Jirka,T., Skala,V.: Stress tensor field visualization, EC&I 2004, 2004.

FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS
OF
THE TECHNICAL UNIVERSITY OF KOŠICE

PROCEEDINGS
of
THE SIXTH INTERNATIONAL SCIENTIFIC CONFERENCE

ELECTRONIC COMPUTERS and INFORMATICS ECI 2004

The conference is organized by
Department of Computers and Informatics

In co-operation with:
Slovak Society for Applied Cybernetics and Informatics
Czech and Slovak Society for Simulation

Sponsoring by:
SIEMENS Program & System Engineering s.r.o. Bratislava
Ing. Milan Roško, TEGH, Information Technology Dep. Toronto, Kanada
ZTS Výskumno-vývojový ústav a.s., Košice

Editors: Štefan Hudák, Ján Kollár

September 22-24, 2004
Košice - Herľany
Slovakia

Editors' Note

This publication was reproduced by the photo process, using the manuscripts and soft copies supplied by their authors. The layout, figures, and tables of some papers did not comform exactly with the standard requirements. In some cases the layout of the manuscripts were rebuilt. All mistakes in manuscripts there either could not be fixed, or could not be checked completely by reviewers and there are a responsibility of authors. The readers are therefore asked to excuse any deficiencies in this publication which may have arisen from the above causes.

Proceedings of the Sixth International Scientific Conference Electronic Computers and Informatics ECI 2004
ISBN 80-8073-150-0
Editors: Štefan Hudák, Ján Kollár
September 22-24, 2004, Košice - Herľany, Slovakia

© Layout & Design: J. Bača, D. Mihalyi, D. Sobotová

Additional copies can be obtained from:
Department of Computers and Informatics of FEI,
The University of Technology Košice,
Letná 9, 04200 Košice, Slovakia
Phone: +421-55-63 353 13
Fax: +421-55-6330115
E-mail: Stefan.Hudak@tuke.sk, Jan.Kollar@tuke.sk

Proceedings of
the Sixth International Scientific Conference
Electronics Computer and Informatics ECI 2004

VIENALA Press
Moldavská 8/A, 040 01 Košice
Edition: 55
September 2004

ISBN 80-8073-150-0

Stress Tensor Field Visualization

Tomas Jirka, Vaclav Skala
Department of Computer Science and Engineering,
University of West Bohemia in Pilsen, Univerzitní 22, 306 14 Plzeň, Czech Republic
E-mail: {tjirka|skala}@kiv.zcu.cz

Abstract

General methods used for visualizing second order tensor fields may fail to provide the user with some of the important features of the underlying physical phenomenon. Due to the high amount of information contained in such data it is hardly possible to display it completely. Therefore, we focus our attention to stress in material as one particular physical problem, whose nature requires symmetric second order tensor field for its description. Our article brings a brief introduction to the mathematical and physical aspects of this issue followed by a description of our visualization method based on surface extraction. The aim is to display certain part of the tensor information in a global and continuous form.

1. Introduction

To describe certain complex natural phenomena as, for instance, stress and strain in the material science, diffusion of water molecules in biological tissues important for the medical science or rate of strain and viscous stresses in studying fluid flows, second order tensors need to be employed. As a second order tensor field in N-dimensional space contains the same amount of information as N or N^2 vector or scalar fields respectively, visualization of all the available information, yet avoiding visual clutter, can hardly be reached. Each of the existing approaches, therefore, concentrates on certain feature, while necessarily sacrificing other. This means, for example, visualizing most of the tensor information in several discrete locations at the expense of continuity and global scope or, on the other hand, providing continuous display of only partial information. We focus our attention to the later case intending to develop a method, which will provide scientists with a continuous and global scope view at the information they find important. We aim it at stress tensor data, but it should be applicable to other second order tensor fields as well. However, the physical interpretation will differ.

The theoretical background will be given in chapters 2 and 3, which bring an overview of the existing visualization approaches and the definition of a stress tensor respectively. In chapters 4 and 5 we introduce our approach and provide the reader with the visual outcomes. In chapter 6 our future plans will be discussed.

2. Existing Methods

In this section, existing approaches to second order tensor field visualization will be outlined. As there is not enough space for an extensive study of this broad topic, we will concentrate on the most well-known approaches, which are related to our own method described later in chapter 4.

2.1 Color Coding

Methods based on color coding try to offer an aid for exploring the tensor data by displaying its individual scalar components. More precisely, the scalar values are mapped to color and then applied on planar slices through the volume. Since three dimensional second order tensors consist of nine independent scalar components, it is necessary to utilize nine images of a slice, each corresponding to one component. These colored slice images are usually arranged to a 3 x 3 grid resembling the matrix notation of a tensor so that the user automatically associates individual images with the relevant components. [4]

As all the scalar values are depicted, the entire information available in the selected slice through the tensor field is included in the final picture. Yet, it is too demanding for human brain to acquire information from multiple images simultaneously and combine it together to construct a comprehensible notion of the phenomenon behind. Although not useless, this method is far from providing intuitively interpretable results.

Another possibility is to use color coding based on the eigenvalues of the tensors, which produces only three images expressing certain geometric information (see [5] for illustration). Yet, it is still difficult to understand the behavior of the data values and rather insufficient for making analyses.

2.2 Glyphs Rendering

Methods falling to this category depict data in discrete locations of the dataset by mapping the values on the properties of a simple geometrical object such as shape, size and color. [7] These objects are usually called glyphs or icons and may take various forms.

Selection of the geometrical object to use must reflect the needs of the visualization. In flow exploration some kind of ring complemented 3D arrow is often the choice. The most frequently encountered shape, however, is probably an ellipsoid, whose axes are parallel to the eigenvectors of the tensor matrix. Length of each axis then corresponds to the magnitude of the appropriate eigenvalue. The power of using such ellipsoids consists in two facts. First, its shape is simple thus making the information easy to understand. Second, the system of eigenvectors represents the situation at a specific location very concisely.

Ellipsoidal glyph as described above works well for symmetric second order tensors, where the eigenvectors are guaranteed to be orthogonal. In case of non-symmetric tensor fields one should rather use tripoids.

The discrete character of glyphs seems to be their main disadvantage. They only allow displaying information in separate locations, whose density, moreover, must not be too high. Otherwise, glyphs might overlap each other and cause visual clutter. To maintain visual clarity some ellipsoids might require rescaling or the tensors might ne be decomposed prior to the visualization i In that case, however, certain par magnitude information may be lost.

2.3 Hyperstreamlines

Unlike glyphs, tensor field lines hyperstreamlines [2] aim to pr continuous display. They were, in developed as a modification of orc streamlines known from vector fields.

Before the computation hyperstreamlines themselves, the syste eigenvectors are found. These vector sorted according to the correspo eigenvalues thus forming three vector one for each eigenvalue. One of these fields is then chosen for integ streamlines.

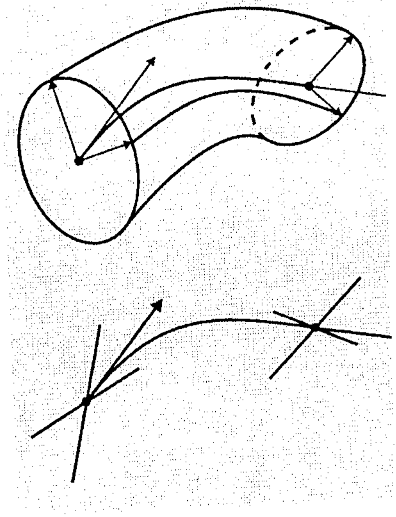


Figure 1: Sketch of a hyperstreamline fo symmetric (top) and non-symmetric (botto

Up to now the procedure d distinguish between symmetric anc symmetric tensor fields. At this however, similarly as in case of glyphs differences can be found in the hyperstreamlines for these two kinds (In the case of symmetric data with th eigenvectors being orthogona hyperstreamline is rendered as a tub elliptical cross-section, whose siz orientation is driven by the two remaining eigenvectors. In the other case, the non-symmetric information is represented by a system of ribbons along the hyperstreamline as illustrated in Figure 1.

Hyperstreamlines give a continuous notion of how the values in the tensor field evolve. To provide complete information, however, one needs to integrate along each of the three vector fields thus producing three images. All the three images must then be taken into account by the user. To avoid visual chaos, seeding must be considered carefully again.

2.4 Topological Approach

To a certain extent, this method is a solution for the hyperstreamline seeding strategy issue. The topological approach [3] first locates points, where at least two of the eigenvalues equal to each other. These locations are called degenerate points and may be classified according to the behavior of the tensor field in their surrounding as, so called, trisectors or wedge points. Degenerate points are then used as seeding locations for integrating hyperstreamlines. Together they express the tensor field's topology, which is, according to the authors, often simpler than that of a vector field.

2.5 Interactive Deformations

Boring and Pang [1] suggest visualizing symmetric second order tensor fields by letting the tensors deform a geometric object as, for instance, plane. First, the so called resolute vector of a tensor, which determines the tensor's impact on an interrogation object I with normal vector **n**, is computed at all points defining I using equation (1) bellow. Where the tensor is unknown, trilinear interpolation is used to approximate it and the resolute vector is then applied on the interrogation object. The displacement of point $I(\mathrm{x})$ of the interrogation object to a new position $O(\mathrm{x})$ is controlled by the following rule:

$$O(\mathbf{x}) = I(\mathbf{x}) + s[\mathbf{T}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})],$$

where **x** denotes position, s is a scale factor, $\mathbf{T}(\mathbf{x})$ tensor at **x** and $\mathbf{n}(\mathbf{x})$ is the user selected normal at the position of **x**. The product $\mathbf{T}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x})$ corresponds to the resolute vector. The deformed object illustrating the impact of the tensor field's influence is then visualized using appropriate visualization techniques.

This concept has been extended later in [9], where volume is deformed instead of the interrogation object.

3. Stress Tensor Definition and Description

The state of stress in location M is defined, if and only if stress on any facet passing through this location is given. [6] A stress vector **p** is then a function of M's position and a normal vector of the considered facet. We can write it as

$$\mathbf{p} = \mathbf{p}(\mathbf{r}, \mathbf{n}),$$

with **r** being the position vector and $\mathbf{n} = (n_1, n_2, n_3)$ a unit normal vector of the considered facet. As there is an infinite number of facets passing through M, this vector representation is unsuitable. Therefore, a tensor representation was derived. It starts from an elementary tetrahedron chosen in the vicinity of M, whose three edges are parallel to the three axis of the Cartesian coordinate system as depicted in Figure 2.

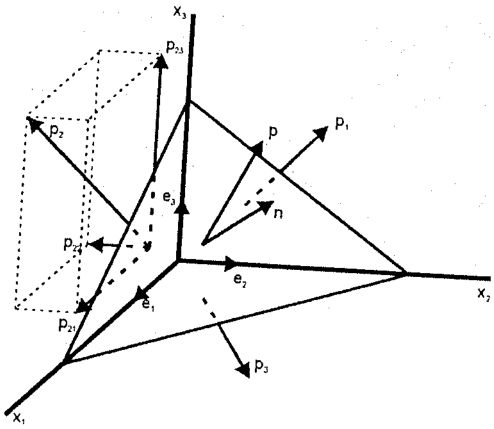


Figure 2: The elementary tetrahedron for stress tensor derivation

Employing the condition of balance of concerned forces, the following relation can be obtained

$$p_1\mathbf{e}_1 + p_2\mathbf{e}_2 + p_3\mathbf{e}_3 = \sum_{i=1}^{3}(p_{i1}\mathbf{e}_1 n_1 + p_{i2}\mathbf{e}_2 n_2 + p_{i3}\mathbf{e}_3 n_3)$$

where p_k and p_{ik} denote the components of vectors $\mathbf{p}$ and $\mathbf{p}_i$. The above equation can be rewritten as follows

$$p_k = \sum_i p_{ik} n_i \,.$$

The state of stress in location M is thus uniquely described by the nine values p_{ik}, which are independent of the chosen facet and together make up an object called *stress tensor*. The three components for $i = k$ represent the, so called, *normal stresses* on the three perpendicular facets at location M and they are referenced as σ_i in material science. The remaining components for $i \neq k$ represent the *shear stresses* on these facets and are referenced as τ_{ik}. The whole equation can be expressed in matrix notation as well

$$\mathbf{p} = \Sigma \cdot \hat{\mathbf{n}}\,, \qquad (1)$$

where $\Sigma = [p_{ik}]$ is a 3 x 3 matrix representing the stress tensor.

Moreover, the conditions of balance also imply that the stress tensor is symmetric, which is an important property for some of the visualization methods.

4. Our Approach

All the existing methods, we have found, visualize the information contained in a tensor field by addressing the query to display the stresses acting in the user defined region. Whether this region covers the whole dataset or just a segment of it and whether the visualization is continuous or not, represents two main aspects, which distinguish these approaches from each other.

4.1 Theory

In our work, however, we visualize the tensor field information the opposite way, answering the query to locate and display the region, where the user defined stresses act. In other words, we let users specify a stress vector they are interested in and see the region, where such stress appears within the tensor field dataset.

For this reason we first invert equation (1) to obtain

$$\mathbf{n} = \Sigma^{-1} \cdot \mathbf{p}\,, \qquad (2)$$

where $\mathbf{p}$ is the user defined stress vector. Equation (2) thus yields vector $\mathbf{n}$ that is normal to the facet, on which $\mathbf{p}$ acts. It is, however, necessary to note that in this case $\mathbf{n}$ may not be unit vector any more. Instead, its magnitude provides us with the information about the magnitude of the stress that is really present. In fact, if the magnitude of $\mathbf{n}$ equals to one, the actual stress magnitude matches the user specified request precisely, as can be verified by back substituting this vector to equation (1). If $|\mathbf{n}| > 1$, the actual stress parallel to the usei defined vector is weaker than the user definec one. In other words, if we normalize $\mathbf{n}$ anc back substitute it to (1) again, we will obtain a stress vector in the same direction as the use requested stress vector $\mathbf{p}$, but with lowe magnitude. Similarly, if $|\mathbf{n}| < 1$, the actua stress parallel to the user defined vector i stronger than the user requested.

Under certain conditions, matrix Σ^{-} may not exist. Such a case corresponds to th situation, when vectors $\mathbf{p}_i$ (see Fig. 1) ar coplanar or collinear and, therefore, the row of matrix Σ become linearly dependen Obviously, no solution exists if the us chooses a stress vector, which does not reclir in the same plane or line defined by the: vectors while, on the other hand, an infini number of solutions can be found for one th does. We then need to find out, if some these solutions is of unit length. Mathematic background of this issue can be found in [8].

The above procedure provides us wi the possibility to locate regions, where t stress in the requested direction is higher lower than the user defined one. We can th visualize the surface separating these regic as a kind of isosurface.

4.2 Implementation Issues

There are several implementation issues that deserve mentioning here. The first concerns matrix inversion. We use singular value decomposition as described in [8], which offers several advantages. First off, this procedure assures sufficient accuracy in locations with regular tensor matrix. Second, it provides good diagnostics in areas, where the tensor matrix becomes close to singular. Third, it allows us to treat both the above cases in a similar way. In other words, we can perform SVD of all the tensor matrices in advance, regardless from whether they are regular or not. Then, after the user specifies the stress vector of interest, it allows us to find a solution in the singular location as well, if one exists.

Another issue to be pointed out is the extraction of the separating surface. At present, we first compute the magnitude of **n** for all the nodes in the whole dataset and then extract an isosurface for an isovalue equal to one. To accomplish this, interpolation must be utilized. Mathematically, however, the succession of the operations of inversion, length computation and interpolation matters. That is, we do not get the same results as if we first interpolated the tensor matrix and only then computed the inverse and calculated the normal and its length. Thus we get an approximation of the separating surface. Comparing its accuracy to the case, when interpolation precedes inversion and normal length computation is one of our plans for the future work. We plan to utilize interval subdivision for this purpose.

5. Results

Although static pictures do not provide such a good notion of the stress behavior as may be obtained by interactively changing the requested stress vector, we try to illustrate the results of the method by Figure 3. As expected some of the extracted surface structures take rather complex shapes. Therefore, we use transparency in combination with stereo projection for the final visualization. Unfortunately, this is again impossible to present in a printed form.

6. Conclusion and Future Work

Our procedure, described in this article, provides us with the possibility to locate areas, where the user defined stress acts. We aim to subject the visualization of this extracted region to further investigation focusing on the possibility to complement it with as much additional relevant information as possible and reasonable. For this purpose, we might utilize some glyph rendering or color coding techniques.

We are also going to study the accuracy issue mentioned in the detail in the last paragraph of section 4.2.

7. Acknowledgements

At this point, we would like to thank Mr. Uwe Mahn from TU Chemnitz (Germany) and Mr. Bohuslav Masek from the Department of Material Science of the UWB (Czech Republic) for providing us with the tensor field data sets.

This project was supported by the Ministry of Education of the Czech Republic – project MSM 235200005.

References

[1] Boring, E., Pang, A.: Interactive Deformations from Tensor Fields, Proceedings IEEE Visualization 98, pages 297-304, 1998.

[2] Delmarcelle, T., Hesselink, L.: Visualizing Second-Order Tensor Fields with Hyperstreamlines, IEEE Computer Graphics and Applications, 13(4): 25-33, 1993.

[3] Hesselink, L., Levy, Y., Lavin, Y.: The Topology of Symmetric, Second-Order 3D Tensor Fields, IBEE. Trans. Vis&CG, pp. 1-11, Vol. 3, No. 1, 1997.

[4] Kindlmann, G., Weinstein, D.: Hue-Balls and Lit-Tensors for Direct Volume Rendering of Diffusion Tensor Fields, 10th IEEE Visualization Conference, 1999.

[5] Laidlaw, D. H., Ahrens, E. T., Kremers, D., Avalos, M. J., Jacobs, R. E., Readhead, C.: Visualizing Diffusion Tensor Images of the

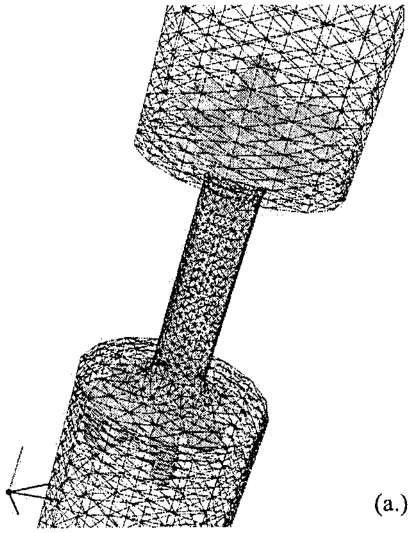
Mouse Spinal Cord, IEEE Visualization 98, pp. 127-134, 1998.

[6] Míka, S.: Matematická analýza III, University of West Bohemia in Pilsen, 1993.

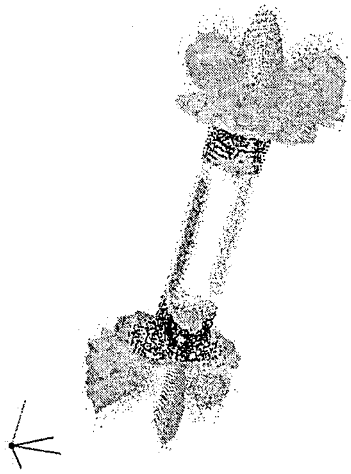
[7] Post, F. J., van Walsum, T., Post, F. H.: Iconic Techniques for Feature Visualization, Proceedings Visualization '95, IEEE Computer Society Press, pp. 288-295, 1995.

[8] Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 1992.

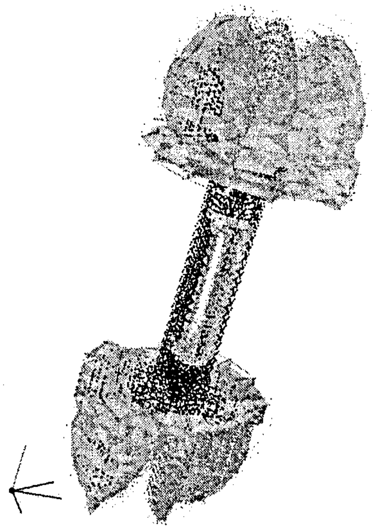
[9] Zheng, X., Pang, A.: Volume Deformation for Tensor Visualization, IEEE Visualization, pp. 379-386, 2002.



(a.)



(b.)



(c.)

Figure 3:

Visualization of the separating surface between the region, where the actual stress parallel to the user defined stress has lower magnitude than requested and the region, where it exceeds the user requested value.

The upper two images (a.) and (b.) were taken for a user defined stress vector (0.5, 0.5, 0.5). (a.) shows the position of the surface within the object. (b.) shows the same surface complemented with nodes shaded according to the stress magnitude, where dark means higher values and light the lower ones.

Picture (c.) shows the same information as (b.) but for a different user defined stress vector (0.1, 0.1, 0.1).