and by the viewer camera parameters.

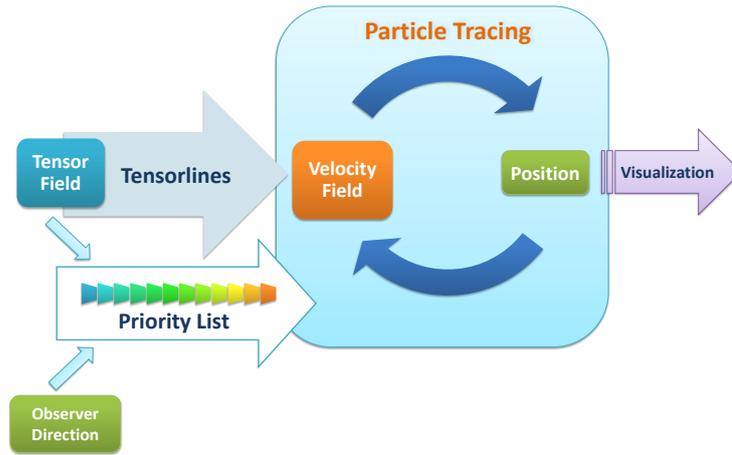


Fig. 1. Schematic representation of the proposed method.

An overview of our method is depicted in Figure 1, having the following steps: extract the best velocity vector from the tensor field, use a priority list to define where new particles will appear, perform the advection of particles.

2 Related Works

In tensor field visualization we can adopt different approaches to represent information. The discrete approach is commonly used when punctual data is sufficient to obtain the required information. A superquadric glyph is an ordinary fashion to represent local information given by the field mapping into geometric primitives, like cubes, ellipses and cylinders this information. Using the concept of glyphs, Shaw et al [2] have developed their work for multidimensional generic data visualization. Their main contribution is to connect the advantages of the visual human being perception and superquadrics intrinsic interpolation feature. In a later extension [3], they propose to measure how many forms assumed by superquadric can be distinguished by human vision system. Westin et al [4] proposed an anisotropic metric to identify and compare a set of glyphs. Kindlmann

[5] defines a linear mapping of shape coefficients in order to view anisotropic and isotropic tensors. They present the problem of ambiguous glyphs that can induce to wrong visual conclusions when the glyphs adopt planar or linear forms. In one hand, all problems that involves symmetry can be solved using ellipsoidal glyphs. In other hand, there are ambiguity situations in the visual identification of the tensor. If the point of view direction is aligned to the main eigenvector, ellipsoidal linear tensors can be identified as spheres. To overcome this problem, he presents a new parametrization of superquadric tensor glyphs to better represent shape and orientation.

There are also continuous methods for tensor visualization based upon the tensor interpolation of two distinct points in a multidimensional space. In [6] the concept of tensor field lines - extended from [7] - is generalized, and the concept of hyperstreamlines is introduced. In that work they represent all information of a tensor field taking into account not points, but the trajectory generated by the tensor using its eigenvectors. This approach is interesting to visualize symmetric tensor fields, where its eigenvectors are real and orthogonal. However, the field becomes hard to visualize for a large number of hypersetreamlines. Delmarcelle et al [8] have presented another problem with hypersetreamlines: the degeneration when a tensor has at least two equal eigenvectors. In [9] is presented a method to avoid degeneration due to planar and spherical tensors in input data. This method was applied in tensors fields obtained from magnetic resonance images.

Zheng and Pang [10] proposed a method to visualize tensor field using the concept of linear integral convolution. Their work is an extension of [11], which uses a white texture noise and hyperstreamlines to generate the visual information.

Dynamical particles walking through a tensor field is a powerful and recent method for visualization. The sensation of movement incites the human perceptual system making easier the understanding of some field properties. Kondratieva et al [1] has proposed a dynamical approach using particle tracing in GPU (Graphic Processing Unit). They argue that particle tracing gives an efficient and intuitive way to understand the tensor field dynamics. The advection of a set of particles in a continuous flow is used to induce particle motion. Through the tensor field, a direction vector field is generated - based on [12] - and then, the advection using this vector field is performed.

3 Fundamentals

3.1 Orientation Tensor

A local orientation tensor is a special case of non-negative symmetric rank 2 tensor. It was introduced by Westin [13] to estimate orientations in a field. This tensor is symmetric and can be saw as a pondered sum of projections:

$$\mathbf{T} = \sum_{i=1}^n \lambda_i e_i e_i^T, \quad (1)$$

where $\{e_1, e_2, \dots, e_m\}$ is a base of \mathbb{R}^n . Therefore, it can be decomposed into:

$$\mathbf{T} = \lambda_n \mathbf{T}_n + \sum_{i=1}^{n-1} (\lambda_i - \lambda_i + 1) \mathbf{T}_i, \quad (2)$$

where λ_i are the eigenvalues corresponding to each eigenvector e_i . This is an interesting decomposition because of its geometric interpretation. In fact, in \mathbb{R}^3 , an orientation tensor \mathbf{T} decomposed using Equation 2 can be represented using the contribution of its linear, planar, and spherical intrinsic features:

$$\mathbf{T} = (\lambda_1 - \lambda_2) \mathbf{T}_l + (\lambda_2 - \lambda_3) \mathbf{T}_p + \lambda_3 \mathbf{T}_s. \quad (3)$$

A \mathbb{R}^3 tensor decomposed by Equation 3, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$, can be interpreted as following:

- $\lambda_1 \gg \lambda_2 \approx \lambda_3$ corresponds to an approximately linear tensor, with the spear component being dominant.
- $\lambda_1 \approx \lambda_2 \gg \lambda_3$ corresponds to an approximately planar tensor, with the plate component being dominant.
- $\lambda_1 \approx \lambda_2 \approx \lambda_3$ corresponds to an approximately isotropic tensor, with the ball component being dominant, and no main orientation present.

For many purposes only the main direction of the tensor is necessary. Furthermore, the shape of the tensor is generally more important than its magnitude. Using the sum of the tensor eigenvalues, one may obtain the linear, planar, and spherical coefficients of anisotropy:

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}, \quad (4)$$

$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}, \quad (5)$$

$$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}. \quad (6)$$

Note that coefficients in Equations 5 and 6 were scaled by 2 and 3, respectively, so that each of them independently lie in the range $\in [0, 1]$ with $c_l + c_p + c_s = 1$ [13].

3.2 Invariants Towards Eigenvalues

The eigenvalues of a tensor \mathbf{D} can be calculated solving:

$$\det(\lambda \mathbf{I} - \mathbf{D}) = 0.$$

Hence:

$$\det(\lambda \mathbf{I} - \mathbf{D}) = \begin{vmatrix} \lambda - D_{xx} & -D_{xy} & -D_{xz} \\ & \lambda - D_{yy} & -D_{yz} \\ & & \lambda - D_{zz} \end{vmatrix} = \lambda^3 - J_1 \lambda^2 + J_2 \lambda - J_3,$$

where,

$$\begin{aligned} J_1 &= D_{xx} - D_{yy} - D_{zz} = \text{tr}(\mathbf{D}), \\ J_2 &= D_{xx}D_{yy} + D_{xx}D_{zz} + D_{yy}D_{zz} - D_{xy}^2 - D_{xz}^2 - D_{yz}^2 = \frac{\text{tr}(\mathbf{D})^2 - \text{tr}(\mathbf{D}^2)}{2}, \\ J_3 &= 2D_{xy}D_{xz}D_{yz} + D_{xx}D_{yy}D_{zz} - D_{xz}^2D_{yy} - D_{yz}^2D_{xx} - D_{xy}^2D_{zz} = \det(\mathbf{D}). \end{aligned} \quad (7)$$

so that $\text{tr}(\mathbf{D})$ and $\det(\mathbf{D})$ are the trace and the determinant of tensor \mathbf{D} , respectively.

The matrix determinant is invariant to basis changing and thus is classified as an algebraic invariant. Another useful invariant used to determine the eigenvalues of a tensor is the squared norm:

$$\begin{aligned} J_4 = \|\mathbf{D}\|^2 &= J_1^2 - 2J_2 \\ &= D_{xx}^2 + 2D_{xy}^2 + 2D_{xz}^2 + D_{yy}^2 + 2D_{yz}^2 + D_{zz}^2 \\ &= \lambda_1^2 + \lambda_2^2 + \lambda_3^2. \end{aligned} \quad (8)$$

3.3 Eigenvalue Wheel

Kindlmann [14] describes other three invariants used to solve a cubic polynomial:

$$Q = \frac{J_1^2 - 3J_2}{9} = \frac{J_4 - J_2}{9} = \frac{3J_4 - 3J_1^2}{18} \quad (9)$$

$$R = \frac{-9J_1J_2 + 27J_3 + 2J_1^3}{54} = \frac{-5J_1J_2 + 27J_3 + 2J_1J_4}{54} \quad (10)$$

$$\Theta = \frac{1}{3} \cos^{-1} \left(\frac{R}{\sqrt{Q^3}} \right). \quad (11)$$

The wheel eigenvalues can be defined as a wheel with three equally placed radii centered on the real number line at $J_3/3$. The radius of the wheel is $2\sqrt{Q}$, and Θ measures the orientation of the first radius [14].

The central moments of a tensor determines the geometric parameters of the eigenvalue wheel. The central moments are defined as:

$$\begin{aligned}\mu_1 &= \frac{1}{3} \sum \lambda_i = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} = J_1/3 \\ \mu_2 &= \frac{1}{3} \sum (\lambda_i - \mu_1)^2 = \frac{2(\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - \lambda_1\lambda_2 - \lambda_1\lambda_3 - \lambda_2\lambda_3)}{9} = 2Q \\ \mu_3 &= \frac{1}{3} \sum (\lambda_i - \mu_1)^3 = 2R.\end{aligned}$$

The second central moment μ_2 is the variance of the eigenvalues, and the standard deviation is $\sigma = \sqrt{\mu_2} = \sqrt{2Q}$. The asymmetry A_3 of the eigenvalues is defined as follows [15]:

$$A_3 = \frac{\mu_3}{\sigma^3} = \frac{\sum (\lambda_i - \mu_1)^3}{3\mu_2\sqrt{\mu_2}} = \frac{R}{\sqrt{2Q^3}}. \quad (12)$$

3.4 Anisotropy

In literature we can find many forms to measure the tensor anisotropy. The fractional anisotropy (FA), relative anisotropy (RA), volume ratio and others, can be computed using the tensor eigenvalues [16].

The FA [16] and RA [17] are defined as following:

$$\begin{aligned}FA &= \frac{3}{\sqrt{2}} \sqrt{\frac{\mu_2}{J_4}} = 3\sqrt{\frac{Q}{J_4}} = \sqrt{\frac{J_4 - J_2}{J_4}} \\ RA &= \frac{\sqrt{\mu_2}}{\sqrt{2}\mu_1} = \frac{3\sqrt{Q}}{J_1}.\end{aligned} \quad (13)$$

3.5 Tensorlines

The tensorlines concept is an extension of the hyperstreamlines method proposed in [6]. Hyperstreamlines is obtained by a smooth path tracing. This is done by using the main tensor eigenvector to perform line integration. The degeneration problem in this method incited Weinstein *et al* [9] to develop an extension called tensorlines. The tensorlines method uses multiple tensor features to determine the correct path to follow. It stabilizes the propagation incorporating two additional terms \mathbf{v}_{int} and \mathbf{v}_{out} given by:

$$\mathbf{v}_{out} = \mathbf{T}\mathbf{v}_{in}, \quad (14)$$

so that \mathbf{v}_{in} is the incoming direction, \mathbf{v}_{out} the outgoing direction and \mathbf{T} the local tensor. The \mathbf{v}_{in} vector corresponds to the propagation direction in the previous step, and \mathbf{v}_{out} is the input vector transformed by the tensor.

The propagation vector used in the integral is a linear combination of e_1 , \mathbf{v}_{in} , and \mathbf{v}_{out} . The next propagation vector, \mathbf{v}_{prop} , depends on the shape of the tensor:

$$\mathbf{v}_{prop} = c_l e_1 + (1 - c_l) ((1 - w_{punct}) \mathbf{v}_{in} + w_{punct} \mathbf{v}_{out}), \quad (15)$$

where $w_{punct} \in [0, 1]$ is a parameter defining the penetration into isotropic regions [9].

3.6 Particle Tracing

The tensorline method generates a vector field that can be visualized using many approaches. One of those is called particle tracing. In this method, massless particles are inserted into the field subspace and their movements are coordinated by its vectors.

It is necessary to compute the particle position \vec{x} in time t over velocity \vec{v} each time-step. The mathematical model for this problem is straightforward. A given particle p , is identified by your initial position \vec{x}_{p_0} with velocity $\vec{v}_p(p, t)$. We must find $\vec{x} \in \mathbb{R}^n$:

$$\begin{cases} \frac{d\vec{x}_p}{dt} = \vec{v}_p(\vec{x}_p, t) & t \in [t_0, T_p] \\ \vec{x}_p|_{t=t_0} = \vec{x}_{p_0} \end{cases}, \quad (16)$$

where T_p is the time for particle p walk through all domain Ω .

4 Proposed Method

One common problem in tensor field visualization is ambiguity. In glyph-based visualization, tensors with different forms may appear similar in 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 can adopt 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. We will apply the benefits of observer metrics to propose a visualization method based on particle tracing.

In our work, particles in motion will represent the features of the tensor field. One critical point in visualization using particle tracing is to define the particle starting point. The most intuitive approach used to insert particles into the domain is to compute new positions randomly. However, a fixed distribution function will generally not insert new particles in most interesting sites.

4.1 Priority Features

Let $\mathbb{T}_{x \times y \times z}$ being a discrete and finite tensor field with lattice given by $x, y, z \in \mathbb{N}$ so that $\mathbb{T} = \{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3 \dots \mathbf{t}_n\}$, and composed by $|\mathbb{T}| = N$ tensors. 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 $\mathbf{t}_i \in \mathbb{T}$.

To correct visualize the tensor field we need to define a criterion to generate and insert particles into the domain \mathbb{T} . This should be done in accordance with field variants and properties. In this work we propose a scalar $\mathcal{Y} \in \mathbb{R}$, which defines the priority of a voxel to have a particle being created on it. The \mathcal{Y} is also used to define a color palette aiming to highlight the desired properties. This priority is calculated using tensors characteristics and geometric features of the scene (Fig. 2).

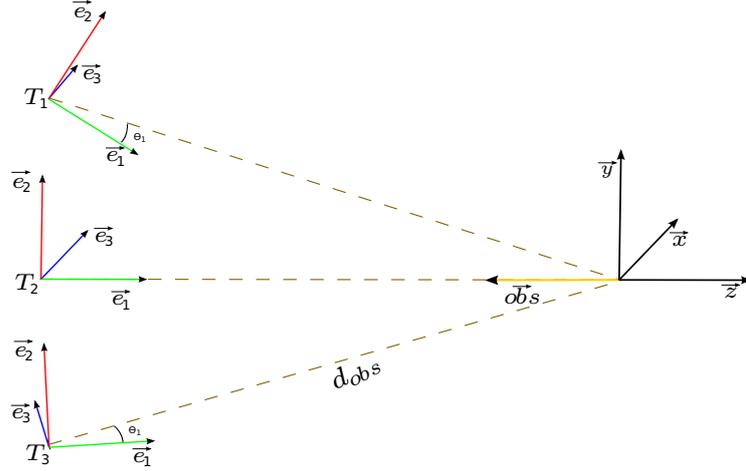


Fig. 2. Simulation space.

To evaluate the tensor position in relation to the observer we propose the coefficients k_1 , k_2 e k_3 :

$$k_1 = 1 - |\mathbf{e}_1 \cdot \mathbf{obs}| \quad (17)$$

$$k_2 = 1 - |\mathbf{e}_2 \cdot \mathbf{obs}| \quad (18)$$

$$k_3 = |\mathbf{e}_3 \cdot \mathbf{obs}|, \quad (19)$$

where \mathbf{e}_1 , \mathbf{e}_2 and \mathbf{e}_3 are the eigenvectors of the tensor and \mathbf{obs} is the vector that corresponds to the camera view.

Another important coefficient is the Euclidean distance between the observer and the tensor d_{obs} :

$$d_{obs} = \frac{|\mathbf{x}_T - \mathbf{x}_{obs}|}{MAX(d_{obs})}, \quad (20)$$

this distance is normalized by the greatest distance in the field $MAX(d_{obs})$.

These coefficients are used together with tensor attributes to evaluate the priority of a voxel receive a particle. For this proposal, we will calculate the scalar \mathcal{Y} as the linear combination of the following terms:

- average of the eigenvalues of the tensor (μ_1): related to the tensor size;

- variance of the eigenvalues of the tensor (μ_2): a bigger variance indicates that the tensor will probably have a planar or linear anisotropy;
- asymmetry of the tensor eigenvalues (A_3): changes from negative to positive as the tensor vary from planar to linear;
- standard square of a tensor (J_4): related to amplification imposed by the tensor;
- coefficient of fractional anisotropy (FA) and relative anisotropy (RA): used to detect anisotropy and isotropic regions;
- coefficient of orthogonally between the observer and the first eigenvalue (k_1), with the second eigenvalue (k_2) and the third eigenvalue (k_3): quantify the relative position of the observer in relation to the tensor eigensystem, so we can prioritize tensors that are parallel or orthogonal to the observer;
- normalized distance to the observer (d_{obs}): reveal tensors closer to the screen;
- coefficient of linear anisotropy (c_l), coefficient of planar anisotropy (c_p): also allow to differentiate anisotropy.

Tensor fields may present multivariate information coming from many different applications. Aiming to generate appropriate results, the scalar \mathcal{Y} will be parameterized by the user in order to focus on the desired characteristics:

$$\begin{aligned} \mathcal{Y}_{\mathbf{t}} = & \alpha_1 \mu_1 + \alpha_2 \mu_2 + \alpha_3 A_3 + \alpha_4 J_4 + \alpha_5 FA + \alpha_6 RA \\ & + \alpha_7 k_1 + \alpha_8 k_2 + \alpha_9 k_3 + \alpha_{10} d_{obs} + \alpha_{11} c_l + \alpha_{12} c_p. \end{aligned} \quad (21)$$

where $\alpha_i \in [-1, 1]$ and $\mathbf{t} \in \mathbb{T}$. So, the $\mathcal{Y}_{\mathbf{t}}$ ponders how much the tensor $\mathbf{t} \in \mathbb{T}$ presents the required information.

4.2 Priority List and Particle Insertion

In the application beginning, a number $N_p \in \mathbb{N}$ of particles will be established by the user. The program will allocate all the necessary memory and particles are initialized, but not immediately inserted into the space. In the next step, all tensors $\mathbf{t} \in \mathbb{T}$ will be sorted and ranked in a list with most important elements (higher $\mathcal{Y}_{\mathbf{t}}$) positioned on the top (Fig. 3).

To insert a particle p_i into the domain, a random number $\kappa \in [0, 1]$ is generated using a standard normal distribution and then we select the correspondent z -th tensor, $z \in [0, N - 1]$, in the priority list:

$$z = \frac{\kappa N}{\varsigma} \quad (22)$$

where N is the total number of tensors and ς defines a Gaussian distribution. A bigger ς implies a higher frequency of choice of the top tensors in the priority list. The particle p_i will be created in the position of the z -th tensor spatial position. The process of particle insertion stops when the domain contains at least N_p particles.

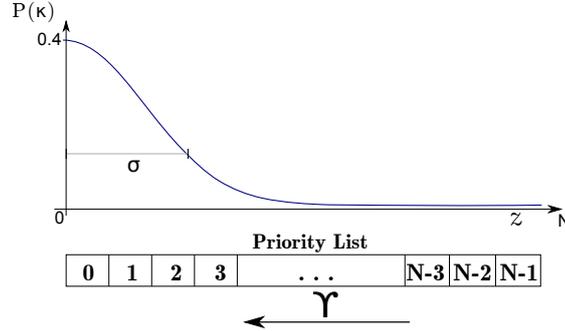


Fig. 3. Normal probability distribution on the priority list.

The next algorithm step is to define a main direction $\vec{v} \in \mathbb{R}^3$ for the new particle. We may define a unique direction for each particle. This can be done using a gradient of a specific field attribute. However, the created particle initially will always move towards the same direction. A better solution is to invert the initial direction of half the number of creations.

The \mathcal{Y} scalar has viewer-dependent terms, so, it is necessary to reorder the priority list on every change of the camera position and orientation. This process can be computationally expensive, and impairs the visualization performance.

To deal with this problem we use the following implementation strategy: after the first iteration, the priority list stays partially ordered only if the camera changes are not abrupt. If a full reordering is needed, the quicksort algorithm with median-of-three partitioning [18] is performed. This algorithm has presented relatively good performance results, leading to a real time visualization.

4.3 Particle Removal

The particle is removed from the visualization space when it reaches one of the following situations: a) it is located at a bigger isotropic region b) get away from the visualization lattice, and c) when the absolute value of the dot product between the entry direction into a voxel and the current voxel propagation direction is equal to zero or smaller than a threshold $\gamma \in \mathbb{R}$. We have found empirically the value $\gamma = 0.3$ as a good parameter to avoid that a particle get stuck between two voxels with opposite directions. It implies that the angle among these two directions should be in the interval $(72.54^\circ, 90.00^\circ]$.

In our implementation, for performance reasons, no particle is deallocated until the application ends. When the stop criterion is reached for a determined particle, its computational resources are reused and it is recreated using the priority list.

In Kondratieva *et al.* [1] work, it is proposed that particles should be restarted in its original initial position. Later, Kondratieva [19] concludes that the previous approach needed modifications. They observed a flicker behavior in the

display due to the presence of high frequencies in the field. This situation may distract the user and disturb the visualization. The authors proposed that particles should always restart at random in the tensor field.

In our method, a fixed number of particles is created, and during the visualization process they are inserted into the lattice taking into account the priority list and removed when it is necessary. When a particle is destroyed, it restarts using the creation criterion. Thus, particles may reborn and highlight different features in the same simulation. Further information will be highlighted if the user, at runtime, manually changes the parameters α_i presented in Equation 21, or changes the observer's point of view.

Our algorithm for tensor field visualization may be summarized as following:

1. select the tensor field to be visualized (domain \mathbb{T}) and the number of particles (N_p);
2. compute $\mathcal{Y}_{\mathbf{t}}$ (Eq. 21) for each tensor;
3. for each tensor $\mathbf{t} \in \mathbb{T}$, sort and rank it in the priority list;
4. select an available particle and insert it into the visualization space using the priority list;
5. perform the particle advection loop;
6. verify what particles must be killed using the stop criterion;
7. if the viewer position or orientation changes too much, perform step 3.
8. if the number of particles in visualization space is smaller than N_p , go to step 4, otherwise go to step 5;

5 Results

In section, we present shots of different types of tensor fields. The particles were represented by a pointer glyph (otherwise specified) and the color gradient adopted flows from blue (minimum) to red (maximum) for a given \mathcal{Y} (Fig. 4).

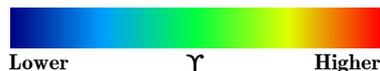


Fig. 4. Color palette for the \mathcal{Y} values.

We have inserted into a 38x39x40 grid three spherical charges, located at (0,0,0), (38,39,40) and (38,0,40). For all voxels in the grid we use a formulation to ponder the influence of each charge in that space region and then compute a local tensor. The Figure 5(a) shows the obtained result using discrete glyphs and in Figure 5(b) we draw a few tensorlines.

An important tensor feature is the anisotropy A_3 (Eq. 12). Thus, if the user want to seek for regions of high anisotropy, the \mathcal{Y} function may be adjusted. It varies from positive to negative as the tensors changes it form from linear to planar.

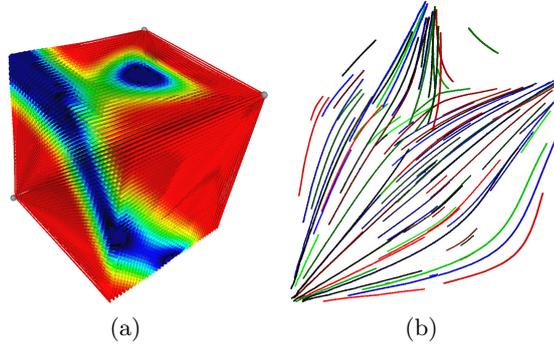


Fig. 5. A tensor field with three charges: (a) represented by superquadric glyphs and (b) by a few tensorlines.

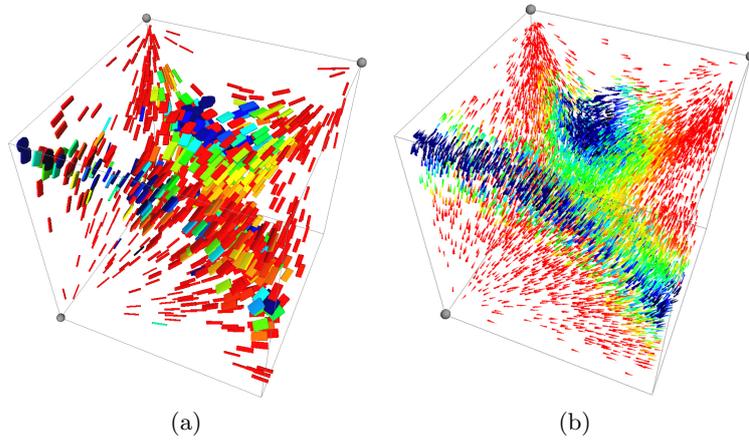


Fig. 6. Three charges field represented by our method: (a) particles assuming superquadric glyph shapes (particles near the charges are more stretched) and (b) pointer glyphs smoothly flowing through the domain.

We have defined $\mathcal{Y} = -A_3 + FA$ and the results are shown in Figure 6. The anisotropy in this field can be seen using the proposed method (Fig. 6). We have used superquadric glyphs (Fig. 6(a)) and pointer glyphs (Fig. 6(b)) as particles to understand the field properties. In Figure 6(a) there are less particles than Figure 6(b). Note that the superquadric glyph particles, in Figure 6(a), are more stretched in regions near to the charges showing a linear behavior. This regions presents a high anisotropy. Using the \mathcal{Y} function in Figure 6(b) one may note: a) a large number of particles are inserted into that region, b) the particles

flows smoothly making the field variation more understandable. We consider the pointer glyph the best way to represent the particles because it makes the visualization cleaner and allows a complete view inside the volume.

The next example is a helical tensor field (Fig. 7) with visualization depending on the observer. The tensors in this field suffer a torsion process along the z -axis. Using k_1 and having the z -axis orthogonal to the observer (Fig. 7(a)), one may see that the tensors in the internal regions tend to have high priority values (reddish colors) as they are orthogonal to the viewer. Using k_1 with the z -axis aligned with the observer (Fig. 7(b)), the now bluish sites (low priority values) represent tensors highly parallel to the viewer, in regard to the new camera orientation. The proposed method is highly efficient and suitable to extract volumetric information from tensor fields.

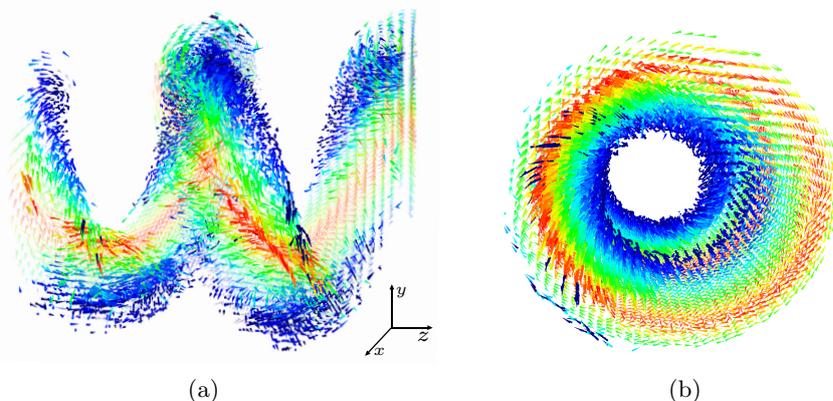


Fig. 7. Helical field: color palette given by k_1 in two different views.

Diffusion tensor magnetic resonance imaging (DT-MRI) is generally used to detect fibrous structures of biological tissues. In this work we have used a diffusion tensor field of a brain available at [20] to test our method. The results are shown in Figures 8 and 9.

The branching and crossing of brain's white matter tracts generates local tensors with high planar anisotropy [5]. To find these brain regions, we adjust the priority and the colorization using $\mathcal{Y} = \mu_2 + A_3 - FA$ (Fig. 8). So, we are searching for tensors with higher variance, amplitude and anisotropy - we are penalizing the isotropic regions with $-FA$. In the central regions of the brain, which is composed by a larger number of fiber and tissues, we can see a concentration of the required information, as expected.

The influence of viewer-dependent terms can be also observed in the brain field (Fig. 9). When we are searching for tensors with eigenvectors orthogonal to the observer, the k_1 and k_2 view-dependent terms could be used. The k_1

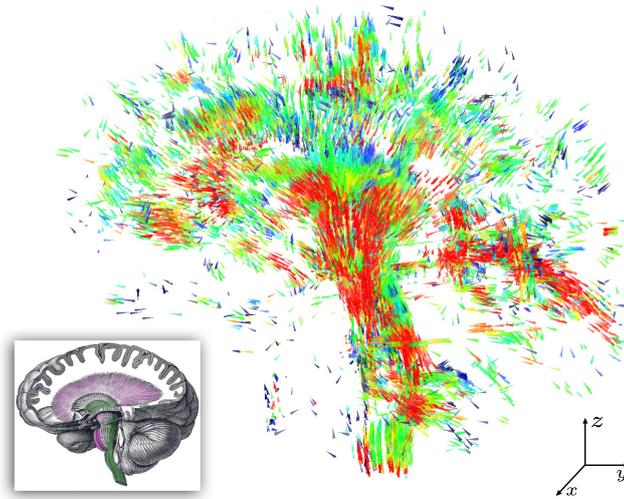


Fig. 8. Brain fiber visualization. Lower left corner: brain image from [19].

coefficient (Fig. 9(a)) highlights tensors which has main eigenvectors orthogonal to the observer. In a opposite fashion, the k_2 coefficient is emphasizing tensors that presents main direction non-orthogonal to the observer, for the same point of view (Fig. 9(b)).

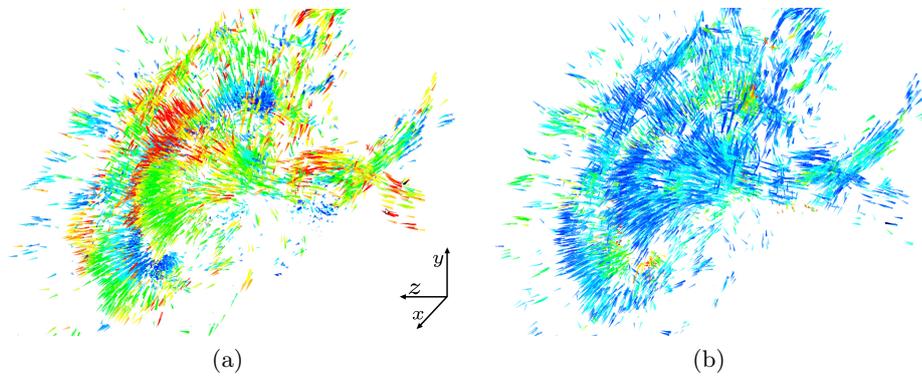


Fig. 9. Influence of the viewer-dependent terms: (a) highlighting tensors orthogonal to e_1 and (b) orthogonal to e_2 .

6 Conclusions

In this paper we presented a tensor field visualization method based upon particle tracing using viewer-dependent terms. We proposed a priority list which defines where particles should be born in the tensor field domain. This is done aiming to highlight regions of interest. We also present a set of observer-dependent coefficients that contribute to the final visualization, generating suitable results. In order to cover a wide range of different tensor fields, we developed a \mathcal{Y} scalar which can be adjusted to the user needs. The \mathcal{Y} quantifies the importance of a tensor in the visualization process for a given set of parameters (Eq. 21). Thus, a previous knowledge of the field is required to achieve a better visual interpretation.

We provide results using three different tensor fields. In each field, the anisotropy analysis showed correctly collinear and coplanar structures formed by the tensors throughout the domain. The view-dependent attributes contributed to the visualization process, highlighting orthogonality and proximity of tensors in relation to the observer (Fig. 7 and 9).

A flicker problem occurs when a new created particle reaches isotropic regions and is instantly destroyed by the removal criterion. This effect can be avoided by filtering the noise present in the tensor field and smoothing the transition between isotropic and anisotropic regions.

Acknowledgment

The authors thank to FAPEMIG (*Fundação de Amparo à Pesquisa do Estado de Minas Gerais*), CAPES (*Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior*) and UFJF for funding this research.

References

1. 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
2. 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)
3. 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 international conference on Information and knowledge management, New York, NY, USA, ACM (1999) 17–20
4. Westin, C.F., Peled, S., Gudbjartsson, H., Kikinis, R., Jolesz, F.A.: Geometrical diffusion measures for MRI from tensor basis analysis. In: ISMRM '97, Vancouver Canada (April 1997) 1742
5. Kindlmann, G.: Superquadric tensor glyphs. In: Proceedings of IEEE TVCG/EG Symposium on Visualization 2004. (May 2004) 147–154

6. 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
7. Dickinson, R.R.: A unified approach to the design of visualization software for the analysis of field problems. In: Three-Dimensional Visualization and Display Technologies, SPIE Proceedings, Vol. 1083. (January 1989)
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. Zheng, X., Pang, A.: Hyperlic. In: VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03), Washington, DC, USA, IEEE Computer Society (2003) 33
11. Cabral, B., Leedom, L.C.: Imaging vector fields using line integral convolution. In: SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM (1993) 263–270
12. Krüger, J., Kipfer, P., Kondratieva, P., Westermann, R.: A particle system for interactive visualization of 3D flows. *IEEE Transactions on Visualization and Computer Graphics* **11**(6) (11 2005) 744–756
13. 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.
14. Kindlmann, G.: Visualization and Analysis of Diffusion Tensor Fields. PhD thesis (September 2004)
15. Bahn, M.: Invariant and Orthonormal Scalar Measures Derived from Magnetic Resonance Diffusion Tensor Imaging. *Journal of Magnetic Resonance* **141**(1) (November 1999) 68–77
16. Masutani, Y., Aoki, S., Abe, O., Hayashi, N., Otomo, K.: MR diffusion tensor imaging: recent advance and new techniques for diffusion tensor visualization. *European Journal of Radiology* **46** (2003) 53–66
17. Basser, P.J., Pierpaoli, C.: Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor mri. *Journal of Magnetic Resonance, Series B* **111**(3) (1996) 209 – 219
18. MAHMOUD, H.M.: *Sorting: a distribution theory*. John Wiley & Sons (2000)
19. Kondratieva, P.: *Real-Time Approaches for Model-Based Reconstruction and Visualization of Flow Fields*. PhD thesis, Institut für Informatik Technische Universität München (2008)
20. Kindlmann, G.: Diffusion tensor mri datasets. <http://www.sci.utah.edu/~gk/DTI-data/>