Tensor field visualization using Eulerian fluid simulation

Marcelo Caniato Renhe *, José Luiz de Souza Filho, Marcelo Bernardes Vieira, and Antonio Oliveira

Universidade Federal de Juiz de Fora, DCC, Brazil
Universidade Federal do Rio de Janeiro, COPPE, Brazil
{marcelo.caniato,jsouza,marcelo.bernardes}@ice.ufjf.br
oliveira@lcg.ufrj.br
http://www.gcg.ufjf.br

Abstract. In symmetric second order tensor fields, the colinearity and coplanarity of the represented structures are properties of major interest. In this paper, we present a method that induces the human perceptual system to extract these structures by using an Eulerian fluid simulation. Differently of previous approaches, our model explores the interaction between the particles to improve the perception of the underlying structures in the tensor field. The main contribution is the introduction of a tensor based advection step and a gradient based external force, suitable for viewing colinear and coplanar structures. Our experimental results show that fluids are suitable to reveal curves, surfaces and their connections.

Keywords: Tensor Field Visualization; Orientation Tensor; Fluid Simulation.

1 Introduction

Among the different tensor field visualization methods, those developed considering a dynamic approach usually present better results. This approach consists in using motion to stimulate the human visual system. Applying some sort of dynamics to the tensor field makes it easier for the observer to identify and analyze specific data of interest in the field. For example, we could define the motion as a function of characteristics of the tensors, in order to enhance structures that they represent.

A viable solution for introducing motion into the field is to use it as a medium for simulating fluid flow. Previous works related to tensor field visualization used advection mechanisms [1–3], which are an essential part of any fluid simulation. Many works in the computer graphics area explored techniques for controlling the fluid direction of flow in order to meet specific application requirements.

* Authors thank to Fundação de Amparo à Pesquisa do Estado de Minas Gerais and CAPES for financial support.

Tensor field visualization means to turn visible some of their properties or somewhat continuous colinear and coplanar structures formed by the tensors. Showing this kind of structures can be challenging due to the huge amount of data. Using particle-tracing methods allows to enhance desired properties by analyzing the paths and arrangement of the particles along the field. Some of these methods are static and does not describe interaction between particles [4, 1].

We propose an interactive and dynamic method for tensor field visualization. It is based on the usage of fluid dynamics to rule the behavior of particles inside tensor fields. An Eulerian method based on Navier-Stokes equations was chosen for that. An external force was proposed to concentrate particles around areas of higher energy. A modification in the the advection is also proposed. This change consists on using properties of the acting tensors to make particles arrange in a way to highlight consecutive colinear structures.

2 Related Work

Most works related to tensor visualization are interested in developing more intuitive forms of visualization. Usually, the problem of analyzing a tensor field is its multivariate information. Methods encountered in literature fall into three categories. The discrete methods often employ some kind of glyph representation to visualize the field. Shaw *et al* [5] proposes using superquadrics to display the tensor data. The idea of superquadrics was later used by Kindlmann [6] to generate a tensor glyph that encodes the geometric shape of the tensor. A tensor shape can be determined based on coefficients introduced by Westin *et al* [7]. The main advantage of using these measures in a visualization is that they avoid ambiguities in the identification of the tensor shape.

The continuous methods, on the other hand, try to establish a continuous path along the tensor field. The hyperstreamlines method [8, **?**], followed by the tensorlines method [1], are examples of them. They define paths that can be used to achieve a smooth visualization of the field in a global fashion. This kind of visualization is very useful for tracking linear features in the field, which are highly present in DT-MRI, for example. Vilanova *et al* [9] presents a thorough explanation of the methods mentioned above and other discrete and continuous methods. More recently, Mittmann *et al* [10] devised a real-time interactive fiber tracking method that can determine new paths automatically based on user feedback, while Crippa *et al* [11] proposed an interpolation method to avoid low-anisotropy regions during fiber tracking.

Finally, we have the dynamic visualization methods. These methods focus on exploring visual cues to stimulate the human perceptual system, making it easier for the observer to analyze the tensor field. Kondratieva *et al* [12] proposes the advection of particles through the directions of a generated tensor field. The goal of the advection is to induce motion in the field to enhance user perception. Leonel *et al* [2] used this idea to create an adaptive method, in which the position and orientation of the observer are taken into account in an

interactive visualization of the field. Souza Filho *et al* [3] extended Leonel's work by using multiresolution data to improve the method, while reducing the number of required parameters and artifacts that were present in previous works.

This work proposes a dynamic visualization of tensor fields using fluid simulation. We use a simple and stable fluid simulator just like the one found in Stam's work [13]. Many other recent methods for fluid simulation exist, but they are focused on achieving physically accurate simulations, which is not our objective. We seek to use fluids as an alternative to introduce motion in the field. To direct the flow along a specific path, previous works used gradient fields to define an external force. In [14], this force is applied to the fluid in a morphing context, making the densities travel from an origin point to a predefined target. Similarly, we define a gradient field that is used in the calculation of the external forces. We also propose a modification in the advection mechanism, where the motion is distorted by the local tensors.

3 Fundamentals

3.1 Tensors

In this work, we are interested in visualizing tensor fields composed by second-order tensors. This type of tensor is represented by a 3x3 matrix, representing a linear transformation between vector spaces. Westin [7] introduces a special case of tensor, which is non-negative, symmetric and has rank 2. This tensor, used for estimating orientations in a field is called a local orientation tensor. It can be defined as:

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

where $e_1, e_2, ..., e_m$ is a base of $\mathbb{R}^n$ and λ_i is the eigenvalue associated to the eigenvector e_i.

In $\mathbb{R}^3$, the orientation tensor can be decomposed into a sum of the contributions of its linear, planar and spherical features. The equation, then, becomes:

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

The above decomposition has an important geometrical meaning. Analyzing the relation between the three eigenvalues, we can determine the approximate shape of the tensor. Basically, we have three possibilities:

- $\lambda_1 \gg \lambda_2 \approx \lambda_3$ indicates an approximately linear tensor
- $\lambda_1 \approx \lambda_2 \gg \lambda_3$ indicates an approximately planar tensor
- $\lambda_1 \approx \lambda_2 \approx \lambda_3$ indicates an approximately isotropic tensor

Usually we are interested in the main direction of the tensor. A tensor, when applied to a vector, distorts the space in accordance with its inherent shape.

Isotropic tensors have no main orientation, reason why we try to avoid them in the visualization process, as we will discuss in Section 4.

In order to identify the shape of the tensor, Westin defines three coefficients, each related to one of the types of tensors mentioned above. The so-called coefficients of anisotropy are defined as:

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3},$$
$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3},$$
$$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3},$$

where each coefficient lies in the range $[0, 1]$. Besides, they have the following property: $c_l + c_p + c_s = 1$.

3.2 Fluid Simulation

Fluids have been extensively researched in the Computer Graphics field for the last decade. Simulations involving fluids evolved to the point of being able to reproduce very complex phenomena. Instead, even a simpler fluid simulation requires the use of many techniques from the Computational Fluid Dynamics field. All these techniques must be put together in order to solve the Navier-Stokes equations, which are responsible for describing the motion of fluids. The Navier-Stokes equations, in their vector form, are presented below:

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$
$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \tag{2}$$

where $\mathbf{u}$ represents the velocity vector, p is the pressure, ν is the fluid viscosity and f is the sum of external forces. The first equation is called the continuity equation, and accounts for the conservation of mass for incompressible flow. Equation 2 imposes momentum conservation [15].

Each term of the Navier-Stokes equations has its own meaning. The first term represents the advection part. The advection is the process of transporting properties through the fluid. These properties include the velocity itself. Other examples of properties or substances that can be advected through the velocity field are the density, the temperature or any other value which might be interesting for the simulation. The second term accounts for the pressure. It plays an important role in the projection step of the simulation. This step is responsible for enforcing Equation 1, i.e., the divergent of the velocity must always be zero, ensuring conservation of mass. The next term accounts for the viscosity, which smooths out the velocities, limiting the fluid motion. That's why numerical dissipation in the fluid simulation is also called numerical viscosity. It makes velocities dampen faster. We will discuss numerical issues in Section 4.3. Finally, the last term represents any external forces acting on the fluid. This term is

often used as a means of controlling the motion direction of the fluid [14]. We could, for example, add a constant wind-like force in a certain portion of the fluid to shape its path.

Usually, in standard fluid simulations, we deal with non-viscous fluids. In these cases, the viscosity term can be suppressed. The resulting equation, suitable for inviscid fluids, is called the Euler equation. While inviscid means that density is constant throughout the fluid, we can still advect density, like any other property, to use it for visualization.

Two approaches are commonly employed with respect to the internal representation of the fluid. The Lagrangian approach focuses on the fluid itself, discretizing it in a number of particles. Thus, the properties of the fluid are represented in each particle. On the other side, the Eulerian approach, which we use in this work, takes the space surrounding the fluid as the reference, discretizing it into a grid. Each cell of the grid stores the values of the fluid properties at that point in the space.

In an Eulerian simulation, in order to solve the Navier-Stokes equations and actually simulate the fluid, we employ a technique called fractional steps, or operator splitting. It consists in breaking the target equation into parts, and solve each part with an adequate numerical method. The advantage is that we simplify the solution of the equation by allowing the conjugation of different methods for the solution of its terms. This means we can choose, for a particular term, the best suitable method to solve it. This technique can be used whenever the equation involves a sum of different terms that present different contributions to the final result. The solution of each term is then takes as input to the next term of the equation.

The first step in the simulation process consists in the application of the external forces. This stage produces a new velocity field, which is then taken as input for the advection step. In this step, we employ an unconditionally stable method presented in [13], which guarantees that no output velocity is bigger than the maximum velocity in the input field. This feature prevents the simulation from blowing off. The process is simple: we trace the advected property back in time to find what was its value in its previous position in the grid. Then, we use that value interpolated with its neighbors to update the current value of the property.

After the advection, we pass on to the last step, called projection. It takes the advected velocity field and makes it divergence free by subtracting the pressure from it. This is accomplished by observing a mathematical statement, known as the Helmholtz-Hodge Decomposition, which says that any vector field can be decomposed into a sum between a divergence free vector field and a scalar field.

4 Proposed Method

In this section, we present our method, which conjugates tensor field visualization with fluid simulation. The tensor field information is used to govern the fluid motion, allowing the observer to detect regions of interest in the field.

4.1 Fluid Representation

As stated in Section 3.2, we implemented an Eulerian fluid representation. We define a grid with a number of cells equal to the dimension of the tensor field. For simplicity, each fluid property is discretized at the center of each cell, as in [13]. Some works in the literature use a staggered arrangement for the velocities [16], in which they are discretized at the cell faces instead of its center. Trottenberg *et al* [17] analyzes the advantages of using the staggered grid, which are mainly related to avoidance of possible numerical instabilities that may arise during the simulation, creating a checkerboard pattern in the visualization (hence the name checkerboard instability). We were able to obtain, though, satisfactory visualizations even without using a staggered arrangement. However, it is important for non-staggered grids, when using central differences, to define pressure values at all boundaries of the grid to achieve second-order accuracy.

4.2 Extenal Force from the Tensor Field

In order to control the direction of the flow, we apply an external force based on tensor information. A more straightforward definition for this force would be to use the main eigenvector from the tensor matrix to set the direction of the force vector. Although this definition could produce rather good results, we decided to use a gradient field based on each tensor weight. The weight of a tensor is defined as:

$$w = \frac{1}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} \tag{3}$$

We then set the external force to be the product between the gradient of the tensor weight at a specified and a parameter used to scale the force modulus. We discretize the gradient as follows:

$$\nabla w = \frac{1}{2} \cdot [w_{i+1,j,k} - w_{i-1,j,k} \quad w_{i,j+1,k} - w_{i,j-1,k} \quad w_{i,j,k+1} - w_{i,j,k-1}]^T \tag{4}$$

where $(i, j, k) \in \mathbb{N}^3$ are the coordinates of the neighbor tensors in the grid. The force from the gradient is applied to all cells in the grid. Since the tensor field is static, the applied force produces a velocity field which quickly converges to a stable configuration through the advection process. This makes the velocity field completely still, which is not interesting, since we want a dynamic visualization. Thus, we employed a different scheme for the application of the force. At each iteration, we apply the force only for a limited amount of tensors. Considering a 3-dimensional grid, these tensors are located in a single plane. So, in each iteration, forces are added to a specific plane in the grid. We sweep the all the planes through the iterations until we get to the first plane again. We then restart the cycle. With this mechanism, we are able to inject energy at varying portions of the grid, generating the illusion of motion in the velocity field. This immediatly allows us to visualize the tensor field simply by updating the velocity field over time, discarding the advection of densities, which is commonly used to visualize the fluid.

4.3 Advection with Tensors

The advection is the transport of properties along the flow path, as mentioned in Section 3.2. We solve the advection term in the Navier-Stokes equations through a semi-Lagrangian scheme. To compute the new velocity at a point x, we imagine a particle at this point and trace its path backwards in time, finding the position x_0 where it was located in the previous iteration. We then interpolate the value at x_0 with the values in its adjacent cells. This new value is used to update the velocity at the point x. Notice that, as we interpolate through already existent velocity values, the advection itself cannot cause the simulation to blow up.

In our case, however, the fluid path is determined by the tensor field. The tensors distort the space in which the fluid is enclosed. So, if in a previous iteration the supposed particle was at x_0, we must take into account the contribution of the tensors in the cells used for interpolation. In other words, we must find the direction determined by these tensors that led the particle to its current position. Thus, at each cell, we apply the tensor located at that site to its associated velocity vector, yielding a new velocity in a distorted space. This distorted velocity is then used to update the current one. This process is depicted in Figure 1.

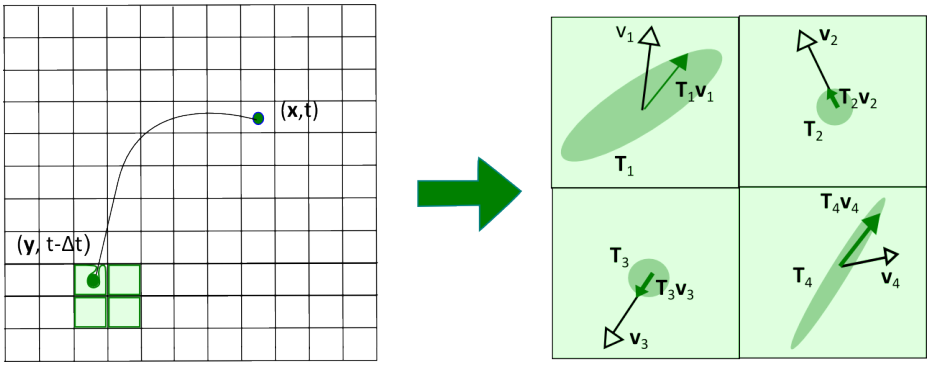


Fig. 1. Advection in the tensor field. The velocity of the cell at position $\mathbf{y}$ in time $t - \Delta t$ is used to define the position $\mathbf{x}$ in time t. The nearest tensors of $\mathbf{y}$ are applied on their respective velocities. The distorted velocities are summed to define the new velocity at $\mathbf{x}$.

Generalizing the example in Figure 1 for threedimensional grids, suppose $\mathbf{y} \in \mathbb{R}^3$ in time $t - \Delta t$ is the point computed from the grid point $\mathbf{x} \in \mathbb{N}^3$ at time t using its current velocity backwards. Let $\{\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3, \mathbf{T}_4, \mathbf{T}_5, \mathbf{T}_6, \mathbf{T}_7, \mathbf{T}_8\}$ be the tensors from the nearest neighbors of the point $\mathbf{y}$ in the grid, forming a 2x2x2 block of cells. We can then define the new velocity $\mathbf{v}_x$ at $\mathbf{x}$ as

$$\mathbf{v}_x = \sum_{i=1}^{8} \alpha_i \mathbf{T}_i \cdot \mathbf{v}_i, \tag{5}$$

where α_i are the weighting factors of a trilinear interpolation.

The semi-Lagrangian method for advection is highly subject to numerical dissipation. Because we are linearly interpolating properties every iteration, the numerical error tends to accumulate. The excessive averaging required by the linear interpolation acts as a low-pass filter, smoothing out values over time [18]. The result is that velocities dampen quite fast, and interesting visual effects, like swirling movements in the fluid, can be lost during the simulation. Possible overcomings include the utilization of sharper interpolation methods [16] and techniques like the vorticity confinement method, first presented by [19] and also used by [16]. As for the first solution, care must be taken to avoid overshooting, because it does not provide the same stability as a simple linear interpolation.

For our purpose of tensor field visualization, however, the usual dissipation that arises from the semi-Lagrangian advection is not exactly an issue. We are more interested in adding some level of fluid dynamics to the field than to accurately and realistically simulate the fluid. The fluid is just an interesting medium for the field visualization. The real problem is that the application of tensors to the velocities deeply aggravates the already present dissipation.

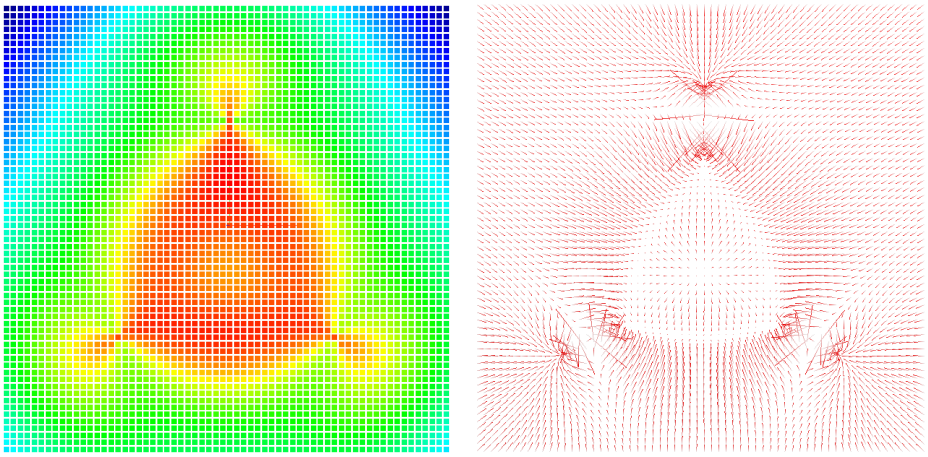
To work around this tensor dissipation issue, we adopted a simple solution. We defined a boost parameter to apply to each tensor in the field before it is used in the velocity calculation. Thus, we counteract the numerical dissipation with a simple scale in the tensor. If correctly adjusted, it is possible to create a balance between the boosting and the dissipation, producing controlled fluid simulations, which allow nice visualizations as the one shown in Figure 3 in the next section. We also can control which tensors receive the boost through checking of its weight or some of its anisotropy coefficients mentioned in Section 3.1. This is useful, for example, if we want to visualize just linear tensors. We then set the boost based on the linear anisotropy coefficient, letting the dissipation naturally smooth out the velocities in the cells with isotropic or planar tensors.

5 Results

In this section, we present some results from our visualization. All experiments were conducted in a computer with an Intel i3 processor running at 2.1GHz, 4GB RAM memory and an nVidia GeForce 540M graphics adapter. We generated a simple tensor field in order to demonstrate our method behavior. In a regular 2D grid of 64×64 elements, with integer coordinates $(i, j) \in [0, 63] \times [0, 63]$, we defined three main points $\mathbf{q}_1 = (16, 16)$, $\mathbf{q}_2 = (48, 16)$ and $\mathbf{q}_3 = (32, 48)$. For all points $\mathbf{p}_{ij} = (i, j)$ in the grid, a tensor is computed as

$$\mathbf{T}_{ij} = \sum_{k=1}^{3} \frac{(\mathbf{q}_k - \mathbf{p}_{ij}) \cdot (\mathbf{q}_k - \mathbf{p}_{ij})^T}{||(\mathbf{q}_k - \mathbf{p}_{ij})||},$$

where $\mathbf{q}_k \neq \mathbf{p}_{ij}$. The result is a tensor $\mathbf{T}_{ij}$ that captures the uncertainty of the direction between the grid point $\mathbf{p}_{ij}$ and all $\mathbf{q}_k$ points, weighted by their Euclidean distances. Figure 2a shows a representation of the tensor norms of



(a) (b)

Fig. 2. (a) Norms of a 64×64 2D tensor field represented by a colored dot going from blue (lowest) to red (highest). (b) The corresponding gradient field of the tensor norms.

this 2D field, varying from lowest (blue) to highest (red). In Figure 2b, the gradient of the tensor norms is shown.

We employ a fluid simulation for the visualization as explained in Section 4. Observing the Figure 2b, we can see that the gradient norms are greater near the main points, since we use the tensor weight at each site to calculate them. The central region has the weakest gradients, which is in accordance with the fact that the tensors in that area are mainly isotropic. In our simulation, it means that, if we apply a force in that area, any velocity generated by this force will soon be reduced to zero.

We can see a screenshot of the aforementioned simulation in Figure 3. The velocity vectors are represented as two-colored lines, with the red edge indicating the direction. We interactively applied a series of external forces to the fluid in different points of the grid. These forces produced motion in the direction of the force, but the flow direction soon converged to the charges. As one can observe, the tensor-guided simulation produces continuous paths between the charges, leading the fluid from one charge to another. The velocities originated from the borders are directed towards the center of the field. In this simulation, we applied a tensor boost equals to 1.2, which avoided rapid velocity dissipation and created a fluid-like motion of the velocity field, with the velocities continuously moving around the charges. The velocities seen at the very center of the field in Figure 3 are not created there, because, as previously said, that area is full of isotropic tensors. Those velocities come from the linearly anisotropic tensors present in the diagonal line connecting the left (or the right) and the top charges.

This field represents a simple example where it is more clear to visualize the wanted behavior. There are two more challenging examples that are a 3D helical field and a DT-MRI brain. In these examples, we did not use interactive force

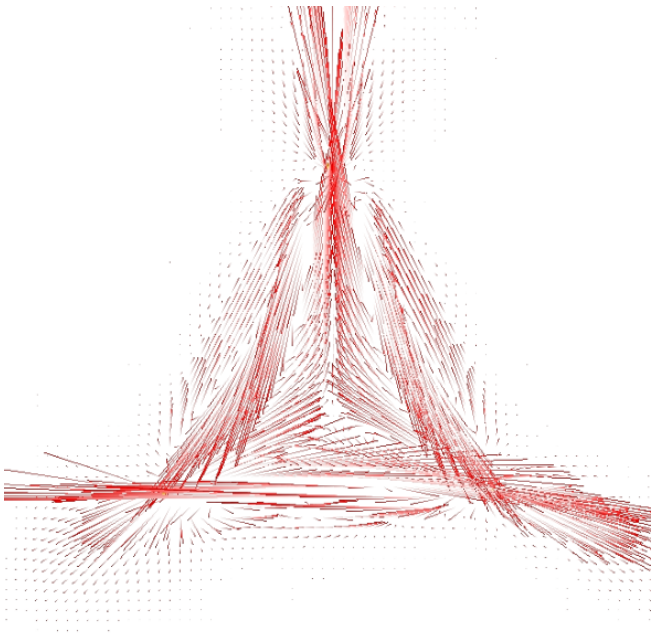
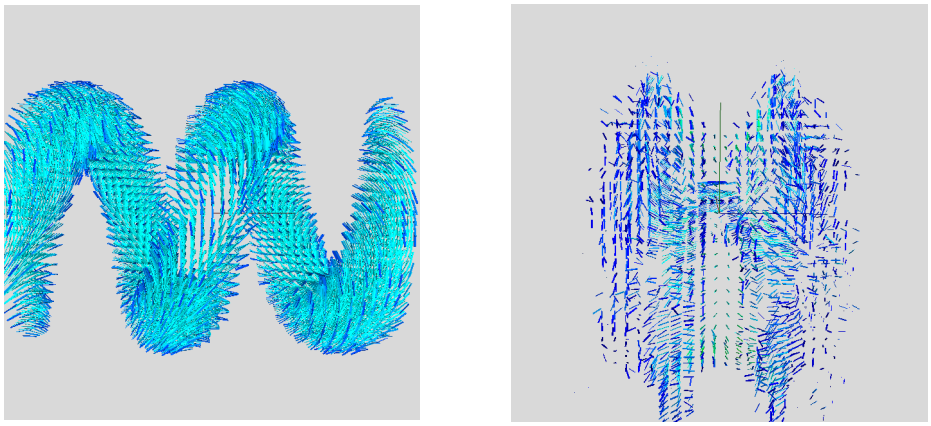


Fig. 3. Screenshot of the 3-point field visualization. Time step = 0.1; tensor boost = 1.2.

placement. The details on how forces were applied are explained in the following discussion.

The 3D helical field is an artificial tensor field, in which we have mainly linear tensors, composing the helix, surrounded by isotropic tensors. In order to better illustrate the tensor field, Figure 4 only shows the tensors below the threshold of 0.7 for the spherical coefficient of anisotropy presented in Section 3.1. Figure 5 shows screenshots of our simulation in different iterations for the helical field. As in the 2D example, we used the velocity field itself to visualize the tensor field. Since tensors do not inherently point in a particular direction, we used parallelepipeds as glyphs to represent the velocity vectors. Besides, to put the fluid into motion we need to apply an external force in the beginning of the simulation. This force starts the velocity advection process. If we do not apply additional forces, the velocities tend to dissipate at some point during the simulation. If we keep the tensor boost on, they will at most reach equilibrium, presenting no movement at all. We cannot arbitrarily scale the boost, because the simulation can become unstable. On the other hand, if we apply a constant force in all the grid in every iteration, the velocities will be directed along the linear tensors path, as expected, but they will still not move. Thus, we employed a scheme in which we apply forces cyclically along varying planes of the 3D grid. In the helical field example, we applied forces, based on the tensor weight gradients, to every cell in the plane $x = k$, with $k = 0..N - 1$. Letting the



(a) (b)

Fig. 4. Snapshots of the tensor fields: (a) $40 \times 39 \times 38$ helical tensor field. (b) $40 \times 38 \times 37$ DT-MRI brain.

tensor boost off, the velocities in the cells not subject to the gradient force tend to smooth out, while the affected cells create new velocities, which are advected through the tensor field. With this method, we were able to induce motion in the field, generating the visualization shown in Figure 5. The colors of the glyphs indicate the velocity norm, which means that the red velocities represent the places where the external force is being applied at the current iteration. This simulation did not run in real-time due to the grid resolution needed for this field. A downscaled version of the field may be used to work around this issue, although we lose some information, producing slightly poorer simulations.

Finally, we simulated the DT-MRI brain tensor field. For this simulation, we used a size-reduced version of the field. Since the brain field is quite large, we were also not able to achieve a real-time visualization, even with downscaling. Plus, the brain field is highly noisy. So, we used a filtered version of the original field. Figure 4 shows this field. Tensors with spherical anisotropy coefficient higher than 0.7 were omitted, as in the previous example. Again, as in the helical example, we applied forces by sweeping the grid planes $x = k$, with $k = 0..N-1$. Also, we let the tensor boost in each cell be set by a gaussian function with respect to the distance between the cell plane and the plane $x = k$. Figure 6 shows 9 subsequent frames from a simulation, with a time step of 0.001. The size of each tensor glyph is defined as a function of its weight w, as defined in Section 4.2, and the velocity norm at the cell. The brain central region, which has many colinear fibers, contains tensors with higher weight, resulting in the red glyphs seen in the figure. The difference between the glyphs can be better seen in Figure 7, which zooms in the mentioned region in the brain. Also, Figure 7 shows alternative views of the brain, like its side and top views, for instance.

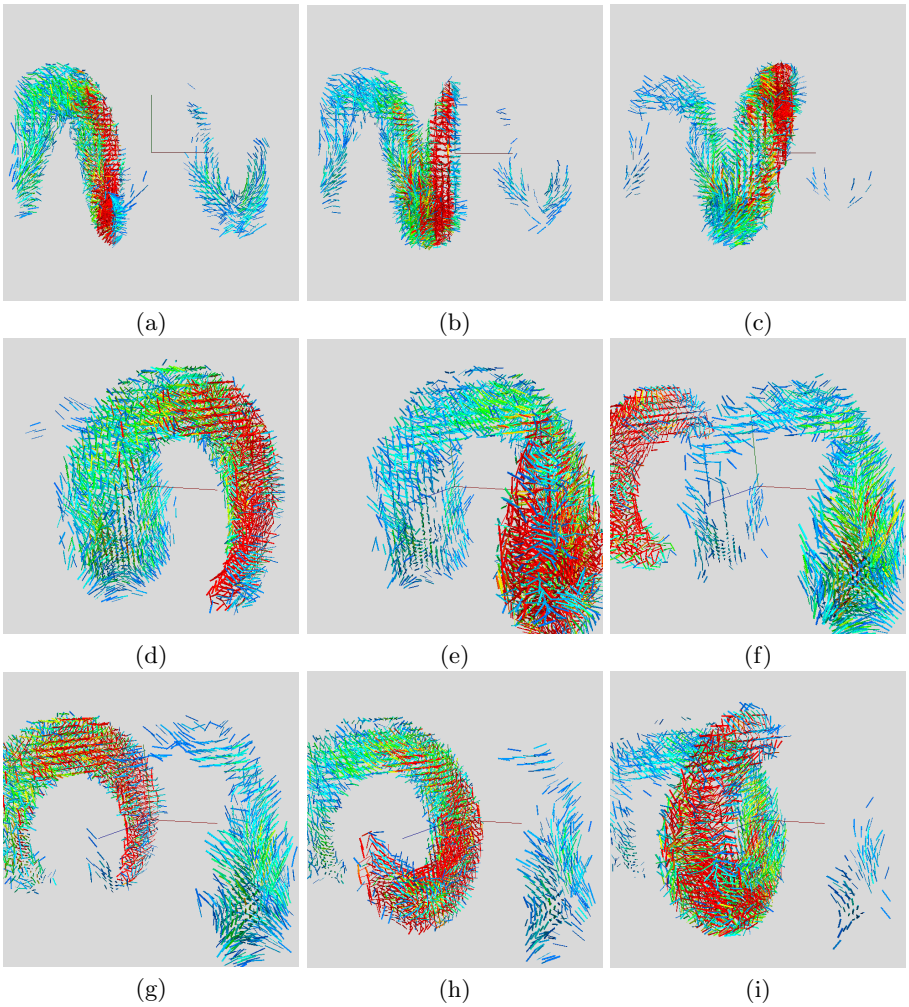


Fig. 5. Helical field visualization in different times and different points of view. Screenshots were taken at each 3 iterations. Time step = 0.001; tensor boost = 1.0.

6 Conclusion

We presented a new visualization method based upon fluid simulation. We use the fluid as a means for visualizing tensor fields. The velocity field generated by the simulation is used to define the paths along linear tensors in the field. We apply external forces to the field in such a way that the velocities produce a continuous movement, enhancing the visualization process. Our method works both for 2D and 3D fields. We have shown our results for two artificial fields and for a DT-MRI brain field.

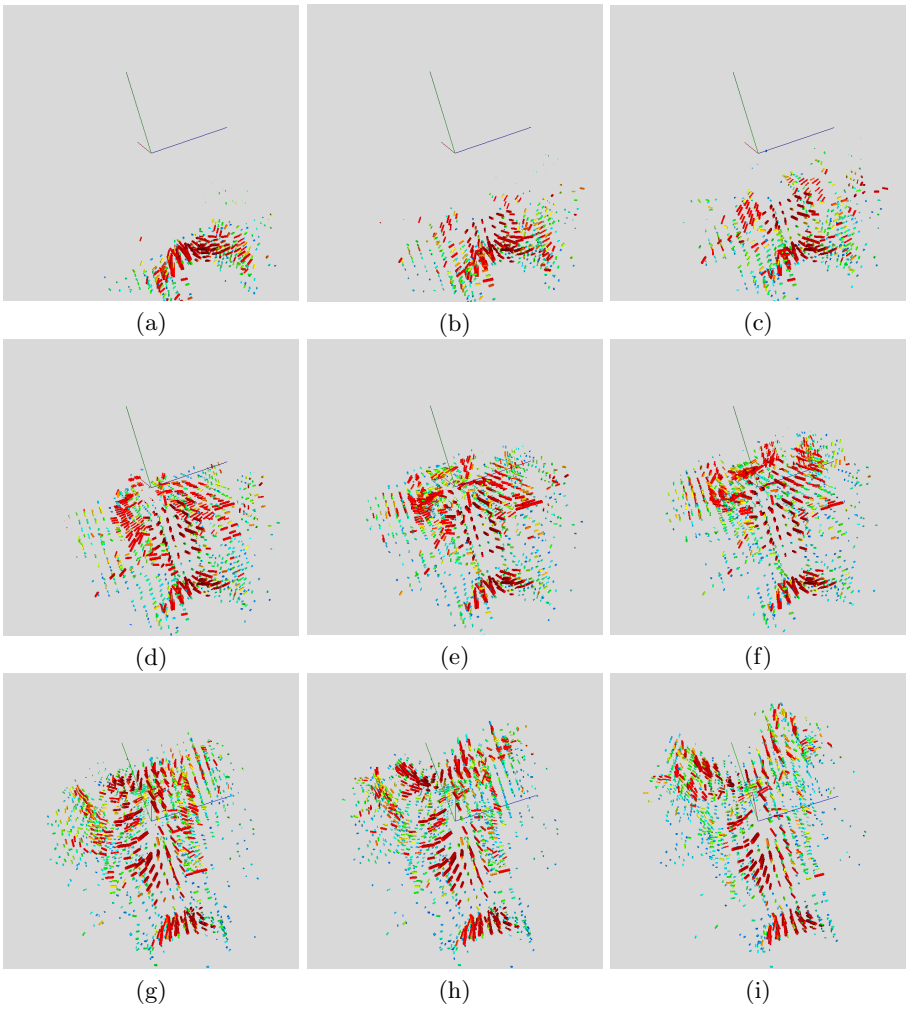


Fig. 6. Visualization of the DT-MRI brain field. Screenshots show 9 sequential frames, taken at subsequent iterations. Time step = 0.001.

As mentioned in Section 5, we were not able to achieve real-time simulations for high resolution grids. Even with downscaling, the brain field visualization had to be generated offline. In an Eulerian simulation, fluid properties are calculated for every grid cell. The original brain field has dimensions 80x95x74. Further research must be done in order to work around this limitation and achieve real-time visualization. Parallelization may be a feasible solution, since it is not difficult to parallelize the stages of the simulation. Another possible experiment includes the utilization of particles through a hybrid fluid simulation method.

We used velocities to visualize the field, which led us to find a way to induce motion due to the reasons presented in Section 5. The use of fluid densities or

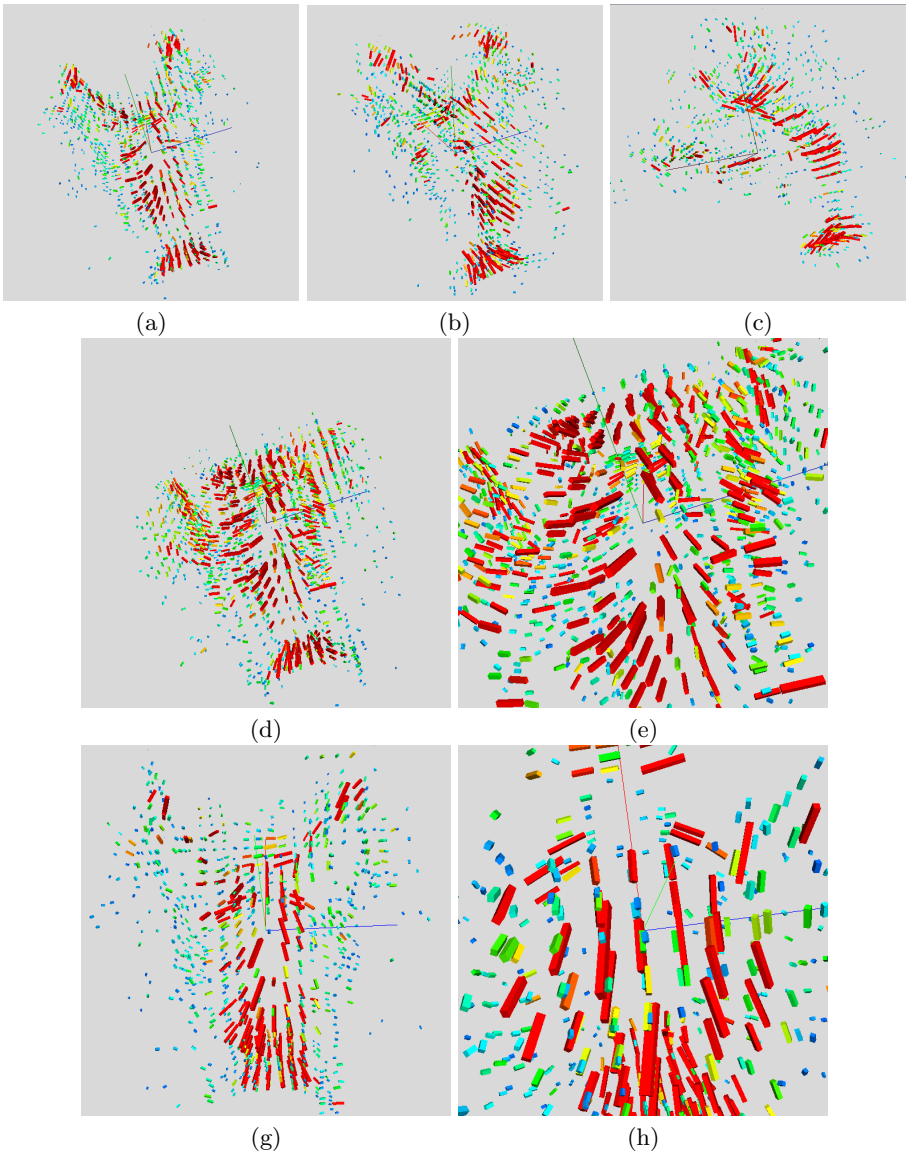


Fig. 7. Different views of the DT-MRI brain visualization depicted in Figure 6. Figure (c) shows a side view of the brain. Figure (g) shows it as seen from the top. Figures (e) and (h) show scaled views of the frames shown in Figures (d) and (g), respectively, to provide a detailed view of the tensor glyphs. Time step

other property like temperature would be a better solution for the movement issue. However, we still need to evaluate the best way to represent the density. More importantly, we would need to determine the best places to insert fluid

density into the grid in such a way that the visualization produced can actually enhance the observer experience. Also, it would be interesting to explore the possibilities created by tweaking other fluid properties, like viscosity for instance.

References

1. 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
2. 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
3. de Souza Filho, J.L.R., Renhe, M.C., Vieira, M.B., de Almeida Leonel, G.: A viewer-dependent tensor field visualization using multiresolution and particle tracing. In: Proceedings of the 12th international conference on Computational Science and Its Applications - Volume Part II. ICCSA'12, Berlin, Heidelberg, Springer-Verlag (2012) 712–727
4. 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
5. 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
6. Kindlmann, G.: Superquadric tensor glyphs. In: Proceedings of IEEE TVCG/EG Symposium on Visualization 2004. (May 2004) 147–154
7. Westin, C., Peled, S., Gudbjartsson, H., Kikinis, R., Jolesz, F.: Geometrical diffusion measures for mri from tensor basis analysis. In: Proceedings of ISMRM. Volume 97. (1997) 1742
8. 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
9. 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
10. 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
11. Crippa, A., Jalba, A., Roerdink, J.: Enhanced dti tracking with adaptive tensor interpolation. Visualization in Medicine and Life Sciences II (2012) 175–192
12. Kondratieva, P., Kruger, J., Westermann, R.: The application of gpu particle tracing to diffusion tensor field visualization. In: Visualization, 2005. VIS 05. IEEE, IEEE (2005) 73–78
13. Stam, J.: Stable fluids. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co. (1999) 121–128

14. Fattal, R., Lischinski, D.: Target-driven smoke animation. In: ACM Transactions on Graphics (TOG). Volume 23., ACM (2004) 441–448
15. Ferziger, J.H., Perić, M.: Computational methods for fluid dynamics. 3 edn. Springer (2002)
16. Fedkiw, R., Stam, J., Jensen, H.: Visual simulation of smoke. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM (2001) 15–22
17. Trottenberg, U., Oosterlee, C.W., Schüller, A.: Multigrid. Academic Press (2001)
18. Bridson, R.: Fluid Simulation For Computer Graphics. Ak Peters Series. A K Peters (2008)
19. Steinhoff, J., Underhill, D.: Modification of the Euler equations for "vorticity confinement": Application to the computation of interacting vortex rings. Physics of Fluids **6**(8) (1994) 2738–2744