CORNELIA AUER

ESSENTIAL STRUCTURES IN SECOND ORDER TENSOR FIELDS

Essential structures in second order tensor fields

VISUALIZATION OF FUNDAMENTAL STRUCTURES IN TWO DIMENSIONAL SECOND ORDER TENSOR FIELDS ON PLANAR AND CURVED SURFACES

Inauguraldissertation

zur Erlangung des akademischen Grades
des Doktors der Naturwissenschaften (Dr. rer. nat.)

eingereicht am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

CORNELIA AUER
Zuse Institut Berlin

March 2013

Erstgutachterin:	Dr. Ingrid Hotz
	Zuse-Institut Berlin
Zweitgutachter:	Prof. Dr. Hans Hagen
	Technische Universität Kaiserslautern
Tag der Disputation:	10.07.2013

ABSTRACT

Tensors provide a powerful and at the same time concise mathematical formalism to encode intricate physical phenomena. They describe multi-linear functions independently of a frame of reference and capture anisotropic properties which vary as function of direction. However, the wealth of information contained in tensor data can be a mixed blessing as in return this can heavily aggravate their interpretation. This thesis is concerned with analysis and visualization methods to support the interpretation of second order tensors per se and fields of such tensors. The focus is on tensor fields from engineering and mechanics. We present visualization concepts for indefinite symmetric tensors as well as asymmetric tensors which both bring about their individual properties and requirements. The aim is to truthfully reflect the specific properties in the tensor fields and at the same time to support an immediate understanding by the user. The presented methods are developed for two dimensional second order tensor fields defined on planar or curved surfaces. These tensor fields naturally occur e.g. on boundary surfaces but also if the data is analyzed on cuts extracted from three dimensional data. This facilitates to inspect the intrinsic properties on these geometries in full detail. One primary constituent of this work is to find expressive structures in order to present the tensor data in a condensed and simplified manner. The results are given as explicit geometries which can be used for further processing such as tracking over time or statistical inquiry. For symmetric second order tensor fields we extract the topology which captures all essential structural features in a strongly reduced graph structure. This graph structure is used as a basis to develop enriched visualization methods. In this vein, a complete segmentation is presented that partitions the field into regions of homogeneous eigenvector and eigenvalue behavior. This segmentation serves as well defined framework for rich visualizations – texture mapping for a continuous and dense depiction and a glyph placement strategy for detailed inspection on demand. Finally, a sketch-like visualization of vector fields and their spatial derivatives, asymmetric tensor fields, is presented. The initial asymmetric tensor data is uniquely decomposed and automatically filtered based on scalar field topology and homological persistence. The extracted prevalent features are depicted in an illustrative, easy to read visualization which facilitates a comprehensive overview of the field characteristics. All contributions in this thesis are topological methods or build on such which guarantees that the results depict the data in a simplified but uncorrupted and consistent manner.

ZUSAMMENFASSUNG

Tensoren bieten ein mächtiges und präzises Konzept zur Formulierung komplexer physikalischer Phänomene. Sie beschreiben multi-lineare Funktionen und erfassen anisotrope Eigenschaften. Diese Mächtigkeit kann jedoch einen gravierenden Nachteil bedeuten – die Interpretation von Tensordaten wird dadurch signifikant erschwert. Diese Arbeit präsentiert Analyse- und Visualisierungsmethoden, die das Verständnis von Tensoren zweiter Ordnung unterstützen. Hierbei liegt der Fokus auf Tensordaten aus den Anwendungsfeldern Physik, Mechanik und Ingenieurswissenschaften, die Visualisierungskonzepte für indefinite symmetrische so wie für asymmetrische Tensoren erfordern. Ziel ist es, die jeweiligen Eigenschaften wahrheitsgetreu wider zu spiegeln und gleichzeitig dem Benutzer ein intuitives Erfassen zu ermöglichen. Die hier vorgestellten Methoden wurden für zwei dimensionale Tensorfelder entwickelt, die auf planaren oder gekrümmten Flächen definiert sind. Dieses beinhaltet physikalische Randflächen, aber erlaubt auch die Analyse dreidimensionaler Datensätze, die zum Beispiel bezüglich ihrer intrinsischen Eigenschaften auf extrahierten Isoflächen untersucht werden. Elementarer Bestandteil dieser Arbeit ist die Extraktion expressiver Strukturen, um Tensordaten in einer kompakten und vereinfachten Art darzustellen. Die Ergebnisse werden als explizite Geometrien definiert, die eine weiterführende Analyse begünstigen, wie die Untersuchung zeitabhängiger Felder oder eine statistische Analyse. Für symmetrische Tensoren zweiter Ordnung präsentieren wir ein Modell zur Berechnung der Topologie. Diese beinhaltet alle essentiellen strukturellen Merkmale in einer stark reduzierten Graphstruktur und dient als Basis für erweiterte Visualisierungsmethoden. Darauf aufbauend wird eine Segmentierung vorgestellt, die das Tensorfeld in Regionen gleichen Eigenvektor- und Eigenwertverhaltens partitioniert. Diese wohldefinierte Struktur erlaubt Visualisierungsmethoden hoher Informationsdichte: Texturmapping als kontinuierliche Darstellung des gesamten Feldes und Glyphenplatzierung für eine detaillierte Untersuchung an ausgezeichneten Stellen. Für asymmetrische Tensoren präsentieren wir eine Visualisierungmethode, die in ihrem Grad an vereinfachter Darstellung Handzeichnungen aus Lehrbüchern ähnlich ist. Die Tensoren werden hierbei in ihre Komponenten zerlegt und automatisch durch eine Kombination von Skalarfeldtopologie und homologischer Persistenz gefiltert. Alle hier vorgestellten Beiträge sind topologische Methoden oder basieren auf solchen. Dies garantiert eine Ergebnisvisualisierungen der Originaldaten in vereinfachter, jedoch in unverfälschter und konsistenter Art.

DANKSAGUNG – ACKNOWLEDGEMENTS

My first thanks goes to my advisor Ingrid Hotz. Thank you for being an outstanding group leader in many ways. Your virtue, scientific curiosity and way of leading people has taught me a lot! You created a space of open, undogmatic scientific thinking where free, collaborative research could grow. Thanks for your encouragement in these years and sharing your knowledge, ideas, and way of thinking with us. I wish you a great future where many other people benefit from your excellence.

I also thank Jaya-Sreevalsan Nair for her collaboration and our explorations in tensor field topology. I am very proud you are professor by now and wish you all the best for your professional career!

Working at ZIB was a pleasure, the diversity of projects made it a lively ground for research. Thanks Andrea for our discussions on life and the rest, I think we came pretty far! Seeing your work and claim on scientific standards but also for sharing good and bad moments with me. Thanks for inspiring discussions and feedback especially to Kai, Jens, Dagmar, Stefan Zachow, Britta, Ulli, Valentin, Alex, Steffen, Matze! To all colleagues at ZIB – from the visualization department to the IT-Service crowd – going to work was also meeting friends. I hope I never have to say I miss you, because we stay in touch.

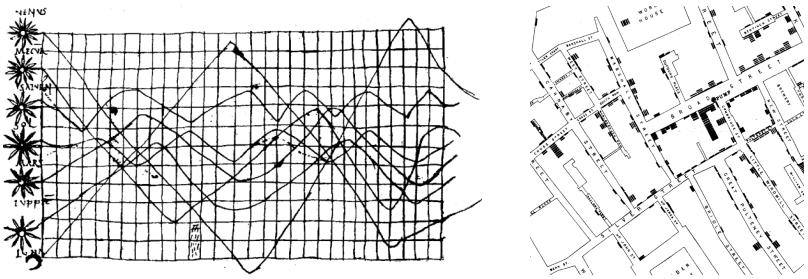
Thanks to my friends – for your support and love all these years, helping me to get my tensor fields going without forgetting about life: Gabriella, Ralf, Ines, Maria, Maxi, Jule, Tomtom, Stefan, Andi, Patricia, Sylvi, Ronny, Simon, and so many more.

My final and very big thank you goes to my parents, Christine and Josef Auer, my brother Josef Auer with his family, and my aunt Gabrielle Lazar with her family for ever since being a source of inspiration, support, and love!

CONTENTS

1 INTRODUCTION

An image is worth a thousand words – this adage is a commonly recognized saying which characterizes that a single still image can sometimes transport a large amount of information that is not feasible to communicate by other means. It can also be used to summarize one of the main goals of *visualization* - the aim of facilitating the absorption of a large amount of data by giving appropriate images, diagrams, or animations. This builds on the fact that the human eye is able to perceive a multitude of parameters synchronously: colors in various shades, patterns, and geometric objects just to name a few. The term visualization itself cannot be assigned to a specific definition as the attempt to depict ideas and facts can be seen throughout the history of mankind, be it ancient cosmological maps (e.g. Fig. 1.1(a)) or the example of a visualization by Dr. John Snow of the cholera outbreak in London, 1854 (Fig. 1.1(b)) collected by Edward R. Tufte [Tuf86]. In the last decades computational power has



(a) (b)

Figure 1: Early Visualizations. (a) Ancient map illustrating the orbits of the planets dated around 10. century. (b) Visualization of Dr. John Snow of the cholera outbreak in London, 1854. Bars plotted in the street map of London indicate the number of deaths in a household and where used to detect sources of the epidemic. Image courtesy: E.Tufte [Tuf86].

strongly increased and the growth of resources available was also extensively used in science. Here, the facility was soon discovered to produce computer graphics to visually assist researchers in understanding their generated data. In 1987, Bruce H. McCormick et al. [MDB87] were the first to formulate the concept and the need for a new and explicit scientific discipline which they called *scientific visualization*. McCormick et al. describe scientific visualization as method of computing, that transforms the data into visual representations

which allow researchers to observe their results and draw expected but also unexpected conclusions.

In scientific computing data sets mostly consist of a geometry and associated attributes. The geometry defines the domain of the data set at discrete coordinates, e.g. on vertices of a computational grid. The attributes are the scientific research results in terms of function values given on the discrete points of the domain. Common types of attributes are scalars, vectors, and tensors which led to the differentiation into *scalar*, *vector*, and *tensor field visualization*. The fourth and younger discipline, *information visualization*, can have attributes of various expression. Further, an explicit geometry as such is mostly not given. This thesis is primarily concerned with the visualization of tensor fields.

The remainder of this chapter is structured as follows: In Sec. 1.1 tensors and tensor fields are characterized and key mathematical concepts are introduced on a rather colloquial level to characterize basic ideas. Sec. 1.2 outlines the basic visualization principles that were guiding the development of the visualization methods presented in this work. Sec. 1.3 summarizes how these principles were implemented. Finally, Sec. 1.4 gives a brief overview of the thesis' structure and Sec. 1.5 lists the publications building the basis of the presented work.

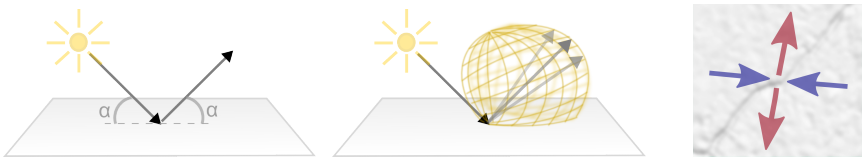
1.1 TENSORS AND TENSOR FIELD VISUALIZATION

Tensors are multi-linear functions that provide an elegant simplicity for the modeling of complex problems and deliver the necessary generalization where vectors and scalars are too limited. Although founded in the work of Carl Friedrich Gauss ($1777 - 1855$) on differential geometry tensors had their breakthrough in the beginning of the last century. Many discoveries were made in the early twentieth century that changed the perception of the world around us: the special and the general theory of relativity, and quantum mechanics might summarize the highlights during that period. It is no accident that this coincides with the time where tensor analysis was developed to its present form. For example, Albert Einstein ($1879 - 1955$) had an enormous breakthrough when he managed to formulate his theory of gravity with tensors. Also the laborious and pioneering work of Emmy Noether ($1882 - 1935$) on the famous theorems predicting the existence and precise nature of conservation laws (applied to physical field theories) can be deduced nowadays with strongly reduced effort by using tensorial notation [LR75]. From that time many other disciplines also incorporated tensor calculus: the areas range from physics and engineering to modern medicine and image processing,

just to name a few.

Tensors are ubiquitous in science. Also scalars and vectors are subsumed by the concept of tensors – a scalar is a tensor of zeroth order and a vector a tensor of first order. In this thesis we concentrate on tensors of second order which are defined to be acting on two vector spaces. The benefit of using tensors can be regarded from two points of view (although they can not be perfectly separated): first they give a concise and versatile framework to model scientific phenomena – especially as transformations from one coordinate system to another can be made effortlessly. Secondly, it can also enhance the understanding of some phenomenon by inspecting the results of simulations or measurements in terms of tensors. For tensors of second order an important property comes into play: they are capable to capture *anisotropic* properties. There is no comprehensive formal definition of anisotropy but it can be summarized to the quality of some regarded phenomenon exhibiting different properties when measured from different directions.

A descriptive case of anisotropic behavior is that of light reflection. In an idealized world a ray of light would be reflected on a perfect mirror in an non scattered manner, without loss of intensity, and the emergent angle would be equal to the incident angle (see Fig. 1.2(a)). However, in reality surfaces are not perfectly mirroring and thus, the light reflection is anisotropic. This means the reflection is scattered and depending on the reflection direction the light intensity varies (see Fig. 1.2(b)). The opposing notion of anisotropy is *isotropy* where properties remain identical independent of the measured direction.



(a) (b) (c)

Figure 2: Schematic illustration of anisotropy. (a) No anisotropy: the reflection of a ray of light in an idealized world causes no anisotropic behavior. (emergent angle equals incident angle). (b) Anisotropy: in reality the ray of light is scattered as it is reflected. The light intensity (yellow distribution function) of the emergent rays is anisotropic. (c) Central topic in the area of mechanics: anisotropic directions and strengths of forces. Pushing (blue arrows) and pulling (red arrows) forces acting in one location can lead to material failure such as cracks.

Another example is more related to the application focus of this thesis. In mechanics, it is of strong interest to understand the forces acting in one location, as for example the occurence of pushing and

pulling forces from different directions at a specific position which can lead to material failure, such as cracks (Fig. 1.2(c)).
Tensors of second order can capture such anisotropic behavior. Directions and strengths occurring at a specific location can be directly inferred and allow valuable insight to a data set.

This wealth of information contained in tensor data however, poses a specific challenge for their interpretation and consequently visualization. Second order tensors are frequently distinguished between symmetric and asymmetric tensors and both types are underrepresented in scientific visualization despite their scientific significance. Comprehensive depictions have to be found, that reflect the tensor properties but which are at the same time intuitive and do not overwhelm the user. For both types of tensors, mathematical concepts exist that allow to derive expressive quantities and structures to ease their analysis. These are introduced below as they are fundamental to the visualization methods developed in this thesis.

Eigenvectors and eigenvalues
Tensors can be decomposed and analyzed in terms of their *eigenvectors* and *eigenvalues*. They are attained by performing an *eigenanalysis* and uniquely represent the original tensor without loss of information. Extremal directions in the transformation encoded in tensors can be identified and visualized by eigenvectors. The strength and quality of the transformation in the respective eigenvector direction is given by the associated eigenvalues.

Topology
Topology is a mathematical approach to reduce some phenomenon to a highly abstract representation and is not restricted to tensor fields. Hereby, the topology contains structures which are rather qualitative than quantitative in nature and properties are captured which are preserved under continuous transformations. Absolute measurements based on an underlying metric such as distance or angles are not regarded. Looking at the Greek origin of topology – *topos* "place" and *logos* "study" – already suggests that topology is the study of positions and relative positions in terms of neighborhood relations. A classical example is the problem of the *Seven bridges of Koenigsberg* – where the question is to find a walk through the city of Koenigsberg by crossing its seven bridges once and only once (Fig. 3). This problem is of topological nature as the solution does neither depend on how long the bridges are nor their mutual distance. Purely the position of the bridges and their relative position to each other defines that – in the case for Koenigsberg – no such walk can be found.

In scientific visualization topology is a meaningful instrument to extract essential structures of a given data set. Topology is repre-

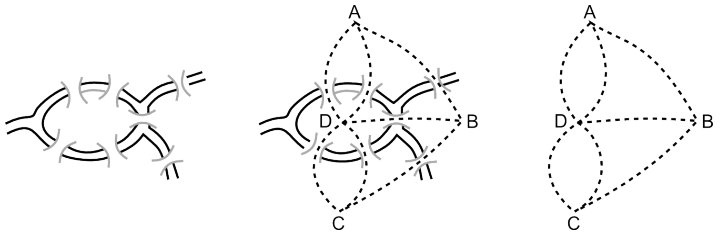


Figure 3: The topological problem of the seven bridges in Koenigsberg: find a walk through the city without traversing a bridge twice. This question can be represented by an abstract graph structure.

sented by a graph structure which partitions a field into regions of similar properties based on well defined criteria. The vertices of the graph represent distinguished locations in the field from which separating lines or shapes are computed, the connections of the vertices. The graph captures the original data in a strongly compressed manner and the resulting skeleton can be used as visualization itself. Surprisingly, the topology of tensor fields is not as unfamiliar as one might think. The growth of the human fingerprint pattern and its directional layout is defined by anisotropic properties and can be represented by a tensor field. In fingerprints the key locations of tensor field topology can be observed (see Fig. 1.4(a)). From those key locations appropriate separating lines can be computed which partition the entire fingerprint into regions of homogeneous behavior (see Fig. 1.4(b), 1.4(c)). In these regions the layout of the skin pattern is continuous and can be qualitatively reconstructed from the separating lines. Similarly, computing the topology of tensor fields for scientific results allows to highly condense the input data to the key topological features. The resulting graph structure gives a continuous and complete view on the essential structures of a tensor field and therefore, is a valuable mean in tensor field analysis. Please note, the topology of a tensor field is purely based on the eigenvector fields and thus, only structural features are captured and eigenvalues are not respected.

1.2 OBJECTIVES AND BASIC VISUALIZATION PRINCIPLES IN THIS THESIS

Visualizing an entire field of a quantity that changes in dependence of direction at each location in the field is a challenging question. It is certainly desirable to present as much information as possible to a user. Still ideally, the final visualization should not overwhelm the user. In summary, an incomplete view is as undesirable as a cluttered image. Any principle to generate a winsome visualization especially holds for the depiction of tensor fields. In the following those con-

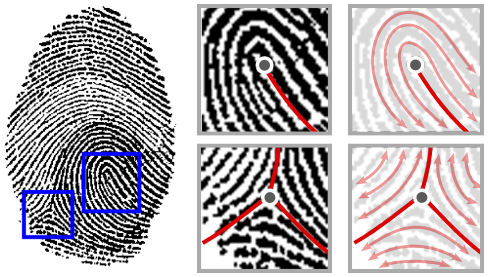


Figure 4: Tensor field topology for fingerprints. a) Key topological locations can be observed in the blue squares. b) Specific separating lines (red) deliver a segmentation of the domain into qualitatively equal directional behavior. c) Surrounding directions (light red) can be reconstructed from the separating topological lines.

cepts pursued in this thesis are summarized.

Simplification and Overview

One obvious way to support the comprehension of tensor fields is to develop sensible simplification strategies to reduce the amount of information to display. The reduction should be reproducible and preserve important features and details in the data set. This is of special importance as naturally scientific visualization has to treat data with great care. But also because visualization is not only about delivering images to specific questions that were posed in advance. A valuable contribution is that the created images can also reveal the unexpected and allow new insights that were or could not be predicted ahead. The eye can discern discontinuities, outliers, or coherence in an image which might be painstaking to find in a data set of numbers. Thus, ideally the final visualization should be *complete* with respect to the regarded quantity.

Full Information on Demand

In contrast to a reduced view on the data is the need of the user to also examine data in full detail. Giving the user an overview visualization at first and showing those details on demand is a common way to provide data at full resolution without overburdening the user. Special examples are focus and context visualizations – these provide a strongly simplified context visualization that allows the user see the data set as a whole. Within the focus region the data is depicted in full detail. This supports the understanding of the entire data set in an interactive and top-down manner. The user can request information bit by bit which is finally less tedious than absorbing the entire content at once.

Comprehensive Depiction of Tensors and their Properties
Often researchers reduce the analysis of their results to derived scalars or vectors as the interpretation of tensors seems unintuitive. This can be helpful for a first understanding however, this may be unsatisfactory as it naturally implies loss of information. Thus, tensor field visualization is not only concerned with supporting the analysis of individual tensor data sets but also with the understanding of tensors themselves. At best, tensor properties and the transformation encoded by tensors are visually explained in the final image.

1.3 ACCOMPLISHMENT OF VISUALIZATION PRINCIPLES

A central part of this thesis is concerned with simplification methods for tensor fields which assist the user in understanding the given data set but at the same time guarantee reliability and reproducibility. Hereby, essential structural characteristics in tensor fields are extracted and depicted.
These reduced structures can be further inspected or are used as basis for extended visualization methods. The aim is to give the user the freedom to explore data sets at a level of finer granularity or enriched with additional information. This provides the flexibility to meet the diverse demands of different tensor types and application areas.

1.3.1 *Dimension Reduction*

The developed methods are designed for two dimensional tensor fields of second order defined on planar or curved surfaces. This can inherently represent a first simplification step – dimension reduction. Often, two dimensional data occurs if dimension reduction is chosen to get a hold of the data which is a common approach for tensor data. The initial three dimensional data is projected onto slices or cuts (e.g. extracted by isosurfacing) and inspected in terms of its intrinsic properties on these geometries. However, two dimensional tensor data also appears for data sets where the quantity of interest is defined on two dimensional manifolds, such as boundary surfaces. Here, the fields are inherently two dimensional.
On these manifolds the depiction of tensors in their full information content is significantly alleviated. On such surfaces depiction elements which cover the anisotropic properties of tensors are easier to arrange and compared to three dimensional space problems like occlusion and clutter are strongly reduced. In this thesis we will attach importance of also providing the necessary concepts to visualize tensor fields on curved surfaces.

1.3.2 *Topological Analysis*

Topology provides a sound mathematical foundation to strongly reduce the amount of information in the field without introducing heuristics about which information should be kept or can be omitted. To extract the tensor field topology several challenges arise. The extraction of the topology is based on a continuous model of the tensor field. However in visualization mostly fields occur that are given at discrete data points. Consequently, a fundamental challenge is to provide a model for continuous reconstruction of the field that allows the detection of topological elements. Chap. 4 proposes a new model to compute the topology of two dimensional symmetric tensor fields defined on planar surfaces. It is founded on a specific interpolation scheme [HSNHH10] decoupling shape and direction of tensors (in terms of a consistent linear interpolation of eigenvectors and eigenvalues). Our model of topology computation is analyzed towards its validity in generating qualitatively correct results. The resulting structure, the *integral topological graph*, captures all directional features of both eigenvector fields and segments the field into curvilinear cells of homogenous directional behavior. Depending on the data the topological graph may result in complex structures which are still hard to interpret. The integral topological graph of the proposed method extracts topological elements of higher dimensionality which is compared to previous approaches more cleaned up. Chap. 7 extends the usability of the topology extraction model by giving the necessary extensions to conduct the topological analysis for piecewise continuous and piecewise constant tensor fields.

1.3.3 *Complete Segmentation*

The integral topological graph is a highly compact representation of all directional features in a symmetric tensor field. In Chap. 5 this graph is the basis for a segmentation that respects the full tensor information. An adaptive refinement workflow incorporates the eigenvalue fields to partition the field into explicit geometric cells that delineate regions homogeneous eigenvector *and* eigenvalue behavior. Alternatively, the refinement process of the topological graph can also be guided by more application specific scalar invariants. The segmentation framework consists of flexible building blocks giving the user the freedom to adjust the process towards divers segmentation criteria or granularity and adapt it to meet the demands of a specific application. The resulting cells can be considered as visualization glyphs where eigenvector and eigenvalue characteristics are subsumed in form of tiles. The cells are delineated by tangent lines to the eigenvector fields. Like this the structure of the segmentation resembles a net and allows immediate interpretation of the directional

behavior in the field. Color coding of the cells encodes the eigenvalue characteristics. Such a segmentation allows to comprehensively inspect the entire tensor field on a higher level of abstraction. The user is assisted in the exploration of the data set as rather similar regions can be distinguished from regions of strong variation. Further, a focus and context visualization can be directly deduced.

1.3.4 *Texture and Glyph Mapping*

Chap. 6 uses the variability of visualization methods like textures and glyphs to enrich the abstract topology based segmentation described above. This combines the accuracy of topological methods with more intuitive visualizations. The segmentation structure serves as basis to map textures and place glyph exponents. The textures encode physical properties of the underlying symmetric tensor field and support the understanding of the field as a whole. The glyph exponents in contrast, can be used to give detailed insight at distinct locations. Any kind of tensor glyph can be placed into the centers of the segmentation cells with homogeneous characteristics. For the texture mapping the cells of the segmentation provide a consistent parametrization. Textures offer a multitude of parameters to encode tensor properties – eigenvector directions, eigenvalues and other derived quantities – combined in a continuous, dense visualization of the entire field. The textures are mapped such that physical properties of the tensor field like compression or expansion are directly reflected in the texture frequency. The field can be explored in great detail and coherences discovered. Further, selective color mapping and post processing are applied to visually assist the users to direct their attention to features of interest.

1.3.5 *Illustrative Visualization*

In Chap. 8 asymmetric tensor fields are used to generate illustrative representations of vector fields. A major advantage of visualizing tensors is their property of being independent of the chosen frame of reference. Visualization methods purely based on vectors have the difficulty that computed features can disappear if the frame of reference is changed. Looking at the spatial derivatives of the vectors leads to asymmetric tensors which allow valuable insight to the properties of the vector field, such as rotation, shear, and isotropic scaling. We present an automatic framework to extract a highly simplified representation of the vector and asymmetric tensor field. The tensors are decomposed into their rotation, shear and isotropic scaling components and processed in terms of these individual components. This allows to decompose and depict the transformation encoded in asymmetric tensors into easier to comprehend elements. In their simplicity

the representations aim to resemble hand-drawn sketches which can be found in subject literature. They are reduced to the most prevalent features and use easy to read icons. Large scale trends are depicted as background visualization and strongly expressed features displayed in the front. The automatic approach uses the extrema of scalar field topology to extract all local features in the field and on the basis of homological persistence those features are filtered to the most prevalent ones. This provides a sound mathematical basis for the simplification process; the results are reproducible and have a clear mathematical interpretation.

1.4 OUTLINE OF THIS THESIS

In Chapt. 2 the mathematical basics are introduced which build the foundation for the content of this thesis. Existing visualization techniques related to the scope of this work are introduced in Chapt. 3. The actual contributions for new methods in tensor field visualization are presented in Chapt. 4, 5, 6, 7, and 8. In Chapt. 9 the work is discussed and a conclusion drawn towards its achievements and options for future work.

1.5 PUBLICATIONS

The content of this thesis builds on the following papers which have been peer-reviewed and published in international conference proceedings, journals, and books: [SNAHH11], [ASNZH11], [AH11], [ASKH12], and [AKKH13].

2 BASICS

This sections introduces the basics and notations used throughout the thesis. If basics are only related to one specific chapter they will be introduced there to keep the reading of this text modular.
Tensors of second order are the main subject of this thesis, accordingly they are introduced in more detail. While scalars (Sec. 2.2) and vectors (Sec. 2.3) are probably familiar to most readers with general scientific background, the notion of tensors might be unknown to readers without specific background in natural science. Also the wide range of applications (engineering, natural science, medicine, and other disciplines) lead to a multitude of notions or interpretations of tensor values. Sec. 2.4 gives a brief overview on the different definitions of tensors to provide a broadened entry point to the specific focus of this thesis. Second order symmetric and asymmetric tensors – both subject of this thesis – are introduced and set in relation and important properties and derived quantities summarized.
Further sections cover key aspects about the interpolation of tensors (Sec. 2.5) and the topology of tensors (Sec. 2.7). Knowledge which is specific to the visualization methods presented in this thesis is also summarized: different application areas and types of tensors (Sec. 2.8), and important derived quantities of tensors (Sec. 2.9).

2.1 NOTATION

If not stated differently the following notation convention will be applied. Scalars and scalar-valued functions are denoted by lower-case letters, i.e. s and $s(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$, respectively. Points, vectors, and vector-valued functions are represented by bold lower-case letters, i.e. $\mathbf{v}$ and $\mathbf{v}(\mathbf{x})$. $\mathbf{v}^T$ denotes the transpose of vector $\mathbf{v}$. Tensors of second order, matrices, and tensor-valued functions are referred to by bold upper-case letters $\mathbf{T}$ and $\mathbf{T}(\mathbf{x})$. The identity matrix is denoted by $\mathbf{I}$ and $\mathbf{T}^T$ is the transpose of $\mathbf{T}$.
Eigenvectors and eigenvalues (Sec. 2.4.5) are important derived quantities of tensors. In this thesis they are foremost relevant for two dimensional symmetric tensors leading to two eigenvectors and two real valued eigenvalues for each tensor. Unless stated differently, eigenvalues are named $\lambda_1, \lambda_2 \in \mathbb{R}$, such that always $\lambda_1 \leqslant \lambda_2$. Accordingly, λ_1 is referred to as the *minor* eigenvalue and λ_2 as the *major* eigenvalue. Due to the direction indeterminacy the eigenvectors are denoted by $\overset{\leftrightarrow}{\mathbf{v}_i}$, with $i = 1, 2$ and $\overset{\leftrightarrow}{\mathbf{v}_i} \in \mathbb{R}^2$. The index i of an eigenvector $\overset{\leftrightarrow}{\mathbf{v}_i}$ is assigned according to its associated eigenvalue λ_i.

In all illustrations tensor lines (Sec. 2.6) integrated along the major eigenvector field are depicted in red and tensor lines of the minor eigenvector field in blue.
New terms which are introduced for the first time in the text are highlighted by italic letters.

2.2 SCALAR FIELDS

A scalar field is defined as a map $s(\mathbf{x})$, $\mathbf{x} \in D$ and $D \subset \mathbb{R}^n$ that assigns a scalar to each point in the domain D. In scientific visualization scalar fields are ubiquitous, either as primary scientific results but also as characteristic derived scalar quantities of vector and tensor fields.

Gradient
The *gradient* of a smooth scalar field $s(\mathbf{x})$ is defined by the first order spatial derivative of the scalar field. It is given by

$$\nabla s = (\frac{\partial s}{\partial x_1}, .., \frac{\partial s}{\partial x_n}), \tag{1}$$

with $\frac{\partial}{\partial x_i}$ denoting the partial derivative in the i-th component of $\mathbf{x}$. The gradient points towards the direction of strongest change in the scalar field.

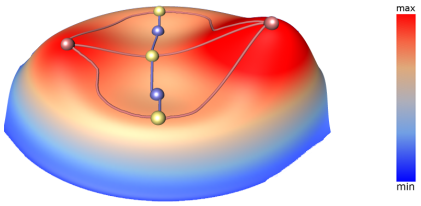
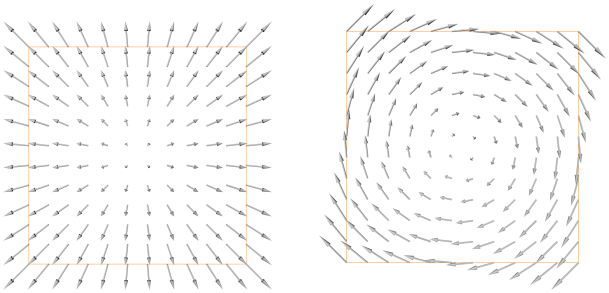


Figure 5: Topology of a scalar field (represented as height field). Critical points are given as small spheres – maxima in red, minima in blue and saddle points in yellow. Separatrices are depicted as connecting lines. Image courtesy: Jens Kasten [Kas12].

Topology
The topology of scalar fields is represented by their *critical points* (minima, maxima and saddle points) and their connectivity, see also Fig. 5. The connectivity is defined by *separatrices* which are specific curves everywhere tangential to the gradient field of $s(\mathbf{x})$. They segment the field into regions where the scalar function is monotonic. Imagining the scalar field as height field these curves would be the path of a ball rolling down a hill. The topology can be computed discretely based on Morse-Smale theory [Mil63] or numerically based on the gradient vector field of $s(\mathbf{x})$.



(a) $\mathbf{v}(\mathbf{x}) = (x, y)$ (b) $\mathbf{v}(\mathbf{x}) = (-y, x)$

Figure 6: Two examples of a discrete vector field given as arrow plots – orientation and magnitude illustrated by direction and length of arrows.

2.3 VECTORS AND VECTOR FIELDS

A vector field is defined as a map $\mathbf{v}(\mathbf{x})$, $\mathbf{x} \in D$ and $D \subset \mathbb{R}^n$ that assigns an n-dimensional vector $\mathbf{v} = (v_1, .., v_n)$ to each point in the domain D (see Fig. 6 for two simple examples). We assume the reader to be familiar with the basics of vector calculus. For an introduction please refer to [MT88] for example. Unless normalized, vectors encode a direction and a magnitude. Vector fields occur in many scientific disciplines, for example as displacement fields, flow fields, or as gradients of scalar fields.

Stream lines

Integrating line structures along a vector field is a common approach to give a continuous impression of the directions in the field. For flow visualization a popular type of such line structures are *stream lines* which provide a snapshot at a specific time step in the vector field. They are tangential everywhere to the vector field and give a sense of the instantaneous flow directions. Decisive criteria besides the precision of the stream line computation approach are the chosen interpolation model and the placement strategy. The definition and the computation approaches of stream lines and tensor lines (Sec. 2.6) are almost equal. Therefore these details are given in the vein of tensor lines (Sec. 2.6).

Gradient

The *vector gradient tensor* of a n-dimensional vector field $\mathbf{v}(\mathbf{x})$, also called the *Jacobian*, is the first order spatial derivative and defined by:

$$\mathbf{T} = \left(\frac{\partial v_i}{\partial x_j} \right), i, j \in \{1, n\}. \tag{2}$$

The vector gradient tensor is independent of the chosen frame of reference for $\mathbf{v}(\mathbf{x})$. This property is of especial importance in the area of

fluid mechanics and will be revisited at the end of this section and in Sec. 2.8.3.

Divergence and vorticity
Both, divergence and vorticity are important quantities that can be intuitively understood by looking at a vector field as a flow of a fluid or gas. However, they are also used for other data types like images, meshes, and scalar and tensor fields.

Divergence for a vector field $\mathbf{v}(\mathbf{x}) : D \to \mathbb{R}^n$ with $n \in 2, 3$ can be defined as:

$$\operatorname{div} \mathbf{v} = \sum_{i=1}^{n} \frac{\partial v_i}{\partial x_i}. \tag{3}$$

If a flow field $\mathbf{v}(\mathbf{x})$ is considered as transporting mass then a positive divergence value at a specific location can be interpreted that mass spreads outward from that point. A negative divergence value indicates a drain location and zero divergence that mass is transported with neither loss nor gain. A vector field that is *divergence free* has neither sources and nor sinks of mass.
Please note, divergence is exactly the isotropic scaling quantity (apart from a constant factor) which will be defined for asymmetric tensors in Eq. 14 .

Vorticity, also called *curl*, for a three dimensional vector field $\mathbf{v}(\mathbf{x}) : D \to \mathbb{R}^3$ can be defined as:

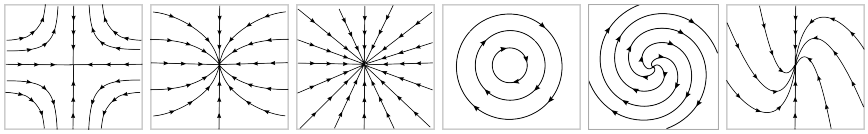
$$\operatorname{rot} \mathbf{v} = \left(\frac{\partial v_3}{\partial x_2} - \frac{\partial v_2}{\partial x_3}, \frac{\partial v_1}{\partial x_3} - \frac{\partial v_3}{\partial x_1}, \frac{\partial v_2}{\partial x_1} - \frac{\partial v_1}{\partial x_2}\right). \tag{4}$$

The resulting vector is related to the axis of strongest rotation at a specific location and its magnitude gives the speed of angular rotation around rot $\mathbf{v}$.
For two dimensional vector fields $\mathbf{v}(\mathbf{x}) : D \to \mathbb{R}^2$ vorticity is defined as scalar quantity:

$$\operatorname{rot} \mathbf{v} = \frac{\partial v_2}{\partial x_1} - \frac{\partial v_1}{\partial x_2}. \tag{5}$$

For the two dimensional case, vorticity is exactly the rotation quantity which will be defined in Eq. 14, apart from a constant scaling factor.

Trying to perceive divergence and rotation from vector illustrations only might be misleading. Looking at Fig. 2.6(a) it seems obvious that the flow is expanding as it spreads out from the center point. However, if the flow would be slowing down as it moves outward we would have a negative divergence value although the vectors point outward and might suggest otherwise.



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

Figure 7: Representatives of first order critical points in a linear vector fields: (a) saddle point, (b) node, (c) focus point, (d) center, (e) spiral, and (f) improper node.

Topology

The topology of vector fields is again given by *critical points* and *separatrices*. The critical points are defined by the zeros in the field $\mathbf{v}(\mathbf{x}) = 0$ and can be characterized by analysis of their gradient tensor (Eq. 2). For two dimensional stationary vector fields the following types of first order critical points occur (Fig. 7): saddle points, nodes, focus points, centers, spirals, and improper nodes. The separatrices are found by integrating stream lines from saddle points. They partition the domain into regions which are topologically equivalent to uniform flow.

Dependence on chosen frame of reference

For vector fields in the application area of fluid mechanics it is of special importance that the interpretation of vector fields depends on the chosen view or frame of reference they are represented in. Two common perspectives on vector fields are the *Lagrangian* and the *Eulerian* view. For the Lagrangian view the field is regarded from a perspective of a particle moving along with the flow, i.e. position or velocity are associated with the particle. In contrast, the Eulerian perspective assumes an observer with a fixed position on a given mesh. Individual particles play no role and properties like position and velocity are associated with the spatial location in the mesh. Changing these perspectives can lead to strongly differing results for the same vector field.

Another important notion is *Galilean invariance* which originates in fluid mechanics. It denotes the quality of some field being independent of the change of reference frame by a relative constant (also called *Galilean transformation*). Tensors like the vector gradient tensor (Eq. 2 and Sec. 2.8.3) or the rate of strain tensor (Eq. 11 and Sec. 2.8.4) in contrast are invariant under the change of coordinate system and consequently Galilean invariant. Misinterpretations due to different perspectives cannot arise. This induces a strong importance of such tensors in the area of fluid mechanics.

This is very compact introduction to vector fields. For further reading on vector fields and their visualization please refer to Telea [Tel08]

for an introductory summary. Also the state of the art reports given in the Related Work chapter (Chap. 3) give a good introduction to vector field visualization for specific application areas.

2.4 TENSORS AND TENSOR FIELDS

Tensors are multi-linear functions that allow to encode intricate phenomena in natural science, engineering, and other disciplines independently of a chosen frame of reference. Tensors generalize and extend the concept of scalars, vectors and matrices. They provide a framework that is highly generic and concise for the formulation of scientific questions. In fact, tensors of order zero are scalars and tensors of order one are vectors. This thesis is concerned with tensors of order two, also called *second order* tensors, which allow to encode anisotropic properties which vary as function of direction.
In this section, first different ways to define a tensor are given to allow access to a broader audience. The subsequent information is then restrained to two dimensional tensors of second order defined on planar or curved surfaces which are primarily treated in this work. Basic concepts and properties of second order tensors are introduced. For further reading extensive introductions about tensors can be found in [AMR88] and [Lan12]. Additionally, more information about tensors from an engineering point of view is provided in [Bra03] and [Dan97] .

2.4.1 *Definition of Tensors*

Due to the power and flexibility of tensors one might encounter various definition formalisms, dependent on the discipline or application area. These definitions describe the same concept and can be equally transferred into another. In the following two common abstract definitions are introduced and along that delimited to the aspects that are relevant for this work. The first definition is that of a tensor as a multilinear map which is highly generic and the second is the definition of a tensor as a linear operator with a possibly clearer interpretation for a reader new to tensor calculus. Both allow to describe a phenomenon without specifying an explicit coordinate basis. Finally, a third definition of a tensor in indicial notation is given which is less generic but occurs frequently as visualization input data.

Tensors as multi-linear maps

$$\mathbf{T} : \underbrace{V^* \otimes .. \otimes V^*}_{n} \otimes \underbrace{V \otimes .. \otimes V}_{m} \to \mathbb{R} \tag{6}$$

where V is a vector space, V^* its corresponding dual space, and $\otimes$ denotes the tensor product. $\mathbf{T}$ is called a tensor with *valence* (n,m) and

of *order* $r = n + m$. Please note, the order of a tensor is independent of the dimension of the given vector space. The order of a tensor denotes the number of (vector and dual) spaces it acts on.
In this thesis V is always identified with Cartesian space, therefore V and V^* do not have to be distinguished explicitly. Related to this is the notion of co- and contra-variance, which can be omitted in this case, as well.
Also the tensors occurring in this thesis are of no order higher than two. When referring to tensors in this text, tensors of second order are denoted.

Tensors as linear operator
The following definition of a tensor as linear operator can be generalized to a definition equally powerful to that of the multi-linear map. However, here we restrict it to second-order tensors and assume V as Cartesian space:

$$\mathbf{T} : V \to V \tag{7}$$

This definition is frequent in engineering and physics. It interprets the tensor as a linear operator that maps one input vector to a result vector.

Tensors in indicial notation
In visualization mostly tensors defined by indicial notation appear. For a beginner, this definition might be more apprehensive. However, it requires a fixed coordinate basis and is less generic. In this vein a second order tensor $\mathbf{T} \in \mathbb{R}^{n \times n}$ can be represented by a $(n \times n)$ square matrix (often also referred to as multidimensional array).

$$\mathbf{T} = T_{ij}, i, j \in \{1, .., n\} \tag{8}$$

This thesis is concerned with two dimensional tensors, which leads to a 2×2 matrix:

$$\mathbf{T} = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix}. \tag{9}$$

2.4.2 *Symmetry of a Tensor*

Given a tensor $\mathbf{T}$ in indicial notation (Eq. 8). $\mathbf{T}$ is called

- *symmetric* if $t_{ij} = t_{ji}$,
- *asymmetric* if $t_{ij} \neq t_{ji}$, and
- *anti-symmetric* or *skew-symmetric* if $t_{ij} = -t_{ji}$.

Symmetric two dimensional tensors are defined by the three independent scalars t_{11}, t_{22}, $t_{12} = t_{21}$. For readers unfamiliar with tensors the effect of a transformation encoded by a two dimensional

symmetric tensor can be depicted by a simple icon in form of a circle that is deformed to an ellipse by a tensor (see Fig. 8).
Asymmetric tensors are defined by all four scalars $t_{11}, t_{22}, t_{12}, t_{21}$ and in contrast to their symmetric counterpart incomparably more complex to handle. A straightforward intuition about the information encoded by an asymmetric tensor cannot be given like for the symmetric case. Therefore existing work in visualization for asymmetric tensors mostly applies some decomposition either to the tensors themselves (see e.g. [ZYLL09] and Sec. 2.4.4 below) or to the tensor field (see e.g. [ZPP05] and Sec. 2.7.2).

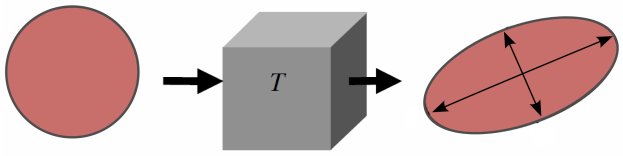


Figure 8: Illustration of the tensor concept: result (ellipse) of applying a tensor to an isotropic element (circle). The resulting eigenvectors are depicted as arrows and the eigenvalues are reflected by the scaling of the arrows. For the isotopic element the eigenvectors are not uniquely defined and every vector is an eigenvector.

2.4.3 *Definiteness of a Tensor*

Given a tensor $\mathbf{T}$. Then if for *all* non-zero vectors $\mathbf{v}$

$$\begin{aligned} \mathbf{v}^T\mathbf{T}\mathbf{v} > 0 \quad & \mathbf{T} \text{ is called } \textit{positive definite}, \\ \mathbf{v}^T\mathbf{T}\mathbf{v} \geqslant 0 \quad & \mathbf{T} \text{ is called } \textit{positive semi-definite}, \\ \mathbf{v}^T\mathbf{T}\mathbf{v} \in \mathbb{R} \quad & \mathbf{T} \text{ is called } \textit{indefinite}, \end{aligned}$$

where $\mathbf{v}^T$ denotes the transpose of $\mathbf{v}$. The notion for negative (semi-) definiteness can be derived of positive (semi-) definiteness, only that the sign has to be changed.

Properties
Positive definite tensors have eigenvalues (Sec. 2.4.5) and a determinant (Eq. 32) greater than zero. For semi-positive definite tensors the same holds for greater or equal than zero. For indefinite tensors non of these assumptions can be made which makes the development of visualization methods more challenging.

2.4.4 *Decomposition of Tensors*

Tensors can be decomposed into unique components to alleviate their processing and interpretation.

2.4.4.1 *Symmetric Tensors*

Naturally symmetric tensors cannot contain a rotational component. However, they can be decomposed uniquely into an isotropic part **D** and a deviatoric part **A**:

$$\mathbf{T} = \mathbf{D} + \mathbf{A} = \begin{pmatrix} \frac{t_{11}+t_{22}}{2} & 0 \\ 0 & \frac{t_{11}+t_{22}}{2} \end{pmatrix} + \begin{pmatrix} \frac{t_{11}-t_{22}}{2} & t_{12} \\ t_{12} & -\frac{t_{11}-t_{22}}{2} \end{pmatrix}. \quad (10)$$

Please note, frequently **D** is also used to denote the deviatoric part of a tensor. In this thesis however, we adopt the notion of Zhang et al. [ZYLL09] and identify the isotropic part with **D**.

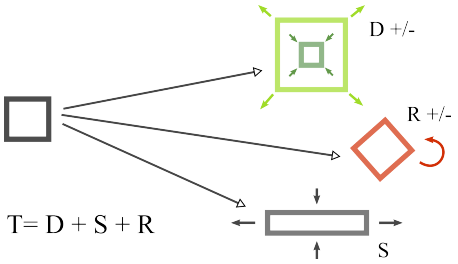


Figure 9: Schematic illustration of the asymmetric tensor decomposition into isotropic scaling, rotation, and shear components. In the image the individual components are applied to the isotropic element on the left side.

2.4.4.2 *Asymmetric Tensors*

A popular approach decomposes asymmetric tensors uniquely into the sum of a symmetric **Q** and an antisymmetric **R** component:

$$\begin{aligned} \mathbf{T} &= \mathbf{Q} + \mathbf{R} \\ &= (\mathbf{T} + \mathbf{T}^T)/2 + (\mathbf{T} - \mathbf{T}^T)/2, \end{aligned} \quad (11)$$

where $\mathbf{T}^T$ is the transpose of **T**. The matrix **Q** measures the scaling and shear, and **R** the rotation induced by the tensor. They are called *rate of strain* and *vorticity tensor*, respectively (see also Sec. 2.8.4).

Further splitting **Q** (Eq. 10) leads to the following equation:

$$\begin{aligned} \mathbf{T} &= \mathbf{D} + \mathbf{S} + \mathbf{R} \\ &= \gamma_d \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \gamma_s \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix} + \gamma_r \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \end{aligned} \quad (12)$$

with

$$\begin{aligned} \gamma_d &= (t_{11} + t_{22})/2, \\ \gamma_s &= \sqrt{(t_{11} - t_{22})^2 + (t_{12} + t_{21})^2}/2, \\ \gamma_r &= (t_{21} - t_{12})/2. \end{aligned} \quad (13)$$

The scalar θ is the orientation of the stretching and is given by the angular component of the vector

$$\begin{pmatrix} t_{11} - t_{22} \\ t_{12} + t_{21} \end{pmatrix}.$$

In the following, these components are denoted that $\mathbf{D}$ represents the *isotropic scaling*, $\mathbf{S}$ the *pure shear*, and $\mathbf{R}$ the *rotational* component of $\mathbf{T}$. The components of the decomposition are illustrated in Fig. 9.

2.4.5 *Eigenvalues and Eigenvectors*

A tensor $\mathbf{T}$ can be represented uniquely by its *eigenvalues* λ_1 and λ_2 and corresponding *eigenvectors* $\overleftrightarrow{\mathbf{v_1}}$ and $\overleftrightarrow{\mathbf{v_2}}$. The eigenvalues $\lambda_i \in \mathbb{R}$, $i \in \{1, 2\}$ of a tensor $\mathbf{T}$ are defined by the solution of the characteristic equation:

$$\det(\mathbf{T} - \lambda_i \mathbf{I}) = 0, \tag{14}$$

where det denotes the determinant of $\mathbf{T}$ (Eq. 32) and $\mathbf{I} \in \mathbb{R}^{2x2}$ the identity matrix. Solving the following equation delivers the eigenvectors $\overleftrightarrow{\mathbf{v_i}}$.

$$(\mathbf{T} - \lambda_i \mathbf{I})\overleftrightarrow{\mathbf{v_i}} = 0. \tag{15}$$

In consequence, the eigensystem of a linear system defined by $\mathbf{T}$ is given by:

$$\mathbf{T} \cdot \overleftrightarrow{\mathbf{v_i}} = \lambda_i \cdot \overleftrightarrow{\mathbf{v_i}}. \tag{16}$$

The eigenvectors represent the directions of maximal variation encoded by the tensor, the eigenvalues give the maximal magnitude of this variation, sometimes also referred to as amplitude (for symmetric tensors see Fig. 8, for asymmetric tensors Fig. 10).

2.4.5.1 *Symmetric Tensors*

In contrast to asymmetric tensors, the eigenvalues of symmetric tensors are always real valued and eigenvectors mutually orthogonal. Positive eigenvalues allude to expansion and negative eigenvalues to compression at a specific location. Since the multiplication of an eigenvector by any non-zero scalar yields an additional eigenvector, eigenvectors should be considered without norm and orientation. The orientation indeterminacy is indicated by the double arrows above $\overleftrightarrow{\mathbf{v_i}}$. As already mentioned above, eigenvalues are named such that always $\lambda_1 \leqslant \lambda_2$, associated eigenvectors $\overleftrightarrow{\mathbf{v_i}}$ are indexed accordingly.

2.4.5.2 *Asymmetric Tensors*

Eigenvectors for asymmetric tensors are in general not orthogonal and eigenvalues may be complex. Depending on the eigenvalues a domain D can be decomposed into real domains (eigenvalues are real valued), degenerate lines (eigenvalues are equal), and complex domains (eigenvalues are complex valued). The degenerate lines form the boundaries between the real and complex domains. Along the degenerate lines the eigenvectors coalesce. In complex domains no real eigenvectors exist, therefore alternative directional clues have been developed as continuous extensions of eigenvectors into the complex domain.

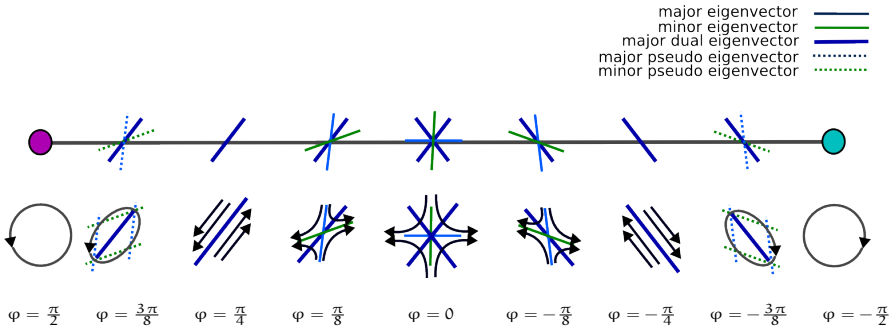


Figure 10: Joint visualization of vectors and derived eigenvector types (example vectors depicted as black (curved) arrows). Evolution of eigenvectors, dual eigenvectors, and pseudoeigenvectors for the exemplary case of $\theta = 0$ (see Eq. 18). Circular point (Eq. 28) at $\varphi = \pi/2$ indicating purely circular behavior of the vectors. For $\varphi = 3\pi/8$ dual eigenvectors and pseudoeigenvectors depict the elliptical flow behavior. In $\varphi = \pi/4$ the eigenvectors coalesce, for $\varphi = \pi/8$ the eigenvectors are not orthogonal, and in $\varphi = 0$ the tensor is symmetric and eigenvectors are orthogonal.

Zheng et al. [ZPP05] introduce *dual eigenvectors*. In the real domain dual eigenvectors are the bisectors of the eigenvectors (Fig. 10). From the viewpoint of flow visualization an intuition about dual eigenvectors in the complex domain is given by interpreting them as the axes of elliptical patterns a flow forms in this regions. The dual eigenvectors $\overset{\leftrightarrow}{\mathbf{j}_1}$ (minor), $\overset{\leftrightarrow}{\mathbf{j}_2}$ (major) are computed in dependence of the eigenvectors $\overset{\leftrightarrow}{\mathbf{v}_i}$ and eigenvalues λ_i according to the following relationship:

$$\overset{\leftrightarrow}{\mathbf{v}_i} = \begin{cases} \sqrt{\lambda_2}\,\overset{\leftrightarrow}{\mathbf{j}_2} \quad \pm \quad \sqrt{\lambda_1}\,\overset{\leftrightarrow}{\mathbf{j}_1}, & \mathbf{T} \text{ in real domain,} \\ \sqrt{\lambda_2}\,\overset{\leftrightarrow}{\mathbf{j}_2} \quad \pm \quad i\sqrt{\lambda_1}\,\overset{\leftrightarrow}{\mathbf{j}_1} & \mathbf{T} \text{ in complex domain.} \end{cases} \tag{17}$$

In addition to the dual eigenvectors Zhang et al. [ZYLL09] introduce *pseudoeigenvectors*. Again the elliptical flow pattern given by a tensor $\mathbf{T}$ shall be represented (Fig. 10). Here, the direction of the pseudoeigenvectors is defined by the minimal enclosing diamond of the ellipse pattern. Since the tensor magnitude and the isotropic scaling

component have no influence on the eigenvectors only the unit traceless tensor is regarded. The tensor $\mathbf{T}$ is re-parametrized as follows:

$$\mathbf{T}(\theta,\varphi) = \sin\varphi \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} + \cos\varphi \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix}, \tag{18}$$

with $\varphi = \arctan\left(\frac{\gamma_r}{\gamma_s}\right)$.
For $\mathbf{T} = \mathbf{T}(\theta,\varphi)$ the *major pseudoeigenvector* is defined as the minor eigenvector of the tensor $\mathbf{T}(\theta, \frac{\pi}{2} - \varphi)$ if $\varphi > \frac{\pi}{4}$ and $\mathbf{T}(\theta, -\frac{\pi}{2} - \varphi)$ if $\varphi < -\frac{\pi}{4}$. The *minor pseudoeigenvector* is defined as the major eigenvector of the above given tensors under the same conditions.

Eigenvector and eigenvalue manifold
Zhang et al. [ZYLL09] expose that the eigenvectors of a tensor $\mathbf{T}$ only depend on γ_r, γ_s, and θ and the eigenvalues can be defined by γ_d, γ_r, and γ_s. They propose to study the occurring strengths by the inspection of *unit tensors*, in their work defined by $\gamma_d^2 + \gamma_r^2 + \gamma_s^2 = 1$. Accordingly, the *eigenvalue manifold* $\mathbf{M}_\lambda$ and the *eigenvector manifold* $\mathbf{M}_\mathbf{v}$ can be defined as follows:

$$\mathbf{M}_\lambda = \{(\gamma_d,\gamma_r,\gamma_s) \mid \gamma_d^2 + \gamma_r^2 + \gamma_s^2 = 1 \text{ and } \gamma_s \geqslant 0\}, \tag{19}$$

and

$$\mathbf{M}_\mathbf{v} = \{(\gamma_r,\gamma_s,\theta) \mid \gamma_r^2 + \gamma_s^2 = 1 \text{ and } \gamma_s \geqslant 0 \text{ and } 0 \leqslant \theta \leqslant 2\pi\}, \tag{20}$$

These manifolds are used to classify the tensor field by dominant behavior. The eigenvalue manifold $\mathbf{M}_\lambda$ in particular will be used in Chap. 8.

2.5 TENSOR INTERPOLATION

Interpolation is a fundamental process in scientific visualization. The definition of scalar, vector, and tensor fields in visualization requires a continuous model of the given fields. However, in visualization they are mostly given as sets of discrete data points with associated values. This happens because the data is measured at discrete locations or simulations results are written out on various grid types, for example. Still, most visualization methods require a model for continuous reconstruction of the data. One very simple reason is for data given on a grid: visualizing it only at the grid points can introduce a pattern by the structure of the grid itself and lead to a misconception of the data (e.g. see recapitulation of works in glyph placement in Sec 3.2.2). Also the concept of topology which has high importance in this work affords a continuous model of the underlying data. Depending on the data and the application the reconstruction can be a

sensitive process. Ideally, the chosen interpolation model optimally approximates the properties of the original data and function. Here, the choice of the interpolation method strongly influences the quality of reconstruction.
Although using a higher order interpolation scheme guarantees the continuity of derivatives (up to the chosen order of interpolation) in visualization mostly linear interpolation is used. This is efficient and also sufficient in most of the cases. Still, especially for data of higher dimension (e.g. vectors, tensors) it has several drawbacks. In the following we address the central issues related to this thesis. A more general discussion of tensor interpolation can be found in [KASH12].
Foremost, by linear interpolation single features of higher order decay to multiple features of linear order, which limits the clarity of representation. In general, the interpolation of tensors requires special care. Interpolating the data linearly in its components may not reflect the nature of the initial data and introduce artifacts, such as the *swelling effect*. Further and more grave, inconsistencies can arise. Special attention has to be given, e.g. to positive-definite tensors. But also for indefinite tensors inconsistencies can be introduced if the interpolation does not respect the topology.

2.6 TENSOR LINES AND INTEGRATION METHODS

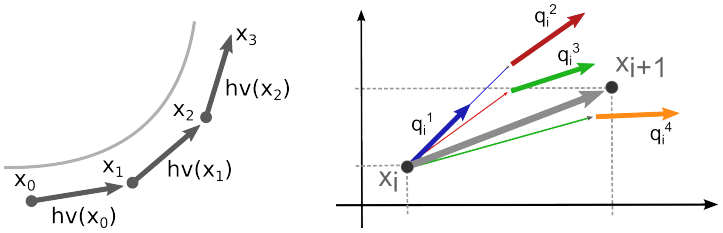
Tensor lines are curves tangential everywhere to the eigenvector fields which are computed for one specific time step of the tensor field. The integration of the eigenvector fields results in two orthogonal families of continuous curves. In the following these curves will be called *major* and *minor tensor lines* according to the eigenvector field integrated. Tensor lines are strongly related to the concept of stream lines for vector fields (Sec. 2.3), only that for tensor lines the integration has to be kept consistent by orienting the bidirectional eigenvectors into integration direction. Thus, the theory described below can be used for both types of integral lines, tensor and stream lines.
In general, the integral lines cannot be described by an analytic expression, but they can be derived by ordinary differential equations. Let $\mathbf{v}(\mathbf{x})$ be a Lipschitz continuous vector or eigenvector field. The integral curve c is parametrized by $s \in \mathbb{R}$ with

$$\begin{aligned} c : I &\rightarrow D,\ 0 \in I \subset \mathbb{R},\ I \text{ interval} \\ s &\rightarrow c(s) \end{aligned}$$

and for a specific time step t_0

$$\frac{\partial c}{\partial s} = \mathbf{v}(c(s), t_0). \tag{21}$$

The solution to Eq. 21 can be obtained by formulating it as an initial value problem. Several numerical techniques exist to compute a numerical approximation of this initial value problem. We will shortly summarize two approaches. Let $\mathbf{x_0}$ be the starting point of the integration and $h > 0$ a chosen step width.



(a) Euler (b) Runge Kutta

Figure 11: (a) Euler integration of first order (three integration steps). (b) Runge Kutta 4th order integration with evaluation of the vector field at four sample points (one integration step).

Euler integration

The Euler integration is a first order numerical procedure (Fig. 2.11(a)). It is defined by

$$\mathbf{x_{i+1}} = \mathbf{x_i} + h\mathbf{v}(\mathbf{x_i}) \tag{22}$$

The Euler integration is rather simple as it only requires one evaluation of the (eigen-) vector field per integration step. Naturally, this delimits the accuracy of the integration scheme.

Runge Kutta integration

A refined integration scheme is the family of Runge Kutta integrators. Runge Kutta integration exists of arbitrary order, where the order denotes the number of sample points for the evaluation of the function. In this thesis integration of fourth order (Fig. 2.11(b)) will be used as we consider it as a good trade off between accuracy and computational costs:

$$\begin{aligned}
\mathbf{q_i^1} &= \mathbf{v}(\mathbf{x_i}) \\
\mathbf{q_i^2} &= \mathbf{v}(\mathbf{x_i} + h/2\ \mathbf{q_i^1}) \\
\mathbf{q_i^3} &= \mathbf{v}(\mathbf{x_i} + h/2\ \mathbf{q_i^2}) \\
\mathbf{q_i^4} &= \mathbf{v}(\mathbf{x_i} + h\mathbf{q^3}) \\
\mathbf{x_{i+1}} &= \mathbf{x_i} + h/6\ (\mathbf{q_i^1} + 2\mathbf{q_i^2} + 2\mathbf{q_i^3} + \mathbf{q_i^4})
\end{aligned} \tag{23}$$

Further, the integration schemes can be combined with step size adaptation. Here, the step size is refined (e.g. divided by two) until the angular difference of the evaluated (eigen-)vectors is less than a predefined threshold. Stream and tensor lines of an individual field

are uniquely defined and cannot intersect each others. However, as they are approximated numerically intersections may occur and if required, need to be treated explicitly.

2.7 TOPOLOGY OF TWO DIMENSIONAL TENSOR FIELDS

In visualization and data analysis topological methods are in general associated with segmentation algorithms. Two classes can be determined for the segmentation of a tensor field. First, the algorithms that generalize common segmentation and clustering techniques known from image processing. These approaches require the definition of appropriate dis-/similarity measures for tensor fields which offers a certain flexibility towards the quantity steering the segmentation process. The entire tensor, but also any plausible derived quantity can be used to partition the field according to the required properties. These methods are mostly applied in diffusion tensor imaging (DTI). For an overview of existing methods see Sec. 3.2.1.
The second class of algorithms are those that decompose the tensor field by extracting the topology. This yields a structural segmentation that is purely based on the eigenvector characteristics of the field.

2.7.1 *Symmetric Tensor Fields*

The topology of two dimensional symmetric tensor fields plays a major role in this thesis. Hence, it will be introduced in more detail.

2.7.1.1 *Topological Graph*

The fundamental topological elements are represented by the *topological graph* or *topological skeleton* which is constituted by:

- *Degenerate elements* (Sec. 2.7.1.2): Locations in the field where the tensor's eigenvalues are equal and the dimension of the eigenvector space is larger than one. For two-dimensional fields, the generic case are degenerate points. They can be classified according to the behavior of the eigenvector field in their vicinity. Typical patterns are shown in (Figure 12).

- *Separatrices* (Sec. 2.7.1.3): Distinctive tensor lines (Sec. 2.6) that emerge from the degenerate elements and that are tangent to the eigenvector fields everywhere. Separatrices segment the tensor field such that they bound regions of qualitative homogeneous eigenvector behavior. Fig. 12 shows the separatrices of both eigenvector fields as bold blue and red lines.

Please note, to complete the topology definition also closed tensor lines would need to be considered. In practice however, these are very rare and will not be considered in the presented methods as such.

2.7.1.2 *Degenerate Elements*

Common definition

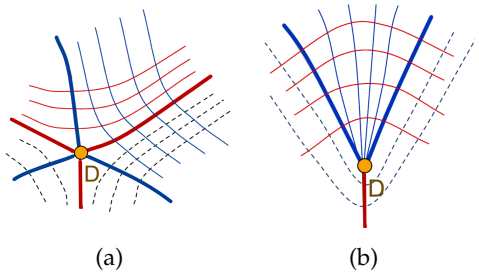
Degenerate points are the basic singularities of the tensor field topology. Depending on the application area degenerate points are also called *umbilics* or *isotropic points*. In these locations the tensor behavior is purely isotropic: the two eigenvalues of the tensor are identical $\lambda_1 = \lambda_2$, and the eigenvectors are no longer uniquely defined. Therefore the tensor is proportional to the identity matrix and all vectors are eigenvectors. Please note, the actual magnitude of the eigenvalues is not of importance for degenerate points. Degenerate points in tensor fields correspond to critical points in vector fields. Due to direction indeterminacy of tensor lines, however, these points exhibit structures different from those seen in vector fields.

Tensor index

The existence of an isolated degenerate point can be detected and the degenerate point categorized by computing its *tensor index*. This is the number of windings an eigenvector performs when encompassing it on a *Jordan curve* [Del94] in counterclockwise direction. A Jordan curve is homeomorphic to a circle, e.g. a closed, smooth, and simple curve. The undirected eigenvector field allows winding-numbers to be multiples of one half. The closed curve can be encompassed arbitrarily close to the degenerate point. Accordingly, the tensor index can also be directly assigned to an isolated degenerate point.
In a tensor field defined by linear interpolation only isolated degenerate points which are simple (tensor index of $\pm\frac{1}{2}$) can exist. A degenerate point with a tensor index of $-\frac{1}{2}$ is called *trisector*; for a tensor index of $\frac{1}{2}$ it is called *wedge point* (see Figure 12).

Multiple degenerate points

In general, *multiple degenerate points* which are degenerate points of



(a) (b)

Figure 12: Basic first order degenerate points for 2D tensor fields: (a) trisector, and (b) wedge. Separatrices for both eigenvector fields are drawn as bold red or blue lines, respectively. Exemplary integrated tensor lines (thin lines) emphasize how the eigenvector behavior within can be reconstructed by the bounding separatrices.

higher order do not occur in linear tensor fields. However, in very rare cases or for artificial data sets they can occur as one of the initially given tensors of the discrete input tensor field (e.g. on the grid vertices). For these points the neighborhood is no longer linear but piecewise linear. Their characterization is defined by $\delta \in \mathbb{R}$, which can be computed in a simplified approach [Del94] by

$$\delta = ac - bd, \tag{24}$$

with

$$\begin{aligned} a &\equiv \tfrac{1}{2}\tfrac{\partial(t_{11}-t_{22})}{\partial x} & b &\equiv \tfrac{1}{2}\tfrac{\partial(t_{11}-t_{22})}{\partial y} \\ c &\equiv \tfrac{\partial t_{12}}{\partial x} & d &\equiv \tfrac{\partial t_{12}}{\partial y}. \end{aligned} \tag{25}$$

at the location of the degenerate point x_0. For $\delta = 0$ the degenerate point is of higher order. To simplify the analysis of multiple degenerate points, Delmarcelle [Del94] proposed the physical formalism of the *far field*. Here, a multiple degenerate point of index I_T regarded in the far field is equivalent to the sum of simple degenerate points whose indices sum up to I_T:

$$W + T = I_T, \tag{26}$$

with W and T the number of wedges and trisectors, respectively.

Exact location of linear degenerate points

To compute the location of degenerate points based on a componentwise interpolation model on a linear planar field only the deviator tensor A (see Eq. 10) has to be considered as the topology is purely structural and the scaling component D has no influence on it [Tri02]. Looking for the zeros of the deviator field, can be solved e.g. by computing the zeros of the functions defined over the tensor components.

For the interested reader, the tensor index of an orientable surface is directly related to the properties of the surface itself. Precisely speaking, the tensor index of an orientable surface with tangent tensor field $\mathbf{T}(\mathbf{x})$ is equal to the Euler characteristic of the surface. This property holds for any tensor field $\mathbf{T}(\mathbf{x})$.

2.7.1.3 *Separatrices*

Common definition

The behavior of the tensor lines in the vicinity of degenerate points follows certain characteristic patterns. Although in the degenerate elements the eigenvectors are not uniquely defined specific tensor lines can be determined. These tensor lines emerge radially from the degeneracy and partition the vicinity into sectors of qualitatively homogeneous eigenvector behavior (see Fig. 22): *hyperbolic*, *elliptic*, or *radial* sectors.

Traditionally only radial tensor lines bounding hyperbolic sectors are called *separatrices* and constitute the edges of the topological graph. However, in this work we also consider the radial tensor lines as structural relevant which bound elliptic sectors (Sec. 4.3) .

Traditional approach - computation
In the approach based on component-wise interpolation the exact directions of the separatrices s_i from a degenerate point are computed in terms of angles θ_i between the x-axis and the separatrices s_i. For simple degenerate points all radial directions are delivered by computing the real roots z_k (including infinity) of the following cubic equation:

$$dz^3 + (t_{21} + 2t_{12})z^2 + (2t_{11} - t_{22})z - c = 0, \tag{27}$$

with $dz_k = tan\theta_k$. Classifying the sectors bounded by the radial lines and only keeping those radial lines that coincide with the boundary of a structurally relevant sector finally delivers all separatrices.
The actual classification is explained in detail in Sec. 4.3.4.

2.7.2 *Asymmetric Tensor Fields*

Here, the topological features of asymmetric tensor fields are briefly summarized, for more detail please refer to the work of Zheng et al. [ZPP05], for example. Degenerate lines where eigenvalues are equal, and eigenvectors coalesce constitute an important topological feature. As mentioned before, degenerate lines form the boundaries between complex and real domains. Like this an asymmetric tensor field can be decomposed into regions where tensors have a stretching effect (real domains) and into regions where tensors have a swirling effect (complex domains). Degenerate points are referred to as *circular points*. Circular points are defined by

$$(t_{11} - t_{22})^2 + (t_{12} + t_{21})^2 = 0. \tag{28}$$

In fluid mechanics these locations indicate purely circular flow patterns.

2.8 AREAS OF APPLICATION AND TYPES OF TENSORS

Tensors occur in a multitude of areas ranging from medicine to image processing. The methods presented here focus on the area of mechanics and engineering. With mechanics we summarize continuum mechanics, fluid mechanics, and geomechanics. Introducing these areas of research in depth is beyond the scope of this thesis. We will briefly characterize them and add a reference that can be used by the interested reader for further reading.

Continuum mechanics examines the internal forces in a body as reaction to external forces. The body is considered to be a continuous mass rather than a set of discrete particles (Segel [SH87]). Fluid mechanics summarizes liquids, gases, and plasmas as fluids. Especially of interest are fluid dynamics (Batchelor [Bat67]) which study fluids in motion (see also Sec. 2.8.4). Finally geomechanics (Elsoufiev [Els07]) studies forces acting within soil and rock. This is strongly related to seismology (study of earthquakes, primarily), or finds its application in geotechnical engineering like tunnel design or rock drilling.
These areas of application share the fact that important results are given by indefinite symmetric tensors and asymmetric tensors. For both types of tensors rather few visualization methods have been developed.

2.8.1 *Stress Tensor*

One very important type of tensor in mechanics is the stress tensor. It describes internal forces occurring in deformable bodies experiencing small deformations. These internal forces arise as a reaction to external forces applied to the body. For researchers an interesting issue is, if these forces can lead to structural failure. However, also liquids, and gases have stress fields. Given on infinitesimal small volume elements the stress tensor is symmetric (see Fig. 2.13(a)).

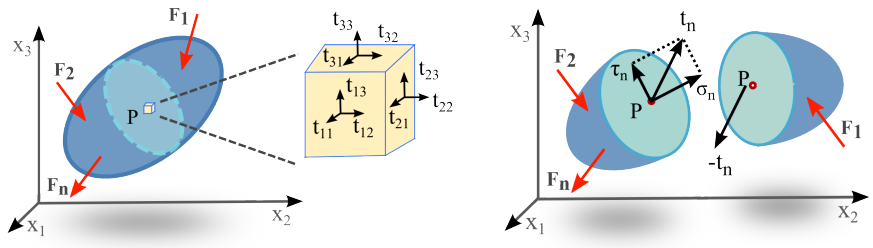


Figure 13: Forces (red arrows) acting on a deformable body. (a) Components of the resulting stress tensor defined on an infinitesimal small volumetric element. (b) Stress vector $\mathbf{t_n}$ on specific cutting plane with surface normal $\mathbf{n}$. The stress vector can be further decomposed into the shear and normal stresses τ_n and σ_n, respectively.

Measures of stress

The *principal stresses* can be directly derived from the stress tensor, they are the eigenvalues of the stress tensor. The eigenvectors are called *principal directions* in this context. Further investigation is done by inspecting stress forces on a specific cutting plane (see Figure

2.13(b)): the stress tensor maps a corresponding surface normal to a stress vector $\mathbf{t}$. Normal (σ) and shear (τ) stresses can be determined from the stress vector $\mathbf{t}$. Another important measure for shear is the maximum shear stress defined in Eq. 31 below. In this thesis we will be concerned with the principal stresses and the maximum shear stress.

2.8.2 *Strain Tensor*

Directly related to the stress tensor is the strain tensor. The strain tensor describes the deformation in terms of a change of shape due to applied forces. The stress-strain relationship is described by constitutive equations, such as Hooke's law for linear elastic materials.

2.8.3 *The Vector Gradient Tensor*

The vector gradient tensor has been formally introduced in Eq. 2. For the sake of completeness of this section we will shortly revisit it here. It can be of interest from many points of view. Inherently, any derivative of a function allows further inference on specific properties. With respect to vector fields especially another quality comes into play – the vector gradient tensor is independent of the chosen frame of reference. A pure vector based visualization might lead to misinterpretations due to the chosen frame of reference which cannot happen for the gradient tensor and its derived quantities (see also Chap. 8).

2.8.4 *Rate of Strain Tensor*

The rate of strain tensor occurs mostly in fluid dynamics and is derived from the vector gradient tensor according to the decomposition of Eq. 11. It reflects the amount of stretching and shearing in a flow. The diagonal entries of $\mathbf{Q}$ are called the normal strain rates. The non-diagonal entries are the shear strain rates.
Its asymmetric counterpart, the vorticity tensor $\mathbf{R}$, allows inference on vortical activity in a flow. In this thesis it will only be visited in terms of its scalar strength quantity γ_r (Eq. 14).

2.9 DERIVED SCALAR QUANTITIES

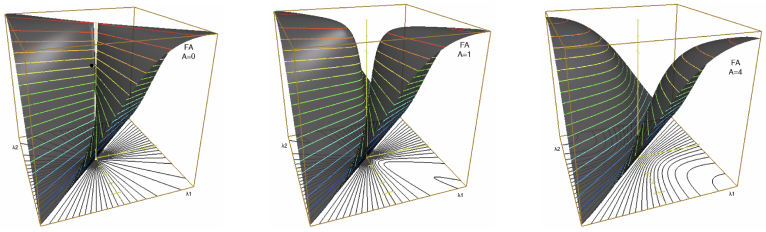
In this section we will shortly summarize derived scalar quantities of tensors which are commonly important and which will partially enrich or guide the visualization methods in the subsequent chapters.

2.9.1 *Fractional Anisotropy*

Anisotropy is commonly used as notion to describe the direction dependency of some phenomenon's properties. The opposing notion is isotropy which indicates that properties are equal in all directions. In diffusion tensor imaging (DTI) fractional anisotropy FA is a measure which describes the strength of anisotropy:

$$\mathrm{FA} = \sqrt{\frac{1}{2}\frac{(\lambda_2 - \lambda_1)^2}{\lambda_2^2 + \lambda_1^2}} \tag{29}$$

The definition results in an anisotropy measure with a range from zero to one for $\lambda_i > 0$, see also Fig. 2.14(a). The minimum value 0 is reached in the isotropic case when $\lambda_1 = \lambda_2$. The maximum value of 1 is reached for $\lambda_2 = -\lambda_1$. For $\lambda_2 = \lambda_1 = 0$ the values are undefined which poses no problem for positive definite tensors like they are used in DTI applications.



(a) (b) (c)

Figure 14: (a) Plot of the original fractional anisotropy definition over the λ_1, λ_2 plane as height functions with highlighted isolines. The function is undefined in $(0,0)$, else the values for the isotropic case $\lambda_2 = \lambda_1$ are zero. Plots in (b) and (c) show the generalized fractional anisotropy measure FA* where the discontinuity in the origin is removed with two different values for the parameter A. The values for isotropic case are still zero.

In this work we require a generalization of this measure to indefinite tensors. A way to eliminate the discontinuity close to zero is to introduce an additional positive constant which is added to the denominator.

$$\mathrm{FA}^* = \sqrt{\frac{1}{2}\frac{(\lambda_2 - \lambda_1)^2}{\lambda_2^2 + \lambda_1^2 + A^2}} \tag{30}$$

This results in low anisotropy values for tensors with eigenvalues with different sign but small absolute value, see Fig. 2.14(b), 2.14(c).

2.9.2 *Maximum Shear Stress*

Maximum shear stress S is defined by the principal stresses as

$$S = |\lambda_2 - \lambda_1| \tag{31}$$

and is related to the anisotropy. Here, absolute values are regarded. The maximum shear stress denotes the maximum difference of the principal stresses that can occur in a specific location. Thus, it is an important measure for possible material failure. Thereby the direction of most likely failure is along the bisectors of the two eigenvectors $\overset{\leftrightarrow}{\mathbf{v_1}}$ and $\overset{\leftrightarrow}{\mathbf{v_2}}$.

2.9.3 *Determinant*

For a 2×2 matrix the determinant is defined as

$$\det(\mathbf{T}) = t_{11}t_{22} - t_{12}t_{21} \tag{32}$$

If the determinant is non zero the matrix is invertible. The transformation can be characterized by the absolute value of the determinant which gives the scaling factor by which an area or volume is multiplied. The sign of the determinant indicates if the transformation is orientation preserving.

2.9.4 *Trace*

The trace of a tensor $\mathbf{T}$ is defined as the sum of its diagonal components

$$\mathrm{tr}(\mathbf{T}) = \sum_{i=1}^{n} t_{ii}. \tag{33}$$

The trace is equal to the sum of the eigenvalues. Its notion is similar to that of the determinant as it also encodes the change of size or volume.

2.10 USED GRID TYPES

All presented algorithms are developed for data sets given on a triangulated domain. Of special importance are Delaunay triangulations (see e.g. de Berg [dBvKOSoo]) that maximize the smallest angle inside a triangle. If data is processed based on a triangulation the triangles themselves introduce a bias as all points within a triangle are processed purely based on the data defined for this triangle.

Here, the Delaunay triangulation can represent the more "natural" triangulation. It reduces elongate triangles that are unfavorable (see Fig. 15). This is also important for the topology computation in Sec. 4.3 where degenerate points are inserted to the Delaunay triangulation with subsequent retriangulation. This facilitates degenerate elements of higher dimensionality and strongly reduces the complexity of the topological graph.

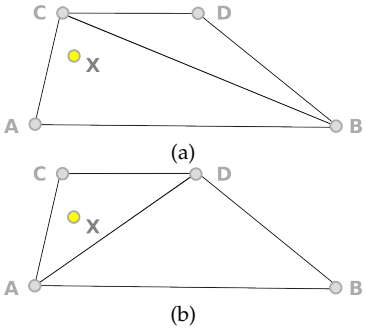


Figure 15: Image illustrating the effect of a Delaunay triangulation for interpolation. (a) Random triangulation ignoring inner angles. (b) Delaunay triangulation maximizing minimal inner angles.

Within the triangles we use linear interpolation by barycentric coordinates (see e.g. Sec. 4.2). For curved surfaces we use a polyhedron structure. It provides a structure for geometric meshes with vertices, half-edges and facets with their incidences. In our case the facet shape is restricted to triangles. We can define piecewise linear interpolation on this structure. For all grid types the implemented structures of CGAL [cga] were used.

3 RELATED WORK

Tensors provide a highly flexible language to capture a multitude of phenomena. Accordingly, visualization methods are often designed for explicit applications as the tensor fields have differing and often very application specific characteristics.
Many visualization and analysis methods for tensor fields are built on derived scalar fields. While this approach is often helpful it is not always sufficient as looking at the isolated scalar values implies loss of information. Thus, tools are required representing the entire tensor information. For this purpose a variety of visualization methods has been developed. However, most existing tensor field visualization work has been done in the area of diffusion tensor imaging (DTI), which is restricted to semi-definite tensors. In contrast, visualization methods for mechanics and engineering need to be applicable for indefinite tensors which poses specific challenges.

In this chapter, we will recapitulate methods that are related to the work presented in this thesis. Hereby, we pursue two major goals:

- To provide an overview over related methods – the main concepts will be summarized and set in contrast.
- Where a direct relation exists the respective chapters of this thesis are referenced and are set into context with the related work.

For further reading we recommend the overviews given by Kratz et al. [KASH12] and Cammoun et al. [CCMMM+09]. The state of the art report by Kratz et al. revises methods for indefinite symmetric tensors with a focus on engineering and physics. Cammoun et al. introduce tensors from the perspective of signal processing and computer graphics.

The remainder of this chapter is structured as follows:

1. An overview about the subject of tensor field interpolation is given which is relevant for the eigenvector-based interpolation presented in Sec. 4.2.
2. Existing work in tensor field visualization is summarized. The methods are subdivided into structure and image focused approaches. Structure focused methods are the most important part for this thesis, the topology computation (Sec. 4.3, Chap. 7) and the topology based segmentation (Chap. 5) are methods

that can be used as visualizations themselves, but they also return explicit geometric structures that characterize the tensor field and that allow the use of multiple further analysis and visualization methods. Image focused methods are employed in Chap. 6: the topology based segmentation of Chap. 5 serves as framework to map textures and glyphs.

3. Methods for vector field simplification and feature analysis are also summarized as in Chap. 8 asymmetric tensors will be used to give a simplified, sketch-like visualization of vector fields. Asymmetric tensors occur as derivatives of vector fields and derived quantities enhance the illustrative depiction of vector fields.

3.1 TENSOR INTERPOLATION

Interpolation is a central subject in scientific visualization. Mostly the phenomena to be visualized are given as sets of data points. To provide an trustworthy analysis and visualization the data has to be reconstructed to a continuous function in a sensible manner. Depending on the type of data and application different demands arise. In tensor field visualization the preservation of tensor properties and/or invariants, consistency, and efficiency are some of the central questions for interpolation models (for more details see Sec 2.5).
Especially in the context of tensor imaging an important requirement is the preservation of positive definiteness of the input tensors. Excellent theoretical properties are given by works based on a metric defined on the Riemannian manifold that guarantees affine invariance [MB06, FJ07, LRDF06, PFA06]. Less computational costly approximations are based on a log-Euclidean metric [AFPA06, FAPA07]. Here again, positive-definiteness is preserved and the determinant is monotonically interpolated along geodesics.
Alternatives to preserve the nature of the tensor data are methods separating direction and shape. Due to the direction indeterminacy of eigenvectors a consistent interpolation of the directions is no trivial task. Many proposed approaches are based on heuristics [BCG+92, CAA01, MFWAL04, BWHK05] which is feasible if isotropic regions are not regarded. Kindlmann proposed an interpolation based on geodesic loxodromes[KEN+07] which interpolates three dimensional tensors in one dimension along a line.

Context to this thesis: The interpolation models decoupling shape and direction are those related to the work in Sec. 4.2. The *eigenvector-based* interpolation for two dimensional tensors interpolates directions and shape separately. Therefore it is shape preserving. No inconsistencies can be introduced by this interpolation schema as degenerate elements are incorporated in the model.

3.2 TENSOR FIELD VISUALIZATION

Although this is no absolute classification, for this thesis it is convenient to split previous work in tensor field visualization into *structure focused* and *image focused* approaches. In this classification structure focused methods are regarded to return some geometric structure or entity (such as partitions) as primary result. These structures can be used as visualization themselves but they also provide a basis for further processing such as tracking, simplification or as preprocessing step for a variety of visualization methods. With image focused methods the main attention lies on the final image.

The major part of this thesis is concerned with structure focused methods. The extent in the review of existing methods is adapted accordingly, especially the overview of topological methods will take a large portion.

3.2.1 *Structure Focused Approaches*

The structure focused methods for tensor fields related to this thesis can be subdivided into two types: tensor lines and their refined variations, and the extraction of structures that segment the tensor field. Such a segmentation is either based on structural properties and returns the topological structure or it is based on an appropriate dissimilarity measure and partitions the tensor field more generally.

3.2.1.1 *Tensor Lines*

To depict the behavior of a single eigenvector field, tensor lines can be used. These are defined as lines that are tangential to the chosen eigenvector field. They are strongly related to streamline methods used for vector fields. Hyperstreamlines introduced by Delmarcelle and Hesselink [DH92] are an extension, which additionally incorporate the eigenvalues and other principal directions.
Hyperstreamlines have been utilized in a geomechanical context by Jeremic et al. [JSF+02]. Wilson et al. [WB05] modified the idea of integrating tensor lines. Instead directions of maximum shear stress in two dimensional tensor fields are integrated. The overall directional behavior in the field can be understood in terms of an evenly spaced stress net.

3.2.1.2 *Tensor Field Topology*

Methods concerned with the structural segmentation of possibly indefinite tensor fields are based on tensor field topology. They concentrate on the structure of the eigenvector fields neglecting the scalar

entities like eigenvalues.

Topology of two dimensional symmetric tensor fields – The idea of using topology to analyze the structures of tensor fields goes back to Delmarcelle et al. [Del94, DH94] and Lavin et al. [LBHL97]. This builds the basis for the tensor field topology methods proposed in this thesis. They have introduced the topological skeleton consisting of degenerate points and connecting tensor lines as basic constituents. The topological skeleton separates the field into regions of qualitatively homogeneous eigenvector behavior.
Simplifying and tracking over time of the resulting structures was started by Delmarcelle [Del94]. Following this work, much effort has been put into simplification and tracking by Tricoche et al. [TSHC01, Tri02]. This results in structures, that are easier to understand and enhances their practical applicability, especially if more complex 2D tensor fields have to be analyzed.
Alliez et al. [ACSD+03] proposed an application to curvature tensors and their topology for polygonal re-meshing of surfaces.

Context to this thesis: Sec. 4.3 presents the topology extraction for two dimensional symmetric tensor fields based on the shape preserving interpolation model (Sec. 4.2). The topology is consistent and delivers qualitatively equal results to the method invented by Delmarcelle et al.(Delmarcelle1994, Delmarcelle1994a). It extracts the integral topological graph which captures both eigenvector fields. This graph is automatically simplified, in contrast to the methods presented by Tricoche et al. [TSHC01, Tri02] which are parameter dependent or require user interaction. Methods presented by Tricoche et al. reduce the number of degenerate points, whereas our method reduces the number of relevant separatrices.
The results of Chap. 5 share many aspects with the re-meshing of Alliez et al. [ACSD+03]. Alliez uses the curvature tensor of a triangulated surface to re-mesh it to polygonal structures. For this purpose initially also the topology of both eigenvector fields is computed, according to [Del94]. Then the topology is refined until the surface is re-meshed to a predefined granularity. In contrast to our work Alliez performs initial smoothing to the tensor field to obtain clean results, which seems feasible if the goal is re-meshing. The only directive for the generation of the final mesh is a distance measure based on anisotropy. Our method is optimized for a variety of measures and applications. Depending on the resolution of the initial surface's triangulation it might be an valuable extension to the re-meshing process to also look in the vertices and edges for degenerate elements as proposed in Chap. 7.

Topology of three dimensional symmetric tensor fields – 3D tensor field topology was initiated by Hesselink et al. [HLL97] . Zheng et al. [ZP04] picked up the work in Their analysis shows that in three dimensions degenerate one-dimensional structures (lines) are stable topological features, while degenerate points are not. A reformulation by Tricoche et al. [TKW08] leads to a definition of these degenerate lines as crease lines of a tensor invariant. This allows the adaptation of methods from scalar field analysis for the extraction of degenerate elements. Still, the understanding and representation of the topology for tensor fields of dimension higher than two is subject to ongoing research.

Asymmetric tensor field topology – Tensor field analysis is also becoming increasingly interesting in context with fluid dynamics. The tensor of interest is the gradient of the vector field, which is not symmetric. Zheng et al. [ZPP05] extend the topological analysis to 2D general (asymmetric) second order tensor fields. Their goal is give a continuous view onto the field by partitioning the domain. In regions where eigenvalues are complex the tensor has a rotational effect and in regions with real valued eigenvalues the tensor has a stretching effect. These regions are separated by degenerate lines, where the eigenvectors coalesce. Zhang et al. [ZYLL09] continued this approach. At the core of their analysis is a reparameterization of the tensor space, which allows to understand the topology of asymmetric tensor fields by studying the manifolds of eigenvalues and eigenvectors.

3.2.1.3 *Vector Field Topology on Piecewise Linear Surfaces*

There has not been much work on the extraction of tensor field topology on piecewise planar surfaces. However a few publications deal with the related problem for vector fields, which is often referred to as 2.5 dimensional topology. Already Helman and Hesselink recognized the importance of surface topology [HH90]. While focusing on the related flow phenomena they are not concerned with issues arising for piecewise planar surface representations. Kenwright [KHL99] proposed to investigate the wall shear stress vector field, which is a tangential vector field, to find separation and attachment lines in a flow around an embedded object. This idea has been further followed by Tricoche et al. [TGS05] who also recognize the problem of discontinuities of the tangent vector field on piecewise planar surfaces. A solution for a consistent integration of streamlines is discussed. Li at al. [LV+06] model arbitrary higher order singularities represented at vertices on simplicial surfaces. The focus lies on the introduction of an appropriate vector interpolation schema on the mesh. Central to this interpolation is an integer called the period jump, which is associated to each dual edge of the triangulation. The primary goal

of this method is to find a good discrete approximation for continuously given vector fields and not the analysis of a given field. The detection of the complete set of singularities is not guaranteed. Another approach is followed by Zhang et al. [ZMT06] with the purpose of vector field design for texture synthesis. To create continuous vector fields on curved surfaces represented as meshes, they use geodesic polar maps and parallel transport to interpolate vector values defined at the vertices of the mesh. This yields a non-linear vector field inside each triangle, loosing the simplicity of linearity. Later this work has been extended to tensor fields [ZHT07]. Garth et al.[GLT+07] extract the boundary topology of engine simulation data. In their work they hint at the necessity of integrating so-called 'singular edges' into the topological analysis. In context with vector field singularity tracking on curved surfaces Wiebel et al. [WTS+07] observe singularities disappearing in vector field discontinuities. They propose to tackle this problem by parameterizing the mesh such that further computation can be carried out in the plane. The singularities are then mapped back onto the triangle mesh. They avoid to compute a global parametrization of the whole surface restricting the analysis to subregions of the triangle mesh, which are homeomorphic to a disc and without sharp edges. However these parametrizations have to be computed involving a distortion of the triangles. For the interested reader an overview of topology-based methods in flow visualization can be found in [LHZP07].

Context with this thesis: In Chap. 7 we will present the model of continuous transition bridges, that is locally applicable (in vertices and edges of a triangulation) to detect simple and higher singularities for vector and tensor fields.

3.2.1.4 *Segmentation by Dissimilarity Measure*

For diffusion tensor imaging (DTI), a lot of effort has been put into tensor field segmentation, mostly with the goal of brain segmentation. Extending methods from image segmentation and clustering, the central research topic is the definition of an appropriate dissimilarity measure for tensors. The simplest dissimilarity measures are based on tensor components separately considering the tensor segmentation as a multi-channel segmentation of scalar values. More elaborate methods use dissimilarity measures based on invariants or comprise the entire tensor data. In general the chosen dissimilarity measure determines the applicability to indefinite tensors.
A review of developments that focus on the segmentation of anatomical structures from DTI can be found in [dLGLW09]. Proposed methods range from active contours [Wan04], level sets [ZMB+03, FWB03] to graph-cut algorithms [WH07, ZTW06]. Used metrics are the angu-

lar difference between principle eigenvector directions, or standard metrics considering the entire tensor, like the Euclidean or Frobenius distance.
Wang et al. [Wan04] introduced a distance measure based on the Kulback-Leibler distance from information theory designed for Gaussian distributions. Although, it is a good representation of the diffusion tensor characteristics, it is limited to positive definite tensors.
A segmentation designed for meshes based on the curvature tensor was introduced by Lavoue et al. [LDB05]. Vertices are clustered according to their principal curvature values using a k-means classification. The boundaries of resulting cells tend to be parallel to lines of minimum curvature but do not exactly represent the principal directions.

Context to this thesis: In Chap. 5 we provide a segmentation that is topology based and offers a variety of scalar quantities (e.g. eigenvalues, anisotropy measures) that are employed as dissimilarity measures to steer the granularity of the segmentation. The boundaries of the segmented regions are given explicitly in terms of tensor lines and consistency is guaranteed. However, inherently segmentation methods that are derived from image processing approaches offer more possibilities to be performance optimized.

3.2.2 *Image Focused Approaches*

In general, the developed visualization methods can be subdivided into local, continuous or hybrid methods. If the goal is to convey the entire information contained in single tensors, local methods might be favored. For the representation of an entire field, however, this can lead to perceptional issues and most importantly overwhelm the user in its information content. Therefore, continuous representations are more appropriate to convey an overview of the data. Mostly these methods build on derived scalar or directional quantities. The combination of both – continuous and local methods – are especially beneficial for 3D tensor fields. They are referred to as hybrid methods.
For all approaches an adept mapping of the quantities to be displayed can strongly support the understanding of the field.

3.2.2.1 *Mapping*

Applying a sensible mapping to the occurring input parameters can do a great deal for the final visualization. Mapping techniques in terms of scaling the data allows to get a hold on a large range of data values. Additionally a specific interval in the data values can be emphasized. Exponential mapping (i.e. [SK10]) can be applied to tensors describing infinitesimal deformations to provide clear differentiation

of the eigenvalues' signs. A generic mapping is given by Hotz et al. [HFH+04] which also allows to distinguish the eigenvalues' sings. Also it can be used to encode physical properties, e.g. compression and expansion, in the final visualization.

Context to this thesis The approach of Hotz et al. is employed and adapted in Sec. 6.1.2.2.

3.2.2.2 *Local Visualization*

Direct tensor visualization approaches focus on displaying tensor values in discrete points. In this context research issues usually deal with the design and placement of glyphs. Shape, size, orientation, color, and texture can be used to encode tensor properties. Evidently, the entire tensor information and derived quantities can be displayed in one geometric object. Still, the design and use of glyphs always has to be guided by a trade-off between richness in information and perspicuity. The applicability of glyphs is mostly restricted to tensor fields defined on two dimensional surfaces. Naturally, glyphs are well suited for hybrid methods and interactive probing at distinctive locations for detailed inspection.
A complete review of all existing work about Glyphs is beyond the scope of this thesis. A collection with related papers can be found in [WW05, LW09].
For an overview about the design of glyphs the reader is referred to the article by Hashash et al. [HYW03] (glyphs for stress and strain tensors), and about perceptional issues such as visual ambiguity to [SK10]. Additionally, the state of the art report [KASH12] thoroughly reviews design-guidelines, appropriate mapping of encoded quantities, and a discussion of the most common glyphs in DTI and mechanics.
In the following a brief summary of glyph types for three dimensional tensor fields is given. Probably the simplest glyph for tensors to depict the entire tensor information is an ellipsoid aligned to the tensor's eigenvectors and scaled according to its eigenvalues (Fig. 8). An important glyph family are superquadric glyphs based on a parametrizable space. Barr [Bar81] introduced the idea of a generic interpolation of complex glyph geometries within the parametrized space defined by simple base types. Improvements were made towards perception [SHB+99, Kin04], applicability for for indefinite tensors [SK10] and a specified glyph variant for traceless tensors [JKM06]. For mechanical applications a broad variety of glyphs is available: Mohr's circle [BN03, CRBC04, CRB+05, KMH11], Lamé's stress ellipsoid, the Haber glyph [Hab90], Reynolds glyph [MSM96, KYHR05], HWY glyph[HYW03], quadric surfaces, and plane-in-a-box-glyph [NJP05].
Another challenge is the placement of glyphs to maximize the in-

formation that is conveyed in an image. Like this a continuous impression of the field can be achieved. Previous methods in this context aimed to create sample distributions that avoid holes and overlaps [SYI96, HSH07, FHHJ08, KW06, KKH11]. Their applicability for visualization is mostly limited to tensor fields defined on two dimensional surfaces.

Context to this thesis: Chap. 6 addresses the question where to place glyphs by exploiting the segmentation of the tensor field into cells – the aim hereby is not to give a continuous representation of the entire field, but to use the homogeneous cells as frame for selective probing for more detailed insight at distinctive locations.

3.2.2.3 *Continuous Visualization*

While glyphs are appropriate for displaying single tensors, they are limited to low resolution and might fail to give insight into the structure of the entire field. First overview or context visualizations can be generated by volume rendering. This approach is based on the definition of relevant derived scalar quantities that are mapped by a transfer function to color and opacity [DGBW09, KW99, HSW+09]. However important information is lost by the reduction to scalar input parameters.

An alternative can be given by employing textures. They provide a multitude of parameters to reflect tensor characteristics without reducing the information content of the tensor field. In return, texture-based methods are generally restricted to two-dimensional surfaces.

Textures based on line integral convolution (LIC) [CL93], for example, have been extended to HyperLIC [ZP03]. HyperLIC uses a multi-pass approach to represent the orientation of one eigenvector field and to highlight anisotropy.

Another extension are fabric textures [HFH+04]. Two LIC images for every eigenvector field are computed and blended, which leads to an image that resembles a fabric encoding central physical properties of stress tensor fields into fiber thickness (thin = compression, thick = expansion).

A novel method based on anisotropic sampling [KKH11] has been introduced. By computing a generalized Voronoi diagram on the basis of a tensor-determined local metric, the resulting regions can be mapped with textures which results in a large variety of possibilities to create visualizations.

Similar to textures a continuous view on a tensor field can be generated by tensor splats [BW03, Ben04]. Tensor splats can also be used for three dimensional tensor fields if the volumetric data can be strongly filtered.

Context to this thesis: In Chap. 6 we present a method to map a variety of textures using the tensor field segmentation of Chap. 5. The segmentation solves the intricate question where to map textures by providing the framework by the explicit cell geometries.

3.2.2.4 *Hybrid Visualization*

Three-dimensional visualization methods are rare. Hybrid methods can help to propose three dimensional context visualization with detailed depictions in regions of interest. Dick et al. [DGBW09] proposed a visualization of stress tensor fields for implant planning. They focus on hybrid methods to emphasize important regions and to avoid clutter. When features of interest are not known in advance, explorative methods can help [KKH11]. Here integrated views help to understand tensor fields from different perspectives.

3.3 VECTOR FIELD VISUALIZATION: SIMPLIFICATION STRATEGIES AND FEATURE ANALYSIS

Chap. 8 will introduce an illustrative visualization of vector fields by analyzing the properties of the gradient vector field in terms of an asymmetric tensor field.
The analysis and visualization of vector fields has already a long tradition in the area of visualization. Accordingly, a large amount of interesting work has been done in this field which is far beyond this paper. Considering different views on the topic, good review articles can be found: Texture and Feature-Based Flow Visualization [EGL+06], Topology-Based Flow Visualization [LHZP07], Partition-based Techniques [SJWS08], and Illustrative Flow Visualization [BCP+12]. For the work in Chap. 8 methods for vector field simplification and feature analysis are of interest which are be summarized below.

3.3.1 *Simplification via Vector Clustering*

One approach to simplify complex data sets is to group regions of similar behaviors and represent them using on icon, e.g. one representative vector or a curved arrow icon. For all methods, the choice of an appropriate similarity measure on which the clustering is based is essential. Mostly, they are a combination of a position and vector (orientation and magnitude) error. Being able to display several levels of details the construction of vector field hierarchies has been the goal of several clustering algorithms. The methods can be mainly distinguished by choosing a top-down or bottom-up approach. An early work by Heckel et al. [HWHJ99] recursively uses planes to split

clusters based on streamline discrepancy as error measure. The result is a hierarchy of convex clusters. Another top-down approach using a generalized normalized-cut algorithm has been proposed by Chen et al. [CBHL03]. The goal of their method is to cluster vectors that are associated with the same critical point. Du et al. [DW05] propose a simplification of vector fields resulting in a predefined number of clusters based on Centroidal Voronoi tessellations (CVTs). Based on the chosen distance function, the vector fields are then naturally clustered by the CVT. A generalization of the K-means clustering has been introduced by Mc Kenzie et al. [MOD05]. Its basic component is a variational clustering algorithm, that minimizes a global error for a given number of cluster sets. The algorithm is applicable to 2D and 3D fields and the error metric is flexibly exchangeable. There are also a couple of approaches pursuing a clustering based on an anisotropic diffusion process. In contrast to other clustering approaches, this is a continuous process without explicit split or merge events. Garcke et al. implemented a phase field model adapting the Cahn Hillard model to vector fields [GPR+01]. Greibel et al. propose to use an algebraic multi grid method to achieve similar results. While generating impressive results, theses methods do not provide a direct error control. The method we build on was developed by Telea et al. [TvW99]. The goal of their work was to produce simplified but suggestive images without much user interaction. The clustering process works bottom-up starting with clusters for every data point. Then clusters are successively merged according to differences in position and orientation and magnitude. A nice feature of the method is the design of the similarity measure which allows to control cluster shapes by adapting to the weights of the various terms in the error measure.

3.3.2 *Simplification via Streamline selection or Clustering*

Another approach is to simplify the vector field representation displaying only selected streamlines. There are several streamline placement algorithms with the goal to achieve a uniform coverage of the domain that is sparse but still gives a continuous impression of the field, e.g. for 2D fields [JL01, RPH+09] and for 3D fields [MCHM10, CYY+11]. Techniques that cluster precomputed streamlines are frequently used in context of fiber tracking for diffusion tensor data. An evaluation of related methods can be found in [MVvW05]. Kuhn et al. [KLG+11] have introduced a clustering technique using streamline properties, as curvature, to detect regions of similarity and visualize them by means of compact cluster boundaries.

A natural segmentation of vector fields in regions of similar streamline behavior is defined by its topological structure. Methods to simplify this structure and use it for visualization purposes have been proposed by Tricoche and Garth et al. [TSH01, GTS+04].

3.3.3 *Feature Extraction for Vector Fields*

The second component of the visualization frame work in Chap. 8 builds on existing work on feature extraction and definition for vector fields. Papers in this area are concerned with the definition, stable extraction, and simplification of features. One can distinguish features that are directly defined on the vector vector field, as vector field topology, and features that facilitate a derived scalar field. Vector field topology thereby focuses on finding features like source, sinks, and saddle points as well as separatrices connecting them [LHZP07]. Scalar typed features are mostly defined as iso-contours for a given threshold or the extremal structure of the field [SWC+08]. Prominent examples are vortex like features defined on basis of vorticity [SPP04, SWTH07], λ_2 [JH95], the Q quantity proposed by Hunt [Hun87], or the acceleration magnitude [KRHH11]. An evolved analysis of multiple scalar and vector-like features has been presented by Zhang et al. [ZYLL09, CPL+11]. They consider the gradient vector field as basic feature carrier, which is composed in three components: isotropic scaling, rotation, and shear. While the first two components have only two distinct expressions, i.e., positive respective negative, the shear factor is also equipped with directional information. Based on this tensor decomposition, the domain is classified into regions where one feature dominates the others. We built on this work but facilitate the strength of the respective tensor components for a layered visualization. A decomposition with similar meaning is the Hodge decomposition which decomposes the field into a a divergence-free, a curl-free part, and a harmonic part. A topological analysis and multi-scale decompositions based on this decomposition has been persuaded in [PP03, TLHD03]. A glyph-based visualization of time-dependent flows has been proposed in [HLNW11].

Context to this thesis: Based on the tensor decomposition of Zhang et al. [ZYLL09], we will present in Chap. 8 a simplified depiction of vector fields. In contrast to previous work on simplified vector fields we also incorporate the information gained by looking at the gradient vector field. This information is independent of the chosen frame of reference and presented to the user in a suggestive manner. Features with clear mathematical definition and interpretation are extracted by using scalar field topology in combination with homological persistence on the strengths of the three respective tensor components.

4

EIGENVECTOR-BASED INTERPOLATION AND SEGMENTATION

In this chapter we propose a segmentation of two dimensional symmetric tensor fields towards eigenvector characteristics. The segmentation is based on the topology of the tensor field. In contrast to previous approaches (see Sec. 2.7.1) we take both eigenvector fields into account to extract the underlying topological graph structure – which is represented by the *integral topological graph*. This graph partitions the domain into cells bounded by separatrices with homogeneous quality of eigenvector behavior. As eigenvector fields of symmetric tensors are orthogonal the cells are always curvilinear and cannot degenerate.

Hereby, we particularly analyze the influence of the interpolation scheme on the resulting topology, considering the component-wise and the eigenvector-based [HSNHH10] linear interpolation models. 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 (Sec. 2.10) leads to connections between close degenerate points. These new connections create degenerate elements of higher dimensionality, degenerate edges and triangles. When comparing the resulting topology per triangle with the one obtained by component-wise linear interpolation the results are qualitatively equal. However, due to the degenerate elements of higher dimensionality the resulting topological structure is strongly cleaned up. Further by simultaneously analyzing both eigenvector fields for the integral topological graph we are able to propose a compound classification for separatrices.

4.1 NOTATION

For convenience in this section we decompose the two dimensional symmetric tensors $\mathbf{T}$ according to Eq. 10 and use the following notation:

$$\mathbf{T} = \begin{pmatrix} d & 0 \\ 0 & d \end{pmatrix} + \begin{pmatrix} \Delta & F \\ F & -\Delta \end{pmatrix}. \tag{34}$$

Eigenvalues are named λ and μ, with $\mu \leqslant \lambda$ and eigenvectors $\overset{\leftrightarrow}{\mathbf{v}}$ (major) and $\overset{\leftrightarrow}{\mathbf{w}}$ (minor). This facilitates the clarity of equations involving triangles indices (e.g. Eq. 36).

4.2 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 use an interpolation based on eigenvectors and eigenvalues, see Figure 16. This method is *shape-preserving* and minimizes the number of eigenvector computations.

The interpolation is defined such that the resulting topology per tri-

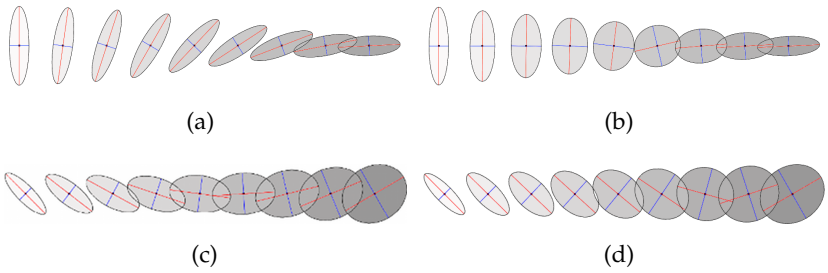


Figure 16: Comparison of interpolation methods : (left images) eigenvector-based (shape preserving), (right images) component-based. Top row: the eigenvector-based interpolation (a) of two strongly anisotropic tensors delivers anisotropic tensors again, thus it preserves the shape. In contrast, for this example the component-wise interpolation (b) induces the *swelling-effect*, meaning that it passes through a highly isotropic tensor (center ellipse in (b)). The interpolation of an anisotropic and an isotropic tensor however is very similar (c,d).

angle is qualitatively the same as for component-wise interpolation, see Figure 20. It is defined on a triangulated domain. In this work we employ a Delaunay triangulation to obtain a clearer topological structure. 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 the fact has to be considered 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

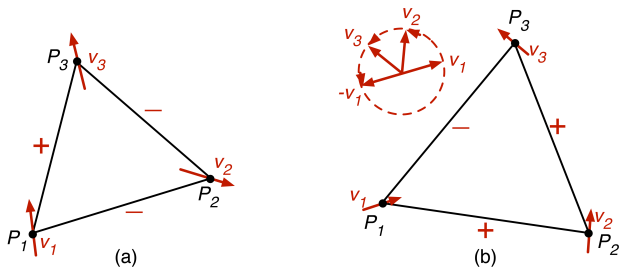


Figure 17: Triangle (a) without and (b) with degenerate point, edge labels indicate whether two adjacent eigendirections match.

the value $F_j\Delta_i - F_i\Delta_j$. If the value is equal to zero, then 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:

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

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 17. It is:

$$\prod_{i=1}^{3} l(e_i) = \begin{cases} 1 & \text{no degenerate point in triangle,} \\ -1 & \text{one isolated degenerate point in triangle,} \\ 0 & \text{there is a degenerate point on at least one of the edges.} \end{cases} \tag{35}$$

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

$$\begin{aligned} \mathbf{v}(\boldsymbol{\beta}) &= \beta_1\mathbf{v}_1 + \beta_2 l(e_3)\mathbf{v}_2 + \beta_3 l(e_2)\mathbf{v}_3, \\ \mathbf{w}(\boldsymbol{\beta}) &= \beta_1\mathbf{w}_1 + \beta_2 l(e_3)\mathbf{w}_2 + \beta_3 l(e_2)\mathbf{w}_3, \\ \lambda(\beta) &= \sum_{i=1}^{3} \beta_i\lambda_i, \quad \text{and } \mu(\beta) = \sum_{i=1}^{3} \beta_i\mu_i. \end{aligned} \tag{36}$$

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 $P(\boldsymbol{\beta}) := \beta_i P_i + \beta_j P_j + \beta_k D$, (cyclic indices) eigenvalues and eigenvectors are interpolated using Equation 36. The resulting eigenvectors are independent from the coordinate β_k which respects the fact that in D eigenvectors are not uniquely defined. Accordingly in a triangle containing a degenerate line the eigenvector fields become constant, see Fig. 18.
Please note, for a consistent interpolation the eigenvectors **v,w** need to be mutually oriented that for each tensor the numerical representation of **w** is in counterclockwise direction to **v**.

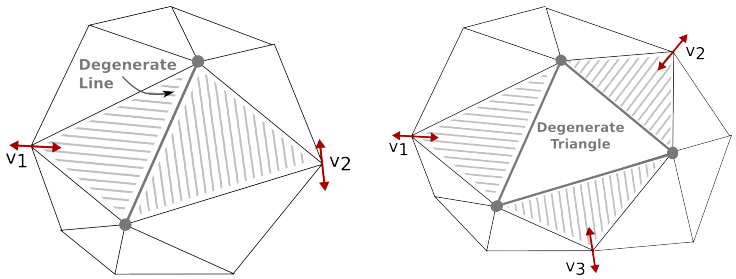
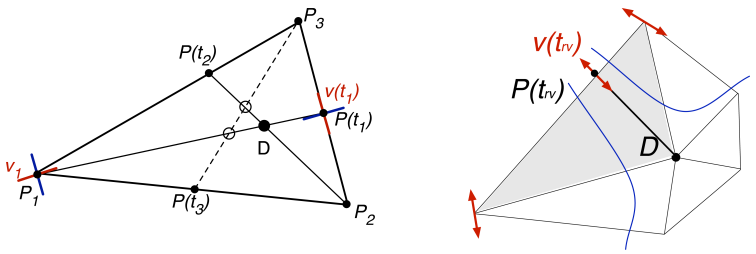


Figure 18: Higher order features: (a) shows a degenerate line; (b) shows a degenerate triangle, with three degenerate vertices. In the interior of the neighboring triangles the vectors are set to constant.

4.3 EIGENVECTOR-BASED SEGMENTATION

The basic steps for extracting the *integral topological graph* are: location and classification of degenerate points, determination of separatrix directions for both eigenvector fields, and their integration. We restrict this section to eigenvector-based interpolation. Topology computation based on component-wise interpolation is summarized in Sec. 2.7.1 and for further details we refer readers to [Del94, Tri02]. The

main difference between the two interpolation schemes is caused by the triangle subdivision and subsequent Delaunay re-triangulation. All degenerate points lie on vertices with piecewise linear behavior in the vicinity and can exhibit structures different from trisector and wedge points. If degenerate points are interconnected we call them degenerate elements of higher dimensionality. These elements approximate degenerate points of higher order (in Sec. 2.7.1 also introduced as *multiple* degenerate points).



(a) (b)

Figure 19: (a) The location of a D is well-defined if the three lines connecting the vertices and their opposite points intersect in one point. (b) Radial tensor line entering degenerate point D.

4.3.1 *Location of Degenerate Points*

Since degenerate points at vertices can be detected easily by comparing the eigenvalues, 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 35, the edge label product has to be -1. We define the location of the degenerate point

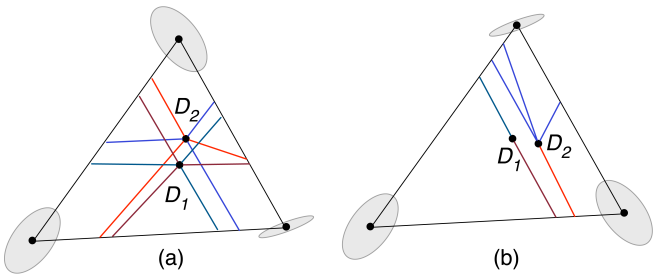


Figure 20: Comparison of interpolation models and resulting topological elements in one triangle: Degenerate points D_1 from componentwise, D_2 from eigenvector-based interpolation, in the case of (a) trisector, and (b) wedge point.

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

$$\mathbf{v}(t) = (1-t) \cdot \mathbf{v}_j + t \cdot l(e_i) \cdot \mathbf{v}_k, \qquad t \in]0, 1[. \tag{37}$$

Even though the resulting vector field $\mathbf{v}$ on the boundary may not be continuous at all vertices, the corresponding un-oriented direction field $\overset{\leftrightarrow}{\mathbf{v}}$ is. It defines a 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 this connection. The parameters t_i are given by

$$\mathbf{v}_i \cdot \left((1-t_i)\mathbf{v}_j + l(e_i)t_i\mathbf{v}_k \right) = 0, \tag{38}$$

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 Fig. 4.19(a)
It can easily be seen that the point D is well-defined. From the definition of t_i in Equation 38, 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.

4.3.2 *Non-isolated Degenerate Points*

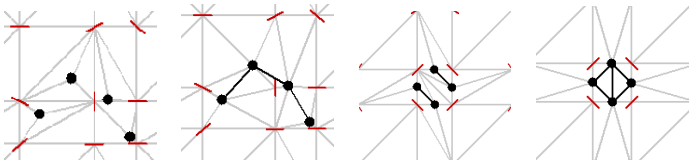
Two degenerate vertices connected by an edge give a degenerate line. The resulting eigenvector field inside adjacent triangles is constant and does not contribute to the final structure (see Fig. 18). Similarly a degenerate triangle, where all vertices are degenerate points, is enclosed by three triangles 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 even see degenerate polylines when applying a subsequent Delaunay re-triangulation (see also Fig. 21).

4.3.3 *Determination of Radial Directions*

The neighborhood of the degenerate point is characterized by segments separated by radial tensor lines. For linear eigenvector interpolation, radial tensor lines are straight lines and are determined by their intersection $P(t_r)$ with the edges of adjacent triangles, see Fig. 4.19(b). For each edge of the triangle,

$$\begin{aligned} &\mathbf{v}(t_{rv}) \times (P(t_{rv}) - D) = 0, \\ &\mathbf{w}(t_{rw}) \times (P(t_{rw}) - D) = 0, \quad t_{rv}, t_{rw} \in [0,1]. \end{aligned} \tag{39}$$

t_{rv} and t_{rw} specify the radial directions for the eigenvector fields v and w respectively. In contrast to component-wise interpolation,



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

Figure 21: Close ups of the one-point load data set (red lines indicate the major eigenvector directions in the vertices). (a) and (c) show the resulting triangulation structure if degenerate points are only inserted in the triangulation, (b) and (d) the result with subsequent Delaunay re-triangulation. In (b) three connected degenerate lines and in (d) 2 degenerate triangles arise. This degenerate elements of higher dimensionality will lead to a strongly cleaned up topological graph structure.

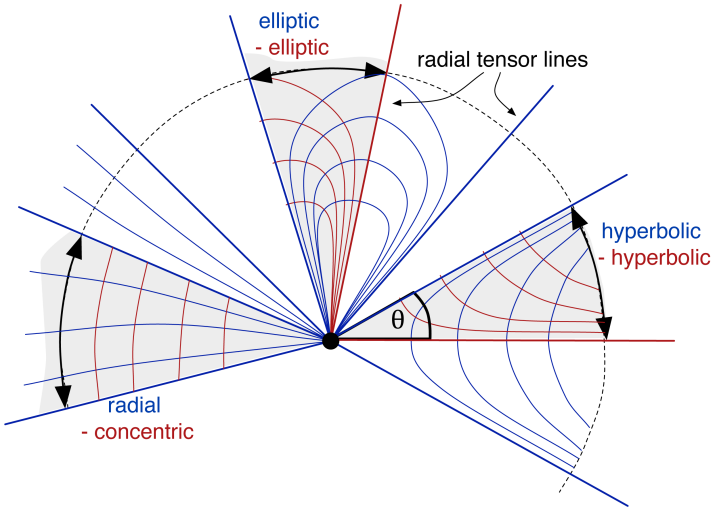


Figure 22: The neighborhood of a degenerate points is characterized by a number of half-sectors with qualitatively equal behavior.

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.

4.3.4 *Half-sector Classification*

In the case of linear component-wise interpolation without re-triangulation a point classification into trisector or wedge points serves as basis for the classification. To cover all possible cases of degenerate elements, for piecewise linear behavior, we built on an immediate sector analysis similar to [Del94]. In contrast we classify *half-sectors*, as we consider the topology of both eigenvector fields together. Half-sectors are radial segments enclosed by two neighboring radial directions, inde-

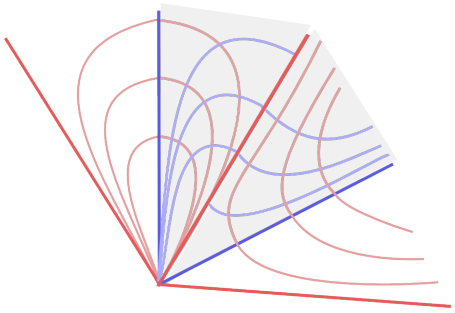


Figure 23: Half-sector classification of radial tensor lines extends traditional classification. Shaded half-sectors can be classified correctly and at one step for both eigenvector fields.

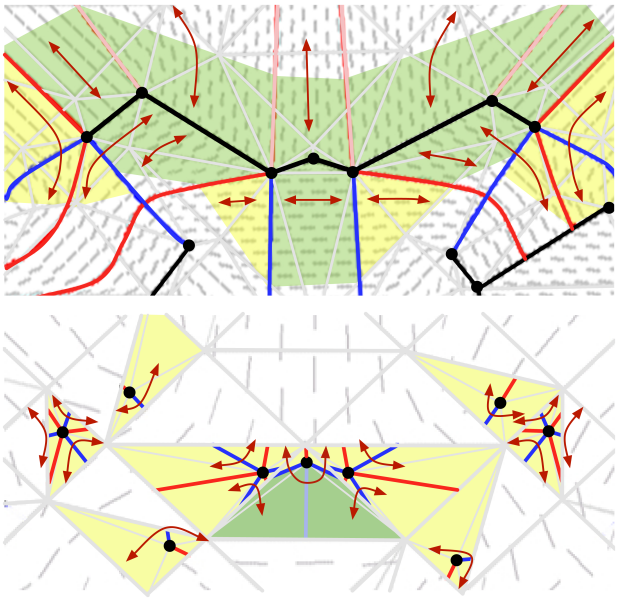


Figure 24: A close-up of sector classification for the one-point load data set using linear interpolation of eigenvectors, with (top) and without (bottom) subsequent Delaunay re-triangulation. Shaded regions show the sectors: green and yellow for non-hyperbolic and hyperbolic, respectively. Red and blue lines show separatrices and light red and light blue radial lines, which were not classified as structurally relevant; black points and lines are the degenerate points and lines.

pendent of the eigenvector field, either red (major) or blue (minor), see Fig. 22. The following types of half-sectors can occur:

hyperbolic sector - bounded by one red and one blue radial line: tensor lines 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 tensor lines, of this color, start from the degenerate point and

then diverge. The tensor lines of the other color enter and leave the sector through bounding lines.

elliptic sector - bounded by one red and one blue radial line: the tensor lines 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 Fig. 22

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

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 Fig. 24.

Traditionally, for the skeleton computation only radial lines, which are boundaries of hyperbolic sectors, are relevant. This was defined in analogy to vector field topology. However, we will consider elliptic sectors as structurally relevant as well and keep radial tensor lines bounding elliptical sectors.

With the classification by the opening angle $\Delta\Theta$ to the actual rotation angle $\Delta\alpha$ given above one can easily see that both eigenvector fields can be correctly classified at once. Due to the orthogonality of the eigenvector fields the quality of the half-sectors induces the same $\Delta\alpha$ for both eigenvector fields (see also Fig. 23).

4.3.5 *Separatrix Computation*

To complete the topological skeleton we integrate all radial tensor lines bounding the hyperbolic sectors using Runge-Kutta 4th-order integration scheme with adaptive step size. Alternatively, an exact tensor line integration for the linear eigenvector field can be used [NJ99]. 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 tensor lines.

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 Fig. 25a.
3. A separatrix describes a circle or spiral and passes itself closely in parallel integration direction , see Fig. 25b and 25c. Circulating separatrices overload the topological graph without adding

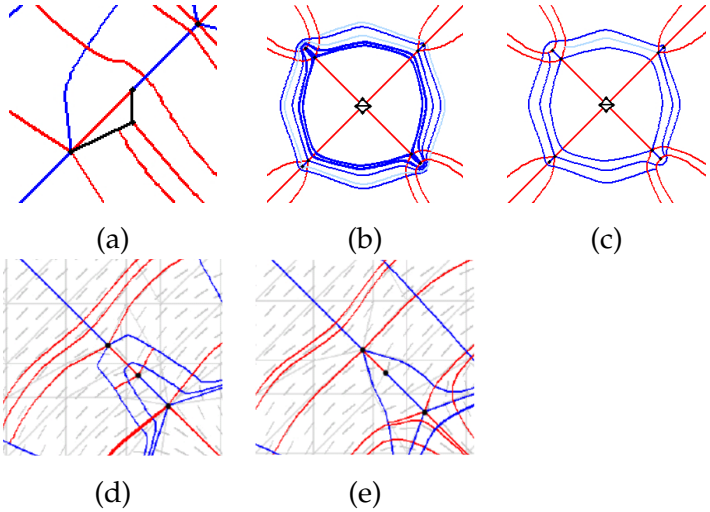


Figure 25: Close-up from one-point load data set: (a) tensor line runs into a degenerate line (black line); circulating tensor line (light blue) (b) before and (c) after clean up; (d,e) separatrix integration for component-wise and eigenvector-based interpolation.

structural information for the final segmentation. We delete circulating tensor lines 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.

4.4 SEGMENTATION

After computing the topological skeleton for both eigenvector fields, we find the intersections of the red and blue tensor lines. The properties of the resulting segmentation as can be seen in Fig. 22 are
(a) Cells without degenerate point are quadrangular with two red and two blue tensor lines as boundary, in an alternating order. All red tensor lines 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.

4.5 RESULTS

We tested our method on two dimensional 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 (Sec. A.1.2 + A.1.3) and multiple forces applied to a notched block (Sec. A.1.4).

Figure 28 is a slice of the one-point load data set. It shows that the results of the topology extraction for component-wise and eigenvector-based interpolation are qualitatively equal. Since the results for the eigenvector- and component-wise interpolation schemas are qualitatively equal we will further inspect the results in close-up views showing the major and typical differences.

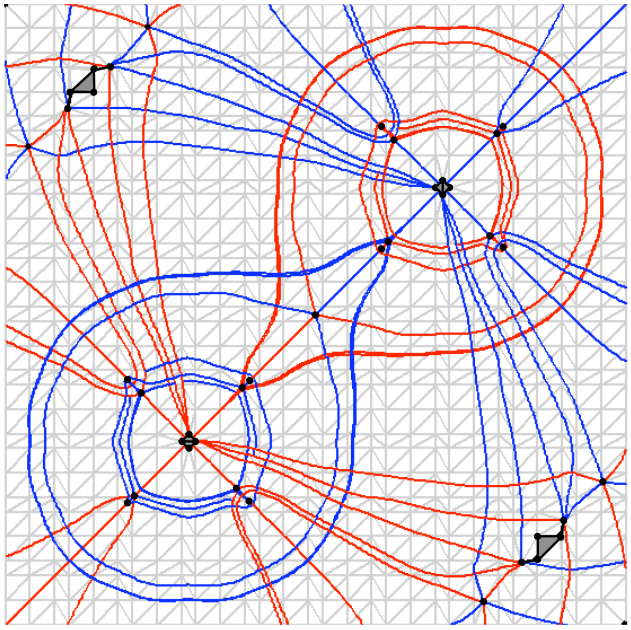
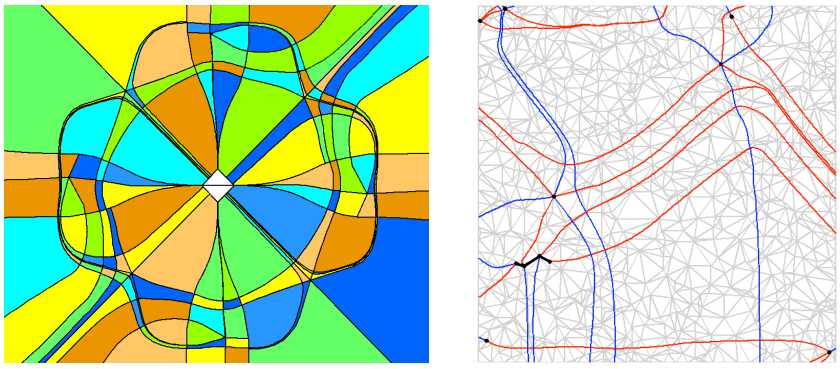


Figure 26: Full segmentation of two-point load data set.



(a) (b)

Figure 27: left: one-point load with randomly colored segments, right: slice of strain simulation of forces on notched block.

In Fig. 25(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 are smoother for the eigenvector-based interpolation, resulting in less curved tensor lines in the vicinity of degenerate points. As a consequence, a proper step size adaption (leading in general to a smaller step size) is especially important for component-wise interpolation to obtain tensor lines 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. Within the triangles interpolation of eigenvectors and eigenvalues can be performed linearly. For the component-wise interpolation the eigen-analysis 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 Fig. 21 and Fig. 24. These images are close-up views of the one-point load data set, both using eigenvector-based interpolation. Fig. 21 depicts the difference in the resulting triangulation, and Fig. 21(b,d) in particular the degenerate elements of higher dimensionality for a Delaunay re-triangulation. Fig. 24 demonstrates the effect on the topological graph. In the bottom image only vertex insertion for degenerate points is performed, whilst the top image was computed using a subsequent Delaunay re-triangulation. The re-triangulation combines seven degenerate points in one degenerate poly-line. The classification of the half-sectors along the degenerate poly-line reduces the number of separatrices from 35 (bottom image) to 14, but still consistently captures the dominant eigenvector patterns in the vicinity. 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 also have applied our method to data sets representing the simulation of different force combinations acting on a solid block. Fig. 26 and 27 show complete segmentations of a slice of each data set. In Fig. 4.27(a) the cells are randomly colored. The other images (Fig 26 and 4.27(b)) display the blue and red tensor lines 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. 4.27(b) emphasizes that this method can be applied to two dimensional tensor data given on a multitude of grid types or even scattered data. The only condition is that the data points are triangulated as preprocessing step.

4.5.1 *Discussion*

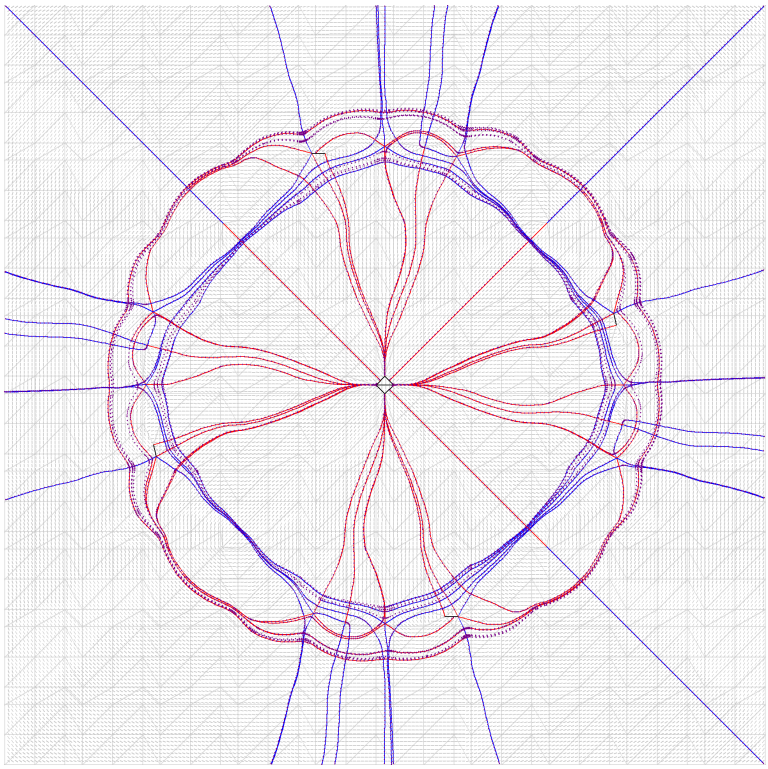
The presented method above allows the extraction of an eigenvector-based segmentation of two dimensional symmetric tensor fields. The segmentation is given by the integral topological graph, which captures the topology of both eigenvector fields. Applying topological methods for tensors has the clear benefit that it is guaranteed that all structural features are captured. The integral topological graph is given as explicit geometrical structure that can be used as visualization itself but also for any further processing or visualization. Tensor interpolation defined with respect to the topology guarantees consistent results.

The simplification of the graph structure induced by the degenerate elements of higher dimensionality can be considered as linear approximation of *multiple* degenerate points (Sec. 2.7.1). It is purely based on the nature of the data, no user interaction, thresholds or heuristics have to be applied to achieve the simplification. In this regard pursuing the idea of a data dependent re-triangulation for an even stronger support towards the formation of degenerate elements of higher dimensionality would be interesting future work. Here, quantities like the isotropy in the field are a considerable measure.

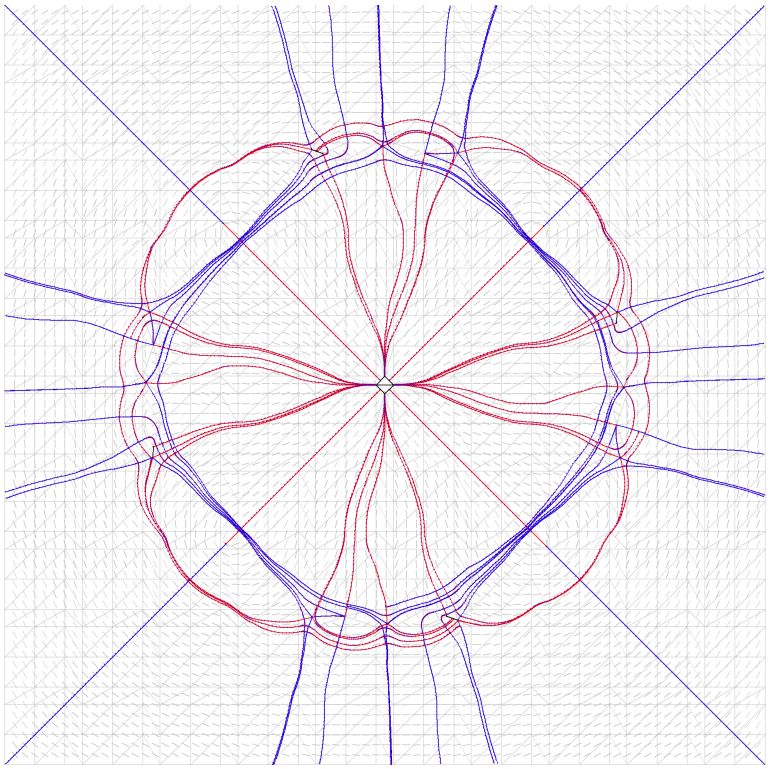
With the help of edge-labels given in a pre-processing step to the underlying triangulation structure the interpolation and the location of degenerate points can be performed very efficiently. Also the computation of start directions of radial tensor lines is strongly alleviated as instead of a third order polynomial only a linear equation has to be solved.

4.5.2 *Conclusion*

We have proposed a method to compute an eigenvector-based segmentation of two dimensional symmetric tensor fields by means of the extraction of the integral topological graph. This graph is an explicit geometric structure that partitions the field into regions of qualitatively homogeneous eigenvector behavior. As both eigenvector fields are considered the topological graph delivers curvilinear cells bounded by separatrices which determine the eigenvector behavior within. However, as mentioned before the tensor field topology ignores eigenvalues characteristics as it is defined by the eigenvector fields. The work in Chap. 5 will build on the structural accuracy of the integral topological graph and extends the segmentation towards eigenvalue characteristics.



(a) Component-wise interpolation



(b) Eigenvector-based interpolation

Figure 28: Slice of one-point load – complete eigenvector based segmentation represented by the integral topological graph with (a) component-wise interpolation (b): eigenvector-based interpolation. Blue dots int the images indicate points in cleaned up circulating separatrices (according to termination conditions in Sec. 4.3.5)

.

5 TENSOR FIELD SEGMENTATION

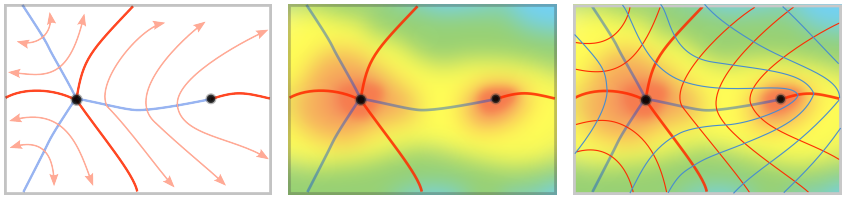
In the following we present a segmentation that is based on the topology extracted in the previous section (Sec. 4.3). As described before the topology of a tensor field is defined by the eigenvector fields and is hardly reflecting eigenvalue characteristics (apart from eigenvalues being equal in degenerate points). This approach presents an adaptive refinement process that extends the extracted topological graph to a full segmentation also respecting the eigenvalue fields.

The segmentation provides a structure that delineates cells of similar (or dissimilar) behavior in the underlying field. This allows a qualitative, focused comprehension of the field properties. The resulting higher-level of abstraction of the field provides valuable analysis and allows a top-down exploration of the given data set. Additionally, the structure of the segmentation is given explicitly and can be used for further statistical analysis inside the cells.

The extraction of the integral topological skeleton using both major and minor eigenvector fields serves as a structural pre-segmentation. The resulting curvilinear cells are bounded by tensor lines and already delineate regions of equivalent eigenvector behavior (Fig. 29(a)). This pre-segmentation is further adaptively refined to achieve a segmentation also respecting eigenvalue characteristics. Cell refinement involves both subdivision and merging of cells achieving a predetermined resolution, accuracy and uniformity of the segmentation. The building blocks of the approach can be intuitively customized to meet the different demands or applications. The adaptive refinement process of the segmentation is guided by the definition of a scalar invariant as similarity or dissimilarity measure, see Fig. 29(b),(c). Depending on the application, a variety of scalar invariants can be used. For example, anisotropy and maximum shear stress reflect the relation of eigenvalues and are of high importance in many applications. A generalization of the notion of anisotropy to non-positive definite fields allows us to extend our analysis to all tensor fields. In our approach, several dissimilarity measures can be applied either singly or as combinations. Application to tensor fields from numerical stress simulations demonstrates the effectiveness of our method.

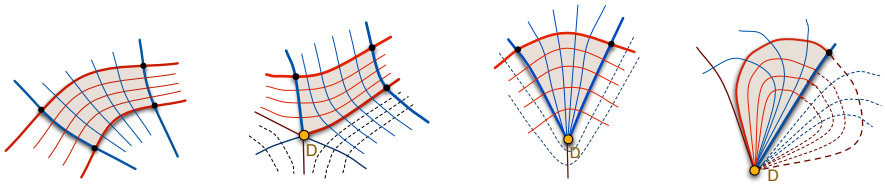
5.1 INITIAL CELL GENERATION

For the pre-segmentation we extract the topology of the tensor field as described in Sec. 4.3. Please note, the segmentation process is not



(a) (b) (c)

Figure 29: Schematic illustration of the segmentation process. (a) Step 1: Integral topological graph with degenerate points as black dots, separatrices as bold lines (major in red, minor in blue). The light red lines depict tensor lines within the segmented regions and exemplary illustrate how separatrices aggregate homogeneous eigenvector behavior. (b) Step 2: Definition of scalar field reflecting the eigenvalue fields. (c) Step 3: Refinement of topological graph according to scalar field.



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

Figure 30: Cells defined by the topological skeleton: (a) regular cells without any degenerate points, (b) hyperbolic sector, (c) parabolic sector, and (d) elliptic sector.

restricted to our topology extraction – the traditional topology computation [Del94] based on the component-wise interpolation model could serve as basis, as well. As the initial cell structure given by the topological skeleton will be refined according to the eigenvalue characteristics the technical aspect of the cell generation and associated data structures is introduced here in more detail.

After computing the topological skeleton for the major and minor eigenvector fields the intersections of the red and blue separatrices define the cells of the pre-segmentation. In detail, the vertices of the resulting curvilinear cells are either intersection points of a red with a blue separatrix, degenerate points, or intersections of separatrices with the boundary. The separatrices form the edges of the initial cells. The cells exhibit one of the following basic structures, see Fig. 30:

1. Cells without a degenerate point are quadrangular with two red and two blue tensor lines as boundary, in an alternating order. All red tensor lines passing through this segment enter at one blue boundary and leave the cell at the opposite boundary. All intersection angles are orthogonal.

2. Cells with one degenerate vertex lying in a hyperbolic sector are quadrangular. The angle at the degenerate point is in general not orthogonal.

3. Cells having a degenerate point in one vertex, lying in a parabolic segment, degenerate to a triangular shape.

4. In elliptic sectors, cells with either two or three vertices are possible.

5. Cells containing degenerate lines as edges can exhibit all kinds of complicated structures.

The edges of the cell are segments of the separatrices and hence are represented as polylines. The edges are ordered in counterclockwise orientation of the cell, and stored in a doubly-linked list, for efficiency in finding neighbors to the cell and adjacent edges in a cell. Each edge is represented using a half-edge data structure.

5.1.1 *Half-edge Data Structure*

A half-edge data structure [cga, dBvKOSoo] is an edge-centered data-structure that maintains spatial information of vertices, edges and cells. Each edge is shared by two cells. An edge can also be considered as two opposite directed half-edges, called twins. Each half-edge stores its start point, the end point of a half-edge however is determined indirectly by referencing to the start point of the twin. The prime advantage of using this data structure is that a half-edge and its corresponding cell share a one-to-one relationship. Consequently, neighbor-searches and an iteration through the cells become very efficient.

Half-edge twins always belong to the same separatrix, except in the cases when the edges are part of either boundaries or degenerate lines. As separatrices are represented by polylines, the half-edge data structure is represented by polylines. Our implementation using CGAL [cga] additionally has to support irregularities in the cell layout, namely T-junctions or hanging nodes, where twins have an n:m relation, such that they share common points of the separatrix polyline but do not share the same start and end points, see Fig. 31(a). To resolve the issue of continuities in irregularities, our half-edge data-structure is modified as follows: (a) We store pointers to points representing the current edge; (b) Two sets of twins are supported for each edge - (i) *a geometric twin*: a single edge to identify the geometrical limits of the edge and (ii) *neighboring twins*: an array of twins to identify all neighboring cells in case of hanging nodes. The geometric twin of an edge is the flipped image of the edge with respect to its starting and end points, which would ideally be the twin but necessarily need not exist in the topological skeleton. The neighboring

twins of an edge is the segmented set of the first twin, which are the edges that actually exist in the skeleton. In the absence of hanging nodes, the second set is a singleton set of the first twin.

5.1.2 *Cell Extraction*

The actual cell creation process involves physically creating the half-edges from the topological skeleton, and using them to build the curvilinear cells. Starting with a single cell as a seed cell, its neighborhood is grown to find the entire set of cells. Convex cells can be found by a strict rotation angle criteria at the vertices in a counter-clockwise orientation. The remaining non-convex cells in the vicinity of degenerate lines or triangles are found by implementing a greedy walk of finding consecutive half-edges that are not associated with any cells. Consecutive half-edges are all that have the current half-edge's end point as start point.

5.2 ADAPTIVE SEGMENTATION WORKFLOW

The segmentation resulting from the topology already decomposes the domain in regions where the eigenvector fields have a qualitatively similar behavior, but it does not yet fulfill all our criteria for a good segmentation. To represent the entire tensor information also the scalar invariants based on the eigenvalues have to be considered. This is achieved by adaptively modifying the cells, characterized by a specified degree of similarity with respect to eigenvalue behavior.

The segmentation strategy on the initial cell structure builds on two basic operations:

- **Coarsening**: Cells that do not exhibit enough structural information on their own get merged with adjacent cells.
- **Subdivision**: Cells which exceed the defined criteria of similarity are subdivided by new tensor lines.

Due to divergence and convergence of tensor lines, adaptive segmentation inevitably causes occurrences of hanging nodes in the edges. To keep these irregularities to a minimum, a growing strategy is employed where we continue to merge or subdivide on consecutive cells as long as possible and necessary, see Fig.s 31(b,c). Algorithms for coarsening and subdivision operations are described in Sec. 5.2.3.

To guide the modification process by eigenvalue behavior one or more scalar fields are derived from the initial tensor field, which directly render the eigenvalue behavior, see Sec. 5.2.1. The degree of similarity and the need of modification is represented by weight functions defined on the edges of the cells, see Sec. 5.2.2. These edge-weights evaluate the derived scalar field but also reflect geometric

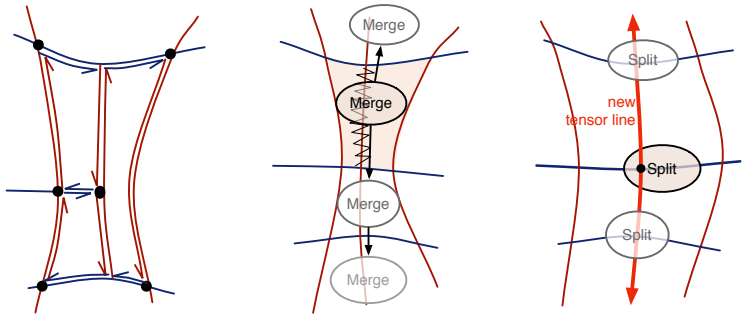


Figure 31: (a) T-junction or hanging node, where neighboring twins in half-edge structure have an n:m relation. Adaptive refinement operations: Recursive strategy for avoiding hanging nodes in (b) coarsening of cells, and (c) subdivision of cells started by insertion of new tensor line.

properties of the cells. Combined with data dependent thresholds the edge-weights serve as decision basis whether cells have to be merged or subdivided. Further the edge-weights help to steer the modification by importance, as their values directly offer prioritization to achieve a smooth segmentation. Using edge-weights for decision-making is an efficient choice as edges are one-dimensional structures, on which the weights are computed.
To extend the capabilities of the adaptive segmentation process, it is designed to have a high degree of flexibility to customize the workflow. For example using this segmentation as preprocessing step for glyph placement would have different demands than using it for texture mapping. We define the variables for customizing the workflow of the adaptive refinement as:

- **Operations:** The operations of the refinement, namely coarsening and subdivision, are the modules of the workflow. They can be repeated and the workflow customized by choosing the number and the order of implementation of the operations.

- **Control Parameters:** The control parameters of the refinement are the chosen edge-weights, the considered scalar fields and the thresholds defined by the demanded accuracy and resolution. The choice of these control parameters impact the priority queues used for the implementation of the operations.

The remainder of this section first presents the operations and control parameters of the approach. Then the basic workflow is demonstrated by calculating a segmentation of the tensor field. The segmentation follows accuracy towards tensor invariant similarity as wells as geometric criteria. Implemented as focus and context visualization, the refinement process is stopped for cells whose size falls below a value proportional to the resolution of the displayed domain. This

provides an overview of the field, on demand the user can specify a focus region to view further detail, see Sec. 5.2.5. The flexibility of the approach is finally shown by the extraction of degenerate regions, see Sec. 5.2.6.

5.2.1 *Choice of Scalar Field*

Several scalar fields can be considered as basis for the refinement depending on the specific application. We can either use both eigenvalue fields, an anisotropy value, the maximum shear stress (Eq. 31) or other tensor invariants.

For the anisotropy we propose the generalized notion of fractional anisotropy FA^* (Eq. 30) which can also be used for non positive definite tensors.

5.2.2 *Edge-weight Definition*

A set of pre-defined functions as weights assigned to the cell edges is provided. It consists of

- geometric measures representing the current cell size and shape as well as
- similarity measures for scalar fields derived from the tensor field.

This set can be extended by user-defined functions. The dominant use of geometric edge-weights favors a more uniform segmentation, whereas the scalar field based weights lead to a higher adaptivity towards accuracy in eigenvalue similarity. In the following two weights of each class are proposed. These weights can be arbitrarily combined.
Let e be an edge of a cell consisting of k segments $(\mathbf{x}_i, \mathbf{x}_{i+1})$, by the virtue of being part of a polyline, where $\mathbf{x}_i$ is the position of the ith point on the edge. Further, let s be a scalar function defined along the edge and $s_i = s(\mathbf{x}_i)$. The edge weights are then defined as given in table 1
We chose the proposed edge-weights to be as intuitive and universal as possible, independent of the various ranges that appear in different data sets. *Variance* is a commonly known statistical quantity and is a similarity measure which is robust to smaller perturbations of scalar values along an edge, such as noise. *Difference of minimal and maximal scalar value* in turn is strict towards any changes of scalar values along an edge and directly renders the absolute difference of scalar values appearing on an edge. *Edge length* can be used to adjust the size of the segmented cells to optimize perceptibility by the user.

$w_v(e)$	Variance of scalar values $s(r\mathbf{x}_i)$ on the edge: $w_v(e) = \frac{\sum_{i=1}^{k}(s_{x_i}-\bar{s})^2\|\mathbf{x}_{i+1}-\mathbf{x}_i\|}{\sum_{i=1}^{k}\|\mathbf{x}_{i+1}-\mathbf{x}_i\|}$, where $\bar{s}$ is the mean of s along e
$w_d(e)$	Absolute difference of minimal and maximal scalar value along the edge: $w_d(e) = abs(s_{min} - s_{max})$
$w_l(e)$	Edge length: $w_l(e) = \sum_{i=1}^{k}\|\mathbf{x}_{i+1} - \mathbf{x}_i\|$.
$w_c(e)$	Change of eigenvector direction along the edge: $w_c(e) = \sum_{i=1}^{k}\|\angle(\mathbf{v}_i, \mathbf{v}_{i+1})\|$, $\mathbf{v}_i$ is the major eigenvector at position x_i.

Table 1: Edge weights.

The eigenvector directions are already well represented by the cell shape and tensor line boundaries, however if uniformity of the cells is required *change of eigenvector direction* represents the curvature of the cell boundaries and is therefore an appropriate measure.
In Sec. 5.2.5 we give a preset of thresholds for these weights, which are calculated as percentages of the given ranges in the field. These presets led to stable results of good quality, which experiments with different data sets showed, see Sec. 5.3. However, they can be intuitively strictened or loosened for different visualization purposes with immediate interpretation.

5.2.3 *Refinement Operations*

If not noted differently the refinement operations always respect the chosen thresholds towards the edge-weights. For example if a minimum edge length for the coarsening operation is specified - no edge subdivision is performed if one of the new edges would fall below the minimum edge length.

5.2.3.1 *Coarsening*

The main goal of the coarsening operation is to get rid of small cells that do not carry enough structural information on their own. Coarsening operation involves merges of cell pairs. Merging a pair of cells requires the merge of up to two pairs of edges and removal of the common edge of the cells (see Fig. 31(b)). For this operation, we build an edge-weight based priority queue of pairs of cells that can be merged. We use queues to follow the FIFO (*first in, first out*) order, ascending or descending priority is fixed by minimum or maximum thresholds respectively.

Merge Prerequisites – Based on the geometry layout, two cells can only be merged if they share *a common edge* that can be deleted to join these cells. Technically, a common edge between two cells means

that one of the cells has an edge whose geometric twin is an edge of the second cell. Edges containing hanging nodes cannot be common edges.

Priority Queue and Sorting – For the coarsening operation a priority queue of pairs of adjacent cells that can be merged is maintained. A multi-pass sort is performed based on the number and order of edge-weights. Like this edge-weights favoring different criteria can be combined.

For the example of the segmentation workflow in Sec. 5.2.5 the priority queue is first sorted by minimum edge length of all cell edges involved and then by maximum edge length of the edges to be merged.

Algorithm

- Check adjacent cells for if they can be merged and sort these into the priority queue, based on the chosen edge-weight prioritization.
- While the priority queue is not empty, the pair with the highest priority is merged.
- Update the data structure by merging the appropriate edges of the pair cells, deleting the common edge and creating a new cell from the new edges.
- Update the priority queue with the new merged cell.
- If contiguous cell pairs in direction of the deleted common edge are to be merged move them on top of the queue to accomplish the recursive workflow. See Fig. 31(b).

5.2.3.2 *Subdivision*

Single cells are subdivided by starting a new tensor line of opposite color on one of its edges that has to be subdivided. Similar to the coarsening operation an edge-weight based priority queue implemented as FIFO.

Start Point of Subdividing Tensor line – Two possibilities for the start point of the new tensor lines are provided. The first option favors the generation of equally sized cells, and starts the tensor line in the midpoint of the edge. The second option starts the tensor line between the extrema of the scalar values on the edge. This choice is more adapted to the data and guarantees to decrease the edge-weight when subdividing. There are no technical prerequisites to subdivide a cell.

Priority Functions and Sorting – Differently from the coarsening operation, a priority queue for edges is used rather than cells. Again the priority queue can be sorted according to multiple edge-weights. The growing strategy in subdividing consecutive cells is implemented by integrating a subdividing tensor line as long as possible and necessary, see termination conditions (a,b) in the algorithm below. No explicit prioritization has to be done.

Algorithm

- Sort edges to be subdivided into priority queue, based on chosen edge-weights.
- While the priority queue is not empty, pop the top edge and start a subdividing tensor line of opposite color.
- Integrate tensor line until one of the following termination conditions is reached:
  (a) It intersects an edge, which is not in the priority queue and therefore should not be subdivided.
  (b) It intersects an edge and it's subdivision would generate edges violating fixed edge weight thresholds, for example minimum edge length.
  (c) It fulfis one of the termination conditions described in Sec. 4.3.5.
- Subdivide all cells corresponding to edges intersected by the new tensor line, as shown in Fig. 31(c). Update data structure by subdividing intersected edges, generating new edges along the tensor line, and finally generating new subdivided cells using the new edges.
- Update priority queue by deleting the original edges intersected by the new tensor line, and adding and sorting the newly generated edges if they are candidates for further subdivision.

5.2.4 *Customizable Adaptive Refinement*

The possibility to customize the decisive quantities and the workflow itself gives a high degree of flexibility in obtaining various analyzes of the same data set. Essentially the workflow consists of modules for operations, which are influenced by the control parameters and strategies adopted for implementation. Variations in the workflow are achieved by changing the number and order of the modules, by adjusting the thresholds used for each operation, and by deciding on the strategies to be used for the control flow of the modules. Strategies include the choice of appropriate edge-weights and scalar fields and choice of position of starting a new tensor line for subdivision of edges.

Basic operations	coarsening	subdivide
Edge-weight prioritisation	geometric	scalar field
Error measure for edge-weight	variance	max difference
Tensor line seeding	middle of edge	between max and min
Level of detail	resolution	accuracy

Table 2: Table to summarize the options when configuring for the segmentation process

For a domain expert the flexibility of the approach ranges from using the presets with the scalar field of his choice over strictening or loosening thresholds to mixing and matching the implemented components to his needs. Developers can extend the basic set by implementing new elements, as e.g. edge-weights or tensor line seeding.

5.2.5 *Workflow: Basic Segmentation*

This workflow delivers a focus and context visualization, calculating an initial context segmentation which can be browsed in detail by selecting a focus region. The field is segmented in regions of similar tensor invariant behavior. We chose FA* (see Sec. 5.2.1) as scalar field to render the eigenvalue characteristics. For all operations the same edge-weights and thresholds are used. Thresholds are denoted by φ_i where i denotes the index of the corresponding edge weight w_i (Table 1). We chose the actual values of the thresholds to be relative (percentages) to the occurring value ranges. This enhances the usability as pre-defined parameters for a variety of data sets with strongly differing ranges of eigenvalues (see also Sec. 5.3).

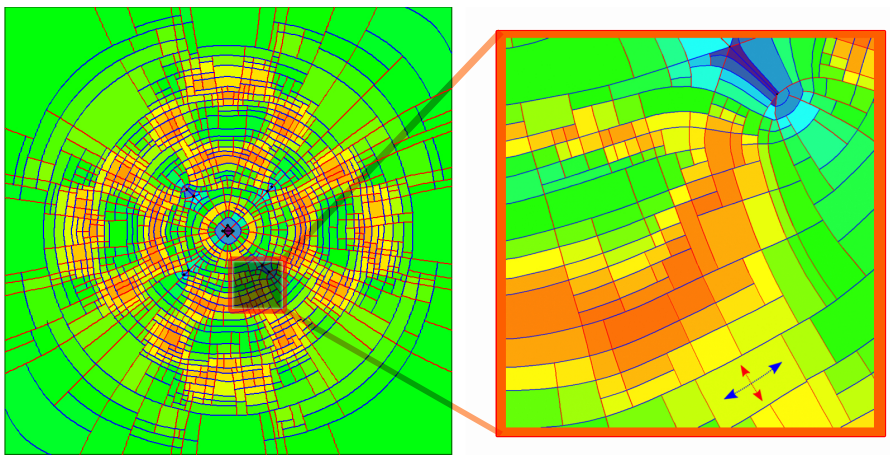


Figure 32: Focus and context visualization. The orange rectangle in the right image is the chosen focus region On the right, the focus region is displayed. As the minimum edge length is adapted according to the current resolution the cells are further refined.

The default parameters are, if not differently noted:
(a) Geometric edge-weight steering the granularity of the segmentation is the edge length w_l with minimum edge length threshold fixed to $\varphi_l = 1\%$ of the displayed domain range. By selecting a focus region the minimum edge length is automatically adjusted and the segmentation is refined displaying further details.
(b) Scalar field edge-weight steering accuracy is the absolute difference of minimum and maximum scalar value w_d, with its threshold set to 10 % of the scalar value range, which is for FA* as scalar field $\varphi_d = 0.1$.
Resolution and accuracy are the basic level of detail parameters for focus and context visualizations, where the geometric edge-weight has higher priority than the accuracy edge-weight. This means for edge lengths smaller than the fixed minimum edge length a merge is performed even if the merged edge exceeds the given accuracy threshold.

The following operations composite the workflow

1. First coarsening: in merging as many similar cells as possible cleans up the pre-segmentation , especially very small cells are removed. The priority queue is first sorted by minimum edge length of the edges involved and then by maximum edge length of the edges to be merged. Merging of small cells with rather large cells favours the goal of a smooth segmentation.

2. Subdivision: the cells are refined to the pre-defined accuracy, unless this violates the resolution criterion. The tensor line seeding is between the extremal points. Experiments showed that the best strategy for a smooth segmentation is to do a 2-pass sort of the priority queue first based on maximum edge-length, the second pass based on maximum scalar edge weight.

3. Final coarsening: cells with highest similarity are merged. The priority queue is sorted first by minimum edge length of the edges involved and then in ascending order by a pre-calculated scalar edge weight w_d of the edges to be merged.

For a chosen focus step 2 and 3 are repeated with adjusted geometric edge weight, the accuracy edge weight remains. As the cells are given as explicit entities, cells exceeding the accuracy edge weight can be highlighted on demand.
This workflow and thresholds can be used in any tensor field segmentation as stable presets. Results for using *variance* as scalar field edge-weight are given in Sec. 5.3.

5.2.6 *Workflow: Degenerate Regions*

In the direct vicinity of degenerate entities tensors are almost isotropic and thus directional behavior is not strongly expressed. Such ar-

eas are collectively represented as *degenerate regions*, following the paradigm of the coarsening operation, that all cells are combined which do not exhibit enough structural information on their own. These degenerate regions are "grown" from the degenerated entities using two passes of the subdivision operation. FA^* is used as scalar field.

1. First subdivision: In the first pass only edges, called *degenerate edges*, emerging from a degenerate entity are considered for subdivision using a specific edge-weight, which is the maximum anisotropy occurring on the edge. The start point for the subdividing tensor line is the point on the edge (given as polyline) with anisotropy below a fixed threshold $\varphi = 0.05$ and with maximum distance from the degenerate entity. prioritisation is done by the least anisotropy value of the start points.

2. Second subdivision: As it cannot be guaranteed that the subdividing tensor lines will intersect other degenerate edges at points with anisotropy below φ, a second subdivision pass is performed, where all edges participating in the intermediate degenerate regions from the first subdivision step are considered.

It should be noted that the generation of degenerate cells may induce subdivision of neighboring cells that may result in small not well shaped cells, which may not get merged in a later step, as seen in Fig. 37. It is still of benefit to extract degenerate regions, as the weak expression of direction in such areas can result in numerical instabilities.

5.3 RESULTS AND DISCUSSION

We tested our algorithm on three data sets from structural engineering, which are finite element simulations of forces acting upon solid blocks resulting in stress tensor data. These are simulations of one (Sec. A.1.2) and two (Sec. A.1.3) forces applied to the top of a solid block and of multiple forces applied to a notched block (Sec. A.1.4). Again, we will refer to them as *one-point load*, *two-point load*, and *notched block*, respectively. For the statistical analysis we will abbreviate them as 1PL, 2PL, and NB, respectively.
Fig. 32 shows the focus and context implementation proposed in Section 5.2.5 on the one-point load data set. The right image shows the selected focus. The focus is automatically further refined to the resolution threshold: the allowed minimum edge length is adapted to 1% of the displayed domain range, the accuracy threshold remains. In this visualization the cells delineate regions of similar eigenvalue behavior, the color coding of the cells renders the relation of the eigenvalues.

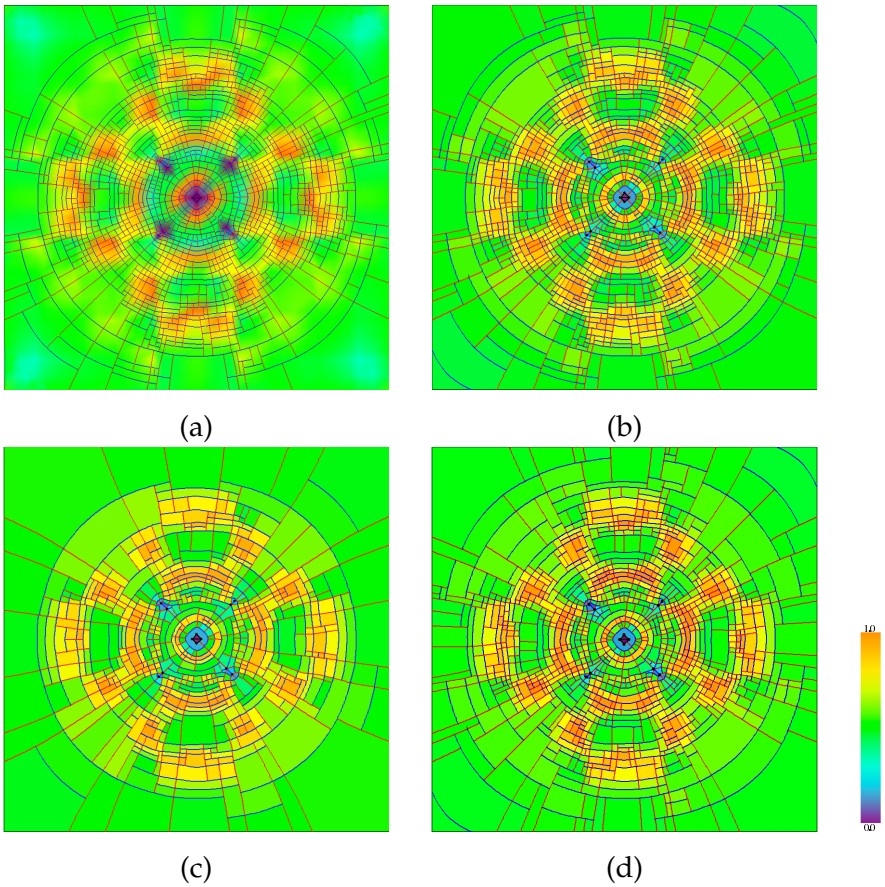


Figure 33: Adaptive segmentation of a slice in the one-point load data set. (a) Superimposition of scalar field with adaptive segmentation of tensor field. Choice of threshold for edge-weight w_d with (b) φ_d=0.1 and (c) φ_d=0.2. (d) Segmentation with edge-weight w_v and $\varphi_v = 0.015$

The two arrows in the lower right corner of the focus image schematically indicate how the eigenvector behavior can be interpreted from the cell boundaries.
We conducted further analysis on thresholds, specific control parameters and strategies. Please note, as mentioned above thresholds are denoted by φ_i where i denotes the index of the corresponding edge weight w_i summarized in Table 1.

To evaluate the quality of the overview segmentation we used the following methods:

- *Image representation of local error*: The error is sampled in an image of resolution $u \times v$ where each pixel (i, j) is mapped to a point (x, y) in the tensor field, and is mapped to error value $err(i, j) = \|s(x, y) - \bar{s}_{cell(x,y)}\|$, where $cell(x, y)$ is the cell containing (x, y) and $\bar{s}_{cell(x,y)}$ is the average scalar value on the edges of cell $cell(x, y)$. Error values are assigned a color from

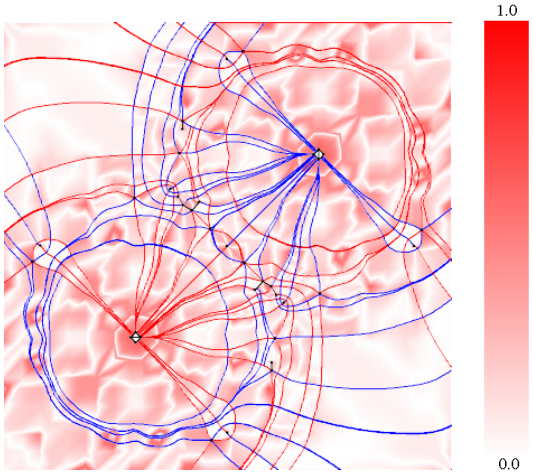
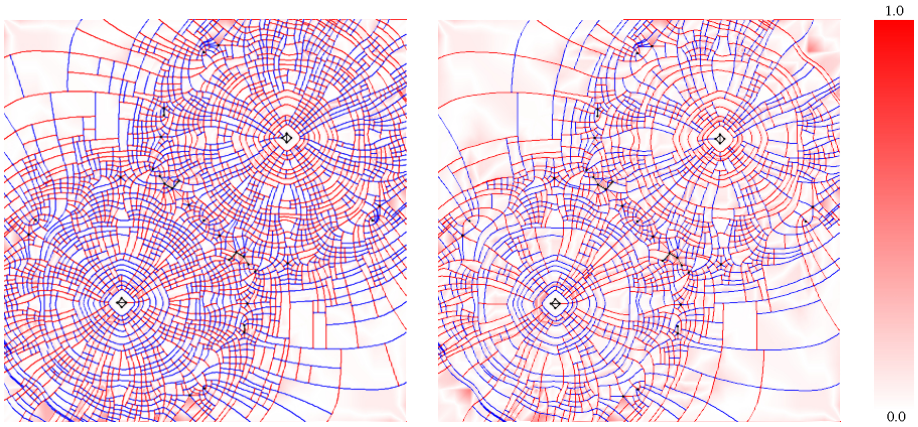


Figure 34: Image representation of error for a specific slice of two-point load data set. Here the error is plotted against the initial segmentation of tensor field (integral topological graph).

a red-shaded colormap. Results are given in Fig. 34, 35, and 36(a).

- *Average error:* While computing the image representation of the local error, we calculate the average error for the field as $\oslash \mathtt{Err} = \frac{\sum_{i,j} \mathtt{err}(i,j)}{u * v}$.
- *Number of cells needed for pre-defined quality:* We aim to have as few cells as possible in an adaptively segmented field, which makes the number of cells needed to achieve a segmentation of predefined quality an important criterion.

The first example examines level of detail according to accuracy, see Fig. 33. In Fig. 33(b) the threshold φ_d for the absolute difference of scalar values along the edge is twice as strict as for Fig. 33(c). Using superimposition of the original scalar field on the segmentation result of the one-point load data set, Fig. 33(a) demonstrates that the segmentation and original scalar field match. Fig. 33(d) was generated by using the variance guided edge weight w_v. The image shows that variance gives a smoother segmentation. This edge weight is not as sensitive to local changes of scalar values like the maximum difference (w_d) is.
The second example focuses on resolution as level of detail. In Fig. 35(b) the minimum edge length φ_l is twice as strict as for Fig. 35(c). All images are superimpositions of error images and the segmented cell boundaries. Fig. 35(a) shows the pre-segmentation, and Fig. 35(b,c) the results for the different levels of detail.



(a) (b)

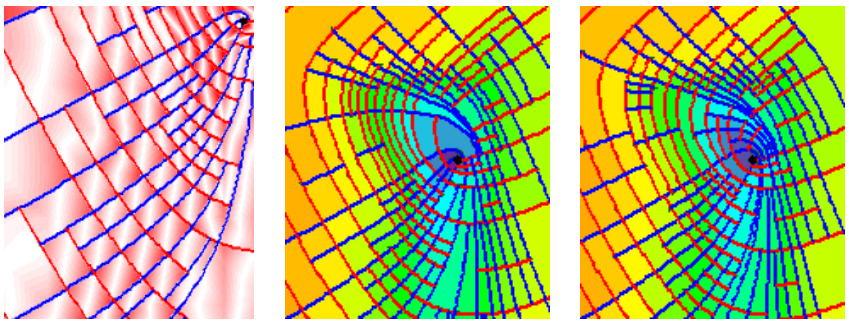
Figure 35: Image representation of error for a specific slice of two-point load data set. (a) Adaptive segmentation using w_d (absolute difference of scalar values along the edge) as edge-weight. (b) Adaptive segmentation using double threshold for minimum edge length.

As differences in images may be hard to perceive we further evaluate the results analytically. Results are displayed in tables with references to associated images, if available. The first block in Table 3 is used as reference for the subsequent evaluation, it lists the values for the overview segmentation of Sec. 5.2.5.
Test 1: Comparison of edge-weights, variance w_v vs. absolute difference of extremal values of scalar field w_d:
As threshold for the variance of the scalar value w_v 0.1% of the scalar value range is used, given by $\varphi_v = 0.001$. The middle block in table 3 and Fig. 33(d) demonstrate that w_v leads to fewer cells at a higher mean error, compared to w_d used in the basic workflow. Variance is a criterion that regards mean values and ignores smaller variations along an edge. Thus, fewer cells have to be subdivided, see Fig. 33(d). The smoothing effect of the variance-based edge-weight can be extensively used to achieve desired results. w_d is an edge-weight that is rather strict and regards any change in the scalar field along the edge, which results in more cells, but higher accuracy.

Test 2: Comparison of strategies in choice of start point of tensor lines for subdivision operation: midpoint of an edge vs. point between extrema of the scalar field along an edge.
Using the midpoint of an edge is a simple, straightforward technique, which leads to a slightly higher mean error, as shown in Table 3 last block. Starting the subdividing tensor lines between the minimum and maximum scalar values guarantees decrease in the edge-weights. This leads to qualitatively higher subdivisions as shown in Fig. 36(b,c).

Data	2PL	1PL	NB
Basic workflow:			
#Cells	2830	1424	420
⌀Err	4.67e-2	3.38e-2	4.02e-2
Ref. Fig.	Fig. 35(b)	Fig. 33(b)	Fig. 36(a,c)
Accuracy edge weight *variance*:			
#Cells	2497	1129	285
⌀Err	5.039e-2	3.64e-2	4.93e-2
Ref. Fig.		Fig. 33(d)	
Start point for subdividing tensor line in the middle:			
#Cells	2870	1414	402
⌀Err	4.75e-2	3.38e-2	4.02e-2
Ref. Fig.			Fig. 36(b)

Table 3: Results for tests on control parameters and strategies 5.2.5.



(a) (b) (c)

Figure 36: Close up of adaptive segmentation of a slice in the notched block data set: (a) Image representation of error shows how the edge-weights only reflect the behavior of the scalar field on the cell edges but not in its interior. Results of the choice of start point for tensor lines in the subdivision operation, at (b) the midpoint of an edge and (c) the midpoint of extrema along an edge.

5.4 CONCLUSIONS

We have presented a segmentation that enhances the interpretation of topology by extending it towards eigenvalue characteristics. The segmentation represents the field in terms of reduced complexity. An essential step to reduce the amount of information is a segmentation that separates the field into regions of similar characteristic behavior. The resulting cells in the final segmentation can be considered as visualization glyphs in form of tiles. They are bounded by tensor lines

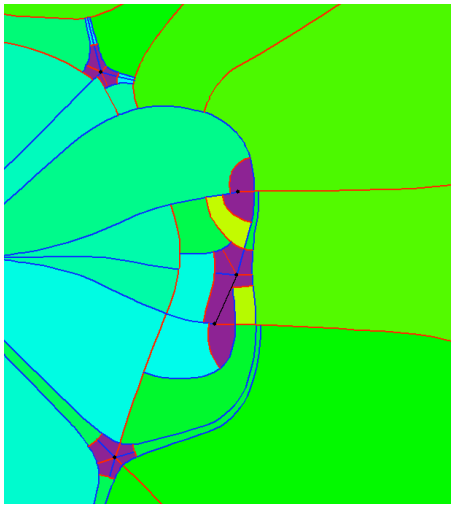


Figure 37: Close up: degenerate regions marked in violet.

which – as the segmentation is based on topology – allows immediate interpretation of the eigenvector behavior within. Appropriate color coding can display the eigenvalue characteristics in the segments. An overview of the data is provided without missing important details and at the same time without overwhelming the observer.
This higher level of abstraction allows a top-down exploration of the given data set. Further, the geometry of the resulting cells is represented explicitly and offers statistical inquiry of properties inside each cell. It has to be mentioned that the generation of the extracted cells requires special care. Numerical rounding errors in the intersection computation of the major and minor tensor lines can make the determination of the correct bounding tensor lines for very small cells difficult. This can be avoided by computing these with integer based arithmetic as provided by CGAL [cga].

We have shown that the presented segmentation approach is able to generate segmentations aligned to the tensor field with low error measures. The weights and strategies can be chosen dependent on the demands of the specific application and allow a wide variety of data representation.

The results in this section are based on the generalized notion of fractional anisotropy FA^* (Eq. 30) of the tensor field. However, it can be easily extended to other scalar fields used individually or as a combination of several fields. The main challenge for a segmentation based on tensor lines is the fact that feature lines resulting from the scalar field are in general not aligned with the eigenvector field and thus they can only be approximated by a step function (see Fig. 36(a)).

The strength of the application lies in its flexibility - restricting the modification process to geometric edge-weights delivers a uniform segmentation, whereas steering it by accuracy edge-weights gener-

ates a highly adaptive segmentation. There is a variety of options to customize the segmentation and use it for example for tensor line seeding or as preprocessing step for glyph placement or texture mapping (see Chap. 6).

During the design of the algorithm at many points decisions were made to balance its efficiency and accuracy. One example is the choice to define weights only on basis of the scalar field along cell edges, which is very efficient. But it cannot be guaranteed that the error contained in a cell is always well represented by the error along its one-dimensional edges. An integrated analysis of all cell edges, which leads to use of cell-weights as opposed to edge-weights is one of the interesting options for the future.

6

GLYPH- AND TEXTURE- BASED VISUALIZATION

Topological methods have a long tradition and for domain experts they serve as visualization itself. However, for untrained users this might be too abstract – especially if the visualization is meant to support a first understanding of the data. In this chapter we show how the variability of visualization methods like textures and glyphs can be used to enhance established methods like topology. The topology-based segmentation of Chap. 5 serves as framework to map textures and place glyph exponents for two dimensional symmetric tensor fields. The textures encode physical properties of the underlying field and support the understanding of the field as a whole, whereas the glyph exponents can be used to give detailed insight at distinctive locations. Textures are a powerful tool to design continuous visualizations. They offer many parameters that can be used to encode features of interest. This makes textures especially interesting for the visualization of tensor fields. Tensor fields contain directional and scalar features which can be encoded in texture parameters. One challenge for many texture-based methods is the definition of appropriate texture coordinates. This is accomplished by the cells of the topology-based segmentation; they serve as consistent framework for the texture mapping. A dense visualization of the underlying tensor field can be generated using every pixel of the image to convey physical properties of the field. Furthermore, a diversity of visualization options arises: To encode directional information, for example, stripe patterns can be used; but also knitting or basketwork patterns are presented and evaluated towards their applicability to depict central tensor properties within the segmentation framework. The goal hereby is not to display all possible features in one image but to offer a flexible framework to a user to switch views for the different points of interest.

Displaying glyphs at discrete locations allows to present a wealth of features with one item. The placement of glyphs is an intricate topic for vector as well as tensor visualization. Here, we use the tensor field segmentation to display glyphs at distinctive locations in the given cells. The advantages of the global structural information provided by the topology-based segmentation and the detailed depiction of representative tensors via glyphs are combined in a single image.

6.1 METHOD

In this work we build upon the results of the segmentation (Chap. 5) and extend it to a texture- and glyph-based visualization. Mapping textures into the segmented cells yields a continuous rendering of the tensor field. This facilitates a comprehension of the field's global nature. The glyph-based approach combines the advantages of the global structural information provided by the topology and the local detailed view of representatives via glyphs.

Cell structure – As explained in Sec. 5.1 cells that are not adjacent to degenerate points or the domain boundary are quadrangular and bounded by two major and minor tensor line segments in alternating order (Fig. 30 and 6.39(a)). Other cells can have more general shapes. The boundaries of the extracted cells are stored as polylines (Fig. 38).

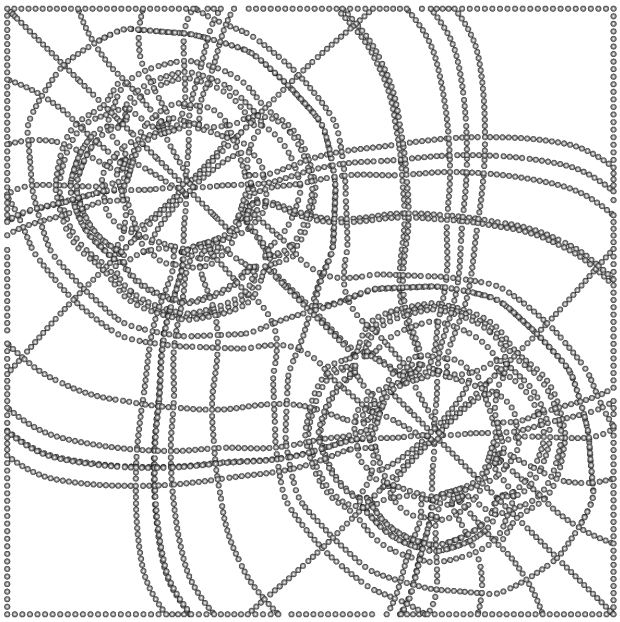


Figure 38: Result of a segmentation, points of the cell bounding polylines are depicted as spheres.

Preprocessing – As mentioned in Sec. 4.3.5, the bounding tensor lines are computed by an integration scheme with adaptive step size. This guarantees accurate results in the segmentation process but leads to irregular distances between the vertices of the polylines. To obtain good results for the texture mapping and glyph locations, however, a more uniform sampling of the cell boundaries is favorable. This is achieved by a pre-processing step deleting respectively adding vertices if the distance to adjacent vertices does not fall in a pre-defined distance interval. An angle criterion guarantees that tensor lines are still sufficiently aligned with the eigenvector fields.

For the texture mapping the segmented cells need to be triangulated. We use the CGAL [cga] implementation of a constrained Delaunay triangulation that maintains non-convex shape.
Please note, the pre-processing only has to be performed once after the segmentation process.

Eigenvalue Mapping – The goal of this work are intuitive visualizations of indefinite tensors. We will use the eigenvalues to control basic texture parameters such as texture density. We adopt a mapping that simulates a texture deformation generated by the underlying tensor field presented by Hotz et al. [HFH+04]. Thus, negative eigenvalues (compression) lead to dense and positive eigenvalues (expansion) to sparse textures. Therefore the eigenvalues are mapped into a restricted positive interval. Hotz et al. define:

$$F(\lambda) = a + \sigma \cdot f(\lambda) \; . \tag{40}$$

The function f is chosen to have a large slope in the neighborhood of zero. In this work, f is the hyperbolic tangent, which preserves the differentiation of negative and positive eigenvalues. The parameter a relates to an offset and σ is an additional scaling factor for the slope. Both can be adjusted by the user.

6.1.1 *Segmentation-based Glyph Placement*

The characteristics of the tensor field – the eigenvectors and eigenvalues – are similar inside each extracted cell (Chap. 5). Thus, the essential tensor properties of each cell can be visualized by one representative glyph. The task is to find an appropriate position within each cell to place this representative.
Since most of the segmented cells are non-convex we follow an algorithm for the computation of *barycentroids* of arbitrarily shaped planar polygons [RLF09]. This algorithm is based on an *interior distance* measure. The barycentroid is defined as the point with minimal average interior distance to the boundary points. Finding this point is a convex optimization problem and can be solved by standard gradient descent routines. The barycentroid has the characteristic that it captures the *semantic center* of the polygon and lies inside any arbitrary shaped planar polygon. See Fig. 41 for results.

6.1.2 *Segmentation-based Texture Mapping*

Using the segmented cells as basis for the texture mapping has several benefits. The cells inherently provide the parametrization for the texture mapping and the underlying topology ensures structural correctness. Also as the segmented cells are bounded by tensor lines the

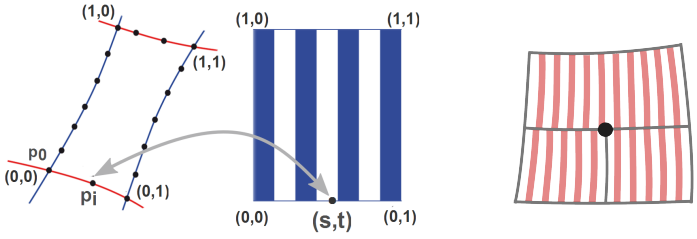
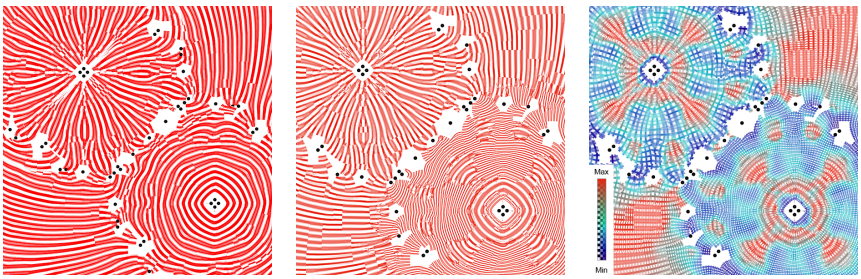


Figure 39: (a) Texture mapping on segmented cell. Mapping of point p_i to texture coordinate (s,t) in quadrangular cell. (b) Illustration of a hanging node for neighboring cells (black dot).

eigenvector directions within are correctly pre-determined. Simple procedural stripe textures already depict one eigenvector field. Thus, the use of textures with one or two orthogonal dominant directions results in continuous representations of the correct eigenvector behavior within these cells (Fig. 40). But also more sophisticated textures, like knitting patterns, lead to expressive representations (Fig. 43). The density of the texture pattern will also be used to reflect physical properties of the tensor field, such as compression and expansion (e.g. Fig. 40(b)).
Cells containing degenerate elements along their boundary can have more complex shapes and the eigenvector behavior cannot be easily represented by simple stripe patterns. In addition, in the proximity of degenerate elements the eigenvector behavior is weakly expressed. For such cells two options are provided. Either these cells are skipped or textured with an isotropic noise pattern encoding information about the isotropic eigenvalues.
For one-directional textures, as stripe patterns, one image per eigenvector field is computed. To depict both eigenvector fields in one image the results for each field are blended (Fig. 40(c), 42).
The texture mapping is performed by vertex and fragment shaders. Texture coordinates (s,t), with $s,t \in [0,1]$ for quadrangular cells are initially computed by mapping the points of a cell boundary to a unit square (Fig. 39(a)).

6.1.2.1 *Rendering of Eigenvector Directions*

All methods that are presented in the following are based on textures with line-like elements to depict directions. To ensure that the cell size does not affect the perception of the pattern, we need a special mapping approach that provides an approximately constant pattern frequency presented by Hummel et al. [HGH+10] (Fig. 40(a)). They



(a) (b) (c)

Figure 40: Data set: Two-point load. Visualization of major eigenvector field (a) with even pattern frequency, (b) with encoded mapped eigenvalues reflecting the physical properties. (c) Superposition of both eigenvector fields, color by fractional anisotropy.

adjust the sampling frequency according to the image-space partial derivatives η_s, η_t at pixel (i, j) of the texture coordinate (s, t):

$$\eta_s(i,j) = \sqrt{\left(\frac{\delta s}{\delta i}\right)^2 + \left(\frac{\delta s}{\delta j}\right)^2}\,, \qquad \eta_t(i,j) = \sqrt{\left(\frac{\delta t}{\delta i}\right)^2 + \left(\frac{\delta t}{\delta j}\right)^2}\,. \tag{41}$$

The initial texture coordinates (s, t) remain unchanged. The evaluation of the input texture P is modified according to the variation of η_s and η_t and steers the pattern frequency in the final image. $\hat{P}_{l_s,l_t}$ gives the frequency adjusted texture values

$$\hat{P}_{l_s,l_t}(s,t) := P(s \cdot 2^{-l_s}, t \cdot 2^{-l_t})\,, \tag{42}$$

with $l_s = \log_2 \eta_s$ and $l_t = \log_2 \eta_t$. Hence, short edges with high partial derivatives yield a low pattern frequency. For large edges this works vice versa. The resulting pattern frequency also interactively adjusts to the current zoom level and resolution of the image. As resolution levels are discretely defined values for neighboring resolution levels are computed and bilinear filtering applied to achieve a smooth pattern frequency. The evaluation of Equation 41 can be done by built-in functionality of the rendering system.

6.1.2.2 *Rendering of Eigenvalue Characteristics*

Line frequency – The approach of Hummel et al. [HGH+10] serves as basis to encode physical properties like compression and expansion

by the pattern frequency (Fig. 40(b)). This is achieved by replacing Equation 41 by the following:

$$\eta_s(i,j) = \sqrt{\left(\frac{\delta s}{\delta i}\cdot\tilde{\lambda}_1\right)^2 + \left(\frac{\delta s}{\delta j}\cdot\tilde{\lambda}_1\right)^2},$$
$$\eta_t(i,j) = \sqrt{\left(\frac{\delta t}{\delta i}\cdot\tilde{\lambda}_2\right)^2 + \left(\frac{\delta t}{\delta j}\cdot\tilde{\lambda}_2\right)^2}, \tag{43}$$

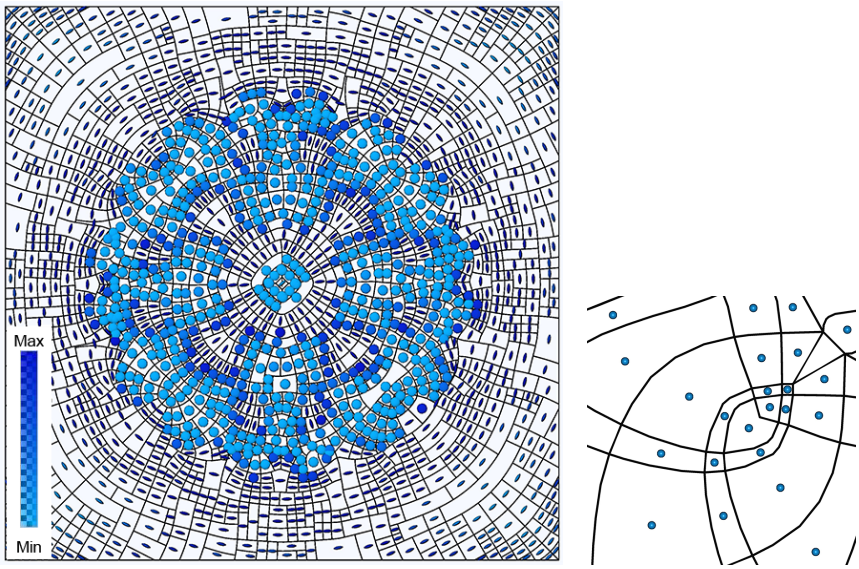
where $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ are the mapped eigenvalues. Thus, the pattern frequency steers the perception of the field: in combination with the mapping (Eq. 40) negative eigenvalues lead to a higher frequency and allude to compression. Mapped positive eigenvalues cause a lower frequency which depicts expansion.

Color mapping – A simple but effective way of conveying additional information is to apply blending of color information on the input textures (Fig. 40(c), Fig. 42(a),(b), and Fig. 43(b),(c)). Coloring can be applied to the eigenvector fields (one color for each field) but also according to derived scalar measures such as the magnitude of one eigenvalue, fractional anisotropy (Eq. 30), maximum shear stress (Eq. 31), or the determinant (Eq. 32).

Blur by derived scalar quantities – If non relevant information should be suppressed in the final image, a post-processing step can be applied. We employ a blur filter, but also the opacity can be modified to hide regions of low information content. Again quantities like fractional anisotropy and maximum shear stress come into question (Fig. 42(b), and Fig. 43(b),(c)) to steer the post-processing. The pixels of the image are convolved with a Gaussian kernel, where the size of the kernel is proportional to the respective value of the chosen quantity. For example, if the blur should be determined with respect to the fractional anisotropy, isotropic regions are convolved with a larger kernel, whereas in anisotropic regions the kernel's size becomes 1 pixel. Like this, isotropic regions are blurred and regions with strongly expressed eigenvector directions are enhanced.

6.2 RESULTS

The results of our approach are presented and evaluated by means of the two stress tensor data sets: the one-point load (Sec. A.1.2) and the two-point load (Sec. A.1.3). We applied the developed visualization methods on planar cuts of the volumetric data sets.



(a) (b)

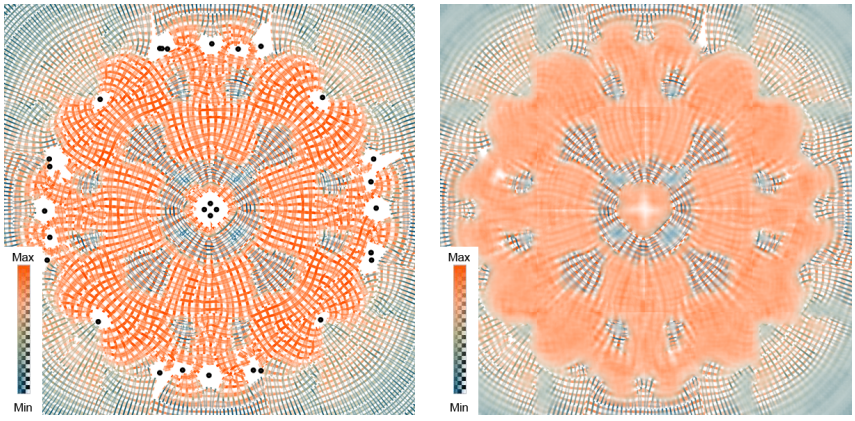
Figure 41: Data set: One-point load. (a) Glyph placement within the segmentation framework providing cells of similar characteristics. (b) Close-up of pre-computed barycentroids that place glyphs within cells at their semantic center.

6.2.1 *Glyph-based Visualization*

Fig. 41 shows the results of using the framework provided by the segmentation for the placement of glyphs (Sec. 6.1.1). This is applied on a slice of the one-point load data set. The glyphs are placed at the pre-computed barycentroids, oriented according to the eigenvectors and scaled by the mapped eigenvalues (Eq. 40). The color is assigned according to the fractional anisotropy (Eq. 30). Isotropic tensors are encoded in light blue and appear as spherical geometries. Anisotropic tensors are encoded in dark blue and result in well-marked ellipses. This work is not concerned with elaborate glyph design or similar. With the glyph placement we rather want to provide a basis for the variety of glyphs provided in the literature (Sec. 3.2.2.2). The close-up in Fig. 6.41(b) nicely shows how the barycentroids capture the semantic center of non-convex polygons.

6.2.2 *Texture-based Visualization*

Fig. 6.40(a) displays the directions of the major eigenvector field. Here the basic approach of approximately constant image space line density (Eq. 41) is applied. Even though the textures are mapped cell-wise the continuous character of the image is harmonious. Only at transitions of cells with hanging nodes (Fig. 39(b)) slight disruptions are noticeable. Zooming in the image automatically adapts the tex-

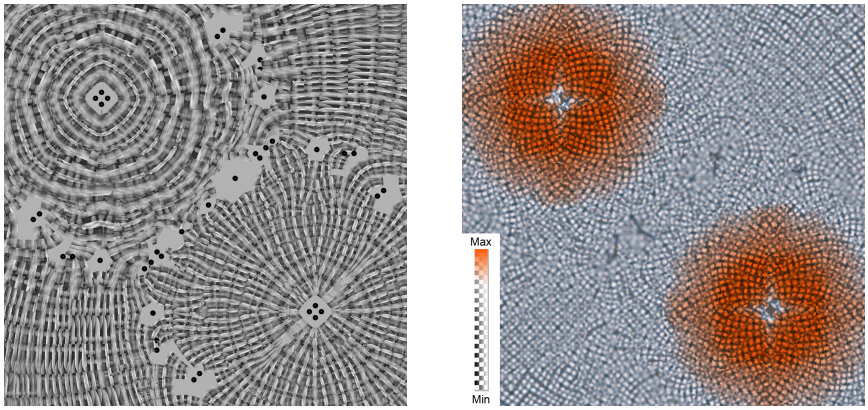


(a) (b)

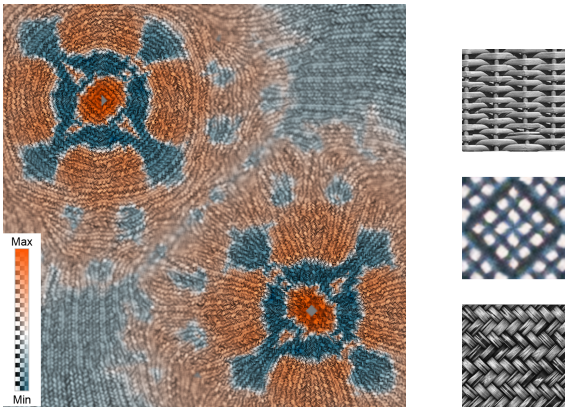
Figure 42: Data set: One-point load. (a) Superposition of both eigenvector fields, color coding according to mapped eigenvalues. (b) Post-processing blur to assist the user in finding regions of interest. Here these regions are defined by high fractional anisotropy .

ture such that the stripe frequency and the even appearance is maintained. Fig. 6.40(b) extends the representation by encoding the physical behavior. The pattern frequency is scaled by the mapped eigenvalues (Eq. 43). In the lower right corner negative eigenvalues are predominant and clearly result in a higher pattern frequency. This resembles to compressive forces and is in contrast to the upper left corner which is characterized by expansive forces. In Fig. 6.40(c) the textures for both eigenvector fields are blended. The pattern frequency is determined again by Eq. 43 and color coding is applied according to the fractional anisotropy. Isotropic regions are colored in blue and characterized by an isotropic pattern frequency. In regions of high anisotropy strongly differing eigenvalues lead to the unequal pattern frequencies for the two eigenvector fields, which is additionally emphasized by the red color.
A similar visualization approach is applied to the one-point load in Fig. 6.42(a). Here the color coding corresponds directly to the mapped eigenvalues (Eq. 40). Anisotropic regions are still discernible as the superposition generates mixed colors for strongly differing eigenvalues. Specific regions can be highlighted additionally by applying the post-processing step: in Fig. 6.42(b) regions with low fractional anisotropy are blurred. The focus of the user is directed to anisotropic regions, where eigenvalues exhibit a large difference.

The versatility and power of textures is demonstrated in Fig. 43. For texture samples with inherent variance discontinuities due to hanging nodes in the original segmentation are less prominent. In Fig. 6.43(a) a weave input pattern is employed to visualize the eigenvector directions. Due to the bidirectional nature of the weave pattern



(a) (b)



(c) (d)

Figure 43: Data set: Two-point load. The comparison of the rendering with different input textures shows the versatility of the framework. (a) Bidirectional weave input pattern, the frequency is adjusted to the mapped eigenvalues. (b) Rendering of the directions of maximal shear. Regions of high maximal shear stress are emphasized in red. Regions of low maximal shear are blurred. (c) Knitting pattern emphasizing major eigenvector directions from far, in detail directions of maximal shear are perceivable. Color coding according to fractional anisotropy. (d) Sample patterns.

both eigenvector fields are visualized at once. In Fig. 6.43(b) the user can switch to investigate directions of maximum shear of the underlying field. Here a texture is used with line structures illustrating the bisectors of the eigenvector directions. Additionally regions of high maximal shear stress (Eq. 31) are emphasized by selective color mapping. The third example, Fig. 6.43(c), generates a texture that resembles a knitted piece of fabric.

6.3 CONCLUSION

We have combined the accuracy of topological methods for two dimensional symmetric tensor fields with the support of more intuitive visualizations. Our approach uses the strength of textures for continuous visualizations and allows to gain insight into detailed information at discrete locations by placing glyphs. The topology-based segmentation framework (Chap. 5) is used to employ these techniques. The cells of this segmentation provide a consistent parametrization for the texture mapping and the bounding tensor lines correctly predetermine the the eigenvector directions within. This offers a multitude of textures that can be used to illustrate a continuous view on the various tensor properties. A selection of textures is presented that encode directional features; simple stripe textures but also textures with higher inherent variance, that support a smooth appearance over uneven transitions (hanging nodes). We believe the latter textures can also be used for other approaches that aim the mapping of directional textures region- or cell-wise without apparent distorted behavior at the boundaries.
Physical properties of the tensor field like compression or expansion are reflected in the texture frequency. Selective color mapping and post processing are applied to direct the users attention to scalar features of interest. Certainly, there remains a large potential to optimally assist the perceptional habits of a user and any development towards this has to be evaluated by user studies to finally prove its effectiveness. We also presented this work as flexible basis for further advancement.

7

COMPLETE TOPOLOGY EXTRACTION ON SURFACES EMBEDDED IN 3D

In the previous chapters we demonstrated how the topology captures all structural characteristics of a tensor field and how it can be extended to a segmentation also respecting the eigenvalue fields. This chapter is concerned with the extraction of the topology of two dimensional symmetric tensor fields on triangulated manifolds embedded in three dimensional space.

Due to the discontinuity of tensor fields on a piecewise planar domain, standard topology extraction methods might result in an incomplete topological skeleton. In particular with regard that topology is favored for accurate results this is not satisfactory. For instance, given an explorative framework for volumetric three dimensional tensor fields (e.g. Kratz et al. [KMH11]) it is highly desirable to provide an investigation of planar or manifold cuts of the entire field with distinctive properties and at reduced complexity. These cuts or also geometries in flow fields are often given as two dimensional triangulations embedded in three dimensional space. Commonly the intrinsic properties of such data sets are investigated on the projected data [Telo8]. On piecewise planar domains this yields piecewise continuous data sets.

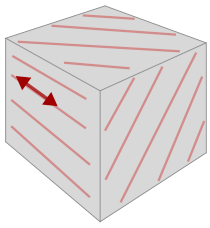


Figure 44: Piecewise constant tensor field (red tensor lines). All degenerate points are hidden in the corner vertices of the cuboid.

The extraction of the complete topology of a data set given on a piecewise planar domain can be a challenging problem. To analyze these data sets by extracting the entire topology, in terms of surface topology, special attention has to be given to the locations of discontinuity on those triangulations – namely edges and vertices. Many achievements have been made on the topic of surface topology for vector and tensor fields, however the fact that topological features are also located in discontinuous locations is mostly out of focus. If they are neglected the extracted surface topology obviously might be incomplete. A simple but evident example is that of a piecewise constant tensor

field defined on a cuboid. On the faces of the cuboid the tensor fields are constant. Approaches only investigating the faces would fail to detect a degeneracy on the cuboid, although the Poincare-Hopf theorem states that no closed surface with a non-constant field defined on it can be without a critical point (Fig. 44).

This chapter extends the work of Chap. 4 and provides a simple model to treat discontinuous fields defined on two dimensional triangulations embedded in three dimensional space. The majority of computations is carried out locally in two-dimensional space. Again, we focus on the topology of symmetric tensor fields, while many ideas are more general and can also be applied to vector fields. We consider two applications. First, continuous three dimensional tensor fields projected onto a surface, which are of interest when analyzing indefinite tensor fields on distinguished geometries. The projected tensor field expresses intrinsic properties to the given surface. Second, piecewise constant two dimensional tensor fields are considered defined per triangle. Such tensor fields naturally arise when considering derivatives of piecewise linear vector fields, e.g. a wall shear flow field of a body embedded in a flow.

The model of continuous transition bridges is introduced (Sec. 7.2) that allows to compute topological features in the locations of discontinuity (Sec. 7.3).

7.1 BASICS AND NOTATION

In the following two and three dimensional objects will be discerned by using normal letters for an object $x \in \mathbb{R}^2$, or $T \in \mathbb{R}^{2\times 2}$ and bold letters for $\mathbf{x} \in \mathbb{R}^3$, or $\mathbf{T} \in \mathbb{R}^{3\times 3}$.

7.1.1 *Tensor Field on Polyhedron Structure*

Input for our topology extraction is a tensor field given on an orientable two dimensional embedded manifold, which is triangulated. To store the manifold we use the CGAL polyhedron structure [cga]. It provides a structure for geometric meshes with vertices, half-edges and facets with their incidences. In our case the facet shape is restricted to triangles. The CGAL polyhedron structure holds vertices per triangle in counterclockwise order with respect to the oriented surface normal. This facilitates global orientation consistency, when oriented angles are calculated across locations of discontinuity. Each triangle has its own local two dimensional coordinate system, explained in Sec. 7.1.2. This paper treats two cases: in the first case three dimensional tensors are given at the triangle vertices and will be projected onto the surface (Sec. 7.1.2). This yields in different tensor values in the vertex for each adjacent triangle, causing discontinuities in the surface tensor field. In the second case piecewise constant

two dimensional tensors per triangle are given. The projected two dimensional tensors are stored locally for each triangle in the vertices and the constant two dimensional tensors for each triangle. If not stated differently vertex points of triangles are denoted by $\mathbf{p}_i$, with $i = 1, 2, 3$ cyclic indices. A half-edge e_i denotes the half-edge opposite to $\mathbf{p}_i$. For technical issues about the polyhedron structure we will refer to facets, for details on the semantic model, e.g. interpolation we refer to triangles.

7.1.2 *Local Coordinate System and Projection Matrix*

Every triangle of the polyhedron structure has its own Cartesian two dimensional coordinate system and two dimensional tensor(s). The local coordinate system is defined by the origin $o = (0, 0)$ and two basis vectors $x = (1,0)$, $y = (0,1)$. Its three dimensional counterpart is denoted by boldface letters with origin $\mathbf{o}$ and basis vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$ and is also stored for each triangle. $\mathbf{o}$ is a randomly chosen vertex of the triangle, $\mathbf{x}$ is parallel to a half-edge starting from the origin vertex and $\mathbf{y} = \mathbf{x} \times \mathbf{n}$ is right-handed orthogonal to $\mathbf{x}$ with respect to the surface normal $\mathbf{n}$ of the triangle (Fig. 45(b)). The projection matrix $\mathbf{P}$: $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ into the local basis systems is defined as $\mathbf{P} = (\mathbf{x}\ \mathbf{y})$. The three dimensional tensors $\mathbf{T}$ in vertices are projected onto the triangles with $T = \mathbf{P}^T \cdot \mathbf{T} \cdot \mathbf{P}$. Locally all necessary operations can be performed in two dimensional space and the basic concepts of Chap. 4 can be applied for the topology extraction.

7.2 CONTINUOUS MODEL

7.2.1 *Continuous Transition Bridges*

This section introduces the concept of continuous transition bridges. They are models for a curved surface connecting two triangles at their common edge imagined as a piece of cylinder. The normal on this surface smoothly varies from one triangle normal to the other triangle normal (Fig. 45(a)). This in turn provides a continuous transition of the tensors between both triangles. As Gaussian curvature on a cylinder is zero, the ensemble of two adjacent triangles with their connecting transition bridge can be flattened without distortion.

7.2.2 *Virtual Edge and Vertex Cells*

Fig. 45(b) shows how the flattened bridges serve as planar *virtual edge cells* on which a continuous interpolation is defined and methods for two dimensional space can be applied. Similarly the model of *virtual vertex cells* is defined as cells bounded by the adjacent virtual edge cells (Fig. 45(c)). The boundary of virtual edge cells and virtual ver-

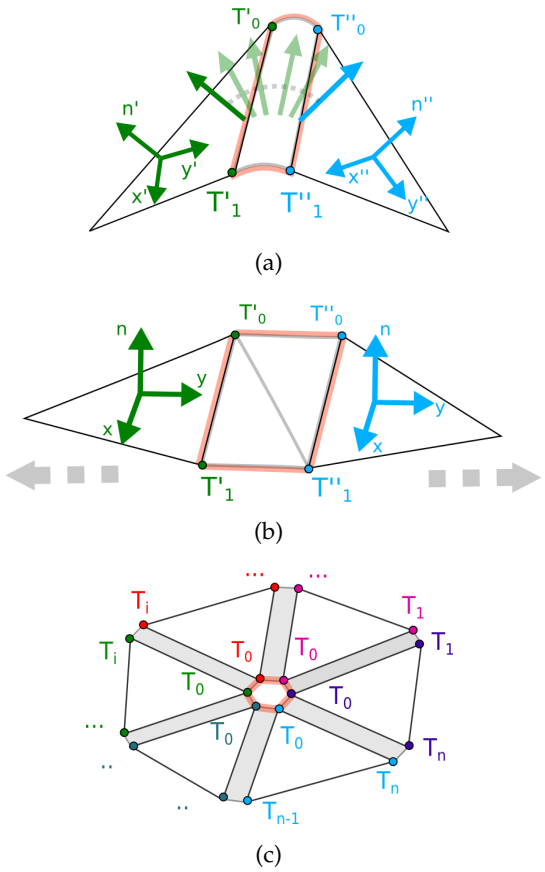


Figure 45: Model for transition bridges: (a) the triangle normals transform along the bridge into each other, (b) virtual edge cell defined by flattened transition bridge, choice of local coordinate system for common parametrization of tensors, (c) virtual vertex cell defined as cell bounded by adjacent virtual edge cells. (a,b,c) Jordan curves are colored in red. Different colors of $T_i, i = 1, .., n$ emphasize different tensors for same vertex point but different triangles.

tex cells are homeomorphic to a piecewise smooth, simple and closed curve defining a Jordan curve. This allows us to detect the existence of degeneracies by calculating the tensor index [Del94] in the locations of discontinuity (Sec. 7.3). The Jordan Curve of the virtual cells is depicted as red lines in Fig. 45(a)-(c). The precise location of degenerate elements has to be computed explicitly for the edges within the virtual edge cells. However, as the location of degenerate points in the virtual vertex cells can only be at the vertex itself there is no need for an explicit definition of the virtual vertex cells.

7.2.3 *Interpolation in Virtual Edge Cells*

To define an interpolation model for the topological analysis in the edges of the triangulation we need to provide a common parametriza-

tion for the two dimensional tensors T_i' and T_i'', $i = 0, 1$ given at the virtual edge cells. Let $\mathbf{e} \in \mathbb{R}^3$ be a normalized vector parallel to the edge constituting the virtual edge cell. Further let $\mathbf{n}', \mathbf{n}'' \in \mathbb{R}^3$ be the surface normals of the two triangles connected by the edge. To represented the given tensors T_i', T_i'', $i = 0, 1$ in matching frames of reference we define the following local coordinate systems in analogy to Sec. 7.1.2:

- $\mathbf{o}' = \mathbf{o}'' = \mathbf{p}_0$
- $\mathbf{x}' = \mathbf{x}'' = \mathbf{e}$
- $\mathbf{y}' = \mathbf{e} \times \mathbf{n}'$, $\mathbf{y}'' = \mathbf{e} \times \mathbf{n}''$

Fig. 45 illustrates how the frames of reference smoothly transform into each other across the transition bridge. By flattening the bridge the frames of reference become equal without distortion.
To summarize, to investigate a virtual edge cell the matching reference frames have to be computed and the tensors given at the vertices of the edge cell transformed to their reference frame. This is a costly operation, be it for computation time or storage. In our approach we will avoid this costly step, whenever possible.

7.2.4 *Eigenvector Rotation Across Transition Bridges*

The computation of the topology in Sec. 7.3 requires the calculation of signed rotation angles of eigenvectors for the transition bridges. To sidestep the costly common parametrization the signed rotation angle $\Delta\alpha$ is composed by two partial angles $\Delta\alpha', \Delta\alpha''$ computed in three dimensional space. Again let $\mathbf{e}$ be a normalized vector parallel to the connecting edge and $\mathbf{n}', \mathbf{n}''$ be the surface normals of the adjacent triangles. The eigenvectors in one of the edge's vertices are numerically represented as vectors and given by $\mathbf{v}', \mathbf{v}'' \in \mathbb{R}^3$. The partial angles $\Delta\alpha', \Delta\alpha''$ are defined as the signed angles of the eigenvectors $\mathbf{v}', \mathbf{v}''$ and $\mathbf{e}$ with respect to the surface normals $\mathbf{n}'$ and $\mathbf{n}''$ (Fig. 46(a)). The vectors $\mathbf{v}'$ and $\mathbf{v}''$ must always be oriented, such that $|\Delta\alpha| \leqslant \pi/2$. The entire rotation angle $\Delta\alpha$ across a transition bridge is given as

$$\begin{aligned} \angle(\mathbf{v}', \mathbf{v}'') &= \Delta\alpha = \Delta\alpha'' - \Delta\alpha' \\ \Delta\alpha' &= \operatorname{sign}(\mathbf{n}' \cdot (\mathbf{e} \times \mathbf{v}')) \arccos(\mathbf{v}' \cdot \mathbf{e}) \\ \Delta\alpha'' &= \operatorname{sign}(\mathbf{n}'' \cdot (\mathbf{e} \times \mathbf{v}'')) \arccos(\mathbf{v}'' \cdot \mathbf{e}) \end{aligned} \tag{44}$$

Please note that the vectors $\mathbf{v}', \mathbf{v}''$ are representations in the global three dimensional coordinate system of the eigenvectors $v', v'' \in \mathbb{R}^2$ of the tensors in the triangles.

7.3 TOPOLOGY EXTRACTION

To compute the complete topology over a non-smooth manifold we build on Chap. 4. In general, all analysis operations carry over to the interior of triangles of the embedded manifold. To achieve a complete topological graph this section provides extensions for the extraction of degeneracies (Sec. 7.3.1) and separatrices (Sec. 7.3.2) at locations of discontinuity – triangle edges and vertices, facilitating the ideas described in Sec. 7.2. In Sec. 7.3.3 the topology extraction of piecewise constant tensor fields is treated as special case of this approach.

7.3.1 *Degenerate Points – Structural and Virtual*

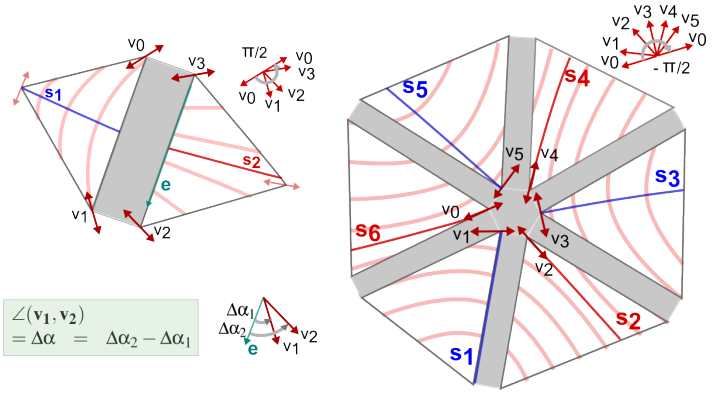
Degenerate points that are detected within triangles of the manifold are called *structural* degenerate points, because they are inserted as new vertices into the triangulation structure. Those that are found on the transition bridges are called *virtual* degenerate points. The determination of degenerate points involves their detection and precise location (Sec. 7.3.1.1), the tensor definition (Sec. 7.3.1.2) in these points and insertion into the polyhedron structure (Sec. 7.3.1.3). First the field is evaluated for *structural* degenerate points. Then edges and finally vertices are examined for *virtual* degenerate points. Only edges and vertices that do not yet contain any degenerate point are considered.

7.3.1.1 *Detection and Location*

Structural degenerate points:
The identification of isolated degenerate points inside triangles is exactly carried out as described in Sec. 4.3.1.

Virtual degenerate points:
In either case, for edges and vertices, the decision if a degenerate point exists is done by calculating the tensor index I_T according to [Del94]. Whilst the location of a degenerate point in a virtual vertex cell obviously is the vertex itself, for virtual edge cells the precise location of the degenerate points has to be calculated.

Virtual degenerate points – existence:
As mentioned in Sec. 2.7.1.2 the tensor index is the number of counterclockwise revolutions made by the eigenvectors traversing an enclosing Jordan curve L in counterclockwise direction.
Delmarcelle [Del94] states about it that it "*is a remarkable fact that the line integral [..] depends only on the nature of the enclosed degenerate point and not on the specific integration path. This very fact grants us considerable freedom in choosing the most appropriate path for computing the index at a degenerate point.*"



(a) (b)

Figure 46: Calculation of tensor index around (a) edge with wedge (b) vertex with trisector. Separatrices depicted in red and blue lines. (a) below: determination of rotation angle across transition bridge.

L has to be homeomorphic to a circle. In our case L is a piecewise linear, simple, closed curve, which is modeled by the transition bridges and triangle edges (see Sec. 7.2 and Fig. 45(a)-(c)) $L = \sum_i l_i$. To determine the index we integrate the change of eigenvector directions $\Delta\alpha$ along L in terms of signed rotation angles $\Delta\alpha_i$ of the eigenvectors $\mathbf{v_i}, \mathbf{v_{i+1}}$ at the end points $\mathbf{p_i}, \mathbf{p_{i+1}}$ of each line segment l_i. If the endpoints of l_i belong to the same triangle signed angles can be computed straightforward. In case l_i is across a transition bridge $\Delta\alpha_i$ is computed according to Equation 44. Computing the tensor index for vertices we have to add the angle defect Θ encountered by the traversal of L due to the surface curvature. It is defined by $2\pi - \sum_i \Theta_i$, where Θ_i are the apex angles of the surrounding triangles. The tensor index I_T is then defined as

$$I_T = ((\textstyle\sum_i \Delta\alpha_i + \Theta)/2\pi) - 1, \tag{45}$$

where $\Delta\alpha_i$ are the signed rotation angles per line segment l_i of L. For $I_T = \pm 1/2 \cdot n$, $n \in \mathbb{N}\backslash\{0\}$, a degenerate point exists, otherwise $I_T = 0$. Please see also Fig. 46

Virtual degenerate points – exact location on edge:
To calculate the exact location of a virtual degenerate point along an edge, we actually compute virtual edge cells; for calculation purposes we set its spatial extent to that of a unit square. The virtual edge cell is subdivided in two triangles. Within these triangles the computation of degenerate points is done as for structural degenerate points. As the virtual edge cells have no spatial extent the detected degenerate points are projected orthogonally onto the half-edges constituting the cell. It can easily be shown that the actual choice of the spatial extent for the stretched planar bridge has no effect on the location of the degenerate points.

7.3.1.2 *Tensor Definition in Degenerate Points*

Structural degenerate points:
Their definition is the same as given in Sec. 4.2.

Virtual degenerate points:
Tensors for virtual degenerate points are not defined as the tensor values themselves have no relevance for the topological structure.

7.3.1.3 *Insertion of Degenerate Points into Polyhedron Structure*

Structural degenerate points:
A degenerate point found inside a triangle is inserted as new vertex into the polyhedron structure to assure consistent eigenvector-based interpolation. The triangle is divided in three new triangles, which share the degenerate tensor in the new vertex. If a structural degenerate point is located on an edge, the new vertex has to be inserted also in the adjacent triangle to provide a consistent triangulation structure. However, the degenerate tensor is not assigned to the new vertex point in the adjacent triangle. Instead the tensor is linearly interpolated using the tensors given in this triangle. If a new structural degenerate point lies at a vertex, the polyhedron structure is unchanged, and only the vertex point of the triangle where the degenerate point was detected is assigned the degenerate tensor.

Virtual degenerate points:
Here, the polyhedron structure remains unchanged.

7.3.2 *Separatrix Computation and Classification*

The integral topological graph contains separatrices for both major and minor eigenvector fields. Thus, the following concepts are applied to both eigenvector fields, respectively. First start directions of the separatrices are determined by finding radial tensor lines emerging from degenerate points. Then the separatrices are computed as tensor lines, integrated along the eigenvector field until one of the termination conditions presented below applies. The determination of start directions differs for structural and virtual degenerate points. Radial directions from structural degenerate points are calculated as in Sec. 4.3.3. For virtual degenerate points two cases can occur. Either the computation of a separatrix can be directly started in an adjacent triangle or radial directions lie within an adjacent virtual edge cell.

Radial directions from virtual degenerate points:
Since virtual edges and vertices have no spatial extent the start direction of separatrices in their interior is irrelevant. Separatrices starting in virtual degenerate points enter adjacent triangles at the projected

degenerate point independently from the explicit start direction. As these entering points not degenerate separatrices can be computed by simply starting the integration in these locations (Fig. 46 (a),(b)).

Radial directions in virtual edges:
For both structural and virtual degenerate points radial directions can also emerge in the virtual edges. If the partial angles $\Delta\alpha_1, \Delta\alpha_2$ of eigenvectors $\mathbf{v_1}, \mathbf{v_2}$ on a common edge (Eq. 44) occur with different sign, the radial direction is in the virtual edge cell. See Separatrix S_1 in Fig. 46(b).

Integration of separatrices:
Within a triangle separtrices are computed by applying a Runge Kutta of 4th order scheme. Eigenvectors are bidirectional, within triangles the consistent interpolation is steered by the edge labels as described in Sec. 4.2 and performed in two dimensional space. If the integration of a separatrix passes a triangle boundary, the separatrix is cut at the intersection point $\mathbf{p}$ with one of the triangle edges. Starting from this point the integration has to be continued in the adjacent cell, which also can be a virtual cell. Three cases can be discerned: the separatrix leaves the triangle through an edge, it leaves through a vertex, or it converges into an edge.
Case 1: A separatrix intersects the interior of an edge in point $\mathbf{p}$ and eigenvector $\mathbf{v}'$. The directed angles $\Delta\alpha', \Delta\alpha''$ of the eigenvectors $\mathbf{v}', \mathbf{v}''$ with the common edge (Eq. 44) are computed, where $\mathbf{v}''$ is the eigenvector at $\mathbf{p}$ in the adjacent triangle. If they have the same sign the integration is continued in triangle T_f. The integration follows $\mathbf{v}''$ pointing to the interior of the triangle.
Case 2: The separatrix leaves through a vertex $\mathbf{p}$. Then all triangles sharing $\mathbf{p}$ are potential follow up triangles. If the vertex contains a degenerate point the separatrix is terminated. Otherwise the triangle containing the "straightest" continuation of the separatrix is chosen. Therefore the eigenvector in $\mathbf{p}$ of the last visited triangle is projected onto the adjacent triangles. For each triangle the angle of the projected eigenvector is computed with the local eigenvector in $\mathbf{p}$. The minimal angle defines the follow up triangle.
Case 3: There is a special case for both edges and vertices. This is if eigenvector behavior symmetrically converges towards an edge (Fig. 47). Looking at the transition bridges the interpolation therein suggests integration parallel to the concerned edge. Technically this case is detected by comparing the directed angles of the edge and the eigenvectors on the respective triangles in the point of intersection $\mathbf{p}$ (Eq. 44). For different signs of the directed angle we can directly integrate the separatrix along the edge. The vertex point of the half-edge performing the smaller angle to the eigenvectors is chosen as next point of the separatrix.

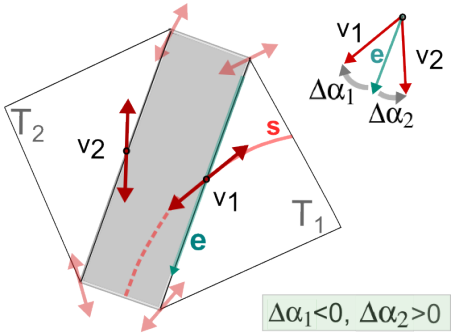


Figure 47: Separatrix S converges into edge e due to different signs of angles $\Delta\alpha_1$ and $\Delta\alpha_2$.

Termination conditions are equal to those given in Sec. 4.3.5. A separatrix is terminated if it fulfills one of the following conditions: 1. leaves the domain. 2. gets close to a degenerate point, line or triangle. 3. describes a circle or spiral and passes itself closely in parallel integration direction.

7.3.3 *Piecewise Constant Tensor Fields*

Piecewise constant tensor fields commonly occur if tensors are only given per triangle. Another example is a tensor field derived from a piecewise linear vector field with vectors defined in the vertices of the manifold. The topology extraction for piecewise constant tensor fields can be regarded as a special case of the approach proposed above and derived in a straightforward manner.

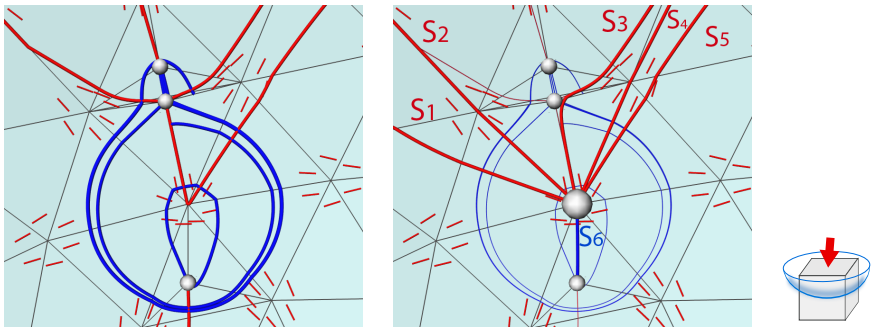
Degenerate points – The calculation of degenerate points is naturally reduced to virtual vertex cells and edge cells. This is done in analogy to Sec. 7.3.1, only that eigenvectors v are defined per triangle and rotation angles $\Delta\alpha_i$ only have to be calculated across edges. This implies that degenerate points in virtual edge cells are only found if eigenvectors $\mathbf{v}', \mathbf{v}''$ of adjacent triangles are orthogonal, $\mathbf{v}' \cdot \mathbf{v}'' = 0$. In this case, the entire edge is degenerate.

Separatrices – Separatrix computation is also done as in Sec. 7.3.2 for separatrices from virtual degenerate points: integration of the separatrices starts in the location of the virtual degenerate point. Radial start directions can be found for the surrounding triangles and in the virtual edge cells applying the same conditions as given in Sec. 7.3.2. The integration within a triangle is performed in one step, the expense of a Runge Kutta integration scheme can be omitted. Again tensor lines can converge into virtual edge cells if the eigenvector behavior of adjacent triangles is symmetric to the edge. The termination conditions for separatrices are also equal.

7.4 RESULTS

7.4.1 *Piecewise Continuous Tensor Fields*

This section demonstrates how the complete topology is extracted – degenerate points and separatrices are also found in the discontinuous locations. First the results are exposed for piecewise linear tensor fields, then for piecewise constant tensor fields. Degenerate points are depicted as spheres and separatrices of the major eigenvector field as red lines, separatrices of the minor eigenvector field as blue lines. For a better evaluation the major eigenvectors are indicated by short red lines in the triangles. However, they are not displayed, if the image would be overloaded and cluttered.



(a) (b) (c)

Figure 48: Analytic one-point load data set cut with sphere given as piecewise linear tensor field (schematic illustration in (c)). Fig. (a) renders the topological graph without additional consideration of locations of discontinuity. The red separatrix in the center seems to be of incorrect behavior. Fig. (b) shows the missing degenerate point (large sphere) and additional radial tensor lines (bold lines).

7.4.1.1 *Analytic One-Point Load Cut with Sphere*

The first example is the cut of a triangulated sphere with the analytic data set of the one point load (see Sec. A.1.1 and schematic illustration in Fig. 48(c)). The data set is sampled at the vertices of the sphere and projected onto the triangles.
This data serves as a simple but clarifying example why locations of discontinuity have to be examined for degeneracies. Fig. 48(a) shows the result of the topology extraction without and 48(b) with additional examination of the locations of discontinuity. The topological graph in Fig. 48(a) is incomplete and the red separatrix in the middle seems to perform an unmotivated turn. Fig. 48(b) exhibits the missing virtual degenerate point, found in the virtual vertex cell in the center. For a clearer arrangement in Fig. 48(b) the additional virtual degenerate point is depicted larger and only separatrices emerging

from the virtual degenerate point are drawn as bold lines. Separatrices from structural degenerate points are drawn as thin lines. Separatrices S_2, S_4 are found in the virtual edge cells. Comparing the separatrices and the major eigenvector field illustrated by the short red lines, it can be seen that starting the integration of tensor lines in the location of a virtual degenerate point delivers radial start directions (Sec. 7.3.2). Thus, the integrated separatrices partition the domain into qualitatively homogeneous regions. S_2, S_3, S_4 could be omitted, as they show radial behavior within the bounding separatrices S_1, S_5 and do not contribute to the structural partition the topological graph renders. They can be identified by a classification in analogy to Sec. 4.3.4. Please note, separatrices are stopped if they describe a spiral and pass themselves closely in parallel integration direction.

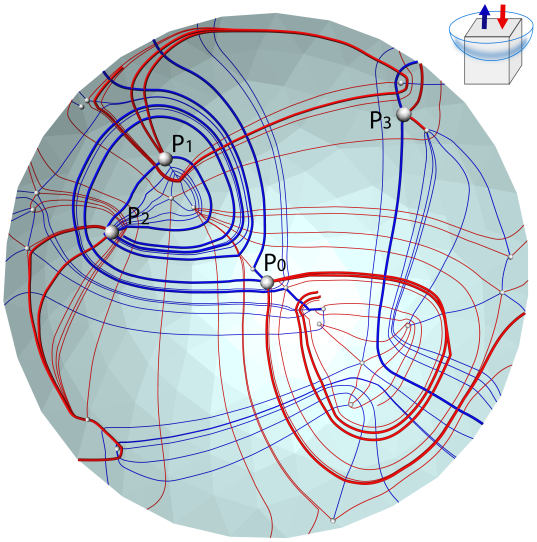
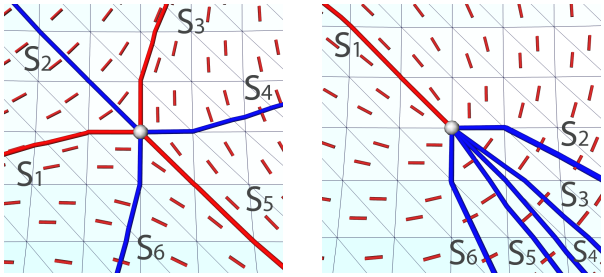


Figure 49: Simulated two-point load cut with a sphere, schematic illustration in top right corner. Virtual degenerate points are depicted as larger spheres and associated separatrices as bold lines.

7.4.1.2 *Simulated Two-Point Load Cut with Sphere*

The second example (Fig. 49) is the cut of a triangulated sphere and with the finite element simulation of the *two-point load* (Sec. A.1.3). Again for better clarity, topological elements from structural degenerate points are drawn smaller. Virtual degenerate points are shown as large spheres and associated separatrices as bold lines. As this data set is simulated it has a more complex topological structure. Point P_0 is a virtual degenerate point found in a virtual vertex cell, P_1, P_2 and P_3 are found in a virtual edge cell. All of them are wedge points. The blue separatrices below P_1 demonstrate how our approach correctly integrates tensor lines along the edges: due to symmetric eigenvec-



(a) (b)

Figure 50: Topology computation for piecewise constant tensor field – (a) trisector and(b) wedge point (depicted as spheres). Both found in virtual vertex cells.

tor behavior in the adjacent triangles the tensor lines converge in the edge (Sec. 7.3.2, Case3).

7.4.2 *Piecewise Constant Tensor Fields*

7.4.2.1 *Analytic Trisector and Wedge data sets*

Fig. 7.50(a) and 7.50(b) show two analytic data sets and present how the topology extraction also works for constant tensor fields. In Figure 7.50(a) a degenerate point is in the center virtual vertex cell, which is a trisector. Due to the piecewise constant tensor field the separatrices appear rather jagged, but correct. Separatrices S_4, S_6 are found by starting the integration in the adjacent triangles. The separatrices S_1, S_2, S_3, S_5 found in the virtual vertex cells are necessary to complete the topological graph. Fig. 7.50(b) shows a wedge point. Separatrices S_1, S_2, S_4, S_6 are found in the virtual edge cells, S_3, S_5 by integration in the triangles. Again the tensor lines S_3, S_4, S_5 could be omitted as they are radial tensor lines.

7.4.2.2 *Rate of Strain Tensor Field on Aneurysm data set*

In the area of flow analysis and visualization an increasing interest in the rate of strain tensor (Sec. 2.8.4) can be observed. For piecewise linear flow fields this results in piecewise constant tensor fields, which are based on the derivative of the flow field. This motivated the next example, which investigates the blood flow inside a cerebral aneurysm (Sec. A.2.1). This example exposes the topology extraction on the rate of strain tensor field. The tensor field is derived from the original vector field and piecewise constant.

In Fig. 51, top right corner, the original vector field is illustrated by streamlines [RPH+09] on the surface. For better orientation the streamlines are repeated in the central image, which shows the complete surface topology of the rate of strain tensor field. In flow visu-

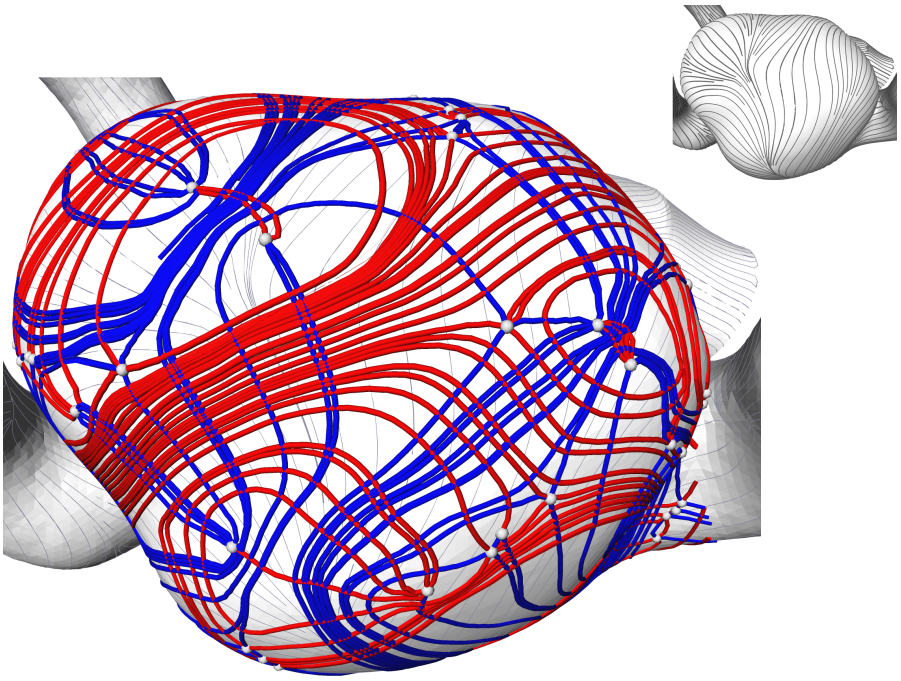


Figure 51: Derived rate of strain tensor field on an aneurysm data set. Top right corner: the streamlines on the surface render the initial velocity gradient vector field.

alization the divergence of a flow is one of the interesting quantities. Evaluating the rate of strain tensor by means of scalar quantities, like the major eigenvalue, determinant (Eq. 32) or trace (Eq. 33) of the tensor gives insight on the strength of the divergence in a field. The topological graph of such a field additionally renders the directionality of the divergence. The major and minor tensor lines indicate directions of strongest divergence respective convergence. This allows to distinguish regions where major flow separation occurs along or across streamlines.

7.5 CONCLUSION

We presented an approach to extract the complete surface topology on piecewise continuous and piecewise constant two dimensional symmetric tensor fields on two dimensional triangulations embedded in three dimensional space. The core idea is to include the locations of discontinuity into the topological analysis. For this purpose the model of continuous transition bridges is introduced, which allows to capture the entire topology on the discontinuous field. The continuous transition bridges close the discontinuous gaps of the tensor field in the edges and vertices of the triangulation. The strength of this work lies in its simplicity compared to previous work. Degenerate points and separatrices hidden in the edges and vertices are

extracted and a complete surface topology is achieved. The existence of hidden degenerate points is detected in an efficient and correct way by computing the tensor index. While the position of degenerate points in vertices is clear, the precise location for degenerate points in edges can be computed. This is done by the model of the flattened transition bridges. The values of the tensor index computation are integral multiples of 1/2 independently from the curvature of the manifold. Thus, no thresholding has to be applied for the detection of degenerate points. Please note, we propose a numerical method with inherent precision problems, such as rounding errors. For example the separatrix integration is performed by a Runge-Kutta of 4th order scheme; the result will always be affected by the chosen integration step width for example.

This work also proposes results for piecewise constant rate of strain tensor fields, which are of high interest in flow visualization. The directionality of flow divergence and convergence can strongly support the analysis of complex flows. The surface topology of the rate of strain tensor field yields a skeleton were all such directional characteristics are illustrated.

The model of continuous transition bridges can be extended to the extraction of the surface topology of vector fields as well. We hope that this paper enriches work in that research area, as well.

8

AUTOMATIC, TENSOR-GUIDED ILLUSTRATIVE VECTOR FIELD VISUALIZATION

Visualization tries to embody much information of a data set within a single image. This often results in complex depictions that cannot be understood by domain experts without further knowledge of the visualization methods. As a remedy, illustrative visualization tries to mimic hand-drawn sketches [Dal83, SCK10] of schematic illustrations as used by the domain experts themselves.
This chapter presents an automatic visualization tool similar to these hand drawn sketches. In particular, it is concerned with the illustrative representation of properties in asymmetric tensor fields. In this work we present the benefit of such an illustrative visualization approach for the simplified depiction of properties in vector fields. Hereby, the asymmetric tensors occur as the spatial derivatives of the vector field. These derivatives contain valuable information that is independent of the chosen frame of reference and that are hard to deduct from vector only visualizations.
Our major motivation is a sketch drawn by domain experts on the basis of a complex visualization [CPL+11] derived from an earthquake simulation [Vin98]. Abstracting the complex original visualization results in a simple, easy-to-read image containing the most important information (Fig. 52). The goal of this work is to use computers to automatically generate similar abstracted, highly simplified representations of vector data and its prevalent features, pursuing the idea that showing less can sometimes be more. This can serve as a bridge towards other elaborate and more complex visualization methods and also eases the comparison of data sets, e.g. for different time steps.

Displaying less certainly requires a careful choice of what to show. This also implies that the abstraction process should only depend on a few intuitive parameters. Therefore, the design of our visualization is guided by the following ideas: generate a visualization that is simple in its concepts, intuitive in its single elements, conveys trends and strongly expressed features in the data.

Visualization concept – The visualization consists of two major components: (a) a background visualization that displays large scale trends, and (b) a feature visualization that highlights locations of strongly expressed field characteristics. Both components are built from multiple layers considering the vector field itself and the gradient tensor field, an asymmetric tensor field. The derivative supplies valuable additional information about the vector field. The derivative, an asymmetric tensor, is decomposed according to Zhang et al. [ZYLL09] into

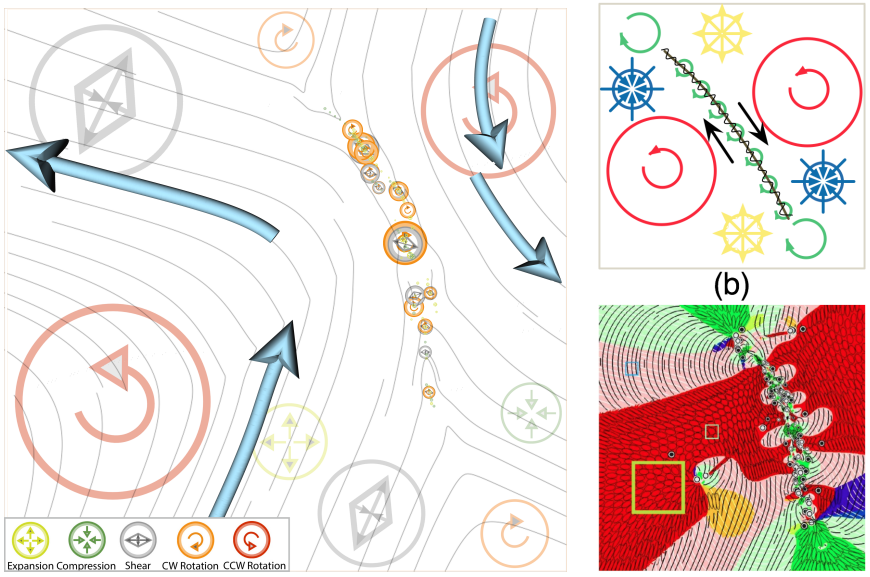


Figure 52: Visualization of a vector field from a simulation of co-seismic displacements [Vin98] (Sec. 8.3.2). The image (a) shows the result of our automatically generated visualization including a rough representation of the context as background and the strongly expressed features on top. Those occur mostly along the fault moving from top to down. The image (c) shows the detailed hybrid visualization technique developed by Chen et al. [CPL+11] consisting of hyper-streamlines and elliptical glyphs. Image (b) is the sketch drawn by domain experts on basis of (c) and motivated our work.

three components expressing the strengths of isotropic scaling, the rotation and the shear. In contrast to the work by Chen et al. [CPL+11], we do not display the entire tensor information but restrict ourselves to the magnitude of the components, which are scalar fields describing the respective feature strength.

The idea of the background visualization is to cluster regions of similar behavior. Thereby, only large scale clusters are considered for the visualization. The clustering method depends on the respective field. For vector fields, we introduce an efficient clustering approach, which is almost parameter free. For the clustering, we adopted the similarity measure proposed by Telea et al.[TvW99] adding a component that considers the strength of the derived tensor field. As result, we obtain large clusters in mostly homogeneous regions while clusters are small in regions with high variations. For the scalar fields, we propose two approaches: the first delivers a clear partition of the domain into the dominant components of the scalar fields. However, for its clarity it might loose important details. The second approach conveys more details and builds on scalar field topology exploiting

basins of the maxima. To complement this trend visualization, we add a detailed feature visualization in the foreground. Local features are defined as maxima of the respective scalar field and offer the possibility to steer the level of detail for this representation via homological persistence. Features as well as background trends are represented by icons. The combination of these elements involves few plausible parameters which can be used to move the focus from global to detail or to specific features of interest.

In summary, the algorithm consists of the following components:

- Tensor decomposition
- Vector field clustering
- Scalar field topology extraction and persistence computation
- Cluster center computation for background glyph placement
- Rendering

The method is relevant for a multitude of strongly differing application areas. We demonstrate its effectiveness using three data sets: A two dimensional flow behind a cylinder, a wind vector field from a multi-parameter weather simulation and the displacement fields of an earthquake simulation.

8.1 BASICS AND NOTATION

In the following let $\mathbf{v}(\mathbf{x}) : D \to \mathbb{R}^2$ be a vector field given on the domain $D \in \mathbb{R}^2$. $\mathbf{T}(\mathbf{x})$ denotes the asymmetric tensor field representing the gradients of $\mathbf{v}(\mathbf{x})$, see also Eq. 2.

8.1.1 *The Gradient Tensor Field*

In this approach the gradient plays a fundamental role for the analysis of the vector field. In the area of fluid dynamics it is often referred to as a Galilean invariant entity which expresses shear, rotation, and isotropic scaling of a fluid. In terms of a linear Taylor approximation of the field it appears as coefficient of the second term:

$$\mathbf{v}_1(\mathbf{x}) = \mathbf{v}(\mathbf{a}) + \nabla\mathbf{v}(\mathbf{a})(\mathbf{x} - \mathbf{a}) \tag{46}$$

Transferring this to the task of vector field analysis and visualization leads to the two guiding ideas of our approach:

1. In locations with derivative $\nabla\mathbf{v}(\mathbf{a})$ close to zero the field is already well represented by the given vectors only, Sec. 8.2.3.2.
2. Adding information given by the derivative helps to further/deeper understanding of the field, Sec. 8.1.2, 8.2.2, 8.2.3.3.

8.1.2 *Decomposition of Derivative*

To analyze and visualize the spatial derivative we pursue the decomposition of asymmetric tensors proposed by Zhang et al. [ZYLL09] into the uniquely defined tensor components: $\mathbf{T} = \mathbf{D} + \mathbf{S} + \mathbf{R}$ which we introduced in detail in Sec. 2.4.4.2, Eq. 12. Without explicitly specifying the application area of the vector field $\mathbf{v}(\mathbf{x})$ the components can be classified as follows: $\mathbf{D}$ is the isotropic scaling, $\mathbf{S}$ the shear, and $\mathbf{R}$ the rotational part of $\mathbf{T}$. The derived scalars γ_d, γ_s, and γ_r (Eq. 14) are measures for the strengths of $\mathbf{D}$, $\mathbf{S}$, and $\mathbf{R}$ and allow to classify the vector field. In the following $\gamma_d(\mathbf{x}), \gamma_r(\mathbf{x})$, and $\gamma_s(\mathbf{x})$ denote the respective scalar fields.

Whilst the strength of the shear is always positive, $\gamma_s > 0$, the sign of γ_r and γ_d distinguishes their quality. With respect to the sign of γ_r and γ_d we can specify five *deformation modes*:

- Positive isotropic scaling for $\gamma_d > 0$
- Negative isotropic scaling for $\gamma_d < 0$
- Pure shear with $\gamma_s > 0$
- Counterclockwise rotation for $\gamma_r > 0$
- Clockwise rotation for $\gamma_r < 0$

The scalars γ_d, γ_r, and γ_s and the absolute tensor magnitude $m = \gamma_d^2 + \gamma_s^2 + \gamma_r^2$ will be used to guide and extend the vector field visualization as proposed above (Sec. 8.1.1).

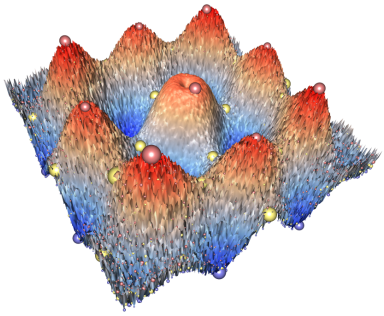
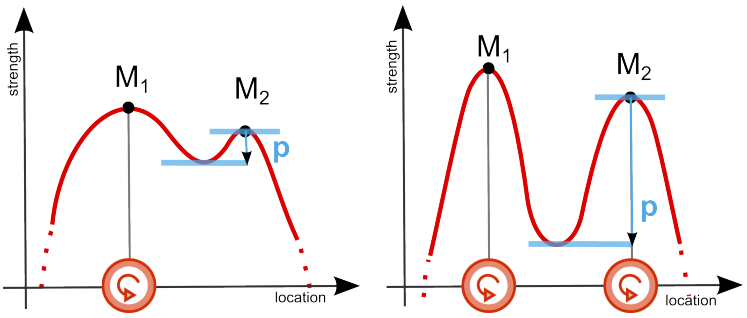


Figure 53: Critical points of a scalar function. The size of the spheres is determined by homological persistence. There are 38000 critical points contained in the dataset, but homological persistence classifies most of them as unimportant.



(a) low persistence (b) high persistence

Figure 54: Simplified schematic illustration of the persistence concept. In both images M_1, M_2 are the maxima of a 1-D scalar function. In (a) M_2 has a relatively low, whereas in (b) M_2 has a high persistence value. Filtering by homological persistence would induce that for the function in (a) only at M_1 an icon would be drawn and the final depiction naturally cleaned up. In contrast, in (b) at both extrema an icon would be placed due to their high persistence. However, we are convinced that although this might induce clutter both extrema are of such high importance that both ought to be displayed.

8.1.3 *Scalar Field Topology and Homological Persistence*

For the background as well as the feature visualization we exploit concepts of scalar field topology. The features are defined as local extrema of the derived scalar fields $\gamma_d(\mathbf{x})$, $\gamma_s(\mathbf{x})$, and $\gamma_r(\mathbf{x})$ and the scalar field clustering uses basins of the extrema as fundamental structure (see also Fig. 62 (b). A basin of such a point is that part of the domain for which all points are connected with the respective extremum when integrating gradient lines. For instance, all points of a valley belong to the basin of its minimum. The basins of all minima present a segmentation that covers the whole domain. The same is true for the basins of all maxima.

To extract the scalar field topology we employ the combinatorial framework by Reininghaus et al. [RGH+10] which is robust and avoids the computation of further derivatives. Within this framework, we use homological persistence as introduced by Edelsbrunner [ELZ02]. Homological persistence measures the stability of an extremal point against perturbation of the data values. It can be used to remove noise induced extremal points, but it also serves as a natural importance measure for critical points. Introducing homological persistence in its full extent is beyond the scope of this thesis. Instead we will summarize the basic ideas about this concept. For further reading we refer the interested reader to [ELZ02]. An example is shown in Fig. 53, where the importance of the extremal points is determined by homological persistence. In this image, there are outstanding extremal

points – however, with very different persistence values. Selecting these exponents by mere thresholding could not provide such a well defined hierarchy like persistence filtering does (see also Fig. 54). Homological persistence therefore provides a natural and sensible hierarchy of the extremal points. For our application, we use homological persistence to control the level of detail for the represented feature points. We split the scalar fields $\gamma_s(\mathbf{x})$, $\gamma_r(\mathbf{x})$ which can also have negative values into two positive fields, respectively. Let $\gamma_i(\mathbf{x})$, $i \in \{d, r\}$ be the original scalar field. Then the two fields $\gamma_{i+}(\mathbf{x})$, $\gamma_{i-}(\mathbf{x})$ are defined as follows:

$$\gamma_{i+}(\mathbf{x}) = \begin{cases} \gamma_i(\mathbf{x}), & \gamma_i(\mathbf{x}) > 0 \\ 0, & \text{else} \end{cases}$$

$$\gamma_{i-}(\mathbf{x}) = \begin{cases} 0, & \gamma_i(\mathbf{x}) >= 0 \\ -\gamma_i(\mathbf{x}), & \text{else} \end{cases}$$

Like this only maxima are extracted by the topology and persistence values unambiguously reflect their importance (see also Fig. 54).

8.2 VISUALIZATION CONCEPTS

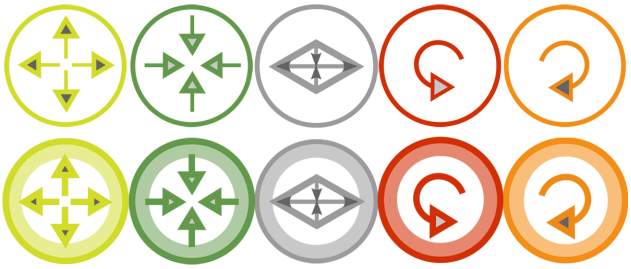


Figure 55: Top: Background icons for different deformation modes: positive, and negative isotropic scale, shear, counterclockwise rotation, and clockwise rotation. Bottom: Respective extrema icons.

8.2.1 *Icons*

The essence of the visualization approach is to provide a composited image, where the most prominent features are presented in the foreground and in the background a context visualization is given.

The most prominent features are identified as the persistence filtered extrema of the scalar fields $\gamma_i(\mathbf{x})$, $i \in \{r+, r-, s, d+, d-\}$ (see Sec. 8.2.2). The background visualization comprises vector representation (see Sec. 8.2.3.2) and a rough simplified visualization of the quantities in

the scalar fields $\gamma_i(\mathbf{x})$. To serve the different needs for understanding vector fields, we provide multiple strategies (see Sec. 8.2.3). Essentially they differ in their focus on either clarity of presentation or completeness of information.

Both extrema and background information deduced by the scalar fields are depicted by icons. The following goals were guiding their design (Fig. 55): the icons should be simple to obtain a clear, harmonious image. Also the colors of the icons should be well discriminable, whilst avoiding too many colors in the image which might distract the user from the actual content. All icons are bounded by a circle, the base size of all icons is equal. The icons themselves only encode qualitative properties, the type of deformation mode. The isotropic scaling and rotation icons are rather self-explanatory. The shear icon qualitatively illustrates the deformation of a unit square under shear. The major arrow inside is oriented according to direction of the major eigenvector at the current location. The icons used for extrema are bounded by a bold circle to differentiate them from the background icons. Please note that the icons were designed that they could also work for color blind people.

8.2.2 *Extrema and Foreground Visualization*

Features defined by the scalar fields $\gamma_i(\mathbf{x}), i \in \{r+, r-, s, d+, d-\}$ are extracted as maxima of the scalar field topology (Sec. 8.1.3). Filtering them by homological persistence allows to extract the most prominent features on a sound mathematical basis without corrupting the data (see also Fig. 56). Maxima due to topological noise are removed in a pre-processing step. The persistence value of the remaining extrema can be used to interactively determine the granularity of the representation. Note that the feature strength itself cannot be deduced from the persistence value.

To display the extracted features extremal-icons (Fig. 55) are placed at the location of the extrema. They are scaled according to their magnitude to encode the feature strength. Icons that fall below a predefined size (in our implementation one twentieth of the domain size) are depicted as colored dots.

8.2.3 *Background Visualization*

For the background visualization of the vector data and the scalar data a clustering is performed. The resulting clusters are given as a region bounded by a polyline and are visualized using representative icons. The respective clustering algorithms, the definition and placement of the representatives are described in the following.

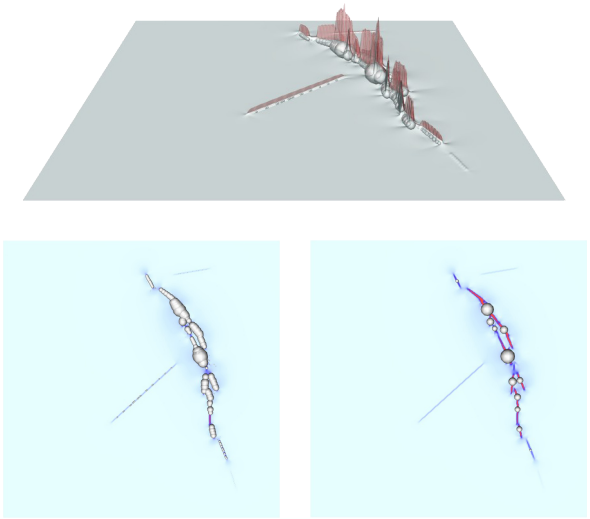


Figure 56: Earthquake data set (Sec. 8.3.2) (a) Height field of shear scalar field. (b) All extrema of scalar field displayed as spheres scaled by their persistence value. (c) Persistence filtered extrema.

8.2.3.1 *Segment Voronoi Diagram*

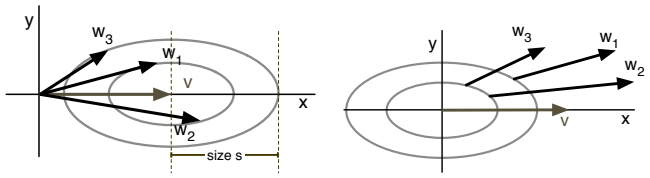
For the placement of the icons (arrows and background icons for the scalar fields) the segment Voronoi diagram by CGAL [cga, Kar04] is employed which extracts the medial axis of the clusters. It takes line segments as input and supports non convex polygons and polygons with holes. Further it provides the minimum distance to the bounding polylines for each point on the medial axis. The use of the medial axis for the placement of the representatives is described in the respective sections.

8.2.3.2 *Vector Field Clustering and Visualization*

For the purpose of generating a vector field clustering we decided to follow an approach similar to Telea et al. [TvW99]. But in contrast to [TvW99] we do not perform a hierarchical clustering of the complete field. Instead can we employ an efficient region growing method. The method detailed below.

Choice of cluster region The aim of the vector clusters is to convey trends in the background. Therefore we restrict the clustering to regions where the tensor magnitude falls below a predefined threshold τ and the field is sufficiently represented by its vector values (Sec. 8.1.1). To respect the data set specific variations we set τ to a multiple of the maximal value m_{max} that the tensor magnitude reaches. In our implementation it is set to $\tau = 0.01 \cdot m_{max}$.

Similarity measure We adopt the anisotropic similarity measure for vectors of Telea et al. [TvW99] which offers the option to trade-off



(a) directionError (b) positionError

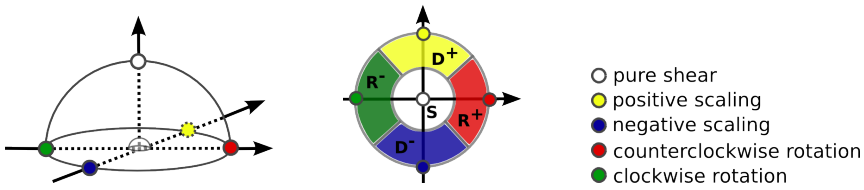
Figure 57: (a) Isocontour errors as defined by [TvW99].

between exact parallelism of vectors in the final clusters or clustering that follows streamlines.
We will shortly summarize the essential properties of the similarity measure. For more detail please refer to the original paper [TvW99]. In this paper an error space is introduced which combines feature and spatial space. Elliptically shaped iso-contours define similarity functions for vector directions and vector positions, see Fig. 8.57(a), 8.57(b). The eccentricity of a reference ellipse and its total size define the "strictness" of the similarity measure. The direction similarity compares the directions of two vectors which for extreme values of the reference ellipse eccentricity would return clusters of parallel vectors. The position similarity reflects how much the *positions* of the vectors is aligned with the vector field. For extreme ellipse eccentricity this would return clusters that are integrated streamlines. The final similarity function e allows to trade-off between directional or position clustering the two error values e_d and e_p:

$$e = (1 - \alpha) \cdot e_d + \alpha \cdot e_p, \ \alpha \in [0, 1] \tag{47}$$

All results in Sec. 8.3 where generated with an aspect ratio of the ellipses of 1:2 and $\alpha = 0.9$. A further variation compared to the method by Telea is that the size s of the reference ellipse for the position error is set in relation to the mean magnitude of the tensor m_{mean} (Eq. 48) inside the cluster, see Fig. 61 (c). This generically allows that large homogeneous regions are clustered even though the actual position distance becomes large.

Region growing As vectors are close to constant in the considered regions the clustering can be performed by a simple cell-based region growing algorithm. The clusters are stored as a list of cells and a bounding polyline. For the region growing a maximum error for the similarity measure (Eq. 47). is set to 0.01. On demand this value can be changed by the user but was not necessary for all results in Sec. 8.3. The region growing starts in the cell with minimum tensor magnitude m_0, which is the location with the most homogeneous neighborhood. For the evaluation of the similarity function each cluster holds its mean vector, its center and the mean tensor magnitude m_{mean}. The



(a) (b) (c)

Figure 58: (a) Eigenvalue manifold of 2×2 tensors. Extremal configurations are indicated by colored dots explained in (c). The small hemisphere in the center excludes the undefined case of a zero unit tensor. (b) Eigenvalue manifold (top-down view along axis of pure shear). Colored regions depict the Voronoi diagram for the classification into dominant components.

size s of the reference ellipse in the similarity function is set in relation to the mean magnitude of the cluster:

$$s = \max(1, \log(1/m_{mean} + 1)) \tag{48}$$

The clustering continues until adding a new cell would exceed the maximum error. In this case the clustering process recursively starts for a new cluster again in the (unclustered) cell with the minimum value of m. Cells with vectors close to zero are left out to avoid the clustering of singularities.

Visualization The extracted clusters are represented by a curved arrow icon. The arrow's tail is constituted by a streamline integrated within the cluster and the head represents the direction of the vector field in that location. The start point for the streamline integration is located in the center of the medial axis (Sec. 8.2.3.1) of the clusters. It is defined as the center point on the longest path of the medial axis. From the start point an exact streamline integration [NJ99] is performed in both directions (on $\mathbf{v}$ and $-\mathbf{v}$). For the display the arrows are shortened by 80%.

8.2.3.3 *Scalar Quantities*

The background visualization of the scalar quantities is also based on clustering of regions of similar behavior. Thereby we offer two different approaches.

- The first is a fully automatic classification of the field characteristics with the advantage of a clear distribution of the representative icons in the image.
- The second approach supports the depiction of the full information by considering the absolute strength of the scalar values.

Classification – visualizing the dominant component
To provide a first overview on the overall behavior in the field the classification into dominant components is performed according to see Chen et al. [CPL+11]. Here the relative ratios among the strengths γ_s, γ_d, and γ_r are compared. By normalizing the strengths $\gamma_s^2 + \gamma_d^2 + \gamma_r^2 = 1$ all possible configurations can be modeled as a unit hemisphere (see Fig. 58). A Voronoi diagram on this hemisphere allows to classify the possible configurations to

- Isotropic scaling dominant $\mathbf{D}^+$ (positive), $\mathbf{D}^-$ (negative),
- Shear dominant $\mathbf{S}$,
- Rotation dominant $\mathbf{R}^+$ (positive), $\mathbf{R}^-$ (negative).

In our approach we exclude tensor magnitude values m close to zero from the derivative visualization, to avoid the undefined case, where the unit tensor is zero. Cells are labeled according to the dominant strength type and depicted according to the background visualization described below. The strongest advantage of this method is that the final image has a clean arrangement. However, the drawback is that this classification does not consider the absolute feature strength. Further it is not possible to display regions where two characteristics play an important role.

Basin visualization
To offer a visualization that overcomes the limitations of an absolute classification this approach treats each scalar field separately. It uses the concept of topological basins as described in Sec. 8.1.3 which can be directly used as clusters for the icon placement. All icons are assigned the scalar value of their position which determines their transparency value in the final visualization. By moving a slider for a magnitude threshold the user can interactively blend in and out the icons of the regarded scalar field. The shear icons are additionally oriented in the direction of the major eigenvector. Please note that with the basin visualization background icons of different scalar fields may overlap, which is not the case for the classification approach.

Icon placement
All approaches for the background visualization are technically treated equal. In a preprocessing step the boundaries of the extracted clusters are strongly simplified according to an angle criterion. For the simplified boundaries the medial axis is computed (Sec. 8.2.3.1). and the points on the medial axis are processed recursively. For the background icons of the scalar fields the placement starts in the point with the maximal distance to the boundary. Recursively we place the

icons along the medial axis, by interpolating their optimal size by the distance values given at each point of the medial axis, see Figure 59.

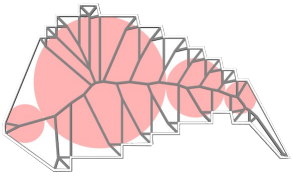


Figure 59: Placement of icons along medial axis of a cluster. The light gray polyline indicates the boundary of the cluster, the dark gray lines depict information by the segment Voronoi diagram (medial axis and distances to the boundary).

8.3 RESULTS

To verify the automatic generation of the sketch based visualization we use three data sets with very different characteristics. All three data sets are inherently two-dimensional. For the two time-dependent data sets the analysis is performed on single time slices. For all data sets the same parameters are used.
Performance – our code basis has not been optimized towards performance yet. The most costly step is the persistence computation that is done for each scalar field [RGH+10]. For the data sets used in the results we denote the resolution and the processing time required for the persistence computation for a single scalar field on a QuadCore i7 processor with 2,6 GHz. The computation time for the scalar fields is similar.

8.3.1 *Cylinder Data Set*

Data set – Incompressible flow behind a cylinder with no isotropic scaling in the field (Sec. A.3.1). We use two versions of this data set: the original data and a version with changed reference frame with subtracted the average velocity.
(Resolution: 242×242, Timing for scalar field topology: 0.47 sec)

Analysis – Figure 60 and 61(a), show all locations of strongest shear and rotation forces defined by the scalar field extrema. This is combined with the result of the classification into dominant components in the background. The image is further composited with the results of the vector clustering and a LIC visualization as context information. Traditional continuous vector visualization (e.g. LIC, or streamlines) does not encode direction nor velocity of the vectors. Thus important qualities like shear or vorticity might not be perceivable. They are made visible in a suggestive manner within our visualization framework. Further, a change in frame of reference might give a very dif-

ferent impression of the same vector field. This is demonstrated by Fig. 61(a) which could imitate the change of an observer's position. The LIC visualization in the background of Fig. 61(a) clearly differs from the one in Fig. 60. In contrast, the schematic illustrations deduced by the tensor decomposition remain independent of the frame of reference. In Fig. 61(a) the LIC might suggest strong vortical behavior throughout the flow from left to right. However, the size of the extremal icons (or dots) clarifies that the magnitude of rotation at the right end is rather low. We consider it as an extension for the vector clustering methods described in Chap. 3 that the level of abstraction for the extrema has a clear mathematical reference and interpretation. This also supports the reproducibility of simplified and abstracted view on different data sets.
The close-up in Fig. 61(b) illustrates the results of the vector clustering, clustered regions are represented by arrows. This image demonstrates how the clustering approach supports the generation of large clusters in homogeneous regions while still providing the necessary granularity in regions with higher variation. This sensitivity is a consequence of the adaptation of the position error of the similar-

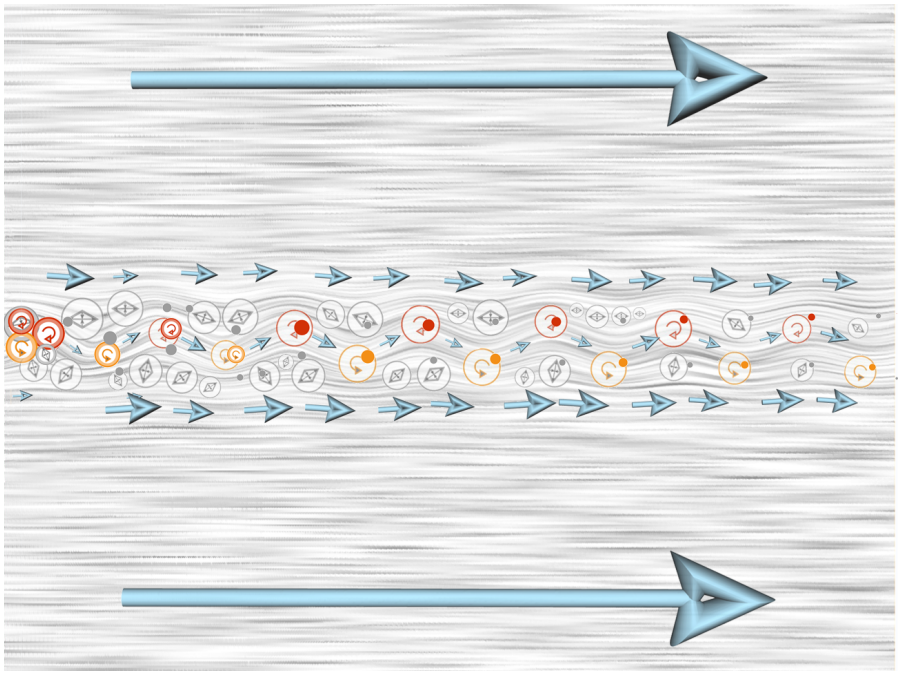


Figure 60: *Cylinder data set* – Information about shear and vorticity made visible in the sketch-like representation (in combination with the vector field visualization method LIC [CL93] as context information). The extrema of the scalar fields show the locations of highest strengths. If the size of extrema is below the fixed threshold they are rendered as colored dots. The background visualization depicts the dominant components according to the classification: regions of dominant rotation are distinguished against dominant shear. Regions of low tensor magnitude are illustrated by arrows.

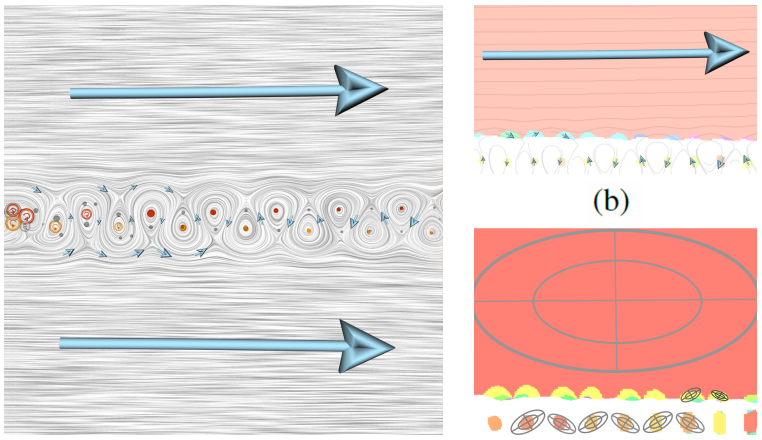


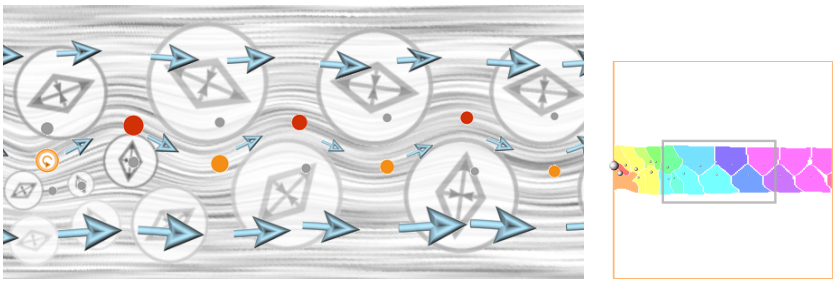
Figure 61: *Cylinder data set* – (a) Same data set as in Fig. 60 but with change of reference frame. Result shows the independence of the information given by the scalar fields from the chosen frame of reference. (b) Schematic depiction of the vector field clustering process with respect to magnitude of derivative (resulting clusters randomly color coded). White denotes regions with magnitude $m > \tau$, which are excluded from the clustering. (c) Close-up: schematic illustration of the size adaption of the reference ellipse for the position error.

ity function to the underlying tensor magnitudes (Sec. 8.2.3.2) (see Fig. 61(c). This would not be possible with a uniform measure as proposed in [TvW99].
In context with flow analysis the mixing of flow is an interesting quality. For this purpose flow data sets are also inspected towards their shear properties. After giving a first overview by the classified background visualization (Fig. 60) the user can switch in the background to the basin visualization to inspect the overall behavior of the shear properties, see Fig. 62(a). In Fig. 62(b) the extrema (dots with size scaled by persistence value) of the shear field and their respective basin regions are shown exemplarily.

8.3.2 *Earthquake Data Set*

Data set – Displacement field encoding the ground deformation associated with a simulation of the Landers, CA earthquake (Sec. A.3.2). (Resolution: 450×450, Timing for scalar field topology: 1.19 sec)

Analysis – Figure 52(a) shows all extrema filtered by persistence (0.3) and scaled by magnitude. Among the extrema the location of the strongest forces can be distinguished by their size. The most prominent extrema align with the fault, where clockwise rotation and shear are dominant. The extrema of the isotropic scale only appear as col-



(a) (b)

Figure 62: *Cylinder data set* – (a) Close-up: background visualization of shear basins (transparency of shear icons adjusted to strength, orientation to major eigenvector) for a more detailed inspection. The background visualization reveals properties of the shear which could not be deduced from the LIC visualization only. (b) Extrema (spheres with size adjusted to persistence value) and basins of shear scalar field. The gray rectangle is the close-up area in (a).

ored dots. For the background visualization, the classification into dominant components according to Sec. 8.2.3 is done. This gives a very clear non overlapping image of the overall behavior in the field. The vector clustering captures the main directional features in the data set, additionally sparsely seeded streamlines are displayed in the background.

As one can see in the shear height field in Fig. 56(a), the shear forces are very high in the vicinity of the fault, whereas they are comparatively small in the remainder of the data set. With the foreground visualization of the extrema icons, the attention is clearly drawn to the fault. Amongst these extrema the locations of strongest forces can be determined by the absolut size of the icons. The background visualization displays the trends in the surroundings which corresponds to a rather homogeneous but still relevant shear respectively rotational movement. This highly simplified depiction of strongest forces gives a clear first orientation, which can then be used in combination with a highly detailed visualization as in Fig. 52(c). This visualization by Chen et al. focuses more on the directional information of the tensor components based on a classification and does not offer such a clear and simple distinction of strong and weekly expressed features. Our resulting visualization nicely resembles the hand drawn sketch by the domain experts [CPL+11]. They felt the urge to deduce such a strongly simplified schematic depiction of the most relevant features from the detailed visualization in Fig. 52(c) to attain a first understanding. Please note, in contrast to the hand drawn sketch, we also encode shear in the final visualization. Also the color coding of the icons differs.

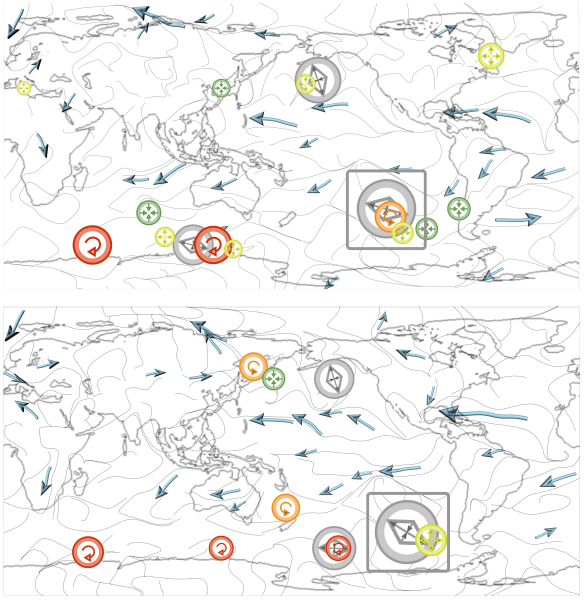


Figure 63: *Climate data set* – Time dependent simulation of wind. Top and bottom image display two consecutive time steps. Due to the high feature density the visualization is extremely simplified – only the most persistent extrema are displayed. Additionally the arrow icons of the vector clustering, contours of the continents, and sparsely seeded streamlines can be seen.

8.3.3 *Climate Data Set*

Data set – Time dependent simulation of wind in a multi-model ensemble forecast system (Sec. A.3.3). The simulation aggregates different climate models with varying parameters. In our results, we inspect two consecutive time steps; the images are computed separately. The data set is of low resolution with high variance and feature density.
(Resolution: 144×73, Timing for scalar field topology: 0.15 sec)

Analysis – Due to the low resolution we chose to compare the evolution of the wind data field and its associated characteristics by only inspecting the persistence filtered extrema (filtering factor: 0.3) and the vector clustering. As further context information sparsely seeded streamlines and the contours of the continents are provided. In the two time steps, related extrema can be visually well correlated due to the high level of abstraction by filtering and the simplicity of the icons (see shear icon in Fig. 63 gray square). Filtering by homological persistence does not avoid overlaps, but within a region of neighboring extrema the most persistent extremum serves as well defined representative. It provides a sound basis for stable and reproducible exponents in form of extrema to compare data over time, or according to varying models or parameters.

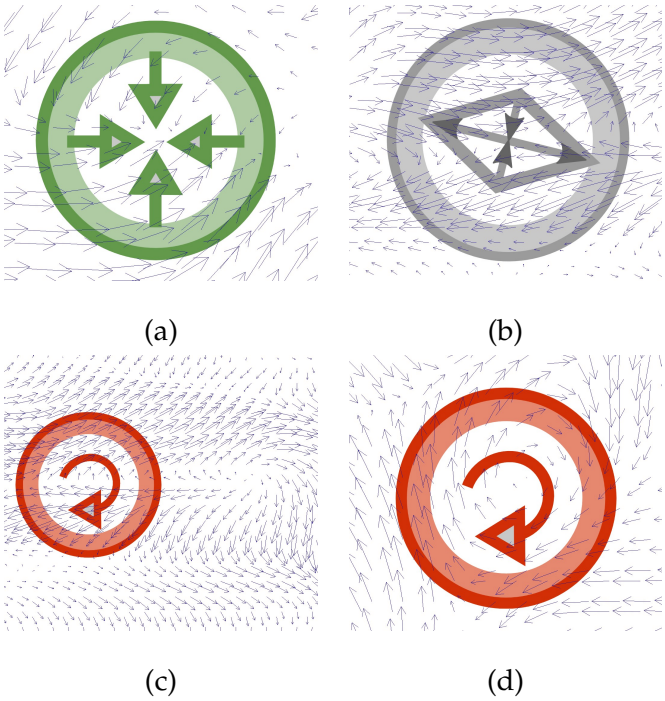


Figure 64: *Climate data set* – Close-ups of Fig. 63 (top row) combined with vector visualization. (a) Isotropic scaling extremum at location where the stream fronts converge from two sides which might indicate an up- or down-stream. (b) Shear extremum where the flow follows to opposing directions. (c+d) The depicted scalar quantities are independent of the chosen frame of reference. In contrast to the extremum in (d) the rotation extremum in (c) is not in a location which the vector visualization would suggest.

A detailed inspection of the climate data sets in combination with simple vector depiction demonstrates that the analysis of the extrema adds new information that cannot directly be deduced from the vector field. Figure 64 shows some close-ups of strong features, which revealed interesting patterns in the climate data set.

8.4 CONCLUSION

We proposed an illustrative visualization of two-dimensional vector fields consisting of two components: a background visualization that serves as an overview of large scale trends and a depiction of details in the foreground. The approach is based on the vector field and its gradient. We decomposed the gradient into shear, isotropic scaling and rotation and analyzed these individual fields. The strengths given by these fields serve as one major component of our visualization approach. In contrast to the approach of Chen et al. [CPL+11] we look at the absolute values of these scalar fields. Our visualization pipeline therefore combines scalar field topology with tensor analysis. The highly abstracted view locates the foreground icons at positions

with clear mathematical interpretation, the extrema of the scalar fields filtered by homological persistence. Homological persistence defines a hierarchy of importance for the extrema. This allows to aggregate features on a solid mathematical basis without the need to employ any heuristics. This could not be accomplished by filtering the extrema by their magnitude only. The visualization is automatic. Only a few easy-to-handle parameters are used. In most cases, they can be set quite similar to default values as shown in the results. The final visualization incorporates quantities that might be hidden in pure vector visualization methods. In addition, it represents information that is independent of the chosen frame of reference. Therefore, it overcomes possible mis-interpretations by only analyzing visualizations based on vectors. The icons are easy to interpret. The aim of this sketch-based visualization is not to provide all possible information in the vector field but rather to give an overview depiction. The resulting sketches are rather sparse and can be composited with more specific visualization methods for in-depth analysis. As future work, we plan to analyze time-dependent fields. The present approach directly allows the application of tracking methods for scalar field topology. This should result in smooth illustrative visualizations of these fields.

9 DISCUSSION AND CONCLUSION

We will conclude this thesis with a brief discussion on questions of general content which cannot be associated explicitly with one of the previous chapters. Finally, the conclusion will summarize the central topics and contributions of this thesis.

9.1 DISCUSSION

In the following we will discuss central aspects that arose during the advancement of this thesis. We present them here hoping that future research can profit from those superordinate aspects we have learned in the vein of this thesis.

9.1.1 *Tensors in Visualization*

As mentioned in the beginning using second order tensors in visualization has evident benefits but it comes with the challenge that the wealth of information encoded in these tensors needs to be handled. Visualizing scientific results in terms of scalars and vectors is attractive as their interpretation is more intuitive than that of second order tensors. Especially the visualization of indefinite symmetric as well as asymmetric tensors is still underrepresented. In the previous chapters we followed the idea to enhance the understanding of tensor data by analyzing them in terms of decompositions. Indefinite symmetric tensor fields are visualized on basis of their eigenvector and eigenvalue fields and asymmetric tensor fields are split into a shear, a rotation, and isotropic scaling tensor field. Hereby, these decompositions uniquely represent the original tensor data but split the tensors into easier to perceive components.

We proposed our approaches to also support the understanding of tensors themselves as the decomposed fields allowed us to provide simplification strategies and easy to read depictions whilst preserving essential tensor characteristics. We presented visualization methods which process a variety of tensor types from different application areas. Yet, we are convinced that there is no universal tensor field visualization method as such, that covers all possible types of tensors and their relevant properties. For the further advancement of tensor fields visualization the users play an important role – they need to become familiar with depictions for tensor data. Thus, we believe the tensor field visualizations should not be overcharged with a maximum of information but rather assist the user in focusing on

the relevant characteristics. We believe topology based segmentations allow to inspect the data at a higher level of abstraction. However, we consider them rather unsuitable for users new to tensor field analysis. As a remedy, we found textures surprisingly powerful to give a intuitive continuous impression of the tensor fields and to display many parameters without overwhelming the user.

9.1.2 *Topology of Symmetric Tensor Fields in Visualization*

As mentioned before topology in general has a long tradition in visualization. Although it might seem quite abstract to someone unfamiliar with this concept it has proven to be beneficial from various perspectives. Topology can be used for indefinite symmetric tensors and represent all key directional features in the field in a continuous manner. Then topology is a mean to achieve a significant data compression without corrupting the initial data or introducing heuristics to achieve a simplification. Further, topology provides a consistent framework for subsequent processing and advanced visualization. Respecting the extracted degenerate points in tensor fields assures that no inconsistencies are introduced in a subsequent processing step. For flow fields the topology of derived symmetric tensor fields (e.g. rate of strain tensors) is independent of the chosen frame of reference. With vector field topology the problem can arise that the critical points and separating lines seem to give a straightforward intuition about the behavior in the field. But changing the frame of reference, e.g. changing the point of view can completely change the key topological features in a vector field. The derived tensor fields are not affected by such a change of reference frame and thus, they provide superordinate information content.
However, the degenerate points of symmetric tensor fields do not have a comparable intuition like the critical points of a vector field (source, sink, etc.). In visualization for the application areas of mechanics and engineering the interpretation of the degenerate points in a symmetric tensor field is limited to the fact that these are the locations where the eigenvectors degenerate and by definition eigenvalues are equal. For asymmetric tensor fields derived from flow vector fields the attempt has been made and it has been showed that topological elements, the circular points, can be directly associated with purely circular flow in the original vector field. It would be a very interesting option for the future to conduct further research if and to what extent topological features can be associated with physical properties. We believe, this would highly add to the usability of topological methods for symmetric tensor fields.

Numerical Issues
One important issue which has to be mentioned with respect to the

proposed topology extraction method here and in previous literature is that all methods are numerical approaches. This induces several effects. As described in Chap. 4 for linear interpolation degenerate points of higher order decay into multiple linear degenerate points. We have presented a method to alleviate this effect by extracting degenerate elements of higher dimensionality, e.g. degenerate lines, or triangles. This approach reduces the complexity of the field as also separatrices can be classified on a higher level and accordingly strongly reduced. Compared to previous approaches this simplification is done without the necessity of user defined parameters. With the extension to detect degenerate points and separatrices in piecewise continuous and piecewise constant tensor fields we can claim that our topology is complete. However, the separatrix computation is performed by a numerical integration scheme which implies it depends on the chosen numerical precision and further parameters (e.g. step width). Here, it would be a highly interesting but challenging option for the future to develop a combinatorial approach for the extraction of the tensor field topology. Especially the fact that separatrices in tensor fields hardly connect degenerate points for numerical integration schemes would be highly interesting to investigate with respect to combinatorial approaches. This would induce a desirable additional simplification of the graph structure and enrich the comprised adjacency information. To the best of our knowledge up to now there has been no attempt towards this aim.
A related topic is that of noise which particularly arises for measured data, but which is inherently given on any computational system due to the representation of numbers and round-off errors during computation. This issue is apparent for every type of visualization. However, topological methods tend to be more sensitive to noise than other methods. Their aim is to generate a strongly reduced view on the initial data set. Thus, newly generated degenerate points and consequently separatrices are more grave if they occur due to noise. In dense visualizations like the texture mapping in Chap. 6 these additional features occur as well, however, they are not as prominent. Here, research would be beneficial towards a filtering strategy based on a clear mathematical foundation to clean up the resulting topological graph.

9.1.3 *Dimension Reduction*

Three dimensional tensor data sets naturally suffer of occlusion if the full tensor information ought to be depicted. Only a strong filtering allows to display such three dimensional tensor fields as reduced sets in their original object space. Approaches like integrated views support the exploration of three dimensional tensor data sets in terms of spaces based on derived attributes (e.g. the eigenvalue space) and

facilitate the choice of relevant filtering attributes (see e.g. [KKH11]). Still, any continuous depiction of tensor fields soon reaches its limits when attempted in three dimensional space. Even the strong reduction attained by the topology computation for three dimensional tensors cannot prevent the problem of occlusion as the regions of homogeneous directions are bounded by separating surfaces which naturally occlude other structures.
We present our methods also to be suitable for three dimensional tensor fields that are investigated in terms of extracted slices or isosurfaces. Hereby, the data is projected onto a planar or curved surface geometry and two dimensional tensors are processed and visualized. Investigating tensor fields in form of two dimensional data sets is a feasible approach to give a continuous view of any indefinite tensor field which preserves the anisotropic properties of the tensors. However, it has to be stressed out that the user needs to keep in mind that observations in two dimensional space cannot be directly conducted to three dimensional space. A simple example is that of a projected two dimensional isotropic tensor which might be highly anisotropic in three dimensional space (see Fig. 65). Hereby, taking reassurance from the original three dimensional data, for example by probing, can provide a way to gain continuous insight at first and demanding full detail at locations of interest.

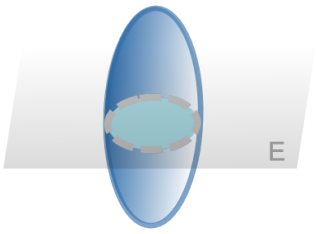


Figure 65: Three dimensional anisotropic tensor (blue ellipsoid) may become an isotropic two dimensional tensor (turquoise ellipse) if projected on surface E.

9.2 CONCLUSION

Second order tensors provide a versatile mathematical formalism to elegantly define and describe complex phenomena. However, they bring about a certain inhibition threshold as a direct interpretation of the encoded information is mostly not given. They encode multilinear transformations and capture anisotropic properties which makes them extremely flexible but in turn also prevents users to investigate them directly. In this work, the focus is on the visualization of tensors occurring in the application areas of mechanics and engineering, which are strongly underrepresented in scientific visualization. Our work comprises indefinite symmetric tensors as well as asymmetric tensors. We chose to design our concepts for two dimensional fields

that are defined on planar or curved surfaces. The visualization of second order tensors on surfaces allows to depict those fields as a whole and discover coherences. But also inconsistencies and other unexpected actualities can be detected which is an important benefit of scientific visualization.

To condense the wealth of information in symmetric tensor fields we provide segmentation algorithms that reflect essential properties in a graph structure. Chap. 4 presents the computation of the integral topological graph which allows the immediate inspection of all directional features. Based on a sound mathematical basis this graph partitions the field into cells of homogeneous eigenvector behavior without introducing any heuristics in the simplification process. Compared to previous approaches the key topological structures are naturally reduced by the extraction of topological features of higher dimensionality. In Chap. 5 this topological basis is refined towards a segmentation that also respects the eigenvalue fields. The segmentation workflow is designed to be flexible and can be adapted to reflect also other scalar quantities or visualization aims. Due to the adaptive character of the segmentation workflow a focus and context visualization can be immediately deduced. This gives the user the freedom to inspect an individual data set on demand at the desired level of detail.

Both segmentations give a strongly simplified view on the initial field. The user can inspect the tensor field as a whole and distinguish resembling regions from regions of strong variation. Additionally, the segmentation results are returned as explicit geometric structures that can be enriched with further information, used for statistical inquiry, and tracking over time.

We attached high importance to broaden the applicability of our methods. In Chap. 7 the necessary extensions are given to conduct this segmentation approaches also for piecewise continuous and piecewise constant tensor fields. This facilitates the investigation of data on curved surfaces, e.g. boundary surfaces or extracted isosurfaces of initially three dimensional tensor data.

The segmentation framework of Chap. 5 is used as a consistent framework to map textures for a rich visualization of tensor fields. The multitude of parameters given by textures and glyphs are used in Chap. 6 to encode further quantities and physical properties of the tensors in the visualization. Important characteristics like compression or expansion can be intuitively perceived by the pattern frequency of the used textures. Further, any type of glyph can be placed in the center of the homogeneous segmentation cells for explicit investigation of the tensor field at distinct locations.

For asymmetric tensor fields we propose an illustrative visualization in Chap. 8 which is based on a decomposition into a rotation, isotropic scaling, and shear tensor component. These components al-

leviate the understanding of the encoded transformation in an asymmetric tensor and can be represented with strongly simplified icons. This facilitates to depict dominant features of the tensor field in the foreground and to subsume large scale trends in the background in a sketch-like manner. The visualization method supports the need of domain experts to have simplified, easy to read depictions for a first understanding and overview of such a highly complex field.

We have presented visualization concepts to further enhance the understanding of second order tensors. A variety of methods has been proposed aimed at domain experts but also at users only becoming acquainted with the tensor concept. Strong simplifications but also highly rich visualizations were developed for a comprehensive depiction of tensors. The concepts are applicable to a multitude of tensor types from different application areas which we have proven by applying them to strongly differing data sets. All our methods build on topology which serves as profound mathematical basis to achieve a reproducible simplification which we consider constitutional for the visualization of scientific data.

APPENDIX

A

DATA SETS

A.1 STRESS TENSOR DATA

Here, three data sets from structural engineering are presented which are simulations of forces acting upon solid blocks resulting in stress tensor data (see also Sec. 2.8). The One- and two-point load are well studied data sets which are especially beneficial to demonstrate basic proof of concepts for new visualization methods.

A.1.1 *One-point load – analytic data set*

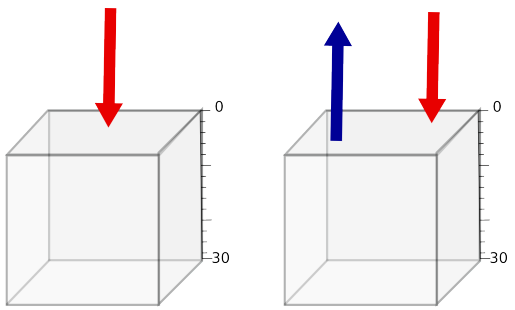
The one-point load is a model of a solid block on which a pushing force acts upon resulting in stress tensor data (Fig. A.66(a)). The analytic formula is given in [SD03].

A.1.2 *One-point load – simulation*

Same data set as in Sec. A.1.1 however given as finite element simulations (FEM) which results in more complex structures (Fig. A.66(a)).

A.1.3 *Two-point load*

Data set generated similar to the one-point load. Here two opposing forces are acting upon a solid block – one pushing and one pulling force (Fig. A.66(b)).



(a) (b)

Figure 66: Schematic depictions of the one- and two-point load data sets. (a) One-point load data set. A pushing force is acting upon a solid block. (b) Two-point load data set. A pushing and a pulling force act upon a solid block.

A.1.4 *Notched block*

Simulation of multiple forces applied to a notched block, using hp-adaptive finite elements.

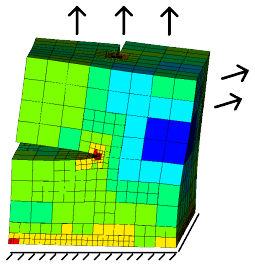


Figure 67: Notched block data set.

A.2 RATE OF STRAIN TENSOR DATA

A.2.1 *Aneurysm*

Computational fluid dynamics (CFD) simulation performed on a tetrahedral mesh. For the rupture risk analysis the wall shear vector field is of special relevance [BRM+08], which is given as three-dimensional vector field defined on the surface vertices. The vector field is projected onto the triangles. The constant rate of strain tensor field is derived locally per triangle. Let $u = (u_1, u_2) \in \mathbb{R}^2$ be a vector in the local coordinate system. The components of the rate of strain tensor are derived as given in Eq. 11.

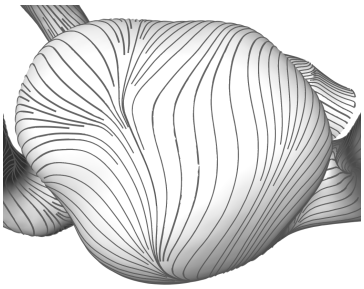


Figure 68: Aneurysm data set: Wall shear vector field depicted as streamlines.

A.3 VECTOR DATA

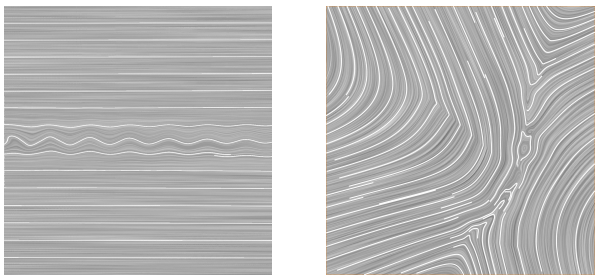
A.3.1 *Cylinder data set*

Time-dependent (32 time steps) two-dimensional CFD simulation of the von-Kármán vortex street [NSA+08, Wil96], the flow behind a

cylinder with Reynolds number = 100. The flow is incompressible, so by definition there is no isotropic scaling in the field. We use two versions of this data set: the original data (Fig. A.69(a)) and a version with changed reference frame with subtracted average velocity.

A.3.2 *Earthquake data set*

Displacement field encoding the ground deformation associated with a simulation of the June, 1992 Mw=7.3 Landers, CA earthquake [Vin98] (Fig. A.69(b)).



(a) (b)

Figure 69: Vector fields depicted as LIC texture and sparse streamlines. (a) Original cylinder data set. (b) Earthquake data set.

A.3.3 *Climate data set*

Time dependent simulation of wind in a multi-model ensemble forecast system for reliable seasonal to inter-annual prediction [PAA+04]. This simulation is freely available by the DEMETER project and aggregates different climate models with varying parameters. The data set is of low resolution with high variance and feature density.

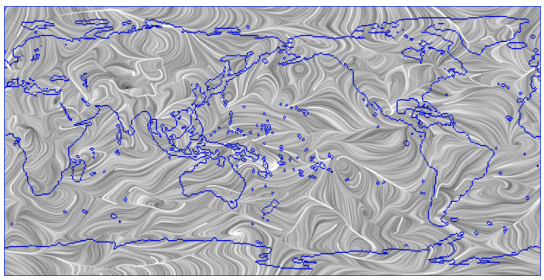


Figure 70: Climate data set: Vector field depicted as LIC texture and sparse streamlines.

LIST OF FIGURES

LIST OF TABLES

BIBLIOGRAPHY

[ACSD+03] Pierre Alliez, David Cohen-Steiner, Oliver Devillers, Bruno Levy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *Siggraph '03 - ACM Transactions on Graphics (TOG)*, 22(3):485–493, Jul 2003. SIGGRAPH 2003 Session : Surfaces.

[AFPA06] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, 56(2):411–421, June 2006.

[AH11] Cornelia Auer and Ingrid Hotz. Complete tensor field topology on 2d triangulated manifolds embedded in 3d. *Computer Graphics Forum (EuroVis'11)*, 30(3):831–840, 2011.

[AKKH13] Cornelia Auer, Jens Kasten, Andrea Kratz, and Eugene Zhang Ingrid Hotz. Automatic, Tensor-Guided Illustrative Vector Field Visualization. *Proc. IEEE Pacific Visualization Symposium 2013*, pages 265–275, 2013.

[AMR88] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis and Applications*. Springer-Verlag GmbH, 2nd edition, 1988.

[ASKH12] Cornelia Auer, Claudia Stripf, Andrea Kratz, and Ingrid Hotz. Glyph- and texture-based visualization of segmented tensor fields. In *International Conference on Information Visualization Theory and Applications (IVAPP'12)*, 2012.

[ASNZH11] Cornelia Auer, Jaya Sreevalsan-Nair, Valentin Zobel, and Ingrid Hotz. 2D Tensor Field Segmentation. In Hans Hagen, editor, *Scientific Visualization: Interactions, Features, Metaphors*, volume 2 of *Dagstuhl Follow-Ups*, pages 17–35. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2011.

[Bar81] A. H. Barr. Superquadrics and angle-preserving transformations. *IEEE Comput. Graph. Appl.*, 1(1):11–23, 1981.

[Bat67] G. K. Batchelor. *An Introduction to Fluid Dynamics*. cup, 1967.

[BCG+92] Alan H. Barr, Bena Currin, Steven Gabriel, John F. Hughes, and Sage Design. Smooth interpolation of orientations with angular velocity constraints using quaternions. In *Proceedings of SIGGRAPH'92*, pages 313–320, 1992.

[BCP+12] Andrea Brambilla, Robert Carnecky, Ronald Peikert, Ivan Viola, and Helwig Hauser. Illustrative flow visualization: State of the art, trends and challenges. *EG 2012*, State of the Art Reports:75–94, 2012.

[Ben04] Werner Benger. *Visualization of General Relativistic Tensor Fields via a Fiber Bundle Data Model*. Ph.d. thesis, Free University Berlin, 2004. ISBN 3-86541-108-8.

[BN03] Mehmet Bilgen and Ponnada A. Narayana. Mohr diagram interpretation of anisotropic diffusion indices in MRI. *Magnetic Resonance Imaging*, 31:567–572, 2003.

[Bra03] Rebecca M. Brannon. *Functional and Structured Tensor Analysis for Engineers*. UNM Book Draft, 2003.

[BRM+08] L. Boussel, V. Rayz, C. McCulloch, A. Martin, G. Acevedo-Bolton, M. Lawton, R. Higashida, W.S. Smith, W.L. Young, and D. Saloner. Aneurysm growth occurs at region of low wall shear stress: patient-specific correlation of hemodynamics and growth in a longitudinal study. *PubMed - Stroke*, 39:2997–3002, 2008.

[BW03] Abhir Bhalerao and Carl-Fredrik Westin. Tensor splats: Visualising tensor fields by texture mapped volume rendering. In *MICCAI (2)*, pages 294–302, 2003.

[BWHK05] Anders Brun, Carl-Fredrik Westin, Steven Haker, and Hans Knutsson. A tensor-like representation for averaging, filtering and interpolation of 3-d object orientation datasets. In *IEEE International Conference on Image Processing ICIP 2005*, volume 3, pages III – 1092–5, 11-14 Sept. 2005.

[CAA01] O. Coulon, Daniel C. Alexander, and S.R. Arridge. Tensor field regularisation for dt-mr images. In *MIUA01, British. Conference on Medical Image*, pages 21–24, 2001.

[CBHL03] Jiann-Liang Chen, Zhaojun Bai, Bernd Hamann, and Terry J. Ligocki. A normalized-cut algorithm for hierarchical vector field data segmentation. In *In Proceedings of Visualization and Data Analysis*, 2003.

[CCMMM+09] L. Cammoun, C.A. Castano-Moraga, E. Munoz-Moreno, D. Sosa-Cabrera, B. Acar, M.A. Rodriguez-Florido, A. Brun, H. Knutsson, and J. P. Thiran. A review of tensors and tensor signal processing. In Santiago Aja-Fernández, Rodrigo de Luis García, Dacheng Tao, and Xuelong Li, editors, *Tensors in Image Processing and Computer Vision*. Springer London, 2009.

[cga] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[CL93] Brian Cabral and Leith Leedom. Imaging vector fields using line integral convolution. In James T. Kajiya, editor, *Proceedings of SIGGRAPH 1993*, volume 27, pages 263–272, August 1993 1993.

[CPL+11] Guoning Chen, Darrel Palke, Zhongzang Lin, Harry Ye, Paul Vincent, Robert S. Laramee, and Eugene Zhang. Asymmetric tensor field visualization for surfaces. *EEE Transactions on Visualization and Computer Graphics*, 17(11), 2011.

[CRB+05] Patricia Crossno, David H. Rogers, Rebecca M. Brannon, David Coblentz, and Joanne T. Fredrich. Visualization of geologic stress perturbations using mohr diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 11(5):508–518, 2005.

[CRBC04] Patricia Crossno, David H. Rogers, Rebecca M. Brannon, and David Coblentz. Visualization of salt-induced stress perturbations. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 369–376. IEEE Computer Society, 2004.

[CYY+11] Cheng-Kai Chen, Shi Yan, Hongfeng Yu, Nelson Max, and Kwan-Liu Masahiro. An illustrative visualization framework for 3d vector fields. *Computer Graphics Forum (Pacific Graphics'11)*, 30(7):1941–1951, 2011.

[Dal83] U. Dallmann. Topological Structures of Three-Dimensional Flow Separations. Dfvlr-ava bericht nr. 221-82 a 07, Deutsche Forschungs- und Versuchsanstalt fÃ¼r Luft- und Raumgfahrt e.V., 1983.

[Dan97] Donald A. Danielson. *Vectors and Tensors in Engineering and Physics*. Addison-Wesley Publishing Company, 2nd edition, 1997.

[dBvKOS00] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry:*

Algorithms and Applications. Springer-Verlag, second edition, 2000.

[Del94] Thierry Delmarcelle. *The Visualization of Second-order Tensor Fields*. PhD thesis, Stanford University, December 1994. diss.

[DGBW09] Christian Dick, Joachim Georgii, Rainer Burgkart, and Rüdiger Westermann. Stress tensor field visualization for implant planning in orthopedics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1399–1406, 2009.

[DH92] Thierry Delmarcelle and Lambertus Hesselink. Visualization of second order tensor fields and matrix data. In *VIS '92: Proceedings of the conference on Visualization '92*, pages 316–323, 1992.

[DH94] Thierry Delmarcelle and Lambertus Hesselink. The topology of second-order tensor fields. In *VIS '94: Proceedings of the conference on Visualization '94*, pages 140–148. IEEE Computer Society Press, October 1994.

[dLGLW09] Rodrigo de Luis García, Carlos Alberola Lopez, and Carl Frederik Westin. Segmentation of tensor fields: Recent advances and perspectives. In Santiago Aja Fernández, Rodrigo de Luis García, Dacheng Tao, and Xuelong Li, editors, *Tensors in Image Processing and Computer Vision*. Springer London, 2009.

[DW05] Qiang Du and Desheng Wang. Anisotropic centroidal voronoi tessellations and their applications. *SIAM Journal on Scientific Computing*, 26(3):737–761, 2005.

[EGL+06] Gordon Erlebacher, Christoph Garth, Robert S. Laramee, Holger Theisel, Xavier Tricoche, Tino Weinkauf, and Daniel Weiskopf. Texture and Feature-Based Flow Visualization - Methodology and Application. In *IEEE Visualization 06 Tutorial*, 2006.

[Els07] Serguey A. Elsoufiev. *Strength Analysis in Geomechanics*. Foundations Engineering Mechanics,. Springer Berlin Heidelberg,, Berlin, Heidelberg :, 2007.

[ELZ02] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological Persistence and Simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.

[FAPA07] P. Fillard, V. Arsigny, X Pennec, and N Ayache. Clinical dt-mri estimation, smoothing and fiber tracking

with log-euclidean metrics. *EEE Transactions on Medical Imaging,*, 26(11):1472–1482, 2007.

[FHHJ08] Louis Feng, Ingrid Hotz, Bernd Hamann, and Kenneth Joy. Anisotropic noise samples. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):342–354, 2008.

[FJ07] P. Thomas Fletcher and Sarang Joshi. Riemannian geometry for the statistical analysis of diffusion tensor data. *Signal Process.*, 87(2):250–262, 2007.

[FWB03] C. Feddern, Jochim Weickert, and Bernhard Burgeth. Level-set methods for tensor-valued images. In O. Faugeras and N. Paragios, editors, *Proc. Second IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, 2003.

[GLT+07] Christoph Garth, Robert S. Laramee, Xavier Tricoche, Jürgen Schneider, and Hans Hagen. Extraction and visualization of swirl and tumble motion from engine simulation dataset. In *Topological Methods in Visualization (TopoInVis'07)*, pages 121–135. Springer, 2007.

[GPR+01] Harald Garcke, Tobias Preušer, Martin Rumpf, Alexandru C. Telea, Ulrich Weikard, and Jarke J. van Wijk. A phase field model for continuous clustering on vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):230–241, 2001.

[GTS+04] Christoph Garth, Xavier Tricoche, Tobias Salzbrunn, Tom Bobach, and Gerik Scheuermann. Surface Techniques for Vortex Visualization. In Oliver Deussen, Charles Hansen, Daniel A. Keim, and Dietmar Saupe, editors, *Data Visualization 2004 - Eurographics/IEEE TCVG Symposium on Visualization Proceedings*, pages 155–164, 2004.

[Hab90] R. B. Haber. Visualization techniques for engineering mechanics. *Comput. Syst. Educ.*, 1(1):37–50, 1990.

[HFH+04] Ingrid Hotz, Louis Feng, Hans Hagen, Bernd Hamann, Kenneth Joy, and Boris Jeremic. Physically based methods for tensor field visualization. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 123–130. IEEE Computer Society, 2004.

[HGH+10] Mathias Hummel, Christoph Garth, Bernd Hamann, Hans Hagen, and Kenneth I. Joy. "IRIS: Illustrative Rendering for Integral Surfaces". *IEEE Transactions*

on Visualization and Computer Graphics, 16:1319–1328, 2010.

[HH90] J. L. Helman and Lambertus Hesselink. Surface representations of two- and three-dimensional fluid flow topology. In *VIS '90: Proceedings of the conference on Visualization*, pages 6–13. IEEE Computer Society Press, 1990.

[HLL97] Lambertus Hesselink, Yuval Levy, and Yingmei Lavin. The topology of symmetric, second-order 3D tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):1–11, January/March 1997. diss.

[HLNW11] Marcel Hlawatsch, P. Leube, W. Nowak, and Daniel Weiskopf. Flow radar glyphs – static visualization of unsteady flow with uncertainty. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1949–1958, 2011.

[HSH07] Mario Hlawitschka, Gerik Scheuermann, and Bernd Hamann. Interactive glyph placement for tensor fields. In *ISVC (1)*, pages 331–340, 2007.

[HSNHH10] Ingrid Hotz, Jaya Sreevalsan-Nair, Hans Hagen, and Bernd Hamann. Tensor field reconstruction based on eigenvector and eigenvalue interpolation. In Hans Hagen, editor, *Scientific Visualization: Advanced Concepts*, volume 1 of *Dagstuhl Follow-Ups*, pages 110–123, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Center for Informatik.

[HSW+09] Mario Hlawitschka, Gerik Scheuermann, Gunther H. Weber, Owen T. Carmichael, Bernd Hamann, and Alfred Anwander. Interactive volume rendering of diffusion tensor data. In D. H. Laidlaw and J. Weickert, editors, *Visualization and Processing of Tensor Fields: Advances and Perspectives*, Mathematics and Visualization. Springer, 2009.

[Hun87] J. C. R. Hunt. Vorticity and Vortex Dynamics in Complex Turbulent Flows. *Transactions of the Canadian Society of Mechnical Engineering*, 11:21–35, 1987.

[HWHJ99] Bjoern Heckel, Gunther H. Weber, Bernd Hamann, and Kenneth I. Joy. Construction of vector field hierarchies. In David S. Ebert, Markus Gross, and Bernd Hamann, editors, *Proceedings of IEEE Visualization '99*, pages 19–26, Los Alamitos, CA, 10 1999. IEEE, IEEE Computer Society Press.

[HYW03] Youssef M. A. Hashash, John I-Chiang Yao, and Donald C. Wotring. Glyph and hyperstreamline representation of stress and strain tensors and material constitutive response. *International Journal for Numerical and Analytical Methods in Geomechanics*, 27(7):603–626, 2003.

[JH95] Jinhee Jeong and Fazle Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 285 1995.

[JKM06] T.J. Jankun-Kelly and Ketan Mehta. Superellipsoid-based, real symmetric traceless tensor glyphs motivated by nematic liquid crystal alignment visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 12:1197–1204, 2006.

[JL01] Bruno Jobard and Wilfrid Lefer. Multiresolution flow visualization. In *WSCG 2001 Conference Proceedings*, pages 33–37, 2001.

[JSF+02] B. Jeremic, Gerik Scheuermann, Jan Frey, Zhaohui Yang, Bernd Hamann, Kenneth I. Joy, and Hans Hagen. Tensor visualization in computational geomechanics. *International Journal for Numerical and Analytical Methods in Geomechanics*, 26:925–944, 2002.

[Kar04] Menelaos I. Karavelas. A robust and efficient implementation for the segment voronoi diagrams. In *Internat. Symp. on Voronoi diagrams in Science and Engineering (VD'04)*, pages 51–62, 2004.

[Kas12] Jens Kasten. *Lagrangian feature extraction in two-dimensional unsteady flows*. PhD thesis, Freie Universität Berlin, 2012.

[KASH12] Andrea Kratz, Cornelia Auer, Markus Stommel, and Ingrid Hotz. Visualization and Analysis of Second-Order Tensors: Moving Beyond the Symmetric Positive-Definite Case. *Computer Graphics Forum - State of the Art Reports*, xx:xx, accepted 2012.

[KEN+07] Gordon Kindlmann, Raul San Jose Estepar, Marc Niethammer, Steven Haker, and Carl-Fredrik Westin. Geodesic-loxodromes for diffusion tensor interpolation and difference measurement. In *Tenth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'07)*, Lecture Notes in Computer Science 4791, pages 1–9, 2007.

[KHL99] David N. Kenwright, Chris Henze, and Creon Levit. Feature extraction of separation and attachment lines. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):135–144, 1999.

[Kin04] Gordon Kindlmann. Superquadric tensor glyphs. In *Proceedings of IEEE TVCG/EG Symposium on Visualization '04*, pages 147–154, 2004.

[KKH11] Andrea Kratz, Nino Kettlitz, and Ingrid Hotz. Particle-based anisotropic sampling for two-dimensional tensor field visualization. In *Proceedings of Vision, Modeling, and Visualization (VMV'09)*, 2011.

[KLG+11] Alexander Kuhn, Dirk J. Lehmann, Rocco Gaststeiger, Matthias Neugebauer, Bernhard Preim, and Holger Theisel. A clustering-based visualization technique to emphasize meaningful regions of vector fields. In *Proceedings of Vision, Modeling, and Visualization (VMV'11)*, 2011.

[KMH11] Andrea Kratz, Björn Meyer, and Ingrid Hotz. A Visual Approach to Analysis of Stress Tensor Fields. In Hans Hagen, editor, *Scientific Visualization: Interactions, Features, Metaphors*, volume 2 of *Dagstuhl Follow-Ups*, pages 188–211. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2011.

[KRHH11] Jens Kasten, Jan Reininghaus, Ingrid Hotz, and Hans-Christian Hege. Two-dimensional time-dependent vortex regions based on the acceleration magnitude. *Transactions on Visualization and Computer Graphics (VisWeek'11)*, 17(12):2080–2087, 2011.

[KW99] Gordon Kindlmann and David Weinstein. Hue-balls and lit-tensors for direct volume rendering of diffusion tensor fields. In *VIS '99: Proceedings of the conference on Visualization '99*. IEEE Computer Society, 1999.

[KW06] Gordon Kindlmann and Carl-Fredrik Westin. Diffusion tensor visualization with glyph packing. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1329–1336, 2006.

[KYHR05] R.D. Kriz, M. Yaman, M. Harting, and A.A. Ray. Visualization of zeroth, second, fourth, higher order tensors, and invariance of tensor equations. Summer Research Project 2003, 2005. University Visualization and Animation Group, Virginia Tech, Blacksburg, Virginia.

[Lan12] J. M. Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, 2012.

[LBHL97] Yingmei Lavin, Rajesh Batra, Lambertus Hesselink, and Yuval Levy. The topology of symmetric tensor fields. *AIAA 13th Computational Fluid Dynamics Conference,*, page 2084, June-July 1997. Snowmass, Colorado.

[LDB05] Guillaume Lavoue, Florent Dupont, and Atilla Baskurt. A new cad mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design*, 37(10):975–987, 2005.

[LHZP07] Robert S. Laramee, Helwig Hauser, Lingxiao Zhao, and Frits H. Post. Topology-based flow visualization, the state of the art. In Hans Hagen Helwig Hauser and Holger Theisel, editors, *Topology-Based Methods in Visualization*, Mathematics and Visualization, pages 1–19. Springer Berlin Heidelberg, 2007.

[LR75] D. Lovelock and H. Rund. *Tensors, Differential Forms, and Variational Principles*. Dover Books on Mathematics Series. Dover Publ., 1975.

[LRDF06] Christophe Lenglet, Mikael Rousson, Rachid Deriche, and oliver Faugeras. Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor mri processing. *Journal of Mathematical Imaging and Vision*, 25(3):423–444, oct 2006.

[LV+06] Wan-Chiu Li, , Bruno Vallet, Nicolas Ray, and Bruno Levy. Representing higher-order singularities in vector fields on piecewise linear surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 12:1315–1322, 2006.

[LW09] David H. Laidlaw and Joachim Weickert. *Visualization and Processing of Tensor Fields: Advances and Perspectives*. Mathematics and Visualization. Springer, 2009.

[MB06] Maher Moakher and Philipp G. Batchelor. Symmetric positive-definite matrices. In Joachim Weickert and Hans Hagen, editors, *Visualization and Processing of Tensor Fields*, Mathematics and Visualization, pages 285–297. Springer, 2006.

[MCHM10] Stephane Marchesin, Cheng-Kai Chen, Chris Ho, and Kwan-Liu Ma. View-dependent streamlines for 3d vector fields. *IEEE Transactions on Visualization and Computer Graphics (Vis'10)*, 16(6):1578–1586, 2010.

[MDB87] B. H. McCormick, T. A. DeFanti, and M. D. Brown. Visualization in Scientific Computing-A Synopsis. *Computer Graphics*, 21(6):61–70, November 1987.

[MFWAL04] M. Martin-Fernandez, C.-F. Westin, and C. Alberola-Lopez. 3d bayesian regularization of diffusion tensor MRI using multivariate gaussian markov random fields. In *7th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'04)*, Lecture Notes in Computer Science, pages 351–359, September 2004.

[Mil63] J. Milnor. *Morse Theory*. Princeton University Press, 1963.

[MOD05] Alexander McKenzie, Santiago V. Ombeyda, and Mathieu Desbrun. Vector field analysis and visualization through variational clustering. In *Proceeding of The Joint Eurographics - IEEE TCVG Symposium on Visualization '05*, 2005.

[MSM96] J. G. Moore, S. A. Schorn, and J. Moore. Methods of classical mechanics applied to turbulence stresses in a tip leakage vortex. *Journal of Turbonmachinery*, 118(4):622–630, 1996.

[MT88] J. E. Marsden and A. J. Tromba. *Vector Calculus, Third Edition*. W. H. Freeman and Company, New York, 1988.

[MVvW05] Bart Moberts, Anna Vilanov, and J.J. van Wijk. Evaluation of fiber clustering methods for diffusion tensor imaging. In *Proceedings of IEEE Visualization Conference*, pages 65–72, 2005.

[NJ99] Gregory M. Nielson and Il-Hong Jung. Tools for Computing Tangent Curves for Linearly Varying Vector Fields over Tetrahedral Domains. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):360–372, 1999.

[NJP05] Alisa Neeman, Boris Jeremic, and Alex Pang. Visualizing tensor fields in geomechanics. In *VIS '05: Proceedings of the conference on Visualization '05*, page 5, 2005.

[NSA+08] B. R. Noack, M. Schlegel, B. Ahlborn, G. Mutschke, M. Morzyński, P. Comte, and G. Tadmor. A Finite-time Thermodynamics of Unsteady Fluid Flows. *J. Non-Equilibr. Thermodyn.*, 33(2):103–148, 2008.

[PAA+04] T.N. Palmer, A. Alessandri, U. Andersen, P. Cantelaube, M. Davey, P. DÃ©lÃ©cluse, M. DÃ©quÃ©, E. DÃez, F.J. Doblas-Reyes, and H. Feddersen. Development of a European multi-model ensemble system for seasonal to inter-annual prediction (DEMETER). *Bulletin of the American Meteorological Society*, 85(6):853–872, 2004.

[PFA06] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *Int. J. Comput. Vision*, 66:41–66, January 2006.

[PP03] Konrad Polthier and Eike Preuss. Identifying vector field singularities using a discrete hodge decomposition. In Hans-Christian Hege, editor, *Visualization and Mathematics III*, volume III of *Visualization and Mathematics*, pages 113–134. Springer Verlag, June 2003.

[RGH+10] Jan Reininghaus, David Günther, Ingrid Hotz, Steffen Prohaska, and Hans-Christian Hege. TADD: A computational framework for data analysis using discrete morse theory. In *Proc. ICMS 2010*, 2010.

[RLF09] R. M. Rustamov, Y. Lipman, and T. Funkhouser. "Interior Distance Using Barycentric Coordinates". In *SGP '09: Proceedings of the Symposium on Geometry Processing*, pages 1279–1288, Aire-la-Ville, Switzerland, 2009. Eurographics Association.

[RPH+09] Olufemi Rosanwo, Christiph Petz, Ingrid Hotz, Steffen Prohaska, and Hans-Christian Hege. Dual streamline seeding. In *IEEE Pacific Visualization Symposium '09*, 2009.

[SCK10] David Schroeder, Dane Coffey, and Daniel F. Keefe. Drawing with the flow: a sketch-based interface for illustrative visualization of 2D vector fields. In *Proc of ACM SIGGRAPH/EG Sketch-Based Interfaces and Modeling*, pages 49–56, 2010.

[SD03] N. N. Som and S. C. Das. *Theory And Practice Of Foundation Design*. Prentice-Hall, India, 2003.

[SH87] Lee A. Segel and G. H. Handelman. *Mathematics applied to continuum mechanics*. Dover Publications, 1987.

[SHB+99] Christopher D. Shaw, James A. Hall, Christine Blahut, David S. Ebert, and D. Aaron Roberts. Using shape to visualize multivariate data. In *NPIVM '99: Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM internation conference on Information and knowledge management*, pages 17–20. ACM, 1999.

[SJWS08] Tobias Salzbrunn, Heike Jänicke, Thomas Wischgoll, and Gerik Scheuermann. The state of the art in flow visualization: Partition-based techniques. In *Simulation and Visualization 2008 Proceedings*, 2008.

[SK10] Thomas Schultz and Gordon Kindlmann. Superquadric glyphs for symmetric second-order tensors. *IEEE Transactions on Visualization and Computer Graphics (to appear)*, 2010.

[SNAHH11] Jaya Sreevalsan-Nair, Cornelia Auer, Bernd Hamann, and Ingrid Hotz. Eigenvector-based interpolation and segmentation of 2d tensor fields. In *Topological Methods in Data Analysis and Visualization*, pages 139–150. Springer, 2011.

[SPP04] Filip Sadlo, Ronald Peikert, and Etienne Parkinson. Vorticity Based Flow Analysis and Visualization for Pelton Turbine Design Optimization. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 179–186, 2004.

[SWC+08] D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. Interactive Comparison of Scalar Fields Based on Largest Contours with Applications to Flow Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1475–1482, 2008.

[SWTH07] J. Sahner, T. Weinkauf, N. Teuber, and H.-C. Hege. Vortex and strain skeletons in eulerian and lagrangian frames. *IEEE Transactions on Visualization and Computer Graphics (Vis07)*, 13(5):980–990, September - October 2007.

[SYI96] Kenji Shimada, Atsushi Yamada, and Takayuki Itoh. Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles. In *In 6th International Meshing Roundtable*, pages 375–390, 1996.

[Tel08] Alexandru C. Telea. *Data Visualization: Principles and Practice*. AK Peters, Ltd., 2008.

[TGS05] Xavier Tricoche, Christoph Garth, and Gerik Scheuermann. A fast and robust method for visualizing separation line features. In G.P. Bonneau, T. Ertl, and G.M. Nielson, editors, *Scientific Visualization: The Visual Extraction of Knowledge from Data*, pages 249–264. Springer Berlin, 2005.

[TKW08] Xavier Tricoche, Gordon L. Kindlmann, and Carl-Fredrik Westin. Invariant crease lines for topological and structural analysis of tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1627–1634, 2008.

[TLHD03] Yiying Tong, Santiago Lombeyda, Anil N. Hirani, and Mathieu Desbrun. Discrete Multiscale Vector Field Decomposition. *ACM Transactions on Graphics (TOG) (Siggraph'03)*, 22(3):445–452, 2003.

[Tri02] Xavier Tricoche. *Vector and Tensor Field Topology Simplification, Tracking and Visualization*. PhD thesis, University of Kaiserslautern, April 2002.

[TSH01] Xavier Tricoche, Gerik Scheuermann, and Hans Hagen. Continuous topology simplification of planar vector fields. In *VIS '01: Proceedings of the conference on Visualization '01*, pages 159–166, Washington, DC, USA, 2001. IEEE Computer Society.

[TSHC01] Xavier Tricoche, Gerik Scheuermann, Hans Hagen, and Stefan Clauss. Vector and tensor field topology simplification on irregular grids. In D. Ebert, J. M. Favre, and R. Peikert, editors, *VisSym '01: Proceedings of the symposium on Data Visualization 2001*, pages 107–116. Springer-Verlag, May 28–30 2001.

[Tuf86] Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, USA, 1986.

[TvW99] Alexandru Telea and Jarke J. van Wijk. Simplified representation of vector fields. In *VIS '99: Proceedings of the conference on Visualization '99*, pages 35–42, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[Vin98] P. Vincent. *Application of SAR Interferometry to Low-rate Crustal Deformation Fields*. PhD thesis, University of Colorado, 1998.

[Wan04] Zhizhou Wang. *Diffusion Tensor Field Restoration and Segmentation*. PhD thesis, University of Florida, 2004.

[WB05] Andrew Wilson and Rebecca Brannon. Exploring 2d tensor fields using stress nets. In *VIS '05: Proceedings of the conference on Visualization '05*. IEEE Computer Society Press, 2005.

[WH07] Y. Weldeselassie and G. Hamarneh. DT-MRI segmentation using graph cuts. In *Medical Imaging 2007: Image Processing. SPIE*, 2007.

[Wil96] C.H.K. Williamson. Vortex Dynamics in the Cylinder Wake. *Annu. Rev. Fluid Mech.*, 28:477–539, 1996.

[WTS+07] Alexander Wiebel, Xavier Tricoche, Dominic Schneider, Heike Jänicke, and Gerik Scheuermann. Generalized streak lines: Analysis and visualization of boundary induced vortices. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):735–1742, 2007.

[WW05] Joachim Weickert and Martin Welk. Tensor field interpolation with pdes. In Joachim Weickert and Hans Hagen, editors, *Visualization and Processing of Tensor Fields*, Mathematics and Visualization, chapter 19, pages 315–324. Springer, 2005.

[ZHT07] Eugene Zhang, James Hays, and Greg Turk. Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):94–107, 2007.

[ZMB+03] Leonid Zhukov, K. Museth, D. Breen, Ross Whitaker, and Alan Barr. Level set modeling and segmentation of DT-MRI brain data. *Journal Electronic Imaging*, 12(1):125–133, 2003.

[ZMT06] Eugene Zhang, Konstantin Mischaikow, and Greg Turk. Vector field design on surfaces. *ACM Trans. Graph.*, 25(4):1294–1326, 2006.

[ZP03] Xiaoqiang Zheng and Alex Pang. "HyperLIC". In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pages 33–, Washington, DC, USA, 2003. IEEE Computer Society.

[ZP04] Xiaoqiang Zheng and Alex Pang. Topological lines in 3d tensor fields. In *VIS '04: Proceedings of the conference on Visualization '04*. IEEE Computer Society Press, 2004.

[ZPP05] Xiaoqiang Zheng, Beresford N. Parlett, and Alex Pang. Topological lines in 3d tensor fields and discriminant hessian factorization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):395–407, 2005.

[ZTW06] Ulas Ziyan, David Tuch, and Carl-Fredrik Westin. Segmentation of thalamic nuclei from dti using spectral clustering. In *Ninth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'06)*, pages 807–814, 2006.

[ZYLL09] Eugene Zhang, Harry Yeh, Zhongzang Lin, and Robert S. Laramee. Asymmetric tensor analysis for flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):106–122, 2009.

SELBSTSTÄNDIGKEITSERKLÄRUNG

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Berlin, March 2013

Cornelia Auer