Tensor Field Visualization Using a Metric Interpretation

Ingrid Hotz[1], Louis Feng[1], Hans Hagen[2], Bernd Hamann[1], and Kenneth Joy[1]

[1] Institute for Data Analysis and Visualization, (IDAV), Department of Computer Science, University of California, Davis CA 95616, USA
[2] Visualization and Graphics Group, Technical University of Kaiserslautern, Kaiserslautern, Germany

Summary. This chapter introduces a visualization method specifically tailored to the class of tensor fields with properties similar to stress and strain tensors. Such tensor fields play an important role in many application areas such as structure mechanics or solid state physics. The presented technique is a global method that represents the physical meaning of these tensor fields with their central features: regions of compression or expansion. The method consists of two steps: first, the tensor field is interpreted as a distortion of a flat metric with the same topological structure; second, the resulting metric is visualized using a texture-based approach. The method supports an intuitive distinction between positive and negative eigenvalues.

1 Introduction

Since the physical interpretation of mathematical features of tensor fields is highly application-specific it is important that visualization techniques are closely driven by the special application. In this chapter, we focus on symmetric tensor fields of second order that are similar to stress strain tensor fields, or the symmetrical part of the gradient tensor. These tensor fields are characterized by the property that they have positive and negative eigenvalues. The sign of the eigenvalues indicates regions of expansion and compression, and it is therefore of special interest. To understand the field behavior, it is important to express these features in an intuitive way. The underlying idea of our visualization method is to transform the tensor field into a metric. This metric is represented using a texture that is aligned to the eigenvector fields, similarly to line integral convolution (LIC) [CL93, SH95]. The eigenvalues are included using the free parameters in the texture generation: the convolution filter length, and parameters of an input noise texture. This approach leads to a fabric-like texture that is dense in regions of compression and sparse in regions of expansion.

2 Related Work

Even though several good visualization techniques exist for tensor fields, they only cover a few specific applications. Many of these methods are extensions of vector field visualization methods, which focus on a technical generalization without providing an intuitive physical interpretation of the resulting images. They often concentrate on the representation of eigendirections and neglect the importance of the eigenvalues. Therefore, in many application areas traditional two-dimensional plots are still used, which represent the interaction of two scalar variables.

One way to represent a tensor field is based on using icons. They illustrate the characteristics of a field at some selected points (see, for example, [Hab90, KGM95, LW93]). Even though these icons represent the tensor value at one point well they fail to provide a global view of the tensor field. A more advanced but still discrete approach uses hyperstreamlines. This approach is strongly related to streamline methods used for vector fields. They were introduced by Delmarcelle and Hesselink [DH92] and have been utilized in a geomechanical context by Jeremic et al. [JSchF02]. Given a point in the field, one eigenvector field is used to generate a vector field streamline. The other two eigendirections and eigenvalues are represented by the cross section along the streamline. This method extracts more information than icons, but it still leaves the problem of choosing appropriate seed points to the user. Thus, both methods have limited usage in exploration complete data sets and are limited to low-resolution due to cluttering.

To generate a more global view, a widely accepted solution for vector fields is the reduction of the field to its topological structure. These methods generate topologically similar regions that lead to a natural separation of a field domain. The concept of topological segmentation was also applied to two-dimensional tensor fields [HD95]. The topological skeleton consists of degenerated points and connecting separatrices. Degenerated points are where the tensor has multiple eigenvalues and the eigenvectors are not uniquely defined. Although the tensor field can be reconstructed on the basis of its topological structure, physical interpretation is difficult.

Following an approach of Pang et al. [BP98, ZP02] a tensor field is considered as a force field that deforms an object placed inside it. The local deformation of probes, such as planes and spheres, illustrate the tensor field. This method only displays a part of the information because it reduces the tensor field to a vector field. To avoid visual clutter only a small number of probes can be included in one picture Zheng et al. [ZPa03] extended this method by applying it to light rays that are bended by the local tensor value.

Another class of visualization methods provides a continuous representation, based on textures. The first one to use a texture to visualize a tensor field in a medical context were Ou and Hsu [OH01]. An approach based on the adaptation of LIC to tensor fields by Zheng et al. [ZP03]. Here, a white-noise texture is blurred according to the tensor field. In contrast to LIC images,

the convolution filter is a two-dimensional or three- dimensional volume determined by the local 2D or 3D tensor field respectively. This visualization is especially good for showing the anisotropy of a tensor field. However, one problem of this method is the integration of the sign of the eigenvalues. Points with the same eigenvalues but with opposite sign are illustrated as isotropic.

There exist some other techniques designed especially for the visualization of diffusion tensors that only have positive eigenvalues. But they are not appropriate for stress, strain or gradient tensor fields.

3 Metric Definition

To motivate our approach we discuss an example for the kind of tensor fields we are interested in. These are stress tensor fields and gradient tensor fields whose behavior is very similar, as a stress tensor is often computed as gradient of a virtual displacement field. It can be observe that for gradient fields or stress and strain tensors, positive eigenvalues lead to a separation of particles or expansion of a probe. Eigenvalues equal to zero imply no change in distances, and negative eigenvalues indicate a convergence of the particles or compression of the probe.

For the symmetric part of a gradient tensor $\mathbf{S}$ of a vector field $\mathbf{v} = (v_1, v_2, v_3)$ with $s_{ij} = \frac{1}{2}(v_{i,j} + v_{j,i})$ this behavior is expressed by equation (1). Here, $v_{i,j}$ denotes the partial derivative of the ith component of $\mathbf{v}$ with respect to coordinate x_j.

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathrm{d}s^2) = \sum_{i,k=1}^{3} s_{ik}\,\mathrm{d}x_i\,\mathrm{d}x_k = \sum_{j=1}^{3} \lambda_j\,\mathrm{d}u_j^2. \tag{1}$$

Here, $\mathrm{d}s = (\mathrm{d}x_1, \mathrm{d}x_2, \mathrm{d}x_3)$ and $\mathrm{d}s^2$ is the quadratic distance of two neighboring points, $\lambda_j, j = 1, 2, 3$ are the eigenvalues of $\mathbf{S}$, and $\mathrm{d}u_j$ are the components of $\mathrm{d}x$ corresponding to the eigenvector basis $\{\mathbf{w}_j, j = 1, 2, 3\}$. If we focus on just one eigendirection $\mathbf{w}_i$, the change of $\mathrm{d}s^2$ is defined by the corresponding eigenvalue λ_i:

$$\lambda_i > 0 \rightarrow \frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{d}s^2 > 0, \quad \lambda_i = 0 \rightarrow \frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{d}s^2 = 0, \quad \lambda_i < 0 \rightarrow \frac{\mathrm{d}}{\mathrm{d}t}\,\mathrm{d}s^2 < 0. \tag{2}$$

A similar behavior can be observed for the deformation of a probe in a stress field (see Figure 1).

Considering a time-independent vector field, a formal integration of Equation (1) results in the following expression for $\mathrm{d}s^2$:

$$\mathrm{d}s^2(t) = \mathrm{d}s^2(0) + \sum_{ik}(s_{ik} \cdot t)\,\mathrm{d}x_i\,\mathrm{d}x_k. \tag{3}$$

Using $\mathrm{d}s^2(0) = a \cdot \sum_i \mathrm{d}x_i\,\mathrm{d}x_i$ we obtain:

$$\mathrm{d}s^2(t) = \sum_{ik} \underbrace{(a\delta_{ik} + s_{ik} \cdot t)}_{=:\, g_{ik}} \mathrm{d}x_i \, \mathrm{d}x_k, \tag{4}$$

where δ_{ik} is the Kronecker-delta. The tensor $\mathbf{g}$ with components $g_{ik} = a\delta_{ik} + s_{ik} \cdot t$ can be interpreted as a time-dependent metric of the underlying parameter space D. The constant a plays the role of a unit length, and t is a time variable that can be used as a scaling factor. This metric definition is the basis of our tensor field visualization method.

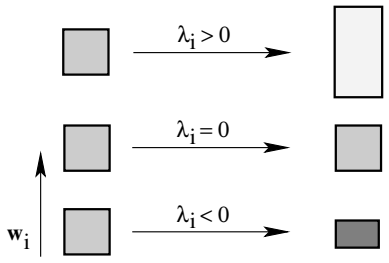


Fig. 1. Deformation of a unit probe under influence of a stress tensor in direction of eigenvector $\mathbf{w}_i$. Eigenvalues larger than zero correspond to a tensile, and eigenvalues smaller than zero to a compressive force in the direction of the eigenvector.

3.1 The Transformation

Based on the observations made in Section (3), we define a transformation of the tensor field into a metric. We do not exactly follow the motivating Equation (4) but use a more flexible approach.

Let $\mathbf{T}$ be a tensor field defined on a domain D. The tensor at a point $P \in D$ is given by $\mathbf{T}(P)$. For each point P, the tensor $\mathbf{T}(P)$ is mapped to a metric tensor $\mathbf{g}(P)$ describing the metric in P. In the most general form, the assignment is achieved by the following three steps:

1. Diagonalization of the tensor field:
   Switching from the original coordinate basis to the eigenvector basis $\{\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3\}$, we obtain a diagonal tensor $\mathbf{T}'$ having the eigenvalues of $\mathbf{T}$ on its diagonal:
   $$\mathbf{T} \mapsto \mathbf{T}' = U \cdot \mathbf{T} \cdot U^T = \mathrm{diag}\,(\lambda_1, \lambda_2, \lambda_3)\,, \tag{5}$$
   where U is the diagonalization matrix.

2. Transformation and scaling of the eigenvalue, to define the metric $\mathbf{g}'$ according to the eigenvector basis:
   $$\mathbf{T}' \mapsto \mathbf{g}' = \mathrm{diag}\,(F(\lambda_1), F(\lambda_2), F(\lambda_3))\,, \tag{6}$$
   where $F : [-\lambda_{max}, \lambda_{max}] \rightarrow I\!R^+$ is a positive monotone function, with $\lambda_{max} = max\{|\lambda_i(P)|; P \in D, i = 1, 2, 3\}$.

3. Definition of the metric g in the original coordinate system by inverting the diagonalization defined in Equation (5):

$$\mathbf{g} = U^T \cdot \mathbf{g}' \cdot U. \tag{7}$$

If the mapping F is linear, the three steps can be combined into one step, and F can be applied to the tensor components, independently of the chosen basis. The resulting metric $\mathbf{g}$ has the following properties:

- It is positive definite and symmetric.
- Its eigenvector field corresponds to the original eigenvector field of $\mathbf{T}$. Thus, the tensor field topology in the sense of Delmarcelle et al. [HD95] is preserved.
- Its eigenvalues are given by $F(\lambda_j)$. Positive eigenvalues are mapped to values greater than a, negative eigenvalues to values smaller than a but larger then zero.The zero tensor is mapped to a multiple of the unit matrix.
- Since the transformation is invertible, we get a one-to-one correspondence of the metric and the tensor field is given.

3.2 Examples for Transformation Functions F:

In this paragraph we suggest some explicit definitions for the function F. Except from the first example all these functions are nonlinear and therefore cannot be directly applied to the tensor components. The functions we discuss can be classified in two groups:

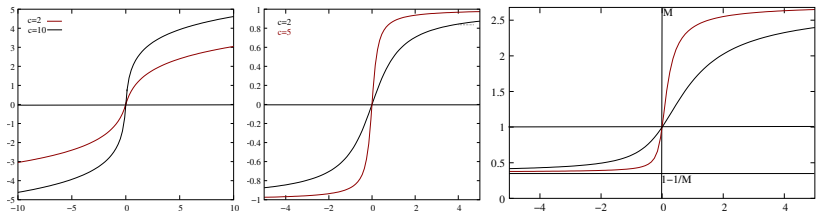
1. Anti-symmetric Treatment of the Eigenvalues
To underline the motivation defined by (4), we can define the transformation function as:

$$F(\lambda) = a + \sigma f(\lambda). \tag{8}$$

Here, $a = F(0)$ defines the unit length, and $\sigma \neq 0$ is an appropriate scaling factor that guarantees that the resulting metric is positive definite. The function $f : I\!R \rightarrow I\!R$ is a monotone function with $f(0) = 0$. If we want to treat positive and negative Eigenvalues symmetrically it is $f(-\lambda) = -f(\lambda)$. From the large class of functions satisfying this condition we have considered three examples:

a. Identity: $f = id$, $f(\lambda) = \lambda$
Since f is linear, the metric g is defined by $g_{ij} = F(t_{ij}) = a + \sigma \cdot t_{ij}$. This equation corresponds exactly to our motivating Equation (4), where σ plays the role of the time variable t. With $\sigma < a/\lambda_{max}$ we can guarantee that the metric is positive definite.

b. Anti-symmetric logarithmic function:
To emphasize regions where the eigenvalues change sign one can choose a function f with a larger slope in the neighborhood of zero.



(a) (b) (c)

Fig. 2. Figures (a) and (b) are examples for an anti-symmetric transformation function f. (a) Logarithmic function; (b) arc-tangent for two different slopes for $\lambda = 0$. Figure (c) is an example of a non-symmetric transformation function F for two different slopes at the origin.

$$f(\lambda; c) = \begin{cases} \log(c \cdot \lambda + 1) \text{ for } \lambda \geq 0 \\ -\log(1 - c \cdot \lambda) \text{ for } \lambda < 0 \end{cases}. \tag{9}$$

If we require $\sigma < a/\log(c \cdot \lambda_{max} + 1)$ the resulting metric is positive definite.

c. Asymptotic function:
A function where the limitation of the scaling factor σ is independent of λ_{max} is

$$f(\lambda; c) = \arctan(c \cdot \lambda), \tag{10}$$

with $\sigma < 2a/\pi$. For both functions, the constant c controls the "sharpness" at the zero crossing. For higher values of c the function becomes steeper, see Figure 2.

2. Non-symmetric function:
As the visual perception of texture attributes is nonlinear, an anti-symmetric approach is not always the best choice. An alternative that takes care of this aspect is defined by the class of functions $F[-\lambda_{max}, \lambda_{max}] \rightarrow [\frac{a}{M}, a \cdot M]$, where

$$F(-\lambda) = \frac{a^2}{F(\lambda)}. \tag{11}$$

The constant a defines again the unit, $a\dot{M}$ the maximum, and $\frac{a}{M}$ the minimum value for F satisfying $M > 1$. Functions with this property can be obtained by using anti-symmetric functions f as exponent:

$$F(\lambda) = a \cdot exp(\sigma \cdot f(\lambda)) \text{ where } f(-\lambda) = -f(\lambda). \tag{12}$$

An example for such a function with $a = 1$ is $F(\lambda; c, \sigma) = exp(\sigma \arctan(c \cdot \lambda))$. The constant c determines the slope of the function in the origin, see Figure 2. The second class of functions produces much better results because the differences in the density of the resulting structure is more obvious. The special choice of the function f does not have a significant influence on the result. Another advantage of this class of functions F is that the resulting metric is

always positive definite and therefore, the scaling factor σ is not limited. By an animation of this parameters we can enhance the impression of stretching and compression.

3.3 Visualization

We now have transformed the problem of visualizing a tensor field to the problem of visualizing an abstract metric. One way to solve this problem is an isometric embedding of the metric [Hot02]. The disadvantage of this approach is that it is restricted to two dimensions, and its existence is only guaranteed locally. In general, several patches are needed to cover a field's entire domain. Since we want to produce a global representation of a field we decided to follow a different approach: Our basic idea is to use a texture that resembles a piece of fabric to express the characteristic properties of the metric. The texture is stretched or compressed and bended according to the metric. Large values of the metric, which indicate large distances, are illustrated by a texture with low density or a stretched piece of fabric. We use a dense texture for small values of the metric. One can also think of a texture as probe inserted into the tensor field.

We generate the texture using LIC, a very popular method for vector field visualization. LIC blurs a noise image along the vector field or integral curves. Blurring results in a high correlation of the pixel along field lines, whereas almost no correlation appears in direction perpendicular to the field lines. The resulting image leads to a very effective depiction of flow direction everywhere, even in a dense vector field. LIC was introduced in 1993 by Cabral and Leedom [CL93]. Since the method was introduced, several extensions and improvements were made to make it faster [SH95] and more flexible.

We compute one LIC image for every eigenvector field to illustrate the eigendirections of the tensor field. For the integration of the integral curves we use a Runge-Kutta method of fourth order, the LIC image is computed using Fast-LIC as proposed in [SH95]. In each LIC image the eigenvalues of every eigenvector field are used to define the free parameters of the underlaying noise image and the convolution. Finally, we overlay all resulting LIC images to obtain the fabric-like texture.

Input Noise Image

We use the free parameters of this input image to encode properties of the metric. Three basic parameters are changed according to the eigenvalues. They are: density, spot size, and color intensity of the spots. Considering these parameters, the standard white-noise image is the noise image with maximum density, minimal spot size, and constant color intensity. It allows one to obtain a very good overall impression of the field; its resolution is only limited by pixel size. Unfortunately, it is not flexible enough to integrate the eigenvalues which represent fundamental field properties besides the directions. For this reason,

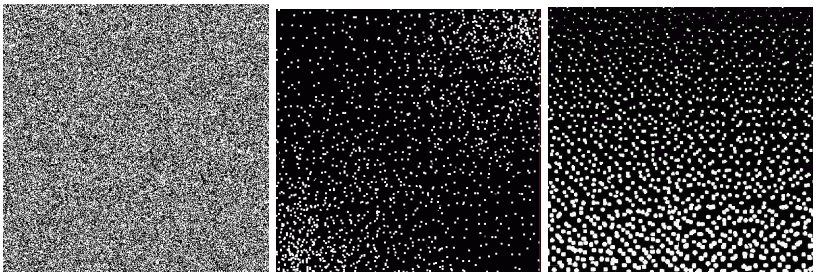
we use sparse noise input images, with lower density and larger spot size even if we obtain a lower resolution. Some examples for different input images with changing density and spot size are shown in Figure 3. The connection of these parameters to the eigenvalues is explained in the following paragraphs:

Density For each direction field $\mathbf{w}_i$, we define a specific density d_i depending on the orthogonal eigenvalues. A compression orthogonal to fibers leads to increasing density, and an expansion to decreasing density. For two-dimensional textures this approach leads to the following definition of a one-dimensional density d_i [spots/cm]:

$$d_i(\lambda) = d_0 \cdot \frac{1}{F(\lambda_j)}, \text{ with } j = \begin{cases} 2 \text{ if } \ i = 1 \\ 1 \text{ if } i = 2|; , \end{cases} \tag{13}$$

where F is defined by Equation (6), and d_0 defines the "unit-density," $d(0) = d_0/F(0)$. In three dimensions, we have two orthogonal eigenvalues and thus obtain a direction-dependent density $d_{i,j}$ for each direction $\mathbf{w}_j$:

$$d_{i,j}(\lambda) = d_0 \cdot \frac{1}{F(\lambda_j)}. \tag{14}$$



(a) (b) (c)

Fig. 3. Example for different input images. (a) White noise image with maximum resolution; (b) spot noise image with changing density; (c) spot noise image with changing spot size.

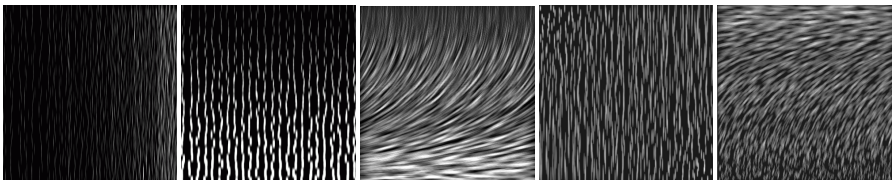
Spot Size Increasing the radius of the underlying noise image leads to thicker; decreasing the radius leads to thinner fibers. This value is controlled by the orthogonal eigenvalues. In three dimensions, we define ellipsoids with three different diameters according to the three eigenvalues:

$$r_{i,j} = \frac{r_0}{d_{i,j}}. \tag{15}$$

Convolution Length The defined noise image only uses the eigenvalues orthogonal to the actual eigendirection field. A stretching or compressing in

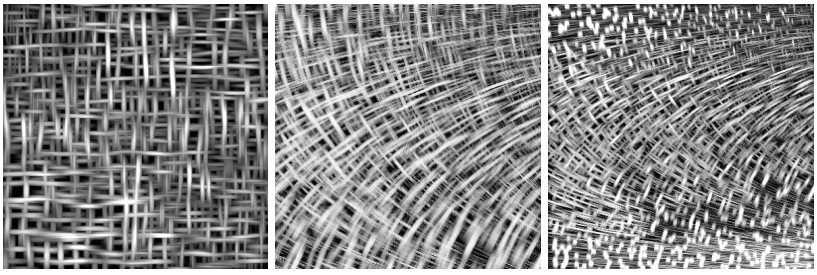
the direction of the integral lines changes the length of the fibers. Fiber length is directly correlated to the length of the convolution filter l_i, i.e.,

$$l_i = l_0 \cdot F(\lambda_i). \tag{16}$$



(a) (b) (c) (d) (e)

Fig. 4. Effect of changing image parameters for one eigenvector field of different simple synthetic tensor fields. In (a)-(c), only the input image is changed corresponding to the eigenvalues of the orthogonal eigenvector field; (a) change of density; (b) change of spot size; (c) change of density and spot size. Images (d) illustrates the effect of changing the convolution length, where the parameters of the input noise image are constant. Image (e) shows a combination of the three parameters (density, spot size, and convolution length).



(a) (b) (c)

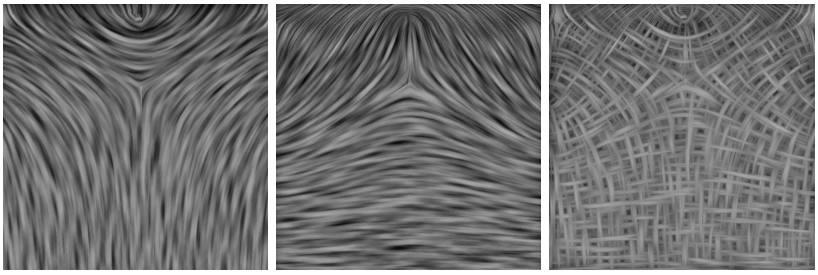
Fig. 5. Combination of two eigenvector fields, each representing both eigenvalues. In (a) and (b), only density and spot size are changed; (c) shows a combination of the three parameters.

Color and Color Intensity In addition to these three "structure" parameters, color intensity can be used to enhance the impression of compression and stretching. We use red for compression and green for tension. We apply a continuous color mapping from red for the smallest negative eigenvalues, white for zero eigenvalues, and green for positive eigenvalues. The definition of the different parameters for three dimensions is summarized in Table 1.

4 Results and Conclusions

We have evaluated our method using synthetic and real data sets. Simple tensor fields, where the eigenvector fields are aligned to the coordinate axes, have allowed us to validate the effect of changing texture parameters. We have obtained similar results for datasets where the eigenvector fields are rotated by 90 degrees. Results where only one eigenvector field is used are shown in Figure 4. Images for the same datasets showing both eigendirections are shown in Figure 5. We have used different input textures and parameter mappings.

The next examples are results for simulated finite element data sets of the stress field resulting from applying different load combinations to a solid block. These datasets are well-studied and therefore appropriate to evaluate our method. For the simulation, a ten-by-ten-by-ten gird had been used. The tensor field resulting from the simulation is continuous inside each cell, but not on cell boundaries. This fact can be observed in our images. Figure 6 and Figure 7 (see color plates) show different slices of the three-dimensional dataset from a single point load. Figure 8 (see color plates) represents a block where two forces with opposite sign were applied. These images provide a good visual segmentation of regions of compression and expansion.



(a) (b) (c)

Fig. 6. Images showing a yz-plane slice of single top-load data set, where a force is applied in z-direction. (a) and (b) illustrate the two eigenvector fields separately; in (c) they are overlaid. In all images, spot size and density are changed according to eigenvalues.

The interpretation of a tensor field as a distortion of a flat metric can be used to produce a visualization based on the real physical effect of the tensor field. The distortion of the texture according to the metric supports a flexible representation of two-dimensional slices of a tensor field, which is easy to understand. An extension to three dimensions is possible but there is still the problem of cluttering which must be solved.

free parameters		eigenvector field $i=1$	$i=2$	$i=3$
density value	$d_{i,j}$	$\frac{1}{\lambda_2}$	$\frac{1}{\lambda_1}$	$\frac{1}{\lambda_1}$
	$d_{i,k}$	$\frac{1}{\lambda_2}$	$\frac{1}{\lambda_1}$	$\frac{1}{\lambda_1}$
color intensity	I_i	$\frac{1}{\lambda_1}$	$\frac{1}{\lambda_2}$	$\frac{1}{\lambda_3}$
convolution length	l_i	λ_1	λ_2	λ_3
spot diameter	$r_{i,j}$	λ_2	λ_3	λ_1
	$r_{i,k}$	λ_3	λ_1	λ_2

Table 1. Assignment of eigenvalues to free parameters for a three-dimensional texture.

References

[BP98] Ed Boring and Alex Pang. Interactive Deformations from Tensor Fields. Proceedings of the Visualization '98 Conference, 1998, pages 297-304.

[CL93] Brian Cabral and Leith Leedom. Imaging Vector Fields Using Line Integral Convolution. In SIGGRAPH '93 Conference Proceedings, 1993, pages 263-272.

[DH92] Thierry Delmarcelle and Lambertus Hesselink. Visualization of second order tensor fields and matrix data. Proceedings of the Visualization '92 Conference, 1992, pages 316-323.

[Hab90] Robert B. Haber. Visualization Techniques for Engineering Mechanics. In Computing Systems in Engineering, 1990, pages 37-50.

[HD95] Lambertus Hesselink, Thierry Delmarcelle and James L. Helman . Topology of Second-Order Tensor Fields. In Computers in Physics, 1995, pages 304-311.

[Hot02] Ingrid Hotz. Isometric Embedding by Surface Reconstruction from Distances. Proceedings of the IEEE Visualization '02 Conference, 2002, pages 251-258.

[HFHHJ04] Ingrid Hotz, Louis Feng, Hans Hagen, Bernd Hamann, Boris Jeremic, and Kenneth Joy. Physically Based Methods for Tensor Field Visualization. Proceedings of the IEEE Visualization '04 Conference, 2004, pages 123-130.

[JSchF02] Boris Jeremic, Gerik Scheuermann and Jan Frey et al. Tensor Visualization in Computational Geomechanics. In International Journal for Numerical and Analytical Methods in Geomechanics, 2002, pages 925-944.

[KGM95] Ron D. Kriz, Edward H. Glaessgen and J.D. MacRae. Visualization Blackboard: Visualizing gradients in composite design and fabrication. IEEE Computer Graphics and Applications, vol 15, 1995, pages 10-13

[LW93] W. C. de Leeuw and J. J. van Wijk. A Probe for Local Flow Field Visualization. Proceedings of the Visualization '93 Conference, 1993, pages 39-45.

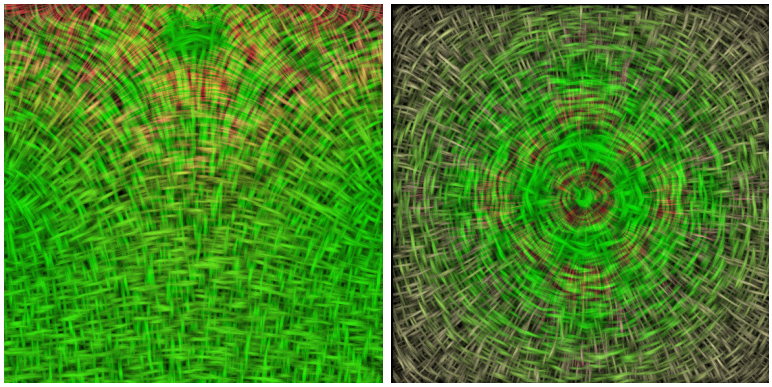
[OH01] J. Ou and E. Hsu. Generalized Line Integral Convolution Rendering of Diffusion Tensor Fields. Proceedings of the 9th Scientific Meeting and Exhibition of the International Society for Magnetic Resonance in Medicine (ISMRM), page 790, 2001.

[SH95] Detlev Stalling and Hans-Christian Hege. Fast and Resolution Independent Line Integral Convolution. In SIGGRAPH '95 Conference Proceedings, 1995, pages 149-256.

[ZP02] Xiaoqiang Zheng and Alex Pang. Volume Deformation For Tensor Visualization. Proceedings of the Visualization '02 Conference, 2002, pages 379-386.

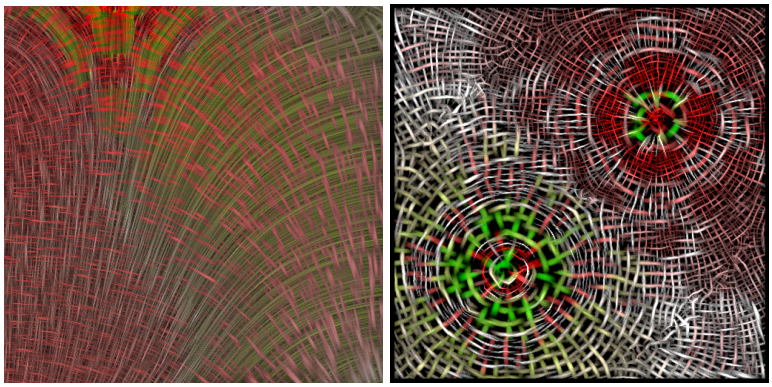
[ZPa03] Xiaoqiang Zheng and Alex Pang. Interaction of Light and Tensor Field. Proceedings of VISSYM '03, 2003, pages 157-166.

[ZP03] Xiaoqiang Zheng and Alex Pang. HyperLIC, Proceedings of the IEEE Visualization '03 Conference, 2003, pages 249–256.



(a) (b)

Fig. 7. This figure shows a single-top-load. Spot size and density of the input images are adapted to the corresponding eigenvectors. Red shows regions of compression, green expansion according the respective eigenvector field: the images are planar slices along the (a) yz-plane and (b) xy-plane slice orthogonal to the force.



(a) (b)

Fig. 8. The images represents a yz-plane (a) and xz-plane (b) slice of a two-force dataset. (a): In the lower-left corner we see a region of compression, a result mainly due to the pushing force on the left; in the upper-right corner expansion dominates as a result of the right pulling force. (b): The left circle corresponds to the pushing and the right to the pulling force. The fluctuation of the color is a result of the low resolution of the simulation.