

Strategies for Visualizing Tensor Fields in more than two Dimensions

Werner Benger^{1,2} and Hans-Christian Hege¹

¹ Zuse-Institute Berlin, Takustrasse 7, D-14195 Berlin-Dahlem, Germany

² Max-Planck Institute for Gravitational Physics (Albert-Einstein Institute), Am Mühlenberg 1, D-14476 Golm/Potsdam, Germany

Abstract. Tensor field visualization aims at the depiction of the full information contained in the underlying data set or the extraction and display of specific features. Here, we focus on the first task and evaluate various methods with regard to their power of providing an intuitive visual representation. Tensor fields are reviewed in a differential geometric context and we provide a coordinate-free derivation of various mathematical properties of tensor fields. An overview and classification of glyph-based methods is given and their usability for time-dependent tensor fields is discussed. Selected innovative methods are presented in more detail (tensor cones, tensor glow, tensor splats, tensor schlieren).

1 Introduction

We consider tensor fields which are given either analytically or on a discrete mesh. For the last kind of data we assume that we are able to reconstruct the underlying continuous field by some interpolation method in a reasonable way. Depicting tensor fields, major problems are the number of degrees of freedom to be displayed at each point and, for data in more than two dimensions, the view occlusion: graphical objects tend to hide each other. The degrees of freedom have to be mapped to graphical degrees of freedom like color, transparency, reflectivity, texture patterns and shape.

Encoding all degree of freedom of a tensor field into the parameter space of just one of these categories is problematic: Ideally we would like to map all degrees of freedom onto each point in space. Texture patterns and shaped objects require spatial extent, and color plus transparency encompass essential four degrees of freedom. The art of tensor field visualization is to find perceptually effective combinations of graphical parameters and suitable mappings to them. To tackle the occlusion problem, one aims to work with sparsely distributed objects, e.g. lines, and transparency. Tensor field visualization techniques can thus be differentiated in those computing integral lines or surfaces, and those which display the field per vertex the entire volume by drawing tiny objects, so-called “glyphs”. We will discuss the benefits of both approaches, concentrating on the mathematical aspects of integral manifolds in the first part, and focusing on rendering glyphs in the second part of the article.

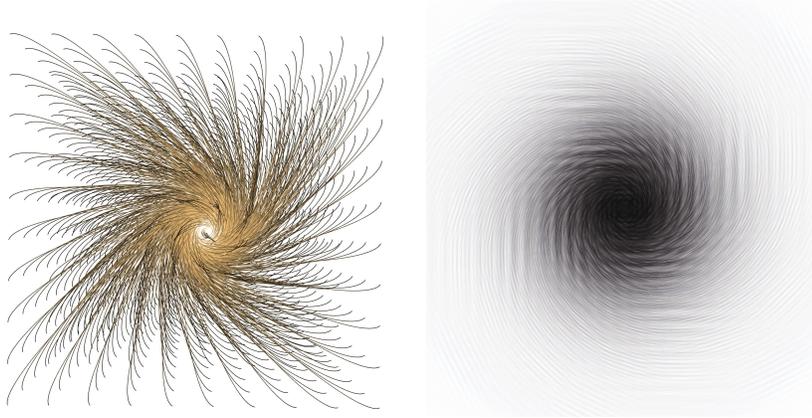


Fig. 1. Maelstrom of spacetime around a rotating black hole, visualized via integral lines (left) and vertex-based glyphs (right).

1.1 Classification of Tensor Fields

At first, we need to determine the symmetry properties of a rank two tensor field: is it symmetric, like the diffusion tensor field in magneto-resonance imaging (DT-MRI), or like the metric tensor field in general relativity; is it antisymmetric or does it contain no symmetries at all, like the Jacobi matrix of a vector field. Any general tensor field can be decomposed into a symmetric and an antisymmetric part, so we can defer the visualization of a generic tensor field into two sub-tasks. In three dimensions, an antisymmetric tensor field of rank two consists of three independent components and is thus equivalent (homeomorph) to a vector field.

An important property of a tensor field that needs to be known before selecting an appropriate visualization method is its definiteness: A multilinear map $g : V \times V \rightarrow \mathbb{R} : (\mathbf{x}, \mathbf{y}) \mapsto g(\mathbf{x}, \mathbf{y})$, with V a vector space, is *positive definite* if $\forall v \in V$ with $v \neq \mathbf{0} : g(v, v) > 0$. If $g(v, v) \geq 0$, then the g is called positive semi-definite. This property is equivalent to requiring the determinant of the tensor field to be positive everywhere.

The number of independent quantities for tensors of rank two is given in the following table:

dim(M)	symmetric	antisymmetric	general
2	3	1	4
3	6	3	9
4	10	6	16
n	$n(n+1)/2$	$n(n-1)/2$	n^2

For positive-definite symmetric three-dimensional tensors of rank two we may compute linear, planar and spherical shape factors [WPG⁺97] from its eigenvalues: $c_l = (\lambda_{max} - \lambda_{med})/tr(g)$, $c_p = 2(\lambda_{med} - \lambda_{min})/tr(g)$ and

$c_s = 3\lambda_{min}/tr(g)$, where $tr(g) = \lambda_{max} + \lambda_{med} + \lambda_{min}$ is the trace of the tensor. The three shape factors obey the relationship $c_l + c_p + c_s = 1$ and can thus be interpreted as barycentric coordinates within a triangle, as discussed later in section 3.1.

2 Visualization via Integral Manifolds

In the following discussion, we provide coordinate-free expressions as far as possible. The respective mathematical objects are not viewed as set of real-valued components, but as objects that allow certain operations like elements of an abstract vector space. This approach is perfectly suitable for an object-oriented implementation that hides the component-wise appearance of a tensor from the user. In particular, the coordinate-free formulation avoids the use of operations that are only valid in specific coordinate systems and thus helps to distinguish features of the tensor field itself from virtual features that are due to the choice of coordinates. Coordinate expressions, as they are required to implement actual numerical computations, are shown at the end of the discussion.

A **curve** q is a continuous map $q : \mathbb{R} \rightarrow M : s \mapsto q(s)$, which provides a point $q(s) \in M$ within a manifold M for a number $s \in \mathbb{R}$. s is called the *curve parameter*. The set of all points of a curve within a manifold is a *line*. A certain line can be described by an infinite number of curves with different parameterizations.

2.1 Integral Lines in Vector Fields

An *integral line* $q \subset M$ on a vector field $v \in \mathcal{T}(M)$ within a spacetime manifold M with starting event $q_0 \in M$ (an “event” is a spatial location together with a specific time) is defined via

$$\dot{q} \equiv \frac{d}{ds}q(s) = v(q(s)) \quad \text{with} \quad q(0) = q_0 \quad . \quad (1)$$

Integral lines are also called *trajectories*, *tangent curves*¹ or *path lines*. They describe the path of a point-like particle in the flow of a vector field. In coordinates, q describes spatial and temporal information; usually only three-dimensional, but possibly time-dependent (non-stationary) vector fields are considered. Then we may use the time coordinate as the curve parameter and equation (1) reduces to three equations

$$\dot{q}^a(s) = v^a(q^t(s), q^1(s), q^2(s), q^3(s)) \quad , \quad q^t(s) = s \quad (2)$$

¹ Usually visualization literature does not distinguish among the terms “lines”, as set of points (one-dimensional subspace) $q \subset M$, and “curves”, i.e. parameterized one-dimensional maps $q(s) : \mathbb{R} \rightarrow M$. What is really meant here for visualization purposes is a *line*, while a *curve* is used for the computation and numerical representation of a line.

whereby $a = 1, 2, 3$ describes spatial coordinates. However, when solving (2) by numerical methods, it is preferable to go back to (1) to get an autonomous system of ordinary differential equations which can be solved using Runge-Kutta methods [DB02].

In the case of a stationary vector field or when investigating a vector field at some instance of time, we may drop the time dependency and by solving

$$\dot{q}^a(s) = v^a(q^t(s), q^1(s), q^2(s), q^3(s)) \quad , \quad q^t(s) = q^t(0)$$

we get lines known as *field lines* or *stream lines*. They correspond to the flow direction of many particles which are spread around in the volume of the vector field. Path lines and stream lines are both one-dimensional manifolds; they can't cross each other, since at each point their direction is uniquely determined by the given vector field. A stream line is a static object, all of its points belong to the same time slice, whereas a path line is constructed by points from different time instances. A path line can be considered as the projection of a stream line within an time-dependent n -dimensional manifold onto an $n - 1$ dimensional spatial submanifold (a path line is the three-dimensional “image” of a four-dimensional stream line) – this projection *may* intersect itself.

Beside the inspection of lines that start from a single event q_0 we can also study the behavior of a bundle of lines that start from a set of events, e.g. some “initial seed” line $q_0(\tau) : I \rightarrow M$ with $I \subset \mathbb{R}$. The *integral surface* $S \subset M$ within a vector field $v \in \mathcal{T}(M)$ with initial seed line q_0 is then constructed from all integral lines that pass through an event on this initial seed line:

$$S = \{q : \mathbb{R} \rightarrow M, \dot{q}(s) = v(q(s)), q(0) = q_0(\tau)\} \quad .$$

It contains a natural parametrization $S(s, \tau)$ by the initial seed parameter τ and the integration length s . The intersection of an 2-dimensional surface within a n -dimensional manifold with a $n - 1$ sub-manifold does not necessarily yield a one-dimensional manifold and thus may lead to lines that can self-intersect.

A commonly used choice is to use a timelike initial seed line $q_0(\tau) = (\tau, q^1, q^2, q^3)$ with fixed spatial coordinates q^1, q^2, q^3 (we can call such a seed line a “location”, since it describes a point in space independent of time). The resulting integral surface will then be spanned by a timelike tangential vector ∂_t and a spacelike tangential vector ∂_s . For a fixed time coordinate t the projection of the integral surface into a time slice $dt = 0$ reveals a line, called a “*streak line*”. It is formed by the location of all particles that have passed (or will pass) through a specific point $q_0(t)$ at some time t . For stationary vector fields, integral lines will be independent of time, and so streak lines will coincidence with stream lines.

Another choice is to use a spacelike initial seed line $q_0(\tau) = (q^0, q^1(\tau), q^2(\tau), q^3(\tau))$. The image of the seed line under evolution, the line $S(s, \tau)|_{s=const}$

is called a “*material line*” or “*time line*”. The surface S is called a *stream surface*. An improved algorithm for computing a streamsurface based on the original algorithm by [Hul92] is given in [Sta98] and [GTS04]

Sometimes higher dimensional initial seed data are used, revealing surfaces or volumes evolving under the flow map of the underlying vector field. For instance, evolving a timelike two-dimensional initial seed surface (spatially a line) yields *streak surfaces* in the spatial projection of the resulting timelike volume. In general, these spatial projections are not manifolds and may penetrate itself like streak lines.

2.2 Eigenvector Stream Lines

As the maximal eigenvectors of a tensor $G \in T_p^{*2}(M)$ play an important role, it is a straightforward approach to employ vector field visualization methods to inspect them. The integration of maximal eigenvectors as stream lines is the basis of one of the widest known tensor field visualization techniques called hyperstreamlines [DH93], which extends the vector field technique by adding an elliptical cross-section encoding the median and minor eigenvalues. However, treatment of eigenvectors needs to consider two relevant aspects:

- The maximal eigenvector is undefined in isotropic regions. Its direction is ambiguous and may vary due to slight numerical instabilities.
- The sign of eigenvectors is undefined, since $-\mathbf{v}$ is a solution of the eigenvalue equation $G \cdot \mathbf{v} = \lambda \mathbf{v}$ as well. We may call a vector which sign is left open a “*pre-vector*” (it is not yet a fully determined vector).

The difference among pre-vector and vector fields is important as it is possible to find a pre-vector field as the tangential vectors of a non-orientable manifold. For instance, consider a three-dimensional pre-vector field that is tangential to a Moebius strip in a volume, and getting smoothly zero farther away from the Moebius strip. Since the Moebius strip is a non-orientable manifold, so also its associated tangential pre-vector field cannot be oriented globally to yield a vector field. Thus in general it is not possible to apply unmodified vector field visualization methods to eigenvectors. Furthermore, due to the eigenvector ambiguity within isotropic regions, unmodified vector field visualization find and display features which are not a property of the data field but stem from the numerical eigenvalue extraction algorithm (isotropy artifacts). Modified interpolation and/or integration methods are required for eigenvector fields:

- When interpolating eigenvectors within a cell, all vectors contributing to the interpolation must be oriented such that they point into the same half-space, i.e. $\mathbf{v}_i \cdot \mathbf{v}_j \geq 0$ (“*local alignment*”).
- Interpolating eigenvectors yields different results than interpolating the tensor field with computation of the eigenvector at each interpolation point (spherical interpolation sustaining tensor shape versus linear interpolation of components).

- Stream line integration advances a point $q(s)$ of the stream line q to the next point $q(s + ds)$ by a small step size ds via

$$q(s + ds) = q(s) + ds \dot{q}(s) \quad ,$$

whereby the new tangential direction is the direction of the vector field \mathbf{v} at the point of interest $q(s)$

$$\dot{q}(s) = \mathbf{v}|_{q(s)} \quad .$$

Here, \mathbf{v} is the solution of the eigenvalue equation $G\mathbf{v} = \lambda\mathbf{v}$ at the point $q(s)$ such that $\mathbf{v} \cdot \dot{q}(s) \geq 0$. This last condition of local alignment during integration is essential and needs to be added to a usual stream line integration algorithm.

- Local alignment does not cure the problems arising from isotropy artifacts. Stream lines of the maximum eigenvector only lead to reasonable results in regions with one dominant eigenvalue. An alternative, less vulnerable integration algorithm is thus to start stream lines in regions with high linearity and to advance it according to the deviation vector ² $G \cdot \dot{q}$:

$$\mathbf{v} = G \cdot \dot{q} \quad \rightarrow \quad q(s + ds) = q(s) + ds \mathbf{v} \quad . \quad (3)$$

Integral lines of deviation vector fields are e.g. used [ZP03]. The method of “tensorlines” [WKL99] combines this method by blending the oriented maximal eigenvector \mathbf{v}_{max} and the deviation vector with the linearity shape factor c_l at the point $q(s + 1)$:

$$\mathbf{v} = c_l \mathbf{v}_{max} + (1 - c_l) [(1 - w) \dot{q}(s) + w G \cdot \dot{q}] \quad .$$

Hereby w is a user-controlled “stiffness” parameter in the range $[0, 1]$ which is said to be selected depending on the type of data.

- Both integral lines as solution of Eq. (3) and tensorlines don’t provide an unique direction at each point in space, thus intersections of lines may occur - in contrast to non-intersecting stream lines.

Visualization of Eigenvector Stream Lines. Hyperstreamlines [DH93] are a widely known technique for visualizing eigenvector streamlines. They encode the median and minimal eigenvectors as elliptical cross-section and the maximal as color along the streamline. However, they severely suffer under isotropy

² This operation actually yields a co-vector, not a vector, which in the case of a non-flat base manifold is not identical. In general, the metric g of the underlying manifold needs to be involved to map the co-vector onto a vector $G \cdot \dot{q} \rightarrow \sharp G(\dot{q}, -) \equiv g^{\mu\nu} G_{\nu\kappa} \dot{q}^\kappa$ - hereby employing the Einstein sum convention in the coordinate expression, which is implicit summation over multiply occurring indices. This operation is also known as the \sharp -operator or “index raising” musical isomorphism.

artifacts and, while they are good for inspecting single lines, they suffer under view occlusion problems when studying a large data volume. An alternative approach is to use the technique of illuminated stream line technique [ZSH96] upon eigenvector fields. As the cross-section is infinitely small here, we may encode the additional tensor field quantities as transparency and line distribution density. To reduce or even avoid anisotropy artifacts, the transparency is set proportional to the isotropy, the spherical shape factor c_s is appropriate. Consequently, lines isotropic regions of undefined maximal eigenvector become invisible, although a stream line of the maximal eigenvector field continues there technically. The seed points for the stream line integration, which determines the density and number of stream lines, are set dominantly in regions where one eigenvector is dominant (density chosen to be proportional to the linear shape factor c_l), because only then there is a unique direction. As a result, stream lines start in highly linear regions, may traverse through planar regions but are less dense there and vanish in isotropic parts of the volume data set.

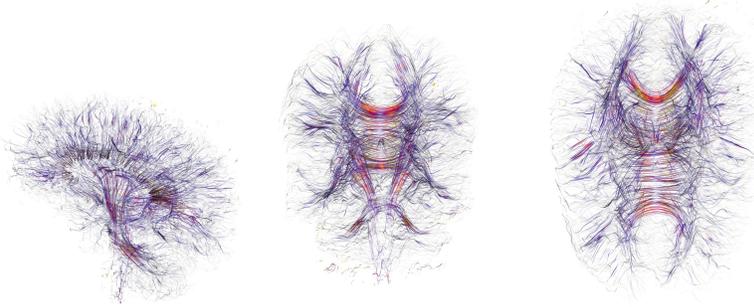


Fig. 2. Front, side and top view of stream lines along the maximum eigenvector in linear regions of the human brain data set.

This approach is very suitable for full three-dimensional visualization of a data set as in Fig. 2. Due to the three-dimensional nature of stream lines, it is not even applicable to two-dimensional slices. The ISL/transparency/line density based eigenvector stream line technique is able to display practically all of the tensor field features, including isotropic and linear regions in a clear way (see Fig. 3). However, it is not suitable for point-wise detailed inspection of a data set. Also, planar regions are not visualized correctly, since eigenvector stream lines visually suggest only one direction there.

2.3 Geodesics

A positive definite tensor field of rank two can be interpreted as a metric tensor field and used to measure distances among points in space and time.

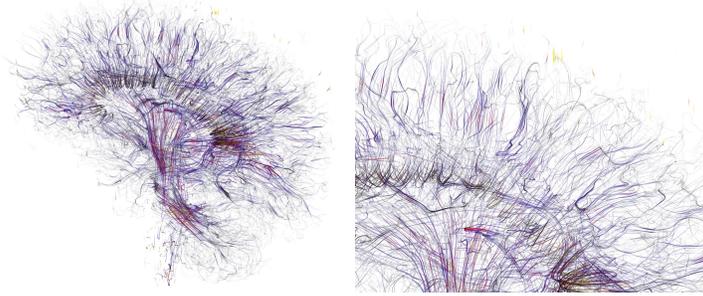


Fig. 3. Stream lines along the maximum eigenvector in linear regions in a portion of the Human Brain.

An *extremal line* is the shortest or longest (i.e. most extreme) connection between two points. It is determined by the metric. A curve $q(s)$ is the most extreme connection between two points $A = q(s_1)$ and $B = q(s_2)$ iff

$$\int_{s_1}^{s_2} \sqrt{|g(\dot{q}(s), \dot{q}(s))|} ds = \text{minimum} \quad .$$

We may employ the Lagrange formalism to derive a differential equation for the curve $q(s)$. If we take the *square* of the length of the tangential vectors as Lagrange function,

$$\mathcal{L}(q^k(s), \dot{q}^k(s)) = g(\dot{q}(s), \dot{q}(s)) = \dot{q}^\mu \dot{q}^\nu g_{\mu\nu} \quad (4)$$

then the parameterization of the curve becomes fixed and the solutions are extremal lines parameterized by their affine parameters (proof not shown here). These curves are called *geodesics*. By inserting (4) into the well-known Euler-Lagrange-equations

$$\frac{\partial \mathcal{L}}{\partial q^k} - \frac{d}{ds} \frac{\partial \mathcal{L}}{\partial \dot{q}^k} = 0 \quad (5)$$

and a little algebra we arrive at the coordinate expression for the geodesic equation

$$\ddot{q}^\lambda + \Gamma_{\mu\nu}^\lambda \dot{q}^\mu \dot{q}^\nu = 0 \quad , \quad (6)$$

whereby $\Gamma_{\mu\nu}^\lambda$ are the so-called *Christoffel symbols*. They abbreviate an expression involving only the metric and its first partial derivatives:

$$\Gamma_{\mu\nu}^\lambda := \frac{1}{2} g^{\lambda\alpha} (g_{\mu\alpha,\nu} + g_{\nu\alpha,\mu} - g_{\mu\nu,\alpha}) \quad . \quad (7)$$

Here, the comma denotes the partial derivative by a coordinate function, i.e. $g_{\mu\alpha,\nu} \equiv \partial g_{\mu\alpha} / \partial x^\nu$. The partial derivative of a tensor $g_{\mu\alpha,\nu}$ (or even of a

vector v^μ , $_{\nu}$) does not yield a tensor again; it may be zero in all components in one coordinate system while non-zero in another ones. Consequently, the Christoffel symbols $\Gamma^\lambda_{\mu\nu}$ do not form a tensor, too. The Christoffel symbols can be used to define the *covariant derivative* of a tensor field that does not depend on the choice of coordinates. This covariant derivative is denoted by a semicolon, e.g. v^μ , $_{\nu}$. Its coordinate expression for a vector field is given by

$$v^\mu$$
, $_{\nu} = v^\mu$, $_{\nu} + \Gamma^\mu_{\lambda\nu} v^\lambda$.

The *directional derivative* of a vector field is just the linear combination of covariant derivatives and the components of the direction of interest. This operation is written as $\nabla_u v$:

$$\nabla_u v = v^\mu$$
, $_{\nu} u^\nu \partial_\mu$.

The covariant derivative has the interesting property that it does not commute in general, whereas the partial derivative always commutes:

$$v^\mu$$
, $_{\nu,\lambda} = v^\mu$, $_{\lambda,\nu}$ but not v^μ , $_{\nu;\lambda} = v^\mu$, $_{\lambda;\nu}$.

As the covariant derivative yields a tensor, so does the difference of two covariant derivatives. This tensor is the Riemannian curvature tensor:

$$K^\mu_{\nu\lambda\sigma} v^\sigma = v^\mu$$
, $_{\nu;\lambda} - v^\mu$, $_{\lambda;\nu}$

It is used in general relativity to form the left-hand side of the Einstein field equations of the gravitational field (via contraction). The Riemann tensor is a map $K : V \times V \times V \rightarrow V$ and is defined in coordinate free notation with $u, v, w \in V$ based on the directional derivative ∇ and the commutator $[u, v] = u(v) - v(u)$:

$$K(u, v)w := \nabla_u \nabla_v w - \nabla_v \nabla_u w - \nabla_{[u, v]} w$$
 . (8)

The Riemann tensor only depends on the metric, its first and second partial derivatives. The first partial derivatives may vanish in a certain coordinate system, but the second ones do not. Thus, the Riemann tensor allows an coordinate-independent classification of the underlying metric tensor fields. If all of its components vanish in one coordinate system, then the metric space associated with the tensor field is said to be *flat*.

2.4 Geodesic Deviation

The difference among close geodesics as depicted by the cross-section of a geodesic bundle depends on differences of the Christoffel symbols and thus directly visualizes the Riemann tensor. We show a short proof based completely on coordinate free notation. Let $\Phi(s, t) : \mathbb{R}^2 \rightarrow M$ denote a two-dimensional family of geodesics such that for fixed parameter t the curves

$\gamma(s) := \Phi(s, t = \text{const.})$ are geodesics. Let $\delta := \frac{d}{dt}\Phi(s, t) \in \mathcal{T}(M)$ denote the deviation vector of points on the geodesics with same parameter $s \in \mathbb{R}$, also known as the *Jacobi field* of the geodesics [O’N83]. Here the dot denotes the derivative by the geodesic parameter s , which is given by the directional derivative along the geodesic:

$$\dot{\delta} := \frac{d}{ds}\delta \equiv \nabla_{\dot{\gamma}}\delta$$

We may also describe the deviation by an vector field $\delta \in \mathcal{T}(M)$ that is transported along the geodesic bundle, i.e. its evolution is described by the flow map along the geodesics. This requires its Lie derivative $\mathfrak{L}_{\dot{\gamma}}\delta$ along the geodesics to vanish (see [Ben04] for illustration):

$$0 = \mathfrak{L}_{\dot{\gamma}}\delta \equiv [\dot{\gamma}, \delta] = \dot{\gamma}\delta - \delta\dot{\gamma} = \nabla_{\dot{\gamma}}\delta - \nabla_{\delta}\dot{\gamma} \quad (9)$$

and we see that

$$\nabla_{\dot{\gamma}}\delta = -\nabla_{\delta}\dot{\gamma} \quad .$$

If we compute the second derivative by the affine parameter we get

$$\ddot{\delta} := \frac{d^2}{ds^2}\delta \equiv \nabla_{\dot{\gamma}}\nabla_{\dot{\gamma}}\delta = -\nabla_{\dot{\gamma}}\nabla_{\delta}\dot{\gamma}$$

Recalling definition Eq. (8) of the Riemann tensor $K(u, v)w$ and inserting $u = \delta$, $w = v = \dot{\gamma}$ yields:

$$K(\delta, \dot{\gamma})\dot{\gamma} = \nabla_{\delta}\nabla_{\dot{\gamma}}\dot{\gamma} - \nabla_{\dot{\gamma}}\nabla_{\delta}\dot{\gamma} - \nabla_{[\delta, \dot{\gamma}]} \dot{\gamma} \quad .$$

$\nabla_{\dot{\gamma}}\dot{\gamma} = 0$ is just the geodesic equation and from (9) we know that $[\delta, \dot{\gamma}] = 0$. Thus we see that the second derivative of the deviation vector is linearly related to the Riemann curvature tensor:

$$\ddot{\delta} = K(\delta, \dot{\gamma})\dot{\gamma}$$

The evolution of the deviation vector in a chart is given by the coordinate expression

$$\ddot{\delta}^{\mu}\partial_{\mu} = K^{\mu}_{\alpha\beta\nu}\delta^{\alpha}\dot{\gamma}^{\beta}\dot{\gamma}^{\nu}\partial_{\mu} \quad .$$

In flat space $K = 0$ in any coordinate system and no focusing happens. The deviation vector then describes just a linear expansion of a geodesic bundle like a cone, depending on its initial cross-section δ and opening angle $\dot{\delta}$. The influence of curved space on a geodesic bundle, e.g. as depicted in Fig. 4, is also known as “Ricci focusing” and plays a central role in gravitational lens theory. An extensive discussion of theory and application can be found in Ehlers et.al. [SSE94,SEF99].

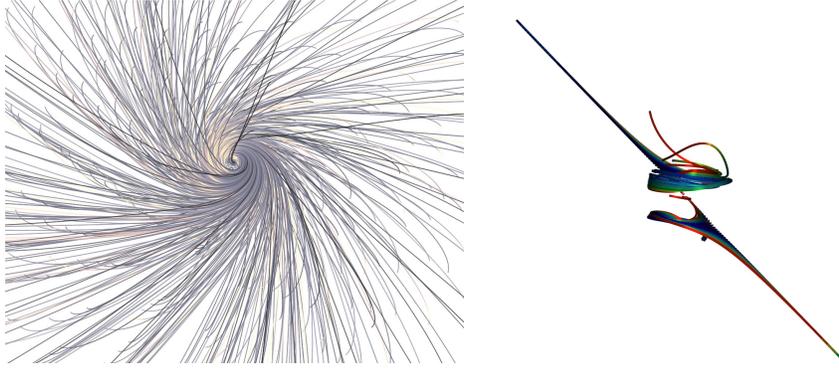


Fig. 4. Particle Geodesics in the vicinity of a rotating black hole. The congruence of the geodesics is a direct visualization of the Riemann tensor, which is a central component of the Einstein equations describing the gravitational field in general relativity. The change of proper distances among geodesic paths thus indicates that the spacetime is non-Euclidean, i.e. it has a non-vanishing Riemann tensor due to some mass distribution.

2.5 Projection of Metric Ellipsoids

For rendering the quadric surface of a tensor, it is sufficient to just draw an two-dimensional ellipse instead of a truly three-dimensional object that is projected by the 3D graphics engine. For drawing such an ellipsoid, we can draw a rectangle with an arbitrary texture on it. This rectangle needs to be oriented perpendicular to the view direction and transformed by a transformation matrix according to the projection of the tensor field in the view plane.

Let \mathbf{z} be the view direction and \mathbf{x}, \mathbf{y} be two orthonormal vectors describing the view plane. A point $\boldsymbol{\varepsilon}$ on the view plane can be computed from two parameters (a, b) via $\boldsymbol{\varepsilon} = a\mathbf{x} + b\mathbf{y}$. We get the projected ellipsoid by considering a ray $\mathbf{p} = \boldsymbol{\varepsilon} + \lambda\mathbf{z}$ that is orthogonal to the view plane (for orthogonal projection, we could model rays for perspective projection as well). Points on the ellipsoid obey $g(\mathbf{p}, \mathbf{p}) = 1$, which yields a quadratic equation in the ray parameter λ

$$1 = g(\boldsymbol{\varepsilon} + \lambda\mathbf{z}, \boldsymbol{\varepsilon} + \lambda\mathbf{z}) \equiv g(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}) + 2\lambda g(\boldsymbol{\varepsilon}, \mathbf{z}) + \lambda^2 g(\mathbf{z}, \mathbf{z}) \quad (10)$$

For the projection of the ellipsoid on the view plane we are interested in the set of parameters (a, b) where the ray is tangential to the ellipsoid, i.e. where the discriminant of (10) vanishes:

$$\frac{g(\boldsymbol{\varepsilon}, \mathbf{z})^2}{g(\mathbf{z}, \mathbf{z})^2} - \frac{g(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}) - 1}{g(\mathbf{z}, \mathbf{z})} = 0 \equiv g(\boldsymbol{\varepsilon}, \mathbf{z})^2 - g(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon})g(\mathbf{z}, \mathbf{z}) + g(\mathbf{z}, \mathbf{z}) = 0. \quad (11)$$

Substituting $\varepsilon = a\mathbf{x} + b\mathbf{y}$ into Eq. (11) yields a quadratic expression in (a, b) :

$$a^2 [g(\mathbf{x}, \mathbf{z})^2 - g(\mathbf{x}, \mathbf{x})g(\mathbf{z}, \mathbf{z})] + 2ab [g(\mathbf{x}, \mathbf{z})g(\mathbf{y}, \mathbf{z}) - g(\mathbf{x}, \mathbf{y})g(\mathbf{z}, \mathbf{z})] + b^2 [g(\mathbf{y}, \mathbf{z})^2 - g(\mathbf{y}, \mathbf{y})g(\mathbf{z}, \mathbf{z})] + g(\mathbf{z}, \mathbf{z}) = 0 \quad (12)$$

The coefficients in (12) for a^2 , $2ab$ and b^2 are the components of a bilinear form describing the *shadow* of the metric ellipsoid in the coordinates (a, b) . Note that in this derivation we never used coordinates on the 3-vectors, i.e. this derivation was completely coordinate-free. We may also write (12) as

$$(a \ b) \underbrace{\left[g(\mathbf{z}, \mathbf{z}) \overbrace{\begin{pmatrix} g(\mathbf{x}, \mathbf{x}) & g(\mathbf{x}, \mathbf{y}) \\ g(\mathbf{y}, \mathbf{y}) \end{pmatrix}}^{=: \pi(g)} - \begin{pmatrix} g(\mathbf{x}, \mathbf{z})^2 & g(\mathbf{x}, \mathbf{z})g(\mathbf{y}, \mathbf{z}) \\ g(\mathbf{y}, \mathbf{z})^2 \end{pmatrix} \right]}_{=: \sigma(g)} \begin{pmatrix} a \\ b \end{pmatrix} = g(\mathbf{z}, \mathbf{z}) \quad (13)$$

whereby $\pi(g)$ is the the intersection of the ellipsoid with the view plane $g(\varepsilon, \varepsilon)$ and $\sigma(g)$ is the “shadow ellipsoid”. With (v, w) the eigenvectors of this 2×2 metric and (λ, μ) the corresponding eigenvalues, i.e.

$$\sigma g \cdot v = \lambda v \quad \sigma g \cdot w = \mu w \quad ,$$

the orientation of the resulting projected ellipsoid in 3D is given by evaluating the eigenvectors as linear combination of the basis $\{\mathbf{x}, \mathbf{y}\}$:

$$\mathbf{p}_1 = v_x/\sqrt{\lambda} \mathbf{x} + v_y/\sqrt{\lambda} \mathbf{y} \quad (14)$$

$$\mathbf{p}_2 = w_x/\sqrt{\mu} \mathbf{x} + w_y/\sqrt{\mu} \mathbf{y} \quad (15)$$

The two three-dimensional vectors $\mathbf{p}_1, \mathbf{p}_2$ are orthonormal with respect to the metric tensor g (i.e. $g(\mathbf{p}_i, \mathbf{p}_j) = \delta_{ij}$) and are completely contained in the view plane \mathbf{x}, \mathbf{y} .

Since the eigenvalue equation of $\sigma(g)$ is just quadratic, it can be solved faster and more precisely than the eigenvalue equation of the full 3×3 tensor matrix. From the visualization side, the advantage of this method is that we can use an arbitrary image as texture on the distorted rectangle.

2.6 Tensor Fields for Analyzing Vector Fields

If the underlying manifold is a vector space, then we may consider a vector field v as displacement field, i.e. to compute an “elastic distortion” d by remapping each point onto another point as given by the vector field:

$$d : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (16)$$

$$p \mapsto p + v(p) \quad (17)$$

A set of equidistant points will no longer remain equidistant under this mapping. If we assign the new point location with the same coordinates as the original points, we need to replace the flat metric with a non-euclidean metric to correctly describe the distances to neighboring points. This non-flat metric is directly related to the underlying displacement vector field. Visualization of this distortion tensor field is an alternative to displaying the vector field or the distorted point locations.

Derivation of the distortion tensor field. Let x_0, x_1 be two points with unit distance, i.e. $\|x_1 - x_0\| = 1$. These points are mapped to $x_0 + v(x_0)$ and $x_1 + v(x_1)$. The difference vector between these mapped points is then

$$\Delta l := (x_1 + v(x_1)) - (x_0 + v(x_0)) \equiv (x_1 - x_0) + (v(x_1) - v(x_0)) =: \Delta x + \Delta v \quad .$$

We now are seeking for a metric that fulfills $g(\Delta l, \Delta l) = 1$, i.e. the metric tensor field that is induced by the displacement vector field v . By insertion we find:

$$g(\Delta l, \Delta l) = g(\Delta x, \Delta x) + g(\Delta x, \Delta v) + g(\Delta v, \Delta x) + g(\Delta v, \Delta v)$$

In a coordinate system, the difference vector Δl is given component-wise as $\Delta l^i = \Delta x^i + \Delta v^i$. To derive a metric tensor field in the tangential space, we write the infinitesimal difference vector as

$$d\Delta l = \Delta l^i{}_{,j} dx^j \partial_i = (\delta_j^i + v^i{}_{,j}) dx^j \partial_i$$

Insertion yields

$$g_{ij} = \delta_{ij} + v^i{}_{,j} + v^j{}_{,i} + \sum_k v^k{}_{,i} v^k{}_{,j} \quad . \tag{18}$$

This tensor becomes the Euclidean metric for a constant vector field. The component $v^i{}_{,j} + v^j{}_{,i}$ is also known in continuum mechanics as the *deformation* or *strain tensor*. Based on this relationship, we may now use tensor field visualization methods to inspect arbitrary vector fields.

3 Vertex-Based Visualization Methods

3.1 Legend for Tensor Glyphs

The following sections contain a collection of various tensor field visualization methods using glyphs (icons) as the basic primitive. For evaluation and comparison of these methods, a “legend” is helpful that displays the extreme cases of tensor shape factors. For such a legend we arrange the tensors within a triangle and construct the tensor shape from its barycentric coordinates:

$$\begin{array}{ccc}
 & c_s = 1 & \\
 c_t=0 \nearrow & & \searrow c_p=0 \\
 c_p = 1 & \xleftrightarrow{c_s=0} & c_l = 1
 \end{array}$$

The appropriate tensor can be constructed as a diagonal tensor with eigenvalues computed from the shape factors. A possible way is to impose constant trace of the tensor along the edges of the tensor legend triangle, i.e. when writing a tensor as an eigenvalue triple $G = (c_s/3, c_s/3+c_p/2, c_s/3+c_p/2+c_l)$ with the maximum eigenvector direction left open for visual adjustments:

$$\begin{array}{ccc} & \frac{1}{3}(1, 1, 1) & \\ c_l=0 \nearrow \swarrow & & \searrow \nwarrow c_p=0 \\ \frac{1}{2}(1, 1, 0) & \xleftrightarrow{c_s=0} & (1, 0, 0) \end{array}$$

However, this choice lead to unpleasant results as in Fig. 5, right, because the according quadric surfaces degenerate at the $c_s = 0$ line: the maximum half axis of the tensor ellipsoid is the inverse square root of the minimum eigenvalue. The minimum eigenvalue is zero at the $c_s = 0$ line, thus the tensor ellipsoid becomes infinitely long in the direction of the minimum eigenvector (and at the same time becoming infinitely thin, therefore sustaining the trace).

A better choice is to normalize the tensors by sustaining the minimum eigenvalue. This provides a more intuitive visual clue of “a sphere flattening to a disc shrinking to a needle”, as in Fig. 5, left. Such a tensor shape is constructed from the eigenvalue triple as

$$(\lambda_{max}, \lambda_{med}, \lambda_{min}) = \left(\frac{3c_l}{c_s} + \frac{3c_p}{2c_s} + 1, \quad \frac{3c_p}{2c_s} + 1, \quad 1 \right) \lambda_{min}$$

whereby we may choose the minimum eigenvalue (the maximum diameter of the tensor ellipsoid) as a free parameter. For $\lambda_{min} = 1$ this tensor shape corresponds to a legend triangle of:

$$\begin{array}{ccc} & \frac{1}{3}(1, 1, 1) & \\ c_l=0 \nearrow \swarrow & & \searrow \nwarrow c_p=0 \\ \frac{1}{2}(\infty, \infty, 1) & \xleftrightarrow{c_s=0} & (\infty, 1, 1) \end{array}$$

We see that the minimum eigenvalue is kept constant by the cost of other eigenvalues becoming infinite. Accordingly the trace of the tensor is not constant along the triangle edges. Instead, it is inversely proportional to the sphericity via $tr(G) = 3\lambda_{min}/c_s$, thus becoming infinite on the $c_s = 0$ edge as well. Nevertheless keeping $\lambda_{min} = \text{const.}$ along the triangle edges is a better choice than keeping $tr(G) = \text{const.}$. For numerical purposes, the $c_s = 0$ line should be avoided by a small $\epsilon > 0$, otherwise glyphs like a quadric surface would become infinitely thin, ergo invisible and thus useless. Extending the tensor legend to include negative eigenvalues has not yet been investigated and is an open issue.

3.2 Selected Methods

Here, we review six glyph-based methods that provide alternatives to quadric surfaces (“metric ellipsoids”).

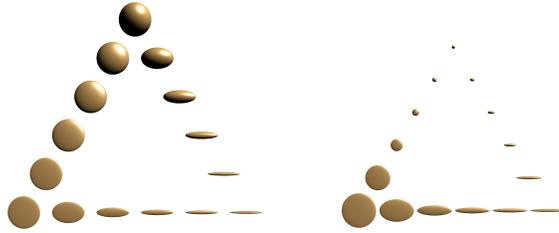


Fig. 5. Legend with constant minimal eigenvalue (left) and constant trace (right).

Reynold Glyphs. Reynold glyphs [MSM95] are an inverse mapping of the metric ellipsoid, mapping each direction v to $v \cdot \sqrt{G(v, v)}$ instead of $v/\sqrt{G(v, v)}$ as with the quadric surface.

Haber Glyphs. Haber [Hab90] used a disc and a rod instead of an ellipsoid to encode the eigenvalues of a tensor at each point. This glyph is useful for depicting anisotropy more easily than ellipsoids, but is also vulnerable to isotropy artifacts.

Tensor Glow. Here, the idea is to avoid rendering three-dimensional objects as a tensor glyph, but instead to only compute the projection of the tensor ellipsoid on the view plane on the fly and depending on the view direction [Ben04]. The actual graphics primitive is just a rectangle which is stretched and oriented according to the visible projection as derived in Eq. (14) and Eq. (15). This rectangle can be rendered very fast and equipped with an arbitrary, even animated texture. It is thus very suitable to provide the impression of a glowing flash of light dissipating into space, which is an intuitive rendering of a metric tensor field.

Tensor Cones. Inspired by the frequently used light cones in general relativity, tensor cones [Ben04] are constructed from little cones with elliptical cross-sections. An arbitrary input vector field has to be provided which forms the original axis of these cones. Their extruded cross-section is computed from the 2×2 tensor in the projection orthogonal to the original axis. Finally, the three left over components of the tensor field along the vector field are used to tilt the cones according to the deviation vector. The tensor cones incorporate the full tensor information content, but depend on a certain input vector field. This allows to display a vector field in addition to a tensor field simultaneously.

Tensor Schlieren. This is experimental technique [Ben04] where the deviation of the view direction by the tensor field is visualized by decreased

transparency at locations of large deviations. The visual appearance is like a fuzzy geometry that changes with view position or rotation of the data volume. This technique is not limited to positive definite tensor fields.

Tensor Splats. This technique has been extensively described in [BH04b] and [BH04a]. The basic idea is to replace the complex geometries of glyphs by transparent splats equipped with a texture-like pattern that incorporates the same information content. As a result, tensor splats are able to visualize entire three-dimensional volumes of a tensor field and intuitively provide a notion of the tensor field’s important properties.

3.3 Comparison of Glyph-Based Visualization Techniques

Application to Brain Data. Diffusion weighted magnetic resonance imaging (DW MRI) is a technique that measures the diffusion properties of water molecules in tissues [ZMB⁺03]. With the availability of such measured tensor field data for medical purposes, the interest of visualizing such data has grown rapidly in the last years [ZDD⁺01, ZDK⁺01, TRW⁺02, HTR⁺02, TWD⁺01]. The inspection of diffusion tensor data is relevant for the segmentation and classification of MRI data to detect the white matter tracts that form the “wiring” of the human brain [KWH00]. We will compare various methods upon an example data set using exactly the same view parameters.

As first approach, we may employ metric ellipsoids with colors indicating the trace of the tensor. We find that this representation clearly depicts the properties of the tensors at each point, Fig. 6, but we need to enlarge the image such that each ellipsoids becomes visible on their own. When inspecting the entire image as an overview, hardly anything can be seen at all because the structures of the ellipsoids fall below the image resolution. Equivalently we could use volume rendering of the trace as a scalar field. But even when zoomed onto an interesting regions, the ellipsoids are hard to interpret because we only see their projected shape.

Employing the method of tensor glow in Fig. 7 reduces the visual clutter. In the variant used here, the projected glow pattern is not normalized, but its transparency is proportional to the trace of the tensor field. Other variants are possible, too. E.g. using an isotropy indicator were a reasonable approach. Employing these settings upon the human brain tensor field enhances regions of high trace. This is the region where water may flow rapidly. Such areas are depicted clearly, in an overview as well as in an enlargement. We also get an glimpse of the orientation of the flow, but it is not too prominent as the anisotropy is not overwhelmingly large. The tensor glow method is thus applicable and helps to enhance certain features, but one gets the impression that it should be possible to do better.

Although specifically developed for relativistic data, using tensor cones for brain data Fig. 8 resulted in a positive surprise: it displays some global

structure information more clearly than both metric ellipsoids and tensor glow. This is due to the larger sensitivity of the appearance of tensor cones to variations of the tensor field. As a consequence, we get a good overview of all structures contained in the data set. However, the interpretation is difficult because we have the vector field probe as an arbitrary input parameter. The structure of this user-chosen vector field is clearly recognizable, so we can study the tensor field properties by visual inspection, but it still requires some mental effort.

Haber glyphs have some history in computation fluid dynamics. They are very sensitive to anisotropy and are thus able to enhance global structures in an overview similar to the tensor cones, but without dependence on an user-chosen input vector field. The enlarged view as in Fig. 9 also gives an hint of some large structures that incorporate an flow. However, to really recognize the details, we require an extreme enlargement such that all glyphs become resolved. An drawback of Haber glyphs are their anisotropy artifacts, as the glyphs are randomly oriented in isotropic areas.

In contrast to tensor cones and Haber glyphs, the technique of tensor schlieren uses transparency as a fundamental part of the visualization technique. Thus, it is more suitable for large-scale overviews. However, transparency is not an invariant quantity here, but depends on the view direction, as the purpose of tensor schlieren is to enhance regions where the maximal eigenvector is perpendicular to the view direction. Tensor schlieren are thus especially suitable for an interactive environment than for static, two-dimensional images. However, even for static images it yields the best overview, Fig. 10, of the brain visualizations discussed so far: it reduces visual clutter by rendering large regions transparent (those where the maximal eigenvector is parallel to the view direction), while strongly displaying the orientation of the minor eigenvector in other regions. We thus get a good structural overview plus directional information in each area.

While tensor schlieren produces view-dependent images encoding the orientation of the tensor field's eigenvectors by intensity, the technique of tensor splats [BH04b] uses colors for doing so and enhances the differences among the median and minor eigenvector by employing an additional one-dimensional high-frequency texturing. Transparency is used to encode the isotropy, i.e. isotropic regions are visually removed from the image. The result Fig. 11 is a strong enhancement for all anisotropic features with clear depiction of difference among minor and median eigenvectors as well. The tensor splat technique intentionally displays various features redundantly in different manners to compensate the reduction of visual information by projection of the glyph geometry onto the two-dimensional view plane. E.g. green indicates a linear region independent from its orientation and is thus clearly distinguishable from a red disk seen from aside. Tensor splats thus appear to provide the best view of the discussed methods and are also appropriate for full three-dimensional volume visualization.

Visualizing Displacement Vector Fields. The tensor splats visualization method is appropriate for distortion tensor fields as in Eq. (18) derived from vector fields because it maps isotropic regions (corresponding to unstretched grid regions) to transparency.

We discuss the method of tensor splats on a data set stemming from elastic registration of image data. It describes the deviations of a individual bee brain from a standardized average bee brain.

Fig. 12, left, displays the resulting non-uniform grid. Fig. 12, right, is overlaid with tensor splats of the induced metric tensor field. Tensor splats are highly sensitive to anisotropy and display even small deformations. Isotropic regions correspond to areas where the grid is constantly shifted and are displayed transparently. Note that the tensor field display reveals full three-dimensional information of the grid distortion, whereas the grid view only provides a two-dimensional slice, so some tensor splats are visible also in regions with apparently no grid distortion.

While we can clearly see the connection among grid compression and tensor field in Fig. 13, we also find some visual ambiguities in the shape of the tensor splats within the left image of Fig. 13: The shape of the green splats in the upper left area is just similar to the shape of the red splats in toward the lower right. However, grid distortion apparently only occurs in the domain of the red splats. Due to the color coding we know that the green splats indicate a linear region. Adding the tensor texturing – right image of Fig. 13 – also reveals the direction of the linearity: it is perpendicular to the projected major axis and thus indicates that the linearity actually indicates grid stretching mostly perpendicular to the view plane. This interpretation is confirmed by a 3D zoom onto the region of interest as in Fig. 14.

4 Summary

Table 1 compares various tensor field visualization methods. The table is ordered according to the tensor field quantities which are used for the visualization. It is not possible to provide an overall evaluation for these methods and to determine the “best” visualization method – each method has advantages that might cause it to be superior to others in special cases. E.g. tensor ellipsoids are straightforward to understand, but suffer from the problem of visual clutter. Tensor splats clearly display relevant features of a tensor field even in 3D volume, but require some experience for understanding the visual effects – as with most tensor field visualization methods.

Nevertheless some criteria might help to select an appropriate method for a particular case:

- *Number of quantities:* Does the method make use of the full information content of the tensor field or does it work by reduction to fewer quantities? Often methods displaying a reduced set include parameters that allow to

browse other quantities as well, such that multiple images are required to get a complete impression of the tensor field.

- *Robustness against visual clutter*: Is a method suitable for three-dimensional data volumes or is it limited to two-dimensional slices only?
- *Isotropy artifacts*: Visualization methods based on eigenvectors have to address/handle ambiguities in isotropic regions.
- *Limitation to positive definite tensors*: Can the method handle tensor fields with negative or zero determinant? Tensor fields like in DT-MRI or Riemannian metric tensors are always positive definite, such that the corresponding quadric surface is an ellipsoid. The stress tensor in CFD or the extrinsic curvature tensor in general relativity may contain negative determinants as well.
- *Limitation to symmetric tensors*: Can the method display asymmetric tensor fields?

5 Acknowledgments

The DT-MRI brain dataset was provided generously by Gordon Kindlmann and Andrew L. Alexander. The displacement vector field for the bee brain was provided by Alex Maye from ZIB. We thank Konrad Polthier and Peter Deuffhard for fruitful discussions.

References

- [Ben04] W. Bengler, *Visualization of general relativistic tensor fields via a fiber bundle data model*, Ph.D. thesis, Free University Berlin, 2004, in preparation.
- [BH04a] W. Bengler and H.-C. Hege, *Analysing curved spacetimes with tensor splats*, 10th Marcel Grossmann Meeting, 2004.
- [BH04b] ———, *Tensor splats*, Conference on Visualization and Data Analysis 2004 (Erbacher, Chen, Roberts, Gröhn, and Börner, eds.), Proceedings of SPIE Vol. #5295, 2004, IS&T/SPIE Electronic Imaging Symposium in San Jose, CA.
- [DB02] P. Deuffhard and F. Bornemann, *Scientific Computing with Ordinary Differential Equations*, Springer Verlag, New York, 2002.
- [DH93] T. Delmarcelle and L. Hesselink, *Visualizing second order tensor fields with hyperstream lines*, IEEE Computer Graphics and Applications **13** (1993), 25–33.
- [GTS04] C. Garth, X. Tricoche, and T. Salzbrunn, *Surface techniques for vortex visualization*, accepted for VisSym '04.
- [Hab90] R.B. Haber, *Visualization techniques for engineering mechanics*, Comp. Sys. in Engineering **1** (1990), no. 1, 37–50.
- [HTR⁺02] J. Hauelsen, D.S. Tuch, C. Ramon, P. Schimpf, V.J. Wedeen, J.S. George, , and J.W. Belliveau, *The influence of brain tissue anisotropy on human eeg and meg*, Neuroimage **15** (2002), 159–166.

Tensor Encoding	Q	Viz Method	⌣	$tr(G)$	“+”	sym
$g_{xx}, g_{xy}, g_{yy},$ g_{yz}, g_{zz}, g_{zx}	6	Quadric surface	bad	good	no	yes
		Metric ellipsoids	bad	good	yes	yes
		Reynolds Glyph	bad	good	yes	yes
$v_{max}, \lambda_{max}, \lambda_{med}, \lambda_{min}$	6	Haber Glyph	bad	bad	yes	yes
		Tensor Schlieren	good	bad	yes	yes
$g_{00}, g_{01}, g_{11},$ g_{12}, g_{22}, g_{20}	6+3	Tensor Cones	bad	bad	yes	yes
g_{00}, g_{01}, g_{11}	3	Deformation Surfaces	bad	good	no	no
$g_{00}, g_{01}, g_{11},$ $tr(g)$	4	Color Coding	good	good	no	no
$v_{max}, tr(g), c_p, c_l$	6	Tensor Splats	good	good	yes	yes
$g(\mathbf{x}, \mathbf{x}), g(\mathbf{x}, \mathbf{y}), g(\mathbf{x}, \mathbf{y})$ $\mathbf{x}(\vartheta, \varphi), \mathbf{y}(\vartheta, \varphi)$	3-6	Tensor Glow	good	medium	yes	yes
$g_{xx}, g_{xy}, g_{yy},$ $tr(g), \lambda_{max}$	5	Van-Gogh keystrokes	bad	good	yes	yes
$v_{max}, \lambda_{max}, \lambda_{med},$ $\lambda_{min}, \partial v_{max}$	9	Hyperstreamlines	good	bad	yes	no
	9	Tensor Lines	good	good	yes	no
$G_{mn}, \partial g_{mn}$	“24”	Geodesic	good	good	yes	yes
$G_{mn}, \partial g_{mn}, \partial \partial g_{mn}$	“78”	Geodesics Bundle	good	good	yes	yes

Table 1. Comparison and assessment of selected tensor field visualization methods. The entries in column “⌣” stand for the robustness against visual clutter, $tr(g)$ indicates how far the methods suffer under eigenvector ambiguities in isotropic regions, the “+”/“sym” columns shows the methods requires the tensor field to be positive definite or symmetric. Van-Gogh keystrokes refers to the method described in [LAK⁺98]. Although geodesics incorporate the derivatives of the tensor field, too, they are only able to display a subset of the full parameter space depending on their initial conditions.

- [Hul92] Jeff P.M. Hultquist, *Constructing stream surfaces in steady 3d vector fields*, Visualization '92, IEEE Computer Society, 1992, pp. 171–178.
- [KWH00] G. Kindlmann, D. Weinstein, and D. Hart, *Strategies for direct volume rendering of diffusion tensor fields*, IEEE Transactions on Visualization and Computer Graphics **6** (2000), no. 2, 124–138.
- [LAK⁺98] D.H. Laidlaw, E.T. Ahrens, D. Kremers, M. J. Avalos, R. E. Jacobs, and C. Readhead, *Visualizing diffusion tensor images of the mouse spinal cord*, June 1998, pp. 127–134.
- [MSM95] J.G. Moore, S.A. Schorn, and J. Moore, *Methods of classical mechanics applied to turbulence stresses in a tip leakage vortex*, Proceedings of the ASME Gas Turbine Conference, Houston, Texas, no. 95-GT-220, 1995.
- [O'N83] B. O'Neill, *Semi-riemannian geometry, with applications to relativity*, Academic Press, Inc., 1983.
- [SEF99] Schneider, J. Ehlers, and Falco, *Gravitational lenses*, Springer-Verlag Berlin Heidelberg, 1999.
- [SSE94] S. Seitz, P. Schneider, and J. Ehlers, *Light propagation in arbitrary spacetimes and the gravitational lens approximation*, Class. Quant. Grav (1994), <http://arXiv.org/abs/astro-ph/950356>.
- [Sta98] D. Stalling, *Fast texture-based algorithms for vector field visualization*, Ph.D. thesis, Free University Berlin, 1998.
- [TRW⁺02] D.S. Tuch, T.G. Reese, M.R. Wiegell, N.G. Makris, J.W. Belliveau, and V.J. Wedeen, *High angular resolution diffusion imaging reveals intravoxel white matter fiber heterogeneity*, Magn Reson Med. **48** (2002), 577–582.
- [TWD⁺01] D.S. Tuch, V.J. Wedeen, A.M. Dale, J.S. George, and J.W. Belliveau, *Conductivity tensor mapping of the human brain using diffusion tensor mri*, Proc. Natl. Acad. Sci. USA (2001), no. 98, 11697–11701.
- [WKL99] D. Weinstein, G. Kindlmann, and E. Lundberg, *Tensorlines: Advection-diffusion based propagation through diffusion tensor fields*, IEEE Visualization 1999, IEEE Computer Society Press, 1999, pp. 249–253.
- [WPG⁺97] C.F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and FA. Jolesz, *Geometrical diffusion measures for mri from tensor basis analysis*, Proceedings of ISMRM, Fifth Meeting, Vancouver, Canada, April 1997, p. 1742.
- [ZDD⁺01] Song Zhang, Cagatay Demiralp, Marco DaSilva, Daniel Keefe, David H. Laidlaw, Benjamin D. Greenberg, Peter J. Basser, Carlo Pierpaoli, E.A. Chiocca, and T. S. Diesboeck, *Toward application of virtual reality to visualization of DT-MRI volumes*, Proceedings MICCAI, October 2001.
- [ZDK⁺01] Song Zhang, Cagatay Demiralp, Daniel Keefe, Marco DaSilva, Benjamin D. Greenberg, Peter J. Basser, Carlo Pierpaoli, E. A. Chiocca, T. S. Deisboeck, and David Laidlaw, *An immersive virtual environment for DT-MRI volume visualization applications: a case study*.
- [ZMB⁺03] L. Zhukov, K. Museth, D. Breen, R. Whitaker, and A. Barr, *Level set segmentation and modeling of dt-mri human brain data*, Journal of Electronic Imaging **12** (2003), no. 1, 125–133.
- [ZP03] X. Zheng and A. Pang, *Interaction of light and tensor fields*, VisSym '03, 2003, pp. 157 – 166.
- [ZSH96] M. Zöckler, D. Stalling, and H.-C. Hege, *Interactive visualization of 3d-vector fields using illuminated streamlines*, Visualization '96, Oct./Nov. 1996, pp. 107–113.

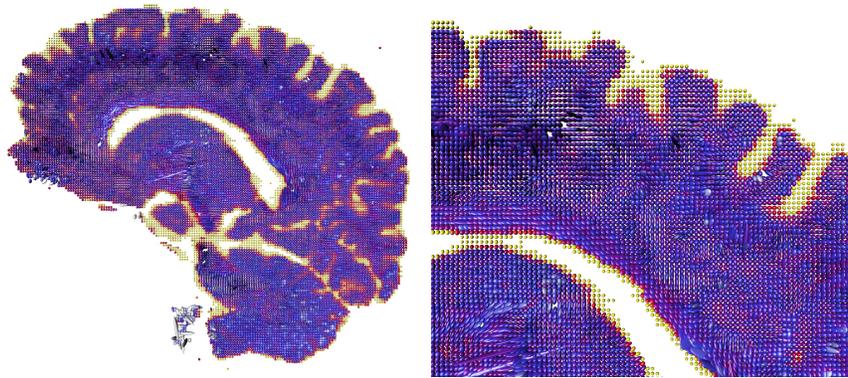


Fig. 6. Metric ellipsoids applied to a slice of the human brain.

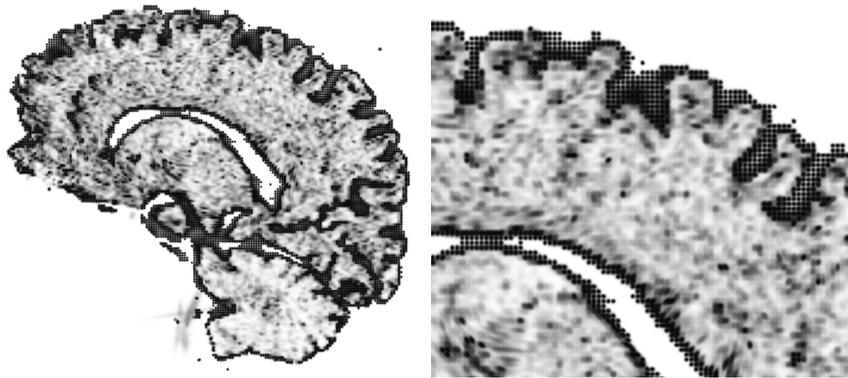


Fig. 7. Tensor glow technique applied to a slice of the human brain.

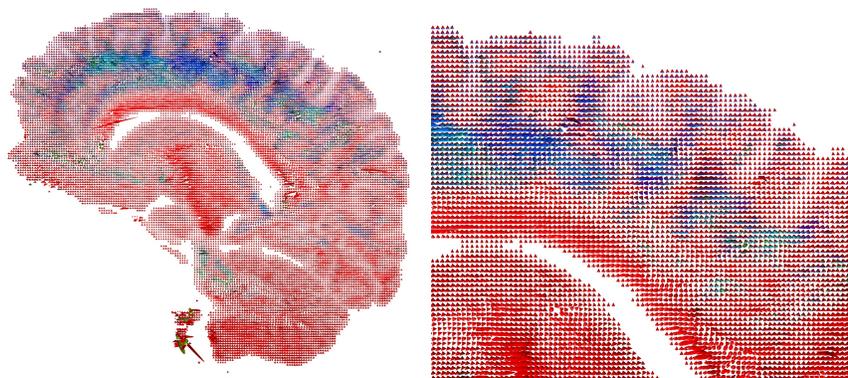


Fig. 8. Tensor cones applied to a slice of the human brain.

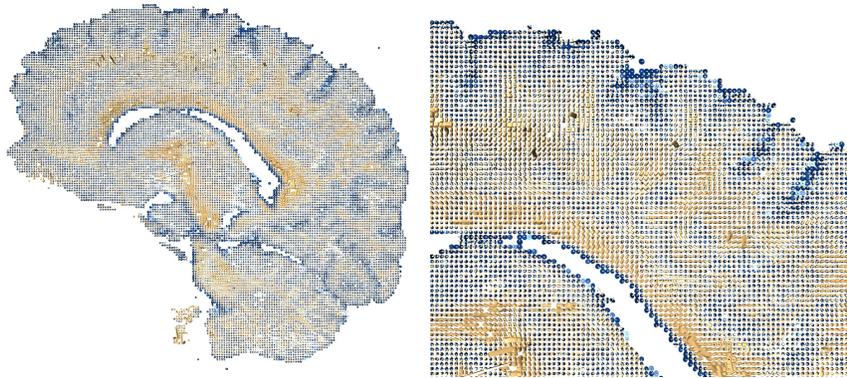


Fig. 9. Haber glyphs applied to a slice of a human brain.

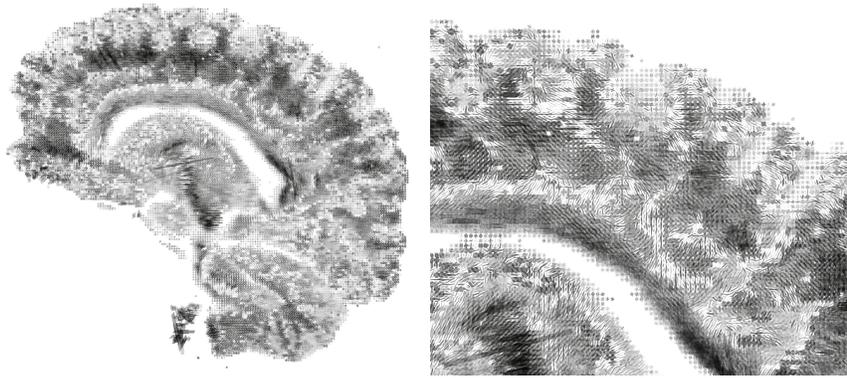


Fig. 10. Tensor schlieren applied to a slice of a human brain.

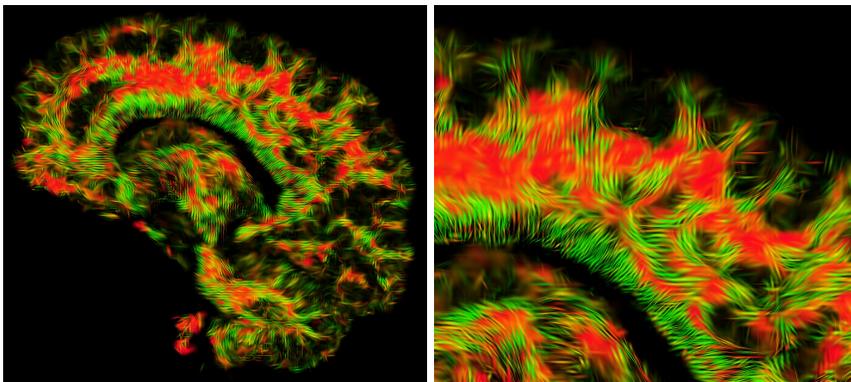


Fig. 11. Tensor splats applied to a slice of a human brain.

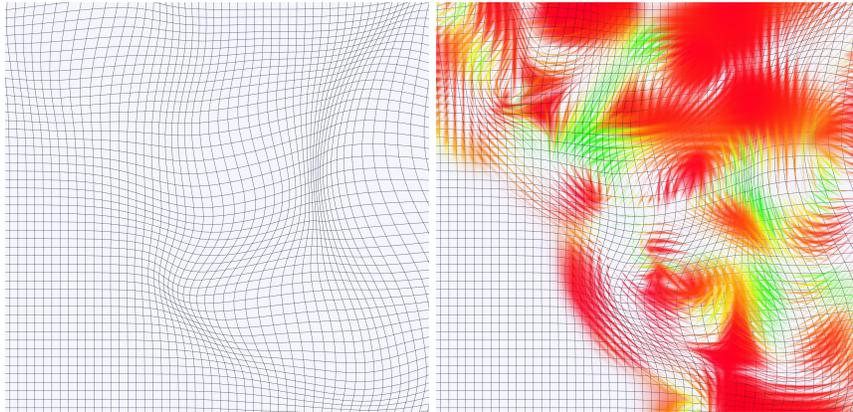


Fig. 12. Distortion of a uniform grid by a displacement vector field (left) and visualization of the induced metric tensor field (right).

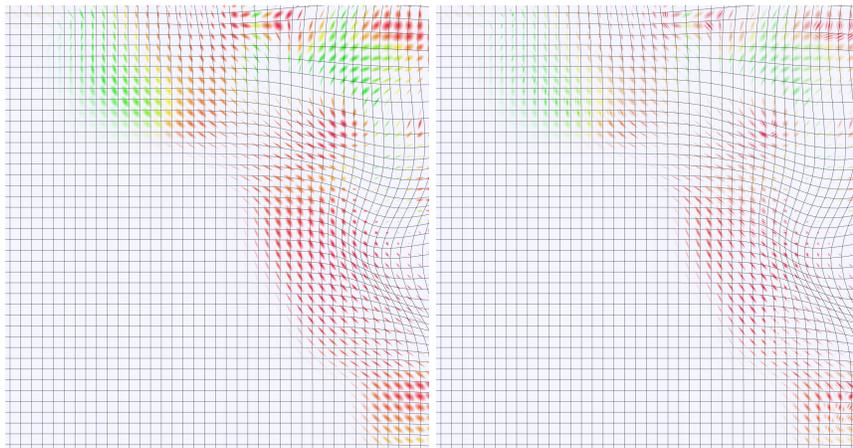


Fig. 13. Detail of the distortion tensor field.

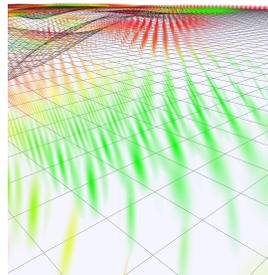


Fig. 14. 3D zoom onto the region of high linearity in Fig. 13, revealing high grid stretching perpendicular to the selected grid plane.