

Eigenvector-based Interpolation and Segmentation of 2D Tensor Fields

Jaya Sreevalsan-Nair¹, Cornelia Auer², Bernd Hamann³, and Ingrid Hotz²

¹ Texas Advanced Computing Center, University of Texas at Austin

² Visualization and Data Analysis, Zuse Institute Berlin, Germany

³ Institute for Data Analysis and Visualization, Dept. of CS, UC Davis

Abstract. We propose a topology-based segmentation of 2D symmetric tensor fields, which results in cells bounded by tensorlines. We are particularly interested in the influence of the interpolation scheme on the topology, considering eigenvector-based and component-wise linear interpolation. When using eigenvector-based interpolation the most significant modification to the standard topology extraction algorithm is the insertion of additional vertices at degenerate points. A subsequent Delaunay re-triangulation leads to connections between close degenerate points. These new connections create degenerate edges and triangles. When comparing the resulting topology per triangle with the one obtained by component-wise linear interpolation the results are qualitatively similar, but our approach leads to a less “cluttered” segmentation.

1 Introduction

Generally tensor fields are not easy to understand owing to their complexity. Segmentation into regions of similar directional behavior offers a way to extract the essential structure of the field. A segmentation based on tensor field topology guarantees that all significant structural irregularities are captured. We build on the basic work by Delmarcelle et al. [5], which introduces tensor field topology to visualization, and the further advancement by Tricoche et al. [17], with the goal of a more complete segmentation. In contrast to previous work we consider the topological structure of both eigenvector fields as a whole. As both eigenvector fields are orthogonal the topological graphs deliver a descriptive cell structure bounded by tensorlines. The field in the neighborhood of degenerate points of the combined topology is characterized by “half-sectors” instead of “sectors” and allows a more general structure. The second focus of our work is the influence of the chosen interpolation. Dealing mostly with discrete data interpolation is an essential step in the visualization process. In order to keep our method simple, we consider only piecewise linear methods defined in a triangulated domain. Besides using the standard component-wise interpolation, we introduce an eigenvector-based approach. This method minimizes the number of eigen-analysis by restricting it to mesh vertices and supports an exact integration of tensorlines. By decoupling “shape” and “direction” we achieve a shape-preserving interpolation. Introducing new vertices at degenerate points

and re-triangulating the domain leads to a simplified global topological structure without any modifications to the initial tensor field.

Our algorithm is designed to be simple and to follow clear rules. The algorithm consists of the following major steps: Eigenvector analysis at the vertices; edge labeling to ensure consistent direction interpolation; localization of degenerate points in each triangle and insertion of new vertices; determination of radial directions for both eigenvector fields; classification of the half-sectors defined by radial lines; integration of separatrices, and finally generation of a segmentation as intersection of the dual tensor field topologies.

2 Related Work

Most existing tensor field visualization methods are either specific to diffusion tensor fields or mechanical engineering applications, the latter being the focus of our work. The needs of domain experts in the area of diffusion tensor fields are well-defined as opposed to the lack of specific questions with respect to the tensors used in mechanical engineering. This difference in the driving force strongly reflects on their respective tensor field visualization techniques. Direct tensor visualization approaches focus on displaying tensor values in selected points. In this context research issues usually deal with the definition and placement of glyphs. Commonly used glyphs are, e.g., ellipsoids, Haber glyphs [7], or superquadrics [9], with improved perceptual properties. Different placement strategies are used to maximize the information displayed per image [6, 11]. While glyphs are appropriate for displaying single tensors, they are limited to low resolution and fail to give insight into the structure of the entire field. A more continuous view of 2D fields can be obtained by using tensor splats [3] or textures based on line integral convolution [8, 19]. The idea of using topological methods to analyze the structures of tensor fields goes back to Delmarcelle [5] and Lavin et al. [12]. Tricoche et al. [17] improvised these ideas for applications of complex 2D tensor fields, by developing algorithms to simplify the tensor topology and to track it over time. Alliez et al. [2] proposed an application to curvature tensors for polygonal remeshing of surfaces. First analysis of tensor field topology 3D tensor fields shows that in 3D degenerate features form lines [20].

In a similar vein, we investigate the influence of different interpolation methods on the topology extraction process and the resulting topological structure. Traditionally the component-wise linear interpolation has been used on tensor fields. However, this interpolation generates artifacts, e.g., the swelling effect. In an effort to alleviate these artifacts, methods separating direction and shape interpolation have gained more attention lately. In context of diffusion MRI data some direction-based interpolation methods based have been proposed for tracing anatomical fibers, [4, 10, 13]. Several other advanced interpolation methods based on components have been developed to achieve noise reduction or feature preservation [1, 14, 18]. The emphasis of our interpolation method is to provide a simple and consistent method, based on eigenvectors and eigenvalues, with a focus on the behavior in the neighborhood of degenerate points.

3 Basics and Notations

3.1 Tensors and Tensor Fields

In the rest of the paper, we will refer to symmetric 2D tensors of second order, defined on a triangulated 2D domain, as tensor field. Using a fixed coordinate basis, each tensor \mathbf{T} can be expressed as a symmetric 2×2 matrix, given by three independent scalars. We use the following notation

$$\mathbf{T} = \begin{pmatrix} d + \Delta & F \\ F & d - \Delta \end{pmatrix}. \quad (1)$$

\mathbf{T} is fully represented by its *eigenvalues* λ and μ and corresponding *eigenvectors* \vec{v} and \vec{w} . Since the multiplication of an eigenvector by any non-zero scalar yields an additional eigenvector, eigenvectors should be considered without norm and orientation. For symmetric tensors, the eigenvalues are real and the eigenvectors are mutually orthogonal. Integrating the eigenvector fields results in two orthogonal families of continuous curves. These curves are called *major* (red) and *minor* (blue) tensorlines according to the eigenvector field integrated.

Usually the tensor data-sets represent a discretized tensor field, whose geometry is represented by a triangulated mesh. Inside a triangle with vertices P_1 , P_2 , and P_3 we use barycentric coordinates $\beta = (\beta_1, \beta_2, \beta_3)$. The edge opposite vertex P_i is named e_i , for $i = 1, 2, 3$. We use \vec{v} and \vec{w} when referring to eigenvectors to allude to the fact that the eigenvectors are bidirectional. We use \mathbf{v} and \mathbf{w} when referring to vectors representing normalized eigenvectors with an arbitrarily but fixed direction, e.g., when using the results of the numerical computation. The direction of \mathbf{w} is defined in such a way that \mathbf{v} and \mathbf{w} form a right-handed system. We assign the names λ and μ , such that always $\lambda \geq \mu$.

3.2 Tensor Field Topology

Similar to vector fields, the structure of eigenvector fields is represented by its topology. It defines a skeleton consisting of distinguished points, so called degenerate points, and connecting edges, the separatrices. This skeleton segments the domain into regions with characteristic tensorline behavior. In contrast to previous work we do not look at the topology of each eigenvector field separately but consider both topologies as a whole. In the following we resume the basics of tensor field topology, concentrating on the aspects that we need later on. For a more detailed discussion we refer the reader to [5, 16].

Degenerate points - Degenerate points are points where the two eigenvalues are identical $\lambda = \mu$, and the eigenvectors are no longer defined uniquely. The tensor is proportional to the identity matrix and all vectors are eigenvectors. Degenerate points in tensor fields correspond to critical points in vector fields. Due to orientation indeterminacy of tensorlines, these points exhibit structures different from those seen in vector fields. A condition for degeneracy of a point is

$\Delta = 0 \wedge F = 0$. Independent of the eigenvalues, an isolated degenerate point can be categorized by the number of windings an eigenvector performs when encompassing it on a closed curve. The undirected eigenvector field allows winding-numbers to be multiples of one half. Only simple degenerate points (winding-number of $\pm \frac{1}{2}$) can exist in a tensor field, defined by linear interpolation of its components.

Separatrices - The behavior of the tensorlines in the vicinity of degenerate points follows certain characteristic patterns, with respect to a characteristic composition of basic sectors, explained further in Section 5.4. These sectors are separated by distinguished tensorlines, which enter the degenerate point radially. Radial tensorlines bounding hyperbolic sectors are called *separatrices* and constitute the edges of the topological graph.

4 Eigenvector-based Interpolation

The definition of tensor field topology is based on continuous data and hence we rely on an interpolation of the available discrete data. The standard interpolation is linear in tensor components. Instead we propose the use of an interpolation based on eigenvectors and eigenvalues, see Figure 1. This method minimizes the number of eigenvector computations. The interpolation is defined such that the resulting topology per triangle is qualitatively the same as for component-wise interpolation, see Figure 3. The main steps of this interpolation are: edge labeling, location and insertion of degenerate points, subdivision of triangles, and vector interpolation.

Assignment of directions to eigenvectors - edge labeling

Using vectors for the interpolation we first have to assign orientations to the eigenvectors to specify the interpolation uniquely. Doing so we have to consider the fact that not all structures occurring in tensor fields can be simulated by global vector fields, e.g., winding numbers of half integers. Thus a consistent global orientation of the eigenvector field is not possible. Therefore we keep the arbitrarily directed eigenvectors at vertices as generated by numerical computations and only encode relative directions between neighboring vertices using edge labels. For the label definition we follow the eigenvector behavior given by the component-wise interpolation. When moving from point P_i to P_j the absolute value of the rotation angle of the eigenvectors is limited to $\pi/2$. The direction of the rotation is given by the value $F_j \Delta_i - F_i \Delta_j$. If the value is equal to zero, then

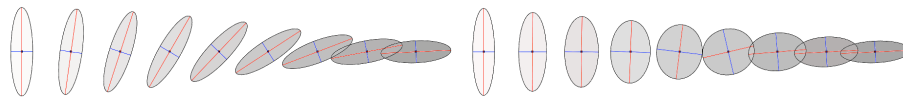


Fig. 1. Comparison of interpolation methods: left: eigenvector-based (shape preserving), right: component-based.

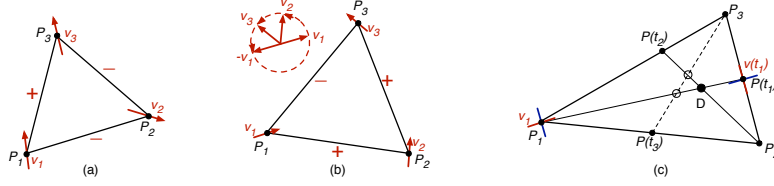


Fig. 2. Triangle (a) without and (b) with degenerate point, edge labels indicate whether two adjacent eigendirections match. (c) The location of a D is well-defined if the three lines connecting the vertices and their opposite points intersect in one point.

either both eigenvectors are the same or they encompass an angle of $\pi/2$ and the rotation direction is undetermined. Then there exists a degenerate point on the edge. The edge label of an edge e_k with endpoints P_i and P_j is defined as:

$$l(e_k) = \begin{cases} 1 & \text{if the directions of } \mathbf{v}_i \text{ and } \mathbf{v}_j \text{ match the direction propaga-} \\ & \text{tion, meaning } \mathbf{v}_i \cdot \mathbf{v}_j > 0, \\ -1 & \text{if the directions of } \mathbf{v}_i \text{ and } \mathbf{v}_j \text{ do not match the direction} \\ & \text{propagation, meaning } \mathbf{v}_i \cdot \mathbf{v}_j < 0, \\ 0 & \text{if there exists a degenerate point on the edge, } \mathbf{v}_i \cdot \mathbf{v}_j = 0. \end{cases}$$

The existence of a degenerate point inside a triangle P_i, P_j, P_k with edges e_i, e_j, e_k , is directly related to the product of its edge labels, see Figure 2. It is:

$$\prod_{i=1}^3 l(e_i) = \begin{cases} 1 & \text{no degenerate point in triangle,} \\ -1 & \text{one isolated degenerate point in triangle,} \\ 0 & \text{there is a degenerate point on at least one of the edges.} \end{cases} \quad (2)$$

If there exist two edges with degenerated points, we have a degenerate line. If there are three degenerate edges, the entire triangle is degenerate.

Interpolation in triangles without degenerate point

The tensor in point $P(\boldsymbol{\beta})$, $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)$, which are the barycentric coordinates of point P inside a triangle P_1, P_2, P_3 is defined by its eigenvectors \mathbf{v} and \mathbf{w} , which are not normalized, and eigenvalues λ and μ given by

$$\begin{aligned} \mathbf{v}(\boldsymbol{\beta}) &= \beta_1 \mathbf{v}_1 + \beta_2 l(e_3) \mathbf{v}_2 + \beta_3 l(e_2) \mathbf{v}_3, \\ \mathbf{w}(\boldsymbol{\beta}) &= \beta_1 \mathbf{w}_1 + \beta_2 l(e_3) \mathbf{w}_2 + \beta_3 l(e_2) \mathbf{w}_3, \\ \lambda(\boldsymbol{\beta}) &= \sum_{i=1}^3 \beta_i \lambda_i, \quad \text{and} \quad \mu(\boldsymbol{\beta}) = \sum_{i=1}^3 \beta_i \mu_i. \end{aligned} \quad (3)$$

Interpolation in triangles with degenerate point

In such triangles it is not possible to define a continuous vector field representing the tensor field structure. However, we can resolve this problem by inserting an additional vertex D at the degenerate point and subdividing the triangle to triangles without interior degenerate point. To determine the eigenvalue at D

we linearly interpolate the mean eigenvalue $d = (\lambda_i + \mu_i)/2$ in the original triangle and set the deviator $\Delta = (\lambda_i - \mu_i)/2$ to zero. Thus we can reconstruct the triangular domain by using piecewise linear interpolation in the subdivided domains. The tensor at point D is defined as a multiple of the unit tensor. The eigenvectors at D are set to zero, in correspondence to vector field singularities. Each new triangle with vertices P_i, P_j, D is interpolated separately. With $P(\beta) := \beta_i P_i + \beta_j P_j + \beta_k D$, (cyclic indices) eigenvalues and eigenvectors are interpolated using Equation 3. The resulting eigenvectors are independent from the coordinate β_k .

5 Topology Extraction

The basic steps for extracting topology are: location and classification of degenerate points, determination of separatrix directions, and their integration. We restrict this section to eigenvector-based interpolation; we refer readers to [5, 16] for further details on linear component-wise interpolation. The main difference between the two interpolation schemes is caused by the triangle subdivision. All degenerate points lie on vertices with piecewise linear behavior in the vicinity and can exhibit structures different from trisector and wedge points.

5.1 Location of Degenerate Points

Since degenerate points at vertices can be detected easily, this section is restricted to triangles without degenerate vertices. Initially, we also assume that there is no degenerate point along edges, and thus from Equation 2, the edge label product has to be -1. We define the location of the degenerate point exclusively dependent on the eigenvector field. Starting with a linearly interpolated eigenvector field on the edges e_i with endpoints P_k and P_j , $i, j, k \in \{1, 2, 3\}$ cyclic, we compute

$$\mathbf{v}(t) = (1 - t) \cdot \mathbf{v}_j + t \cdot l(e_i) \cdot \mathbf{v}_k, \quad t \in]0, 1[. \quad (4)$$

Even though the resulting vector field \mathbf{v} on the boundary may not be continuous at all vertices, the corresponding un-oriented direction field \vec{v} is. It defines a

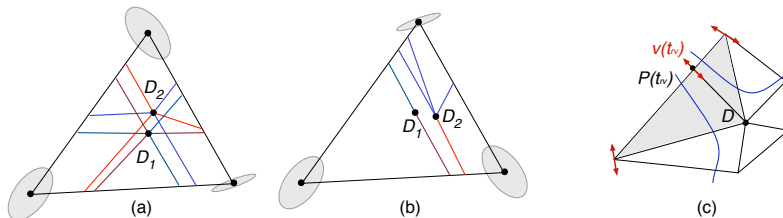


Fig. 3. Interpolation comparison in one triangle: Degenerate points D_1 from component-wise, D_2 from eigenvector-based interpolation, in the case of (a) trisector, and (b) wedge point. (c) Radial tensorline entering degenerate point D .

continuous rotation angle varying from zero to $\pm\pi$. The intermediate value theorem implies that for each vertex a parameter $t_i \in]0, 1[$, $i=1,2,3$, exists such that $\mathbf{v}_i \cdot \mathbf{v}(t_i)=0$. Thus, for every vertex there exists a point on the opposite edge, called *opposite point of the vertex*, with rotation angle $\pm\pi/2$ with degenerate point on its connection. The parameters t_i are given by

$$\mathbf{v}_i \cdot ((1 - t_i)\mathbf{v}_j + l(e_i)t_i\mathbf{v}_k) = 0, \quad (5)$$

where $i, j, k \in \{1, 2, 3\}$ are cyclic indices. This leads to the following definition: *The location of the degenerate point is defined as the intersection of the connections of triangle vertices to their opposite points*, see Figure 2(c)

It can easily be seen that the point D is well-defined. From the definition of t_i in Equation 5, it follows $t_1t_2t_3 = (1 - t_1)(1 - t_2)(1 - t_3)$, which is the condition that three lines connecting the vertices to points on the opposite edge, defined by parameters t_i , intersect in one point. For degenerate points on edges the three connecting lines degenerate to a line. In this case we use the eigenvalues at the vertices to determine the degenerate point.

5.2 Non-isolated Degenerate Points

Two degenerate vertices connected by an edge gives a degenerate line. The resulting eigenvector field inside adjacent triangles is constant and does not contribute to the final structure. Similarly a degenerate triangle, where all vertices are degenerate points, is enclosed by three triangles with constant eigenvector field. Thus from a structural point of view it is enough to consider the vertices of the degenerate entity and ignore the connecting edges. It is not uncommon to see degenerate polylines when applying a subsequent Delaunay re-triangulation.

5.3 Determination of Radial Directions

The neighborhood of the degenerate point is characterized by segments separated by radial tensorlines. For linear eigenvector interpolation, radial tensorlines are straight lines and are determined by their intersection $P(t_r)$ with the edges of adjacent triangles, see Figure 3. For each edge of the triangle,

$$\mathbf{v}(t_{rv}) \times (P(t_{rv}) - D) = 0, \quad \mathbf{w}(t_{rw}) \times (P(t_{rw}) - D) = 0, \quad t_{rv}, t_{rw} \in [0, 1]. \quad (6)$$

t_{rv} and t_{rw} specify the radial directions for the eigenvector fields v and w respectively. In contrast to component-wise interpolation, where radial directions are given by one cubic equation, we obtain one quadratic equation per edge and per eigenvector field. If not trivially fulfilled this leads to a maximum of two solutions per edge and eigenvector.

5.4 Sector Classification

For the skeleton computation only radial lines, which are boundaries of hyperbolic sectors, are relevant. In the case of component-wise interpolation a point

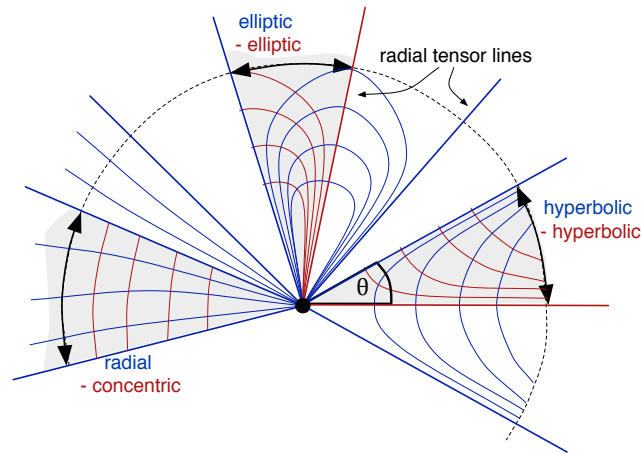


Fig. 4. The neighborhood of a degenerate points is characterized by a number of half-sectors with specific behavior.

classification into trisector or wedge points serves as basis for the classification. To cover all possible cases of degenerate points, for piecewise linear behavior, we built on an immediate sector analysis similar to [17]. In contrast we classify “half-sectors”, as we consider the topology of both eigenvector fields together. These are radial segments enclosed by two neighboring radial directions, independent of the eigenvector field, either red or blue, see Figure 4.

Hyperbolic sector - bounded by one red and one blue radial line: tensorlines approach, sweep past the degenerate point and leave the sector through one bounding radial line.

Parabolic sector - bounded by two radial lines of same color: all tensorlines, of this color, start from the degenerate point and then diverge. The tensorlines of the other color enter and leave the sector through bounding lines.

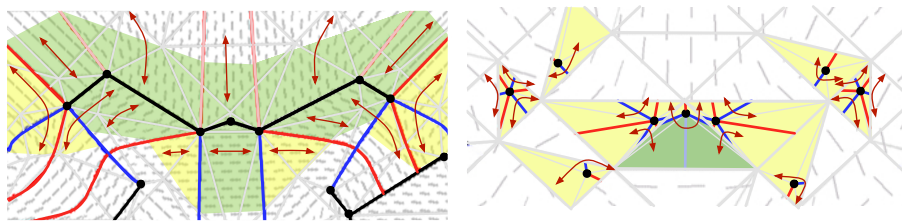


Fig. 5. A close-up of sector classification for the one-point load data-set using linear interpolation of eigenvectors, with (left) and without (right) subsequent re-triangulation. Shaded regions show the sectors: green and yellow for non-hyperbolic and hyperbolic, respectively; red and blue lines show radial lines, which are not integrated; black points and lines are the degenerate points and lines.

Elliptic sector - bounded by one red and one blue radial line: the tensorlines start from the degenerate point, and leave the sector through one of the bounding lines.

To classify the sectors the rotation angle of the eigenvectors $\Delta\alpha$ is compared to the opening angle of a half-sector $\Delta\Theta$ as shown in Figure 4

$$\begin{aligned} \Delta\alpha = \Delta\Theta & \quad \text{radial, concentric,} \\ \Delta\alpha = \Delta\Theta - \pi/2 & \quad \text{hyperbolic,} \\ \Delta\alpha = \Delta\Theta + \pi/2 & \quad \text{elliptic.} \end{aligned}$$

The same sector classification can also be used for degenerate lines and triangles. In this case all radial lines entering one of the participating vertices have to be considered. An example from a real data-set is shown in Figure 5.

5.5 Separatrix Computation

To complete the topological skeleton we integrate all radial tensorlines bounding the hyperbolic sectors using Runge-Kutta 4th-order integration scheme with adaptive step size. Alternatively, an exact tensorline integration for the linear eigenvector field can be used [15]. Direction consistency is not an issue in our approach, as in the case for component-wise interpolation, since eigenvector interpolation gives directed eigenvectors. We implemented the following termination conditions, to obtain a clean integration of tensorlines.

1. A separatrix leaves the domain, a trivial condition.
2. A separatrix gets close to a degenerate point, line or triangle. It is terminated at its intersection with the degenerated entity, see Figure 6a.
3. A separatrix describes a circle or spiral and passes itself closely in parallel integration direction, see Figure 6b and 6c. Circulating separatrices overload the topological graph without adding structural information for the final segmentation. We delete circulating tensorlines in a clean up process, which starts at the end of the separatrix and continues as long as the separatrix has a neighboring separatrix of the same color. The cleanup process ends in a point of intersection with a separatrix of the other color.

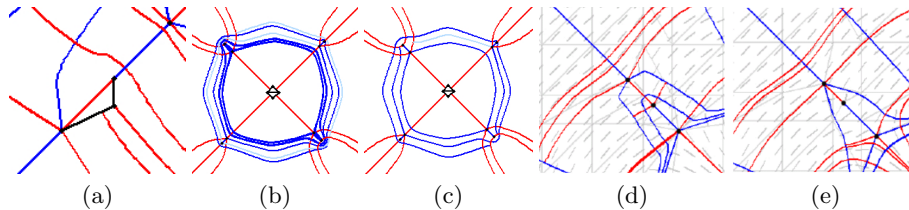


Fig. 6. Close-up from one-point load data-set: (a) tensorline runs into a degenerate line (black line); circulating tensorline (a) before and (b) after clean up; (d,e) comparison of separatrix integration for component-wise and eigenvector-based interpolation.

6 Segmentation

After computing the topological skeleton for both eigenvector fields, we find the intersections of the red and blue tensorlines. The properties of the resulting segmentation as can be seen in Figure 4 are (a) Cells without degenerate point are quadrangular with two red and two blue tensorlines as boundary, in an alternating order. All red tensorlines passing through this segment enter at one of the blue boundaries and leaves the cell at the opposite boundary and vice versa. All angles are orthogonal. (b) Cells with one degenerate vertex lying in a hyperbolic sector are quadrangular. The angle at the degenerate point is in general not orthogonal. (c) Cells having a degenerate point in one vertex, lying in a parabolic segment, degenerate to a triangular shape. (d) In elliptic sectors, cells with either two or three vertices are possible. (e) Cells containing degenerate lines as edges can exhibit more complicated structures.

7 Results

We tested our method on 2D slices of three different data-sets simulating stress behavior in a solid block: one and two forces applied to the top of a solid block and multiple forces applied to a notched block. Since the results for the eigenvector- and component-wise interpolation schemas are qualitatively similar we focus in the results in close-up views showing the major and typical differences. In Figure 6(d,e) a section of the one-point load containing three degenerate points is shown. While the basic structure is the same, the changes of the eigenvector directions is smoother for the eigenvector-based interpolation, resulting in less curved tensorlines in the vicinity of degenerate points. As a consequence, a proper step size adaption is especially important for component-wise interpolation to obtain tensorlines of the same quality. The calculation of the topological skeleton using eigenvector-based interpolation is in general faster than the one based on the component-based interpolation. This speed gain is a result of restricting the eigen-analysis to the vertices. For the component-wise interpolation it has to be performed for each integration step.

The effect of the re-triangulation on the complexity of the resulting topological structure is shown in Figure 5. These images are close-up views of the one-point load data-set, both using eigenvector-based interpolation. The left image was computed using a Delaunay re-triangulation after vertex insertion combining seven degenerate points in one degenerate line, which nicely represents the dominant radial structure of the red tensorlines. The number of separatrices that have to be integrated is reduced from 35 to 14. Details of the local topological structure are often not features of the data-set, but instead are by-products of the chosen interpolation schema. This is an incentive to keep the resulting topological structure simple while still being consistent with the data.

We have applied our method to data-sets representing the simulation of different force combinations acting on a solid block. Figure 7 shows complete segmentations of a slice of each data-set. In the top right image the cells are randomly

colored. The other images displays the blue and red tensorlines bounding the segments. Black dots, lines and triangles show the degenerate entities. The one- and two-point load data-sets are simulated with very low resolution resulting in artifacts that are reflected in the complicated topological structure. An adaptive finite element method was used in the third dataset which results in a much clearer structure, even though the physical configuration is more complex.

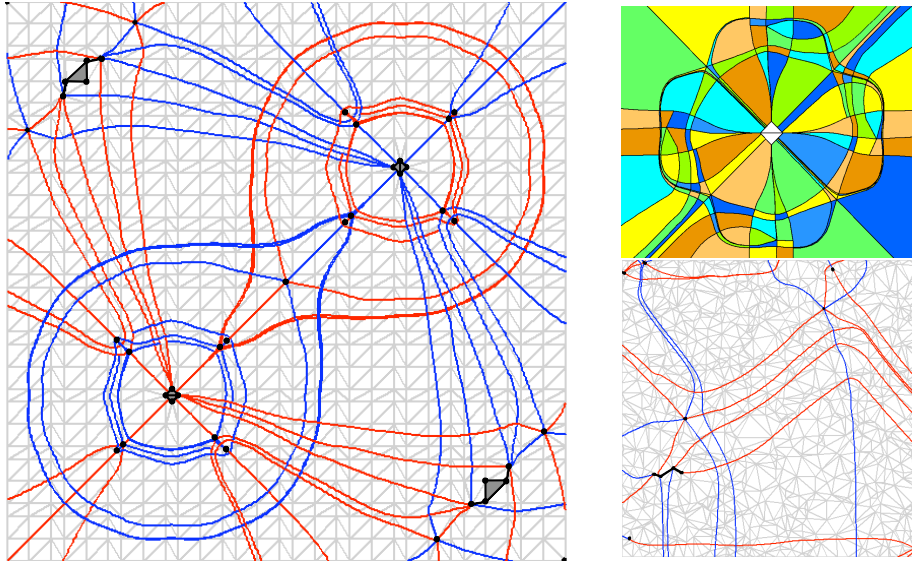


Fig. 7. Full segmentation: left: two-point load, right top: one-point load with randomly colored segments, right bottom: slice of strain simulation of forces on notched block.

8 Conclusions

We have presented a method that delivers a segmentation that capture the eigenvector behavior in a 2D tensor field. With the classification of the sectors we are able to extract separatrices that build the topological structure. Simultaneously the insertion of the degenerated points in the Delaunay triangulation decreases the number of separatrices and therefore the number of segments. The resulting degenerated lines and triangles capture the invariants in the field in a simplified way, without changing the given tensor values. Though the results for the topology extraction of the component-wise interpolation and our eigenvector-based method are qualitatively similar, the latter is faster. Albeit slight differences in the segmentation obtained using both the methods, the quality of the segments is the same. Future work includes a further clean up to simplify the segmentation by combining similar elements and a further refinement of large cells, using the boundary topology.

Acknowledgments This work was supported in part by the German Research Foundation (DFG) through a Junior Research Group Leader award (Emmy Noether Program), and in part by the the National Science Foundation under contract CCF-0702817. We thank our colleagues at the Zuse Institute Berlin and the Institute for Data Analysis and Visualization (IDAV), UC Davis.

References

1. A. Aldroubi and P. Basser. Reconstruction of vector and tensor fields from sampled discrete data. *Contemp. Math.*, 247:1–15, 1999.
2. P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy, and M. Desbrun. Anisotropic polygonal remeshing. *Siggraph'03*, 22(3):485–493, Jul 2003.
3. A. Bhalerao and C.-F. Westin. Tensor splats: Visualising tensor fields by texture mapped volume rendering. In *MICCAI'03*, pages 294–901, 2003.
4. O. Coulon, D. C. Alexander, and S. Arridge. Tensor field regularisation for dt-mr images. In *MIUA01*, pages 21–24, 2001.
5. T. Delmarcelle. *The Visualization of Second-order Tensor Fields*. PhD thesis, Stanford University, 1994.
6. L. Feng, I. Hotz, B. Hamann, and K. Joy. Anisotropic noise samples. *IEEE TVCG*, 14(2):342–354, 2008.
7. R. B. Haber. Visualization techniques for engineering mechanics. *Computing Systems in Engineering*, 1(1):37–50, 1990.
8. I. Hotz, L. Feng, H. Hagen, B. Hamann, B. Jeremic, and K. I. Joy. Physically based methods for tensor field visualization. In *IEEE Visualization 2004*, pages 123–130, 2004.
9. G. Kindlmann. Superquadric tensor glyphs. In *Eurographics Symposium on Visualization*, pages 147–154, May 2004.
10. G. Kindlmann, R. S. J. Estepar, M. Niethammer, S. Haker, and C.-F. Westin. Geodesic-loxodromes for diffusion tensor interpolation and difference measurement. In *MICCAI'07*, pages 1–9, 2007.
11. G. Kindlmann and C.-F. Westin. Diffusion tensor visualization with glyph packing. *IEEE TVCG*, 12(5):1329–1336, 2006.
12. Y. Lavin, R. Batra, L. Hesselink, and Y. Levy. The topology of symmetric tensor fields. *AIAA 13th Computational Fluid Dynamics Conference.*, page 2084, 1997.
13. M. Martin-Fernandez, C.-F. Westin, and C. Alberola-Lopez. 3d bayesian regularization of diffusion tensor MRI using multivariate gaussian markov random fields. In *MICCAI'04*, pages 351–359, 2004.
14. M. Moakher and P. G. Batchelor. Symmetric positive-definite matrices. In *Visualization and Image Processing of Tensor Fields*, pages 285–297. Springer, 2006.
15. G. M. Nielson and I.-H. Jung. Tools for computing tangent curves for linearly varying vector fields over tetrahedral domains. *IEEE TVCG*, 5(4):360–372, 1999.
16. X. Tricoche. *Vector and Tensor Field Topology Simplification, Tracking and Visualization*. PhD thesis, TU Kaiserslautern, 2002.
17. X. Tricoche, G. Scheuermann, H. Hagen, and S. Clauss. Vector and tensor field topology simplification on irregular grids. In *VisSym '01*, pages 107–116, 2001.
18. J. Weickert and M. Welk. Tensor field interpolation with pdes. In *Visualization and Processing of Tensor Fields*, pages 315–324. Springer, 2005.
19. X. Zheng and A. Pang. Hyperlic. In *IEEE Visualization'03*, pages 249–256, 2003.
20. X. Zheng and A. Pang. Topological lines in 3d tensor fields. In *IEEE Visualization'04*, 2004.