Three-Dimensional Second-Order Tensor Fields: Exploratory Visualization and Anisotropic Sampling

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

Andrea Kratz

Berlin, 2013

Erstgutachterin:	Dr. Ingrid Hotz
	Zuse-Institut Berlin
Zweitgutachter:	Prof. Dr. Markus Hadwiger
	King Abdullah University of Science and Technology (KAUST)
Tag der Disputation:	02.05.2013

To B.

Abstract

Tensors provide a powerful mathematical language to describe physical phenomena. Consequently, they have a long tradition in physics and appear in various application areas, either as intermediate product or as output of simulations or measurements. The potential of tensors to describe complex anisotropic behavior, however, concurrently complicates their interpretation. The central research question of this thesis is how three-dimensional tensor fields of second order are visualized effectively so that, as a long term goal, their interpretation becomes easier. The focus of this thesis lies on the class of indefinite tensors.

The methods that are proposed in this thesis fall into two main categories: (1.) the interactive exploration of the three-dimensional tensor data, and (2.) the geometric reduction of the data to two-dimensional planes or triangulated surfaces. In both cases, possible visualization approaches are presented. For interactive exploration of the data, we propose to combine diagram views with three-dimensional hybrid visualizations. We show that this facilitates familiarizing with the data and leads to exciting analytic queries. If a geometric data reduction is possible, we focus on glyph- and texture-based methods. In this context, the thesis is concerned with methods to improve their quality. Therefore, we propose two algorithms for the efficient creation of anisotropic sample distributions. Moreover, we present a novel visualization method that works on planar slices as well as on triangulated surfaces. The basic idea of this method is to use anisotropic sample distributions for the efficient computation of anisotropic Voronoi cells, which then are used as base elements for texture mapping. Hence, the usage of textures to encode the tensor's various degrees of freedom becomes possible.

We evaluate our methods for the interactive exploration on stress tensor fields from structure simulations. To show the effectiveness of novel visualization methods, various datasets are presented.

Zusammenfassung

Tensoren stellen ein mächtiges mathematisches Konzept dar, welches sich zur Beschreibung einer Vielzahl physikalischer Phänomene eignet. Infolgedessen haben Tensoren eine lange Tradition in der Physik und treten in zahlreichen Anwendungsgebieten, entweder als Zwischenprodukt oder Ergebnis von Simulationen und Messungen, auf. Das Potential von Tensoren, komplexes anisotropes Verhalten zu beschreiben, erschwert jedoch zugleich deren Interpretation. Die zentrale Forschungsfrage dieser Arbeit ist, wie sich dreidimensionale Tensorfelder zweiter Ordnung effektiv visualisieren lassen, um auf lange Sicht deren Interpretation zu erleichtern. Der Fokus liegt hierbei auf der Klasse von indefiniten Tensoren.

Um dies zu erreichen, werden in der Arbeit zwei wesentliche Richtungen vorgeschlagen: die interaktive Exploration von dreidimensionalen Tensorfeldern und die geometrische Reduktion der Daten auf zweidimensionale Schnitte oder triangulierte Oberflächen. In beiden Fällen werden mögliche Visualisierungsansätze vorgestellt. Zur interaktiven Exploration der Daten schlagen wir die Kombination von Diagrammansichten und dreidimensionalen hybriden Ansichten vor. Wenn eine geometrische Reduktion der Daten möglich ist, verfolgen wir Glyphen- und texturbasierte Visualisierungsmethoden. Hier stellt die Arbeit insbesondere Methoden zu deren qualitativen Verbesserung mithilfe von anisotropen Samplingmethoden vor. Dazu wurden zwei Algorithmen zur Berechnung von anisotropen Samplingmethoden entwickelt, die sowohl auf planaren Schnitten als auch auf triangulierten Oberflächen anwendbar sind. Darüberhinaus wird eine neue Visualisierungsmethode präsentiert, die auf Schnitten und triangulierten Oberflächen arbeitet. Diese nutzt die mit den zuvor präsentierten Algorithmen erstellten anisotropen Samplingverteilungen für die effiziente Berechnung von anisotropen Voronoizellen, welche dann als Grundelement zur Texturierung verwendet werden können. Auf diese Weise wird die Verwendung von Texturen zur Kodierung der vielen im Tensor enthaltenen Informationen ermöglicht.

Wir evaluieren unsere Methoden zur interaktiven Exploration anhand von Stresstensorfeldern aus Struktursimulationen. Die neuen Visualisierungsansätze werden anhand vieler unterschiedlicher Beispieldatensätze präsentiert.

Publications

This thesis is based on the following publications:

Andrea Kratz, Daniel Baum, Ingrid Hotz,
Anisotropic Sampling of Planar and Two-Manifold Domains for Texture Generation and Glyph Distribution,
IEEE Transactions on Visualization and Computer Graphics (TVCG) 2013, 2013,
Article first published online: 16 May 2013.

Andrea Kratz, Cornelia Auer, Markus Stommel, Ingrid Hotz,
Visualization and Analysis of Second-Order Tensors: Moving Beyond the Symmetric Positive-Definite Case,
Computer Graphics Forum 2013, 2013, Issue 1, pp. 49-74,
Article first published online: 19 Nov 2012.

Andrea Kratz, Nino Kettlitz, Ingrid Hotz,
Particle-Based Anisotropic Sampling for Two-Dimensional Tensor Field Visualization,
Proceedings of Vision, Modeling, and Visualization (VMV) 2011, 2011, pp.145-152

Andrea Kratz, Björn Meyer, Ingrid Hotz,
A Visual Approach to Analysis of Stress Tensor Fields,
Scientific Visualization: Interactions, Features, Metaphors,
Schloss Dagstuhl–Leibniz-Zentrum für Informatik 2011, 2011, pp. 188-211.

This thesis is also related to the following poster:

Andrea Kratz, Markus Hadwiger, Ingrid Hotz,
Improved Visual Exploration and Hybrid Rendering of Stress Tensor Fields via Shape-Space Clustering,
Poster presentation at the IEEE VisWeek 2011, Providence, 2011.

Acknowledgements

This work has been carried out at the Zuse Institute Berlin. It was partially funded by the Emmy Noether research program of the German research foundation (DFG).

I wish to state my sincere gratitude to all those who made this work possible.

First and foremost I would like to thank my advisor Ingrid Hotz. Thanks for your trust, your inspiration, your creative thoughts and for sharing your knowledge, not only in the field of tensor visualization, with me. I would also particularly like to thank Daniel Baum. Daniel, thanks for your steady support and encouragement during the last years. To keep it short: Thank you so much! Markus Hadwiger accompanied my studies in the last years; he also enabled me to spend three months at the King Abdullah University of Science and Technology (KAUST). During this inspiring research period many important ideas of this thesis evolved. I am very thankful for this. Moreover, I am grateful for having written this thesis in a surrounding where I could not only find wonderful colleagues but also valuable friendships: the department of visualization and data analysis of the Zuse Institute Berlin. Special thanks go to Cornelia Auer and Dagmar Kainmüller for an always open ear and for having a great time together! Steffen Prohaska and Morgan Leborgne have supported the last months of writing this thesis by providing me the necessary time to finish it. Thank you for this! Many results that are presented in this thesis, were also supported by the work of two former students: Nino Kettlitz und David Bressler. Thank you for this valuable contribution. I would also like to thank our cooperation partners of Saarland University for providing data sets and for their constant curiosity: Markus Stommel und Marc Schöneich.

It has been a pleasure working with all of you. Thank you.

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Scope . . . 2
1.3 Problems and Objectives . . . 2
1.4 Contributions . . . 4
1.5 Structure . . . 6

2 Tensor Fundamentals **9**
2.1 Introduction . . . 9
2.2 Basic Notations . . . 10
2.3 Tensor Definition . . . 10
2.4 Tensor Invariance . . . 11
2.5 Diagonalization . . . 12
2.6 Tensor Properties and Operations . . . 13
2.7 Tensor Invariants . . . 16
2.8 Tensor Fields and Interpolation . . . 16
2.9 Tensor Decompositions . . . 17
2.10 Projection Tensor . . . 18
2.11 Example Application: Mechanical Engineering . . . 19

3 Categorization and Analysis of Previous Work **23**
3.1 Introduction . . . 23
3.2 Segmentation and Topology Extraction . . . 24
3.2.1 Challenges . . . 24
3.2.2 Similarity-Measure-Based Segmentation . . . 24
3.2.3 Topology-Based Segmentation . . . 26
3.2.4 Discussion . . . 28
3.3 Local Tensor Visualization Methods . . . 29
3.3.1 Challenges . . . 29
3.3.2 Glyph Design . . . 30

3.3.3 Mapping and Normalization . . . 32
3.3.4 Glyph Design based on a Parameterizable Space . . . 33
3.3.5 Glyphs Applied in Mechanical Engineering . . . 34
3.3.6 Glyph Placement . . . 36
3.3.7 Discussion . . . 38
3.4 Global Tensor Visualization Methods . . . 38
3.4.1 Challenges . . . 39
3.4.2 Scalar Visualization Methods . . . 39
3.4.3 Vector Visualization Methods . . . 41
3.4.4 Texture-Based Methods . . . 41
3.4.5 Discussion . . . 44
3.5 Multiple Views for Tensor Visualization . . . 45
3.5.1 Challenges . . . 45
3.5.2 Fiber Exploration . . . 45
3.5.3 Stress Exploration . . . 45
3.5.4 Discussion . . . 46
3.6 Discussion . . . 46

4 A Multi-Perspective Approach to Visual Analysis of Stress Tensor Fields 49
4.1 Introduction . . . 49
4.2 Related Work . . . 51
4.3 Feature Spaces for Tensor Fields . . . 52
4.3.1 Shape Space . . . 52
4.3.2 Direction Space . . . 54
4.3.3 Example Feature Spaces for Stress Tensors . . . 54
4.4 A Framework for the Exploration of Tensor Fields . . . 56
4.4.1 Overview . . . 57
4.4.2 Diagram Views . . . 58
4.4.3 Hybrid Views . . . 65
4.4.4 Shape-Space Clustering . . . 66
4.4.5 Results . . . 69
4.5 Example Application . . . 77
4.5.1 Background . . . 77
4.5.2 Common Practice . . . 77
4.5.3 Insights . . . 78
4.6 Discussion . . . 86

5 Anisotropic Sampling in Planar and Two-Manifold Domains 89
5.1 Introduction . . . 89
5.2 Related Work . . . 91

5.3 Underlying Concepts and Basic Ideas . . . 93
5.3.1 Notations . . . 93
5.3.2 Metric Tensor Fields . . . 93
5.3.3 Isotropic vs. Anisotropic Space . . . 95
5.3.4 Anisotropy Design . . . 95
5.3.5 Anisotropic Triangulations . . . 96
5.3.6 Triangle Fillrate . . . 101
5.4 Particle-Based Approach . . . 102
5.4.1 Overview . . . 102
5.4.2 Initial Sampling . . . 103
5.4.3 Particle Movement . . . 104
5.4.4 Automatic Insert and Delete Operations . . . 107
5.4.5 Results and Applications . . . 107
5.4.6 Discussion . . . 110
5.5 Triangle-Based Approach . . . 112
5.5.1 Overview . . . 112
5.5.2 Initial Sampling . . . 113
5.5.3 Gravitational-Centered Relaxation . . . 114
5.5.4 Back-Projection . . . 116
5.5.5 Volume Slicing . . . 116
5.5.6 Results and Applications . . . 117
5.5.7 Discussion . . . 120

6 Anisotropic Voronoi Cell Rendering 123
6.1 Introduction . . . 123
6.2 Anisotropic Voronoi Diagrams . . . 124
6.3 Rendering and Texturing . . . 126
6.3.1 Pre-Processing and Data Structures . . . 126
6.3.2 Voronoi Cell Computation . . . 128
6.3.3 Texturing of Voronoi Cells . . . 129
6.4 Results and Discussion . . . 129

7 Conclusion and Outlook 137
7.1 Summary . . . 137
7.2 Future Work . . . 138
7.3 Concluding Remarks . . . 139

A Datasets 141
A.1 Two-Point Load . . . 141
A.2 One-Point Load . . . 141
A.3 Rotating Neutron Star . . . 141

A.4 Aneurysm 142
A.5 Cube 143
A.6 Beam Profile 143
A.7 Shear Specimen 144

Bibliography **145**

1 Introduction

> *Tensor mathematics is a beautiful, simple, and useful language for the description of natural phenomena.* [Danielson, 1997]

> *Visualization is fundamental to understanding models of complex phenomena [...].* [Munzner et al., 2006]

This thesis deals with the visualization and analysis of three-dimensional (3D) tensor fields of second order. We present a concept for the visual exploration of these fields. Furthermore, we propose two algorithms to generate anisotropic sample distributions to improve glyph- and texture-based tensor visualization methods. We also show how these sample distributions can be exploited for a novel visualization method for tensor fields that works on two-manifold domains. The goal of this chapter is to motivate our research, discuss the major challenges addressed in this thesis, and to list our main contributions. Finally, an outline of the whole thesis is given.

1.1 Motivation

The major motivation of this thesis was to develop visualization and analysis methods to investigate 3D tensor fields. Tensors are mathematical entities that describe linear relationships between other tensors of zeroth order (scalars), first order (vectors) or arbitrary order (tensors). Second-order tensors, which are the focus of this thesis, are special cases of matrices. They are suited to describe any kind of anisotropic behavior. Tensor fields result from simulations or measurements, either as intermediate product or as the final result. Examples for applications in which tensors play an important role include solid mechanics, continuum mechanics, astrophysics and medicine.

The interpretation of tensor fields is difficult since they encode much information and because the physical meaning of this information is highly application-specific.

Here, scientific visualization can provide insight into the processes that are described by tensors. This has been shown in previous work within the context of diffusion tensor magnetic resonance imaging (DT-MRI) [Vilanova et al., 2005]. In other applications, for example engineering applications, visualization and analysis tools specifically designed for tensor fields are rare. Common practice in engineering applications is to analyze only scalar quantities that were derived from the tensor and to discard the remaining tensor information. But, tensors contain much more information than what is represented by a single scalar value. We believe that the analysis of the whole tensor can lead to deeper insights into the simulations themselves, as well as into the physical phenomena that are simulated.

1.2 Scope

This thesis is concerned with the visualization of tensors as a certain *class of data*. Focus is the class of indefinite tensors, which still are an underrepresented topic in visualization. The thesis has no specific application context. However, most examples are from engineering applications, where indefinite tensors play an important role. Diffusion tensors, the most often encountered example of the class of positive-definite tensors, will not be covered in this thesis. Of course, whenever possible, we try to benefit from the advances made in this field.

1.3 Problems and Objectives

Scientific visualization is an interdisciplinary field of science. Therefore, research questions always arise from three different perspectives: the *application side* involving those people (users) that generate the input data and have questions concerning this data, the *algorithmic side* that is concerned with specific technical problems, and the *visualization side* that needs to find suitable depictions to convey complex data in an intuitive way. All these perspectives need to be considered in this thesis.

The most common method to depict tensors is to encode their properties in shape, orientation and color of geometric entities (glyphs), for example, ellipsoids. While glyphs have the potential to represent the whole information that is contained in a tensor, they are often hard to interpret. Moreover, glyphs cannot be used to create effective visualizations of a complete 3D tensor field. In such fields, every data point is assigned to six independent variables that need to be encoded in the glyph. If every data point is visualized by a glyph, the result are overloaded images that hamper the interpretation of the underlying data. The task of depicting 3D tensor fields, therefore, needs to solve a fundamental visualization problem: the avoidance of clutter and occlusion. One goal of this thesis, therefore, is to minimize the data that is visualized so that only relevant information is depicted. Here, *relevant* means the data that is needed to answer a specific analysis question. In scientific visualization, two approaches are often used for this purpose. These are *feature extraction* and *data reduction*.

Feature extraction means the extraction of *meaningful data characteristics*, hereinafter referred to as *feature*. The goal is to reduce the complex input data to a less complex subset that contains only relevant information with respect to a specific question or task. Since the analysis of tensor fields, besides DTI, is a relatively new research topic in scientific visualization as well as in the application fields where tensors appear, a lack of *specific* questions can be observed, not least due to missing visualization and analysis tools. As a consequence, and in contrast to scalar-, vector- and flow visualization, rarely any feature definitions for tensors exist. In engineering applications, for example, the analysis of tensors has been neglected in the past. Up to now, we therefore do not exactly know what we can expect from the tensor data that arises in these fields and what we are looking for in the data. As a consequence, automatic segmentation or feature extraction methods are in general not applicable. For this reason, we have identified the necessity of exploratory visualization, especially for stress tensor fields. Such an *undirected* visualization concept is needed when we do not know what we are looking for [Bergeron, 1993]. In these cases, exploratory visualization can help to identify relevant features and patterns in the data.

By data reduction we mean two things: First, the reduction of the input data to less complex mathematical entities, such as vectors or scalars. This kind of data reduction can also be classified as feature extraction. Second, the geometric reduction of the 3D input data domain to a two-dimensional (2D) subdomain. Such a subdomain is either a cut through the initial 3D domain, hereinafter referred to as *slice*, or a surface. Previous research in tensor visualization has often focused only on slices, mostly because visualizations become more complex in three dimensions and, hence, are more difficult to understand. However, if this geometric reduction is done too early in the analysis process, the change of tensors over the whole field cannot be communicated [Hashash et al., 2003]. Moreover, if the tensor is solely reduced to a single scalar or vector, relevant information might be lost. In both cases, data reduction can lead to misinterpretations if the reduction is already made in the beginning of the analysis process. Therefore, a goal of this thesis is to develop methods to extract specific tensor properties on-the-fly. These include scalar- or vector-valued features that are derived from the tensor as well as the extraction of single slices or surfaces of interest. With this approach, the tensor data is reduced for the visualization but all the information is available at any time.

If the 3D input domain is restricted to a 2D subdomain, glyphs can be used to depict the tensor field. A fundamental question of such local visualization methods, which depict single tensors at discrete points, is, where to place the glyphs to achieve a high-quality visualization result. This becomes especially important if continuous structures or patterns of the data need to be conveyed. Being placed at discrete grid positions, the pattern of the underlying grid becomes visible in the final visualization, which disturbs the perception of the field's continuous behavior. Moreover, overlapping glyphs and holes should be avoided as they lead to visual artifacts and undesired occlusions. As shown in previous work [Kindlmann and Westin, 2006; Hlawitschka et al., 2007; Feng et al., 2008], intelligent placement algorithms address this issue. However, previous approaches do not work stable if glyph sizes across the sample domain vary strongly. For indefinite tensor fields, however, such a setting is common, because they cannot be easily normalized. For this reason, one goal of this thesis is to develop methods that generate anisotropic sample distributions that guide the placement of glyphs and are stable even for strongly varying glyph sizes.

1.4 Contributions

This section lists the major contributions of this thesis. They have been published in four publications and one poster, on which this thesis is based:

- A. Kratz and D. Baum and I. Hotz, Anisotropic Sampling of Planar and Two-Manifold Domains for Texture Generation and Glyph Distribution, IEEE Transactions on Visualization and Computer Graphics, 2013

- A. Kratz and C. Auer and M. Stommel and I. Hotz, Visualization and Analysis of Second-Order Tensors: Moving Beyond the Symmetric Positive-Definite Case, Computer Graphics Forum, 2013

- A. Kratz and N. Kettlitz and I. Hotz, Particle-Based Anisotropic Sampling for Two-Dimensional Tensor Field Visualization, Proceedings of Vision, Modeling, and Visualization, 2011

- A. Kratz and B. Meyer and I. Hotz, A Visual Approach to Analysis of Stress Tensor Fields, Scientific Visualization: Interactions, Features, Metaphors, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2011

- A. Kratz, M. Hadwiger and I. Hotz, Improved Visual Exploration and Hybrid Rendering of Stress Tensor Fields via Shape-Space Clustering, Poster presentation at the IEEE VisWeek 2011, Providence, 2011

Categorization and analysis of previous work This thesis provides an overview of the state of the art in tensor visualization (Chapter 3). In contrast to most previous work that focuses on diffusion tensors in medical applications (see, for example, [Vilanova et al., 2005]), our review focuses on engineering applications and tensors that are not symmetric positive definite; diffusion tensors are excluded.

Visual data exploration In order to address the need for the visual analysis of tensor fields, this thesis proposes a concept that adapts the idea of *multiple linked views* to tensor fields, where statistical plots and spatial depictions are presented side-by-side (Chapter 4). Interactive selections of specific data properties that are highlighted simultaneously in all views (brushing-and-linking) allow first insights into the data. Although tensors have a multivariate nature [Vilanova et al., 2005], the adaption of methods from multivariate data analysis is not straightforward. The major difference of tensor data and multivariate data is that the individual tensor components are not independent, whereas individual components of multivariate data are independent in the sense that they are not linked via transformation rules. In this work, the application of a multi-perspective approach to tensor fields was achieved by interpreting the *shape space* (compare the work of [Bahn, 1999]) as feature space. In our framework, a shape space is a scalar-valued feature space that serves as a link between the abstract tensor and its depiction in diagrams and 3D spatial visualizations. With this approach, the following is achieved:

- The difficulty to depict all the degrees of freedom of a second-order tensor in a single depiction is solved by distributing this information in various views (diagrams and spatial depictions), which all are presented side-by-side.
- The fact that we do not yet know what we can expect from the data is faced by the concept of brushing-and-linking, which enables visual data exploration.
- The problem of clutter and occlusion in spatial depictions is addressed via hybrid rendering that only depicts selected tensor properties.

Anisotropic sampling To improve the visual quality of glyph- and texture-based tensor visualization methods, we have developed two approaches for the generation of anisotropic sample distributions.

The first algorithm (Section 5.4) is a particle-based method for the generation of anisotropic sample distributions on planar domains. In contrast to previous methods [Kindlmann and Westin, 2006; Hlawitschka et al., 2007], our approach can handle samples that strongly vary in size across a given domain. We show the applicability of the resulting sample distribution for glyph placement and as *noise image* to improve texture-based visualization methods that are similar to *line integral convolution*

(LIC) [Cabral and Leedom, 1993].

In order to extend the sampling to work also on two-manifold domains, we developed an algorithm that solely depends on an anisotropic triangulation of the domain to be sampled (Section 5.5). To create stable results even if sample sizes and sample anisotropies vary greatly within a given domain, we propose the use of an anisotropic triangulation that serves as basis for the creation of an initial sample distribution as well as for a gravitational-centered relaxation. Thus, this approach is not only an extension to two-manifold domains. It also significantly speeds up the generation of anisotropic sample distributions for the planar case compared to previous approaches and, thus, enables interactive slicing through tensor volumes. To demonstrate the potential of our approach, we present results for several examples covering the planar as well as the two-manifold case.

Texturing of anisotropic Voronoi cells To depict tensor fields on arbitrary 2D domains, this thesis presents a novel visualization method (Chapter 6). Based on anisotropic sample distributions, we present an interactive rendering approach that depicts surface tensor fields in terms of anisotropic Voronoi cells. Through this approach, the use of textures to encode tensor properties becomes possible. We show that textures provide many possibilities to design tensor visualizations and to *explore* tensor fields using texturing. We present several examples including different types of tensor fields to demonstrate the flexibility of this visualization method.

1.5 Structure

Following this introduction, Chapter 2 presents the mathematical fundamentals of second-order tensors that are needed for the subsequent chapters. Chapter 3 reviews and categorizes previous work. It covers methods for the segmentation of tensor fields (Section 3.2) as well as local (Section 3.3) and global visualization methods (Section 3.4) and finally multiple view systems for tensors (Section 3.5). This is a review and categorization of state-of-the-art tensor visualization methods with a focus on tensors that are not positive definite. Diffusion tensors are not covered in this review. Chapter 4 concentrates on methods for the visual data exploration of tensor fields. It introduces the basic framework that implements multiple linked views for tensor fields and that was developed within the scope of this thesis (Section 4.4). Chapter 5 presents two approaches for generating anisotropic sample distributions on planar domains (Section 5.4) and two-manifold domains (Section 5.5). The main part of this thesis ends with the presentation of a novel visualization method for tensor fields on two-manifold domains (Chapter 6). The thesis concludes with a summary of the presented methods and a discussion of future work in Chapter 7.

2
Tensor Fundamentals

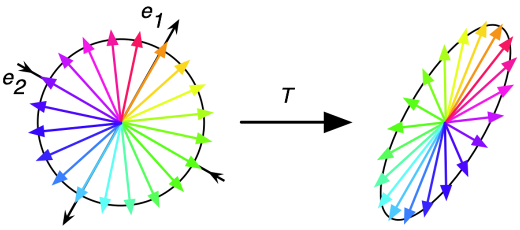


Figure 2.1: Illustration of a symmetric second-order tensor as linear operator. The tensor is uniquely determined by its action on all unit vectors (left). The eigenvector directions are highlighted as black arrows. In this example, one eigenvalue (λ_2) is negative. As a consequence, all vectors are mirrored at the axis spanned by eigenvector e_1. The eigenvectors are the directions with strongest normal deformation but no directional change.

2.1 Introduction

This chapter introduces the mathematical fundamentals of second-order tensors that are needed for the following chapters. The relevant fundamentals and formulas are provided with respect to the goal of this thesis, which is the *visualization* of tensor fields. More comprehensive introductions to tensors in general are found in the following textbooks: A concise mathematical introduction to tensor calculus is given in [Hagen and Garth, 2006]. A more detailed introduction from the viewpoint of physics and engineering can be found in [Danielson, 1997; Brannon, 2003]. As a reference for a solid mathematical treatment of the topic see [Abraham et al., 1988] and the more recent book [Landsberg, 2012]. Parts of this chapter are also published in [Kratz et al., 2013a].

2.2 Basic Notations

This thesis uses the following notations: Tensors and matrices are denoted using bold capital letters, for example, $\mathbf{T}$ and $\mathbf{M}$. If not stated otherwise, $\mathbf{T}$ describes a second-order tensor and $\mathbf{M}$ a three-dimensional (3D) matrix, that is, $\mathbf{M} \in \mathbb{R}^{3\times 3}$. The identity tensor is $\mathbf{I}$ and the capital letter V denotes a vector space. Vectors are denoted as bold lower-case letters, for example, $\mathbf{v}$. Small letters denote scalars, for example, α or n, and small letters with subscript $_{ij}$ denote matrix components, for example, m_{ij}. The letter $\mathbf{e}$ denotes eigenvectors and the small Greek letter λ eigenvalues.

2.3 Tensor Definition

This section starts with a general definition of tensors as a multi-linear map (Equation (2.1)). This is the most general definition for tensors that encompasses every following definition. However, there are various valid definitions for tensors, which mainly differ with respect to the context in which they appear. The goal of this section is to briefly introduce the most common definitions and to present the different perspectives on tensors. The focus are second-order tensors.

Tensor as a multi-linear map Let V be a n-dimensional vector space over $\mathbb{R}$, and let V^* be its dual space defined as the space of all linear maps $f : V \to \mathbb{R}$. The operator $\otimes$ is the tensor product (Equation (2.19)). Then, a multilinear map from q copies of V and p copies of its dual space V^* into the space $\mathbb{R}$ of real numbers

$$\mathbf{T} : \underbrace{V^* \otimes \ldots \otimes V^*}_{p} \otimes \underbrace{V \otimes \ldots \otimes V}_{q} \to \mathbb{R} \tag{2.1}$$

is called a tensor $\mathbf{T}$ of order $r = p + q$; covariant order of p and contravariant order of q. In this thesis, we only deal with finite-dimensional Euclidean vector spaces. Then, the dual space V^* can be identified with V and it is not necessary to distinguish co- and contravariant order [Abraham et al., 1988] [1]. In the following, we only consider tensors of order $r = 2$, which are the focus of this thesis.

Second-order tensor as a bilinear map A second-order tensor $\mathbf{T}$ is defined as a bilinear function from two copies of a vector space V into the space $\mathbb{R}$ of real numbers

$$\mathbf{T} : V \otimes V \to \mathbb{R}. \tag{2.2}$$

[1] Note, that in the more general case of non-Euclidean spaces (e.g., curved surfaces), the distinction between covariant (upper) indices and contravariant (lower) indices is necessary.

Second-order tensor as a linear operator Another perspective defines a second-order tensor $\mathbf{T}$ as linear operator that maps any vector $\mathbf{v} \in V$ onto another vector $\mathbf{w} \in V$. Here, the tensor is a map from the vector space V onto itself:

$$\mathbf{T} : V \to V. \tag{2.3}$$

The definition of a tensor as a linear operator is prevalent in physics. See Figure 2.1 for a 2D illustration.

Second-order tensor as a matrix In computer science, tensors are generally represented with respect to a specific Cartesian basis $\{\mathbf{b}_1, \dots, \mathbf{b}_n\}$ of the vector space V. In this case, the tensor is uniquely defined by its components and is represented as a matrix. Thus, in matrix notation, a 3D tensor ($n = 3$) of order $r = 2$ is represented by a matrix $\mathbf{M} \in \mathbb{R}^{3\times 3}$ with $n^r = 9$ components $(m_{ij})_{i=1,\dots 3, j=1\dots 3}$. Considering definition (2.2), we have

$$\mathbf{T}(\mathbf{v}, \mathbf{w}) = \mathbf{w}^T \cdot \mathbf{M} \cdot \mathbf{v} = \sum_{i,j=1}^{n} m_{ij} w_i v_j, \quad \forall \mathbf{v}, \mathbf{w} \in V, \tag{2.4}$$

with $\mathbf{v} = \mathbf{v}_1\mathbf{b}_1 + \dots + \mathbf{v}_n\mathbf{b}_n$ and $\mathbf{w} = \mathbf{w}_1\mathbf{b}_1 + \dots + \mathbf{w}_n\mathbf{b}_n$. Here '$\cdot$' denotes the standard matrix multiplication. Considering definition (2.3), we have

$$\mathbf{T}(\mathbf{v}) = \mathbf{M} \cdot \mathbf{v}, \quad \forall \mathbf{v} \in V. \tag{2.5}$$

Please note that the matrix representation assumes a Cartesian coordinate system.

2.4 Tensor Invariance

The representation of a tensor as a matrix is prevalent in computer science and, hence, also in the field of visualization. However, the strength of tensor mathematics is that tensors are independent of specific reference frames, that is, they are *invariant* under coordinate transformations. More precisely, the tensor components change according to the transformation into another basis but the characteristics of the tensor are preserved. Consequently, tensors can be analyzed using any convenient reference frame. Invariance qualifies tensors to describe physical processes independent of the coordinate system.

In the following, we assume that a reference frame is given and mainly use matrix notation. Nevertheless, the reader should still keep in mind that the following concepts are independent from the chosen frame of reference. Furthermore, from now on the term *tensor* is used equivalently to *tensor of second order.*

When we talk about *invariant quantities* (see also Section 2.7), we always mean *rotational invariance.* Affine invariance will be denoted explicitly.

2.5 Diagonalization

A tensor is independent of any chosen frame of reference. For specific reference frames, however, the tensor representation becomes especially simple. Symmetric tensors can be represented as diagonal matrices [2]. The basis for such a representation is given by the eigenvectors corresponding to the diagonal matrix. A vector $\mathbf{e}$ is called eigenvector of $\mathbf{T}$ according to the eigenvalue λ if $\mathbf{Te} = \lambda\mathbf{e}$. The eigenvalues are defined as the roots of the *characteristic polynomial* (Equation (2.9)). The complete transformation of $\mathbf{T}$ from an arbitrary basis into the eigenvector basis is given by

$$\mathbf{U}\;\mathbf{T}\;\mathbf{U}^T = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}. \tag{2.6}$$

The diagonal elements $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues and $\mathbf{U}$ is the orthogonal matrix that is composed of the eigenvectors, that is

$$\mathbf{U} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3). \tag{2.7}$$

$\mathbf{U}^T$ defines the transposed matrix of $\mathbf{U}$. For symmetric tensors, the eigenvalues are all real, and the eigenvectors constitute an orthonormal basis. The diagonalization generally is computed numerically via *singular value decomposition* (SVD) or *principal component analysis* (PCA). Note, that in the following we assume a descending ordering of the eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \lambda_3. \tag{2.8}$$

Characteristic polynomial A scalar $\lambda \in \mathbb{R}$ is eigenvalue of $\mathbf{T}$ if the characteristic polynomial is equal to zero

$$\det(\mathbf{T} - \lambda\mathbf{I}) = 0. \tag{2.9}$$

For 3D symmetric tensors, it is a polynomial of order three given as

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0. \tag{2.10}$$

Characteristic invariants The coefficients I_1, I_2, I_3 are called *characteristic invariants*

$$\begin{aligned} I_1 &= & \mathrm{tr}(\mathbf{T}) &= & \lambda_1 + \lambda_2 + \lambda_3 \\ I_2 &= & \frac{1}{2}((\mathrm{tr}(\mathbf{T}))^2 - \mathrm{tr}(\mathbf{T}^2)) &= & \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1 \\ I_3 &= & \det(\mathbf{T}) &= & \lambda_1\lambda_2\lambda_3. \end{aligned} \tag{2.11}$$

[2] Asymmetric tensors can have complex eigenvalues for which no real eigenvectors exist (see, for example, [Zhang et al., 2008]).

For 3D symmetric tensors, the characteristic polynomial has $n = 3$ real solutions. The multiplicity of an eigenvalue λ determines the dimension of the corresponding eigenvector space $\{\mathbf{e} \in V | \mathbf{Te} = \lambda \mathbf{e}\}$. The eigenvector spaces are mutually orthogonal. Tensors having eigenvalues of higher multiplicity are also called *degenerate* (see also Section 3.2.3). In the case of three distinct eigenvalues, the eigenvectors are determined uniquely and form an orthogonal basis.

2.6 Tensor Properties and Operations

This section provides the tensor properties and operations, which are needed for the following chapters, and which play an important role throughout the whole thesis.

Tensor Properties

The following list introduces the most relevant mathematical properties of second-order tensors. The physical interpretation of these properties varies between application areas. Section 2.11, therefore, discusses these properties and their meaning in the context of engineering applications.

Symmetric tensors A tensor $\mathbf{S}$ is called *symmetric* if it is invariant under permutations of its arguments

$$\mathbf{S}(\mathbf{v}, \mathbf{w}) = \mathbf{S}(\mathbf{w}, \mathbf{v}) \quad \forall \mathbf{v}, \mathbf{w} \in V. \tag{2.12}$$

Written in matrix notation, $s_{ij} = s_{ji}$ for any coordinate basis.

Antisymmetric tensors A tensor A is called *antisymmetric* or *skew-symmetric* if the sign flips when two adjacent arguments are exchanged

$$\mathbf{A}(\mathbf{v}, \mathbf{w}) = -\mathbf{A}(\mathbf{w}, \mathbf{v}) \quad \forall \mathbf{v}, \mathbf{w} \in V. \tag{2.13}$$

Written in matrix notation, $a_{ij} = -a_{ji}$ for any coordinate basis. From this, it follows $a_{ii} = 0$.

Traceless tensors The trace of a tensor $\mathrm{tr}(\mathbf{T})$ is defined as the sum of its diagonal components (Equation (2.18)). Tensors $\mathbf{T}$ with zero trace, i.e. $\mathrm{tr}(\mathbf{T}) = 0$, are called *traceless*.

Positive (semi-) definite tensors A tensor $\mathbf{T}$ is called positive (semi-) definite for any non-zero vector $\mathbf{v} \in V$ if

$$\mathbf{T}(\mathbf{v}, \mathbf{v}) > (\geq) \, 0. \tag{2.14}$$

It follows that for positive (semi-) definite tensors their eigenvalues and their determinant are greater than (greater than or equal to) zero.

Negative (semi-) definite tensors A tensor $\mathbf{T}$ is called negative (semi-) definite for any non-zero vector $\mathbf{v} \in V$ if

$$\mathbf{T}(\mathbf{v}, \mathbf{v}) < (\leq) \, 0. \tag{2.15}$$

For 3D tensors, it follows that for negative (semi-) definite tensors their eigenvalues and their determinant are smaller than (smaller than or equal to) zero. This generalization does not hold for two-dimensional tensors of second order.

Indefinite tensors Each tensor that is neither positive definite nor negative definite is indefinite. For indefinite tensors, $\mathbf{T}(\mathbf{v}, \mathbf{v})$ results in positive as well as negative values and the eigenvalues have different signs. No statement can be made about the determinant.

In summary, we distinguish tensors that describe a deformation and those that are generators of a deformation. Tensors that describe a deformation are positive definite.

Operations

The following list provides the most important operations for second-order tensors $\mathbf{T} \in \mathbb{R}^{3\times3}$. These can be defined componentwise and are given with respect to matrix representation in the following.

Transpose The transpose of a tensor $\mathbf{T} = (t_{ij})$ is given as

$$\mathbf{T}^T = (t_{ji}). \tag{2.16}$$

For symmetric tensors, thus, $\mathbf{T}^T = \mathbf{T}$,

Determinant The determinant maps a tensor onto a scalar

$$\det(\mathbf{T}) = \sum_{i=1}^{n} (-1)^{i+j} t_{ij} \det(\mathbf{T}_{ij}). \tag{2.17}$$

Here, $\mathbf{T}_{ij} \in \mathbb{R}^{2\times 2}$ is the submatrix of $\mathbf{T}$ obtained by removing the i^{th} row and j^{th} column of $\mathbf{T}$. Note that there is no summation over j, which can be chosen arbitrarily. Matrices of tensors with determinant not equal to zero are invertible. The sign of the determinant is positive if the orientation of any basis is preserved. Geometrically, the determinant's absolute value can be interpreted as the volume ratio of a volume element before and after transformation.

Trace The trace maps a tensor onto a scalar. The trace of a tensor $\mathbf{T}$ is defined as the sum of its diagonal components

$$\mathrm{tr}(\mathbf{T}) = \sum_{i=1}^{n} t_{ii}. \tag{2.18}$$

The determinant and trace are often used as a measure of the tensor's *size*.

Tensor product The tensor product $\otimes$, or dyadic product, maps two tensors $\mathbf{T}_1$ and $\mathbf{T}_2$ of order r_1 respective r_2 onto a tensor $\mathbf{T}_1 \otimes \mathbf{T}_2$ of order $r_1 + r_2$. As example, we consider the identity tensor $\mathbf{I}$, which is the sum of the tensor products of the three unit vectors $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$

$$\mathbf{I} = \mathbf{u}_1 \otimes \mathbf{u}_1 + \mathbf{u}_2 \otimes \mathbf{u}_2 + \mathbf{u}_3 \otimes \mathbf{u}_3. \tag{2.19}$$

Dot product The dot product $\cdot$, or inner product, denotes the consecutive application of two tensors. The dot product between two tensors $\mathbf{T}_1$ and $\mathbf{T}_2$ of order r_1 respective r_2 results in a tensor $\mathbf{T}_1 \cdot \mathbf{T}_2$ with order $r_1 + r_2 - 2$.

In matrix representation, it is given by standard matrix multiplication of $M \cdot N$. The dot product is associative.

Double dot product The double dot product $:$, or double inner product, maps two tensors $\mathbf{T}_1$ and $\mathbf{T}_2$ onto a scalar. It is defined as the sum of the products of their components

$$\mathbf{T}_1 : \mathbf{T}_2 = \sum_{i,j=1}^{n} t_{1ij} t_{2ij} = \mathrm{tr}(\mathbf{T}_1 \cdot \mathbf{T}_2^T), \tag{2.20}$$

where $\mathbf{T}_2^T$ is the transpose matrix (Equation (2.16)) of $\mathbf{T}_2$.

Frobenius norm The tensor norm maps a tensor onto a scalar. The Frobenius norm, which is induced by the inner product (Equation (2.20))

$$\|\mathbf{T}\| = \sqrt{\mathbf{T} : \mathbf{T}} = \sqrt{\sum_{i,j=1}^{n} t_{ij}^2}, \tag{2.21}$$

is the most common norm used for tensors.

2.7 Tensor Invariants

Tensor invariants

$$I = f(\lambda_1, \lambda_2, \lambda_3) \tag{2.22}$$

are defined as scalar quantities that do not change under orthogonal coordinate transformation. They describe tensor characteristics that reflect the fundamental property of tensor invariance (Section 2.4). In general, any scalar function $f(\lambda_1, \lambda_2, \lambda_3)$ that only depends on the eigenvalues again is an invariant. As a consequence, also every scalar function of invariants is an invariant itself. Most common examples for tensor invariants are the tensor's eigenvalues (Equation (2.6)), determinant (Equation (2.17)) and trace (Equation (2.18)), which are, in terms of the eigenvalues, $\operatorname{tr}(\mathbf{T}) = \sum_{i=1}^{3} \lambda_i$ and $\det(\mathbf{T}) = \prod_{i=1}^{3} \lambda_i$, respectively. See also the characteristic invariants given in Equation 2.11.

2.8 Tensor Fields and Interpolation

Tensor fields In visualization, usually not only a single tensor but a whole *tensor field* is of interest. A second-order tensor field over some domain Ω is a map that assigns to every point $x \in \Omega$ a second-order tensor $\mathbf{T}(x)$. In this work, Ω is a subset of the 3D Euclidean space and $\mathbf{T}$ is represented with respect to a specific Cartesian basis.

Interpolation Typically, a tensor field is given as sampled data at discrete positions on various grid types (e.g. rectilinear or tetrahedral grids). However, intrinsically, these data describe a continuous quantity. Therefore, an important step is to reconstruct a continuous function of the sampled signal. For the reconstruction of a continuous function, different interpolation schemes are employed. In this thesis, component-wise linear interpolation is used, because it is still the most stable method, it is applicable to 3D tensor fields, and it is computationally efficient.

However, when using component-wise linear interpolation, some tensor properties might not be preserved. This can be observed in the *swelling effect* (see, for example, [Hotz et al., 2010]). The swelling effect describes the case when an interpolated tensor has a determinant that is larger than the determinant of the original tensors. For example, the interpolation of two tensors of linear shape results in a tensor that exhibits spherical shape.

Advanced tensor interpolation methods have mainly been developed for diffusion tensors with the goal to maintain the directional information contained in the tensor data (e.g., [Moakher and Batchelor, 2006; Fletcher and Joshi, 2007; Lenglet et al., 2006; Pennec et al., 2006]). A consistent interpolation in 2D focusing on the preservation of tensor field topology was introduced in the work of Hotz et al. [2010]. To preserve the determinant of the original tensors and, hence, to avoid the swelling effect, their method builds upon the separation of shape and orientation for the interpolation. A general and efficient solution of their approach for the 3D case, however, is still missing.

Overall, which properties of the tensor are of interest and should be preserved during interpolation heavily depends on the questions addressed by the visualization task. Since no interpolation method has yet been developed that preserves the tensor as a whole, tensor interpolation remains a critical part in the visualization and analysis process of tensor fields. For data from simulations, the most natural choice is to use the interpolation method that was used for the simulation. Unfortunately, these methods are not always known and if so, they are often of higher order and hence conflict with the demand for efficiency. In many cases, however, the resolution of the simulation is high enough such that a multi-linear componentwise interpolation, as it is used in this thesis, is sufficient.

2.9 Tensor Decompositions

The decomposition of tensors into distinctive parts is a common step prior to their analysis. The parts can then be analyzed separately to reveal information that cannot be easily obtained from the original tensor. This section lists the most common tensor decomposition schemes that are used in visualization. It should be noted that some of these decompositions are limited to positive-definite tensors and some to symmetric tensors.

Symmetric/Antisymmetric For non-symmetric tensors $\mathbf{T}$, the decomposition into a symmetric part $\mathbf{S}$ and an antisymmetric part $\mathbf{A}$ is common practice:

$$\mathbf{T} = \mathbf{S} + \mathbf{A}, \tag{2.23}$$

with $s_{ij} = \frac{1}{2}(t_{ij} + t_{ji})$ and $a_{ij} = \frac{1}{2}(t_{ij} - t_{ji})$. Physically, the antisymmetric part contains rotational information and the symmetric part contains information about isotropic scaling and anisotropic shear. The latter part is often subject to further decompositions described below. For more information about antisymmetric and symmetric tensors and their applications, see also Table 2.1.

Stretch/Rotation Another useful decomposition of non-symmetric, positive-definite tensors $\mathbf{T}$ (e.g. deformation gradient tensors) is the *polar decomposition*. It decomposes the transformation represented by $\mathbf{T}$ in a two-stage process: a rotation $\mathbf{R}$ and a right stretch $\mathbf{U}$, or a left stretch $\mathbf{W}$:

$$\mathbf{T} = \mathbf{R} \cdot \mathbf{U} = \mathbf{W} \cdot \mathbf{R} \tag{2.24}$$

A tensor is called *stretch* if it is symmetric and positive definite. A tensor is called *rotation* if it is orthogonal with determinant equal to one.

Shape/Orientation Via eigenanalysis (Equation (2.6)), symmetric tensors are separated into *shape* and *orientation*. Here, shape refers to the eigenvalues and orientation to the eigenvectors. It is important to note that the orientation field is not a vector field due to the bi-directionality of eigenvectors. For the definition of features and for interpolation (Section 2.8), it is often advantageous to consider shape and orientation separately.

Isotropic/Anisotropic Symmetric tensors can be decomposed into an isotropic $\mathbf{T}_{iso}$ and an anisotropic (deviatoric) part $\mathbf{T}_{dev}$

$$\mathbf{T} = \underbrace{\frac{1}{3}\mathrm{tr}(\mathbf{T})\,\mathbf{I}}_{\mathbf{T}_{iso}} + \underbrace{(\mathbf{T} - \mathbf{T}_{iso})}_{\mathbf{T}_{dev}}. \tag{2.25}$$

From a physical point of view, the isotropic part represents a direction independent transformation (e.g., a uniform scaling or uniform pressure); the deviatoric part represents the distortion.

2.10 Projection Tensor

Some steps of the algorithms that are presented in Chapter 5 require the projection of the tensor $\mathbf{T}$ onto the tangent plane of a given surface (Figure 2.2). The tensor $\widehat{\mathbf{T}}$ projected to the surface defined by the surface normal $\mathbf{n}$ is given by

$$\widehat{\mathbf{T}} = \mathbf{P}(\mathbf{n}) \cdot \mathbf{T} \cdot \mathbf{P}^T(\mathbf{n}). \tag{2.26}$$

Here, $\mathbf{P}$ is the *projection tensor* given by

$$\mathbf{P}(\mathbf{n}) = \mathbf{I} - (\mathbf{n} \otimes \mathbf{n}) = \begin{pmatrix} (1-\mathbf{n}_x^2) & -\mathbf{n}_x\mathbf{n}_y & -\mathbf{n}_x\mathbf{n}_z \\ -\mathbf{n}_x\mathbf{n}_y & (1-\mathbf{n}_y^2) & -\mathbf{n}_y\mathbf{n}_z \\ -\mathbf{n}_x\mathbf{n}_z & -\mathbf{n}_y\mathbf{n}_z & (1-\mathbf{n}_z^2) \end{pmatrix}, \tag{2.27}$$

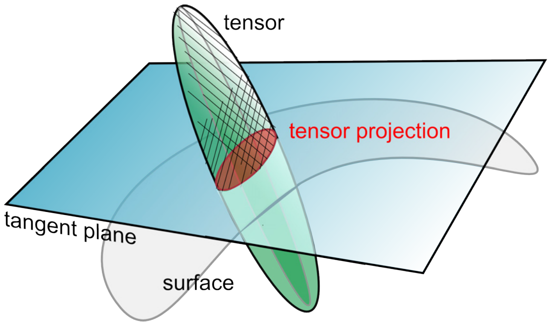


Figure 2.2: Projection of a 3D tensor onto the tangent plane of a surface.

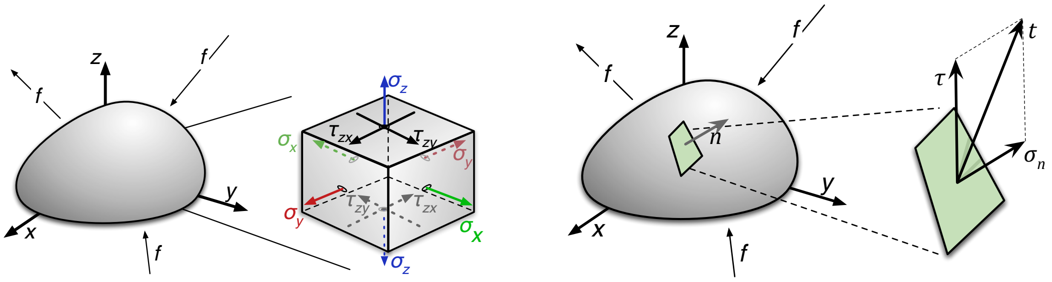
where $\mathbf{I}$ is the identity map. The projection tensor is symmetric, that is, $\mathbf{P}^T = \mathbf{P}$. It has one eigenvector in the direction of the surface normal $\mathbf{n}$ with eigenvalue zero and two orthogonal eigenvectors, lying in the tangent plane. The eigenvectors of the projected tensor $\widehat{\mathbf{T}}$ are in general not eigenvectors of the original tensor $\mathbf{T}$.

2.11 Example Application: Mechanical Engineering

> *Tensors provide [. . .] a natural and concise mathematical framework for formulating and solving problems in areas of physics such as elasticity, fluid mechanics and general relativity.* [Cammoun et al., 2009]

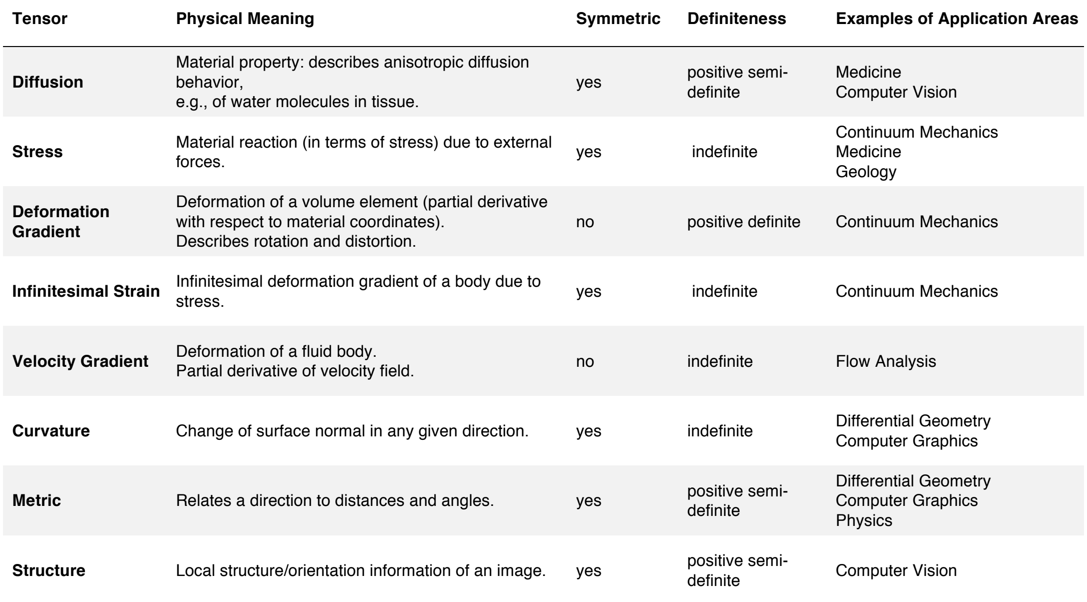
Tensors play an important role in numerous areas ranging from mathematics, physics, various engineering disciplines to medical applications. We list examples of second-order tensors and their applications in Table 2.1. A short overview over related applications with focus on computer vision can be found in [Cammoun et al., 2009]. All these applications build on the well-founded mathematical framework that was introduced in Section 2.3. However, the mathematical properties (Section 2.6) that are exhibited by their tensors of interest require diverse interpretations. The goal of this section is to enhance the intuition for these properties and to introduce second-order tensors that appear in mechanical engineering (see also Section 1.2 and Section 4.5).

Stress tensors describe internal forces or stresses that act within deformable bodies as reaction to external forces (Figure 2.3a). *Strain tensors* are related to the deformation of a body due to stress by the material's constitutive behavior. Together, they specify the behavior of a continuous medium under load, which allows one to deduce information about the strength of a technical part. Stress tensors convey information about the stresses acting on cutting planes through the material: their direction, their magnitude, their anisotropy and whether stresses are compressive (negative eigenvalues)



(a) Stress tensor **(b)** Traction vector

Figure 2.3: External forces f are applied to a deformable body. Reacting forces are described by a 3D stress tensor that is composed of three normal stresses σ and three shear stresses τ (a). Given a surface normal n of some cutting plane, the stress tensor maps n to the traction vector t, which describes the internal forces (normal and shear stresses) that act on this plane (b).



Tensor	Physical Meaning	Symmetric	Definiteness	Examples of Application Areas
Diffusion	Material property: describes anisotropic diffusion behavior, e.g., of water molecules in tissue.	yes	positive semi-definite	Medicine Computer Vision
Stress	Material reaction (in terms of stress) due to external forces.	yes	indefinite	Continuum Mechanics Medicine Geology
Deformation Gradient	Deformation of a volume element (partial derivative with respect to material coordinates). Describes rotation and distortion.	no	positive definite	Continuum Mechanics
Infinitesimal Strain	Infinitesimal deformation gradient of a body due to stress.	yes	indefinite	Continuum Mechanics
Velocity Gradient	Deformation of a fluid body. Partial derivative of velocity field.	no	indefinite	Flow Analysis
Curvature	Change of surface normal in any given direction.	yes	indefinite	Differential Geometry Computer Graphics
Metric	Relates a direction to distances and angles.	yes	positive semi-definite	Differential Geometry Computer Graphics Physics
Structure	Local structure/orientation information of an image.	yes	positive semi-definite	Computer Vision

Table 2.1: Examples of second-order tensors. Common properties of all tensors are: they are independent of their describing reference frame, they linearly relate vectorial quantities and they encode a quantity that inherently exhibits anisotropic behavior. They differ in their physical meaning and characteristic properties. If tensors contain rotational information, they are not symmetric. If anisotropy of the encoded quantity is the same in directly opposing directions, they are symmetric. Tensors that only have positive eigenvalues are positive definite. If eigenvalues are signed, the tensor is called indefinite. Then, it additionally encodes the *type* of the encoded quantity, for example, if stresses are compressive or tensile or curvatures are elliptical, hyperbolic or parabolic. The properties listed in the table follow form the definition of the tensor. For measurement data they might be violated due to noise.

or tensile (positive eigenvalues). If there is no rotation (which is, in general, fulfilled for infinitesimally small volume elements), the stress tensor is symmetric. Corresponding to Equation (2.3), the stress tensor is a linear function that maps an input vector (surface normal $\mathbf{n}$) to the stress vector (traction vector $\mathbf{t}$) (Figure 2.3b):

$$\mathbf{t} = \mathbf{T}_\sigma \cdot \mathbf{n} = \begin{pmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{pmatrix} \cdot \mathbf{n} = \tau_n + \sigma_n. \tag{2.28}$$

The traction vector is further decomposed into a part normal to the cutting plane $\sigma_n = (\mathbf{n}^T \cdot \mathbf{T} \cdot \mathbf{n}) \cdot \mathbf{n}$ representing normal stresses, and into a tangential part τ_n representing shear stresses.

The *strain tensor* is a normalized measure based on the deformation gradient tensor. There are different strain definitions depending on the respective deformation theory. For example, the Cauchy's strain tensor dealing with infinitesimal deformations is defined as the symmetric part of the infinitesimal displacement-gradient. Compared to the deformation gradient tensor, it loses the information about rotation. The relative infinitesimal volume change is expressed by the trace (Equation (2.18)) of the strain tensor.

Questions and features In mechanical engineering, it is a common task to model the mechanical material behavior by constitutive equations or material models. For material models that are concerned with elasto-plastic material behavior, the cutting planes that exhibit a maximum shear stress τ_{max} give an indication of the onset and evolution of yielding

$$\tau_{max} = \frac{\lambda_1 - \lambda_3}{2} \ . \tag{2.29}$$

Another example is the von Mises yield criterion that is often used in the field of ductile materials. It states that material changes its behavior from elastic to plastic when the von Mises stress exceeds a critical value. The von Mises stress is defined as the magnitude of the tensor's deviator

$$\sigma_v = \sqrt{\frac{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_1 - \lambda_3)^2}{2}} \ . \tag{2.30}$$

Both quantities are measures for anisotropy. These two simple criteria should be considered as examples for the numerous criteria used in engineering to rate a 3D tensor by a scalar value.

The motivation to use such criteria is that one can easily measure the onset and evolution of yielding by a simple uniaxial material test that delivers one scalar stress value. We analyze relevant questions in more detail in Section 4.5.

Many questions motivating the simulations of stress and strain tensor fields are related to information about the strength of a technical part. If the internal forces exceed certain limits of material strength, there is the risk for structural failure of the part. The analysis of the resulting stress tensor fields facilitates different failure models depending on the workpiece's materials. These failure models are often of the same structure than the above mentioned criteria for yielding of materials and deduce a scalar value from the 3D tensor data.

3 Categorization and Analysis of Previous Work

3.1 Introduction

Looking at tensors from a visualization point of view, the extraction of relevant information from the literature is a non-trivial task. Several workshops have been held with the goal to bring researchers from different disciplines together and to foster new collaborations and research. As a result, several books [Weickert and Hagen, 2005; Laidlaw and Weickert, 2008; Aja-Fernández et al., 2009] have been published that give insight into the multiple related research areas. However, a clear focus on DTI and image processing (structure tensor as feature descriptor during data analysis) can be observed. Recently, a tutorial on tensors in visualization was presented [Kindlmann et al., 2010] that also covers tensor topology, asymmetric tensors and higher-order tensors for diffusion tensor imaging.

This chapter surveys, categorizes and lists the major challenges of analysis and visualization methods of second-order tensors. The focus is the class of *indefinite* (Section 2.6) tensors. Such tensors are, for example, generated as result of simulations in engineering applications but they are also used as feature descriptors in flow analysis. This chapter is based on the survey article *Visualization and Analysis of Second-Order Tensors: Moving Beyond the Symmetric Positive-Definite Case* [Kratz et al., 2013a].

Structure Section 3.2 covers segmentation and topological methods that have been developed to reduce the complex information of tensor fields. The succeeding sections focus on visualization methods for indefinite tensor fields. These methods are categorized into local methods depicting single tensors at discrete points (Section 3.3) and continuous methods that provide an overview of the data (Section 3.4). Section 3.5

complements this chapter by providing a summary of methods that adapt the concept of *multiple views* for tensors. The chapter concludes with a thorough discussion of open problems and possible research directions in Section 3.6.

3.2 Segmentation and Topology Extraction

The goal of tensor segmentation algorithms is to aggregate regions that exhibit similar data characteristics to ease the analysis and interpretation of the data. Two basic classes can be distinguished:

1. Algorithms that generalize common segmentation and clustering techniques from image processing. These approaches require the definition of appropriate dissimilarity and similarity measures for tensor fields.

2. Algorithms that decompose the tensor field by extracting the topology. These approaches aim for a structural segmentation that builds on the directional part of the field. First distinctive locations (degenerate elements) in the tensor field are computed. From these locations, separating structures are identified (separatrices) that provide a segmentation of the field into regions with qualitatively equal eigenvector behavior.

3.2.1 Challenges

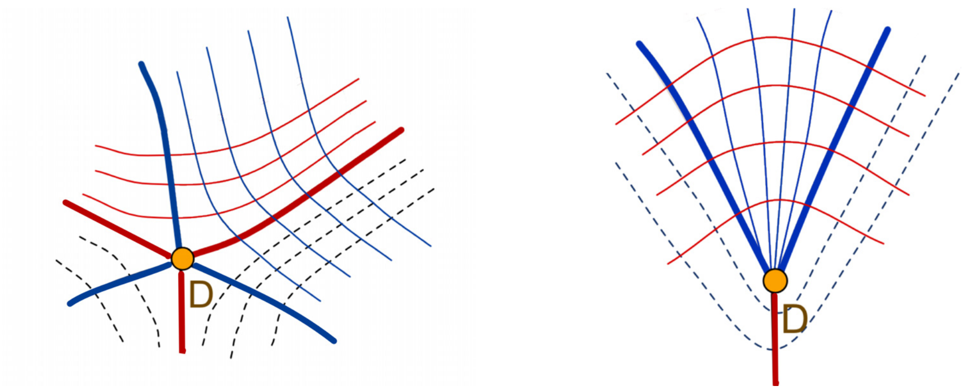
Depending on the chosen algorithm, various challenges arise.

Similarity measure A first step comprises the choice of appropriate quantities that should be segmented. These quantities can be features that are derived from the tensor data or the original tensor data itself. The choice of quantities then determines the choice of an appropriate similarity measure (Section 3.2.2).

Simplification of complex structures Topology-based segmentations may result in very complex structures, which are hard to interpret. Therefore, algorithms for simplification and tracking over time play a crucial role (Section 3.2.3). The understanding and representation of the topology for tensor fields of dimension higher than two is subject to ongoing research.

3.2.2 Similarity-Measure-Based Segmentation

In order to adapt methods from image segmentation and clustering, an appropriate dis-/similarity measure for tensors needs to be defined. Thus, questions posed by



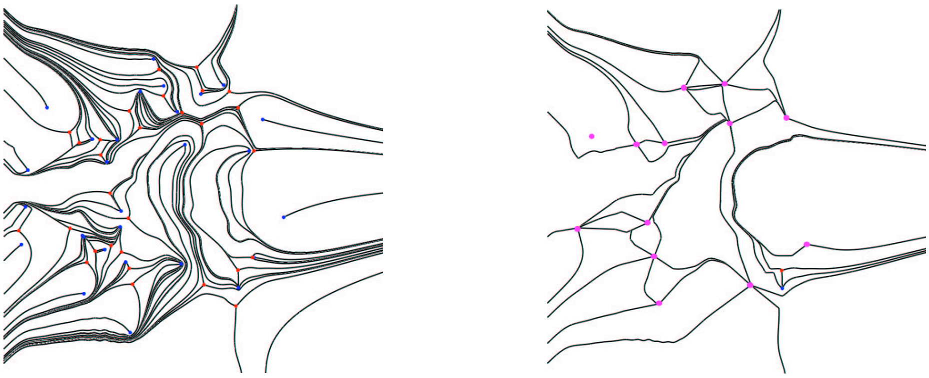
(a) Trisector

(b) Wedge

Figure 3.1: Basic first order degenerate points for 2D tensor fields: (a) trisector, (b) wedge. Separatrices for both eigenvector fields are drawn as bold red or blue lines, respectively. Exemplary integrated tensorlines (thin lines) emphasize how the eigenvector behavior within is reconstructed by the bounding separatrices (bold lines). Image courtesy [Auer et al., 2011].

segmentation are strongly related to those in interpolation (Section 2.8).

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 (Section 2.7) or comprise the entire tensor data. Most existing research efforts have been motivated by DTI data. For a review of recent developments that focus on the segmentation of anatomical structures from DTI, we refer the reader to [de Luis-García et al., 2009]. Proposed methods range from active contours [Wang, 2004] and level sets [Zhukov et al., 2003; Feddern et al., 2003] to graph-cut algorithms [Weldeselassie and Hamarneh, 2007; Ziyan et al., 2006]. Used metrics are the angular difference between principal eigenvector directions, or standard metrics considering the entire tensor, like the Euclidean or Frobenius distance. Wang and Vemuri [2004] introduced a distance measure using the Kulback-Leibler distance from information theory designed for Gaussian distributions, which is limited to positive-definite tensors. In general the chosen dis-/similarity measure determines the applicability to indefinite tensors. A segmentation designed for meshes based on the curvature tensor was introduced by [Lavoue et al., 2005]. Vertices of the mesh are clustered according to their principal curvatures values using a k-means classification.



(a) Initial topological structure

(b) Simplified topological structure

Figure 3.2: Rate of deformation of a swirling jet CFD simulation. Initial topological structure (a) and its simplification (b). Image courtesy [Tricoche et al., 2001].

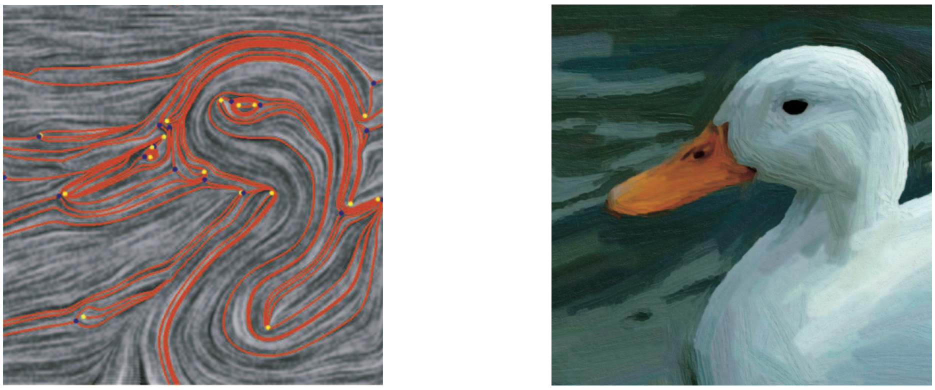
3.2.3 Topology-Based Segmentation

The topology of a tensor field represents its fundamental structure that is uniquely defined by the underlying eigenvector fields. These structures are represented by the *topological graph* or *topological skeleton* consisting of:

- *Degenerate elements:* Locations in the field where the tensor's eigenvalues have a multiplicity of two or three and the dimension of eigenvector space is larger than one (Section 2.5). For 2D tensor fields, the generic case are degenerate points. They can be classified according to the behavior of the eigenvector field in its vicinity. Typical patterns are trisector (Figure 3.1a) and wedge (Figure 3.1b). It is important to note that these patterns cannot occur in vector fields.

- *Separatrices:* Distinctive tensorlines or surfaces (Section 3.4.3) that emerge from the degenerate elements and that are tangent to the eigenvector fields. For 2D tensor fields, separatrices segment the tensor field such that they bound regions of qualitative homogeneous eigenvector behavior.

2D symmetric tensor fields The basic idea to use topology to segment tensor fields with respect to the structure of their eigenvector fields goes back to [Delmarcelle, 1994; Delmarcelle and Hesselink, 1994; Lavin et al., 1997a,b]. To improve the applicability of topological methods, Tricoche et al. [2001] have developed algorithms to simplify tensor field topology. In correspondence with vector field topology, [Tricoche et al., 2001] have extended these methods for time-dependent fields to track the structure over time (Figure 3.2).



(a) Topology-based edge field

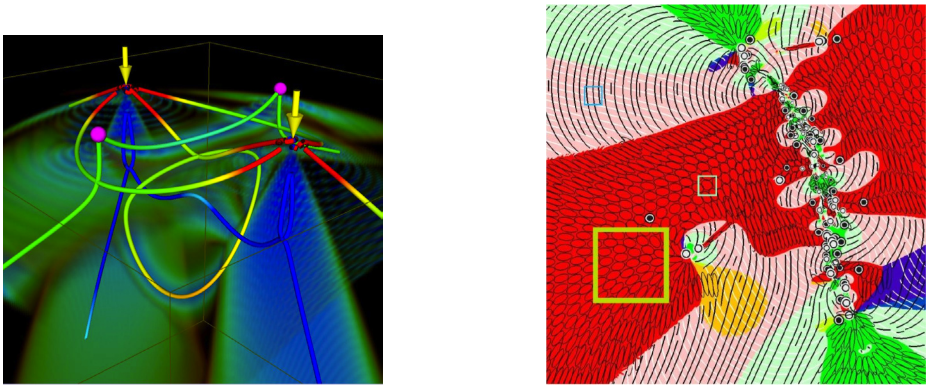
(b) Painterly rendering

Figure 3.3: Tensor-topology-based image-edge field (a) for painterly rendering of an image of a duck (b). Image courtesy [Zhang et al., 2007].

Another approach resulting in a simplified and consistent topology was introduced by Sreevalsan–Nair et al. [2010], which then was extended to curved surfaces [Auer et al., 2011]. Both methods are based on the shape-preserving interpolation model [Hotz et al., 2010] which emphasizes the strong connection of direction interpolation (Section 2.8) and tensor topology. Since purely topology-based segmentation neglects the importance of the eigenvalues, Auer et al. [2011] extend the topology extraction in this manner. To achieve a more intuitive visualization, they also used this segmentation to generate textures [Auer et al., 2012].

Topological methods are also beneficial in computer graphics. Examples for the application of tensors in computer graphics and related research work are remeshing [Alliez et al., 2003], tensor field design for the generation of textures (Figure 3.3) [Zhang et al., 2007], and procedural street generation [Chen et al., 2008].

2D asymmetric tensor fields Regarding flow analysis, asymmetric tensor fields become increasingly important. The most obvious approach to deal with such data is to decompose the tensor into its symmetric and antisymmetric part (Section 2.9) The antisymmetric part then corresponds to a rotation and is visualized using methods from vector visualization. Alternative approaches were presented by [Zheng and Pang, 2005; Zhang et al., 2008]. Zheng and Pang [2005] analyzed asymmetric tensor topology by introducing *dual-eigenvectors*. Zhang et al. [2008] build upon this idea by a reparameterization of tensor space such that tensor fields can be studied by their eigenvalue and eigenvector manifolds. They decompose the tensor into isotropic scaling, rotation, and pure shear and determine the predominant behavior. With respect to this segmentation,



(a) 3D Topology

(b) Hybrid visualization

Figure 3.4: 3D topology of point-load dataset (a). Yellow arrows indicate point load directions, while the two magenta spheres show the location of the triple degenerate points. Degenerate lines are colored by the Eigen difference. Image courtesy [Zheng and Pang, 2005]. Hybrid visualization of the displacement-gradient tensor field of an earthquake simulation (b). Image courtesy [Chen et al., 2011].

different visualization methods (tensorlines and glyphs) are applied [Chen et al., 2011] (Figure 3.4b).

3D symmetric tensor fields Research in 3D tensor topology is still in its infancy and the applicability of existing methods to real-world problems is challenging. Hesselink et al. [1997] started 3D topology, which was extended by Zheng *et al.* [Zheng and Pang, 2004; Zheng et al., 2005, 2006]. They show that in 3D, degenerate elements form one-dimensional structures (Figure 3.4a). Tricoche et al. [2008] demonstrate that these degenerate lines can be defined as crease lines of a tensor invariant. While research for degenerate elements has been initiated, separatrices, which occur in 3D as separating surfaces, increasingly moved out of focus.

3.2.4 Discussion

In the following, the presented methods are discussed in terms of the challenges.

Similarity measure The choice of a certain similarity measure for a segmentation is essential for the usability of the results. This question is closely related to the question of feature definitions. Many existing methods from clustering or scalar and vector field segmentation may be applied if the *right* notion of similarity is chosen.

Tensor field topology Topological methods have their strength in a well-founded mathematical basis. They are well suited to distill structural information of the direction field. Nevertheless, the use of topological methods in real-world applications is still very limited. This is partially due to the complexity of the resulting structures (already for 2D) but also due to the missing interpretations for the respective applications. First attempts have been made to build more intuitive visualizations facilitating the topological structure [Chen et al., 2011; Auer et al., 2012], but there is certainly much more potential that could be exploited.

3.3 Local Tensor Visualization Methods

Local methods depict single tensors sampled at discrete positions across the field. Geometric objects (glyphs) are used to encode tensor properties in terms of shape, color and orientation. Schultz and Kindlmann [2010] recently revealed the following useful applications for glyphs:

- Debugging: For example, during design of new visualization methods.
- Evaluation of data quality: For example, when tensors appear as intermediate product during simulations.
- Visualization overview: For example, to get a first clue of the data and reveal patterns in it.

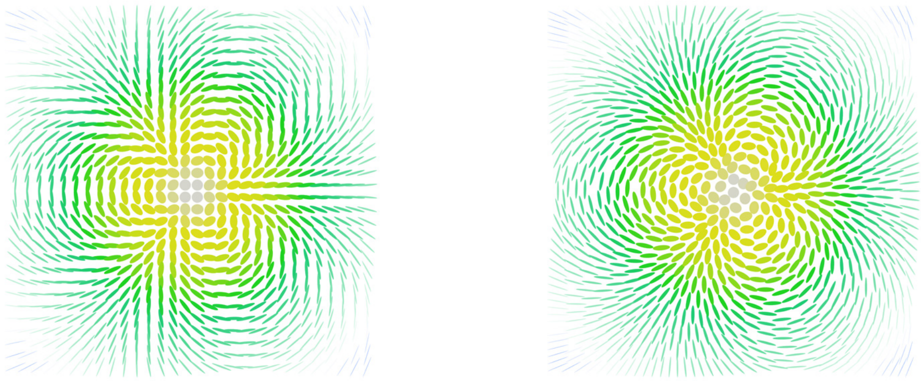
We would like to add *probing* to this list:

- Probing: Complex glyph geometries can be used for the detailed analysis of single tensors.

A comprehensive overview of glyphs focusing on stress and strain tensors is given in [Hashash et al., 2003]. Furthermore, Schultz and Kindlmann [Schultz and Kindlmann, 2010] recently discussed common glyph types with respect to specific design guidelines. The goal of this section is to reveal the major challenges of glyph-based visualizations and to emphasize those papers that we consider to be most inspiring for creative future work. Interactive, high-quality rendering (e.g., [Sigg et al., 2006; Hlawitschka et al., 2008]) will not be discussed.

3.3.1 Challenges

Design The design of glyphs comprises the questions *which* properties of the tensor should be mapped to the various degrees of freedom that geometries provide and *how*. The goal is to find a geometry that encodes all these properties without being too



(a) Uniform sampling

(b) Anisotropic sampling

Figure 3.5: Comparison of uniform ellipse placement (a) and placement based on an anisotropic sample distribution (b). In (b) the continuous structure of the tensor field is much more emphasized whereas in (a) the pattern of the underlying grid becomes visible. The placement was computed with the method described in Section 5.5.

complex. Another criterion is whether a representation for single tensors is desired (complex geometries can be used) or if the goal is to provide an overview of the data to see patterns in it or to get an intuition for the data (in that case, less complex geometries are preferable).

Rendering Concerning glyph rendering, visual ambiguity becomes an issue for 2D projections. This is the case when the glyph's shape depends on the viewpoint, which can result in wrong interpretations and hampers the perception of orientation.

Placement Another challenge is *where* to place the glyphs. This becomes especially important if continuous structures or patterns of the data need to be conveyed. Being placed at discrete grid positions, the pattern of the underlying grid becomes visible in the final visualization, which disturbs the perception of the field's continuous behavior (Figure 3.5). Moreover, overlapping glyphs and holes should be avoided as they lead to visual artifacts and undesired occlusions.

3.3.2 Glyph Design

From a purely theoretical perspective, a 3D tensor is perfectly represented by an ellipsoid scaled by the eigenvalues and oriented by the eigenvectors. However, ellipsoids suffer from visual ambiguity. Furthermore, *pure* ellipsoids do not enable a distinction between the sign of eigenvalues, which encode important physical properties (Section 2.11). This is particularly important for tensors that are not positive definite. The major goal of glyph design, therefore, is to decide which properties are essential to achieve a specific visualization goal and how these are mapped to a geometry. The following common guidelines can support these decisions.

Which tensor quantities should be used? For the choice of feature descriptors that are mapped to glyph parameters (shape, color, transparency), the following guidelines can help:

- Use orthogonal tensor invariants (e.g. [Ennis and Kindlmann, 2006]).
- Use a set that describes the tensor uniquely.
- Focus on specific questions.
- Use quantities users are familiar with.

Whereas the first two points comply with mathematically elegant choices, the last two points can enforce to break these rules. Moreover, glyphs that are designed for a specific application field should follow the conventions that are shared by users in this field. This might also force the visualization expert to break the first two rules.

How should the tensor quantities be depicted? Schultz and Kindlmann [2010] have proposed the following general design guidelines for choosing geometries as glyphs:

- Symmetry preservation: Glyphs exhibit the same symmetries as the underlying tensor.
- Continuity and disambiguity: Glyph geometries look similar for similar tensors. That is, no abrupt changes of the geometry are allowed when neighboring data values differ only marginally. On the other hand, different tensors should result in distinguishable glyphs.
- Scale invariance: Uniform scaling of the tensor by changing its norm results in a uniform scaling of the glyph geometry. For traceless tensors or tensors with small norm, however, the corresponding feature descriptors are undefined and unstable, respectively.
- Invariance under eigenplane projection: The projection of a tensor to a plane spanned by two eigenvectors results in a corresponding orthogonal projection on the glyph geometry.

The first two rules are very general, whereas the third rule (scale invariance) depends on the application. Especially if the tensor's deviator is of interest, glyphs that are

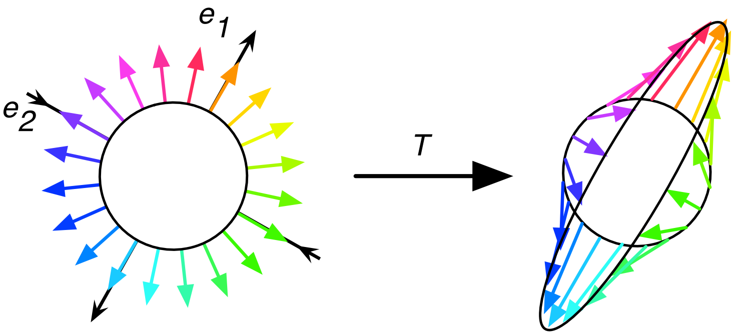


Figure 3.6: Illustration of an indefinite symmetric second-order tensor interpreted as generator of a deformation of a unit sphere. The resulting glyph is the deformation ellipsoid.

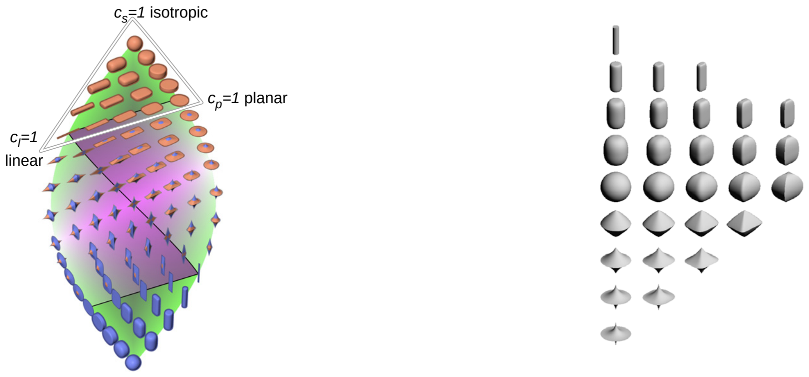
translation invariant are more important. An example for such a glyph is Mohr's circle (Section 3.3.5). We, therefore, add translation invariance as alternative to scale invariance to this list.

3.3.3 Mapping and Normalization

Due to space-filling reasons, glyphs can be normalized [Laidlaw et al., 1998; Zhang et al., 2003]. Especially for diffusion tensors, generally the normalized eigenvalues $\widetilde{\lambda}_i = \lambda_i/\|T\|$ are chosen as feature descriptors, where $\|\cdot\|$ is the Frobenius norm (Equation (2.21)). For traceless tensors or tensors with small norm, a different scaling needs to be applied. Overall, glyphs should only be normalized in applications in which the norm does not play a role.

Many glyph shapes require positive feature descriptors and do not enable the distinction of the eigenvalues' sign (e.g., ellipsoids). Alternative approaches use glyphs that are especially designed to depict positive as well as negative eigenvalues (Section 3.3.5). However, they are often prone to suffer from visual ambiguity. Another possible way is to encode the sign of the eigenvalues in the surface color or define a mapping of feature descriptors into the positive domain. The simplest mapping uses the magnitude of the eigenvalues. However, as discussed in Section 2.11, the sign encodes important information. Therefore, mapping techniques have been developed that preserve the physical meaning of the tensor.

Tensors encoding infinitesimal deformations (e.g., the infinitesimal strain, rate of strain and stress tensors) can be naturally mapped to positive-definite tensors by applying an exponential mapping (see also [Schultz and Kindlmann, 2010]). Considering



(a) Superquadric glyphs

(b) Nematic liquid crystal glyphs

Figure 3.7: Superquadric tensor glyphs extended to the indefinite space (a) for three-dimensional tensors. The upper triangle in (a) corresponds to the space of the original superquadric glyphs. Nematic liquid crystal tensor glyphs (b). Image courtesy [Schultz and Kindlmann, 2010; Jankun-Kelly and Mehta, 2006].

an infinitesimal line segment ds, these tensors act as generators of a deformation (Figure 3.6). This is expressed in the following differential equation

$$\mathrm{d}(\mathrm{d}s) = \mathbf{T} \cdot \mathrm{d}s \ , \tag{3.1}$$

which motivates an exponential mapping. Several related mappings $F : \lambda \to F(\lambda)$ for the eigenvalues have been proposed [Hotz et al., 2004], where F is a positive monotone function. Examples are

$$\begin{aligned} F(\lambda) &= a + \sigma \cdot f(\lambda) && (a) \\ F(\lambda) &= a \cdot \exp(\sigma f(\lambda)) && (b) \end{aligned} \tag{3.2}$$

with $f(\lambda) = -f(-\lambda)$ being a monotone function (e.g., $f = \mathrm{id}$ or $f = \arctan$). The parameter $a > 0$ defines the unit and $\sigma > 0$ controls the slope in the neighborhood of $\lambda = 0$. Thus, regions where the eigenvalues change sign can be emphasized. While (b) is positive by definition, for (a) the parameter σ has to be chosen appropriately. Defining the mapping for the eigenvalues ensures rotational invariance.

3.3.4 Glyph Design based on a Parameterizable Space

> *A new and powerful family of parametric shapes extends the basic quadric surfaces and solids, yielding a variety of useful forms.* [Barr, 1981]

The basic idea of the approaches presented in this section is founded in geometric modeling [Barr, 1981]. Barr introduced *superquadrics* as an extension of quadrics to produce a parameterizable continuous space of varying forms. Thus, only few base geometries need to be designed from which more complex shapes can be interpolated through parameterization. The adaptation of superquadrics to encode data properties with respect to perceptional issues was presented by Shaw et al. [1999]. For positive-definite tensors, superquadric tensor glyphs [Kindlmann, 2004] address the issue of visual ambiguity. Cylinders, cuboids and ellipsoids, therefore, are combined in a barycentric space that encodes the basic diffusion tensor shapes (linear c_l, planar c_p, spherical c_s), which are the anisotropy measures introduced in the work of Westin et al. [1997]:

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}, c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}, c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}.$$

The base geometries for perfectly planar (a flat disk), perfectly linear (a sustained thin cylinder), and isotropic tensors (a sphere) then build the corners of the barycentric shape space (Figure 3.7a, upper triangle). Recently the *superquadric shape space* has been extended for indefinite tensors [Schultz and Kindlmann, 2010] (Figure 3.7a). For symmetric traceless tensors, a special variant of superquadric glyphs has been introduced in [Jankun-Kelly and Mehta, 2006] to visualize the orientation of liquid crystals (Figure 3.7b).

3.3.5 Glyphs Applied in Mechanical Engineering

The following list summarizes the most common glyphs in the context of mechanical engineering.

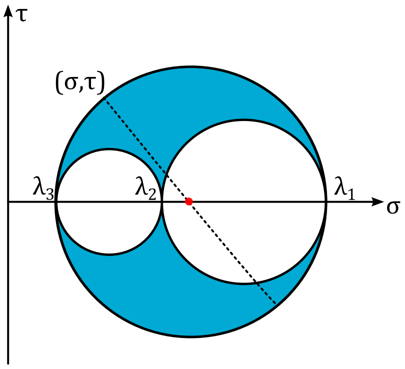
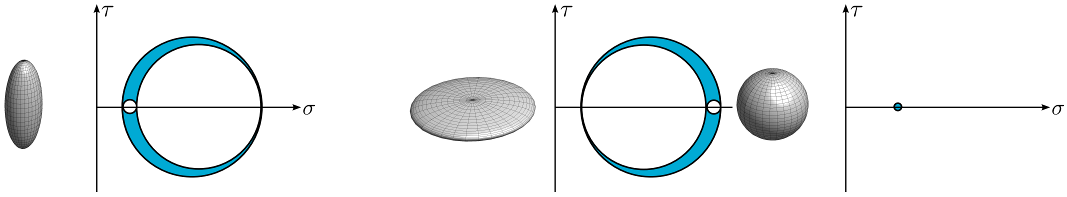


Figure 3.8: Mohr's circle.

Mohr's circle Mohr's circle is common in continuum mechanics and geology. It has been used for stress tensors [Crossno et al., 2004, 2005] as well as for diffusion tensors [Bilgen and Narayana, 2003] by the visualization community. Originally, Mohr's circle was developed as a 2D graph that plots normal stresses σ against shear stresses τ (Figure 2.3). The horizontal axis represents normal stresses σ and the vertical axis represents shear stress τ (see Equation (2.28)). The blue shaded area in Figure 3.8 represents all possible combinations of normal and shear forces. Each point within this region then corresponds to one orientation of the normal of a cutting plane (see also Section 2.11). The finding was that all possible (σ, τ)-combinations fall within the domain of a circle. Traditionally, Mohr's circle is used to find the principal coordinate system, graphically. If the eigenvalues are already known,



(a) Linear $\lambda_1 > \lambda_2 \approx \lambda_3$ **(b)** Planar $\lambda_1 \approx \lambda_2 > \lambda_3$ **(c)** Isotrop $\lambda_3 \approx \lambda_2 \approx \lambda_1$

Figure 3.9: Lamé's stress ellipsoid and Mohr's circle in comparison.

Mohr's circle is constructed from the three eigenvalues, which determine the position and radius of the circle. The greater the circle's outer radius, the higher the maximum shear stress. Circles that degenerate to single points represent isotropic pressure. It is common to display only half of the circle, which we will also do in Chapter 4. Mohr's circle has also been used in multiple-view systems (Section 3.5), where they are used as basic element within a diagram to explore and to validate 3D finite-element simulations.

All following glyphs are aligned with the principal directions of the tensor. Alternatively, they can be represented as surfaces (implicit or as deformed sphere) that do not require an explicit eigenanalysis.

Lamé's stress ellipsoid or PSN-Glyph Lamé's stress ellipsoid displays all possible traction vectors. It can be obtained by applying the tensor to the unit sphere $\{T \cdot n | n \in \mathbb{R}^3, \|n\| = 1\}$. The resulting glyph is an ellipsoid aligned to the tensor's principal directions and scaled by the magnitude of the eigenvalues (Figure 2.1). With respect to the principal frame of reference, the glyph is defined as implicit surface by

$$\frac{x^2}{\lambda_1^2} + \frac{y^2}{\lambda_2^2} + \frac{z^2}{\lambda_3^2} = 1 \ .$$

However, without additional color coding, the sign of the eigenvalues is not represented. Figure 3.9 shows Lamé's stress ellipsoid in comparison with Mohr's circle.

Haber glyph The Haber glyph encodes the same information as Lamé's stress ellipsoid, but emphasizes the major principal direction. It is a combination of an elliptical disc (displaying intermediate and minor eigenvector and eigenvalue, respectively) and a rod in direction of the major eigenvector scaled by the major eigenvalue [Haber, 1990].

Reynolds glyph The Reynolds glyph highlights the normal stresses σ_n thereby neglecting the shear τ (Equation (2.29)). The basic shape of the glyph allows one to distinguish definite and indefinite tensors [Moore et al., 1996; Kriz et al., 2005] (Figure 3.10). The glyph is defined by the set of all normal directions scaled by the magnitude of the normal stresses in that direction $\{\sigma_n \cdot n | n \in \mathbb{R}^3, \|n\| = 1\}$.

HWY glyph The HWY glyph focuses on the magnitude of the shear stress and is a sort of counterpart to the Reynolds glyph [Hashash et al., 2003]. Its surface represents the normal of all possible cutting planes, scaled by the magnitude of the shear force τ_n with $\tau_n^2 + \sigma_n^2 = \|T \cdot n\|$. The glyph is defined by the set of all normal directions scaled by the magnitude of the shear stresses acting in the plane perpendicular to the normal n $\{\tau_n \cdot n | n \in \mathbb{R}^3, \|n\| = 1\}$. The direction of the shear stress in that plane is not represented by the HWY glyph.

Quadric surface Quadric surfaces encode the entire tensor information. They are defined as implicit surfaces by

$$\lambda_1\, x^2 + \lambda_2\, y^2 + \lambda_3\, z^2 = \pm 1\ .$$

Quadric surfaces are mostly used in context with the curvature tensor of a surface corresponding to the Dupin indicatrix. Note that the resulting surfaces are not bounded for the case of indefinite tensors.

Plane-in-a-box glyph The plane-in-a-box glyph has been especially developed for the analysis of stress and strain with application in geomechanics, where the question of failure plays an important role [Neeman et al., 2005]. The plane-in-a-box glyph consists of a plane, defined by the two larger eigenvectors, limited by a box. This glyph is used in combination with other visualization methods to convey a global overview.

3.3.6 Glyph Placement

The goal of glyph placement techniques is to determine positions within some domain Ω where it is effective to place glyphs. Here, effective means that the glyphs are densely packed while holes and overlaps between them are avoided. This task is comparable with questions from sampling theory. The following approaches all start with an initial point distribution that is subsequently refined in an iterative manner.

Feng et al. [2008] presented a geometric approach that is based on anisotropic Voronoi cells. Through Lloyd relaxation, particles are iteratively shifted into the direction of each cell's centroid. This finally leads to a sampling that fulfills blue-noise properties. Since this approach only depends on attractive forces, it provides a stable solution. A limitation is that it does not provide an explicit control of overlaps.

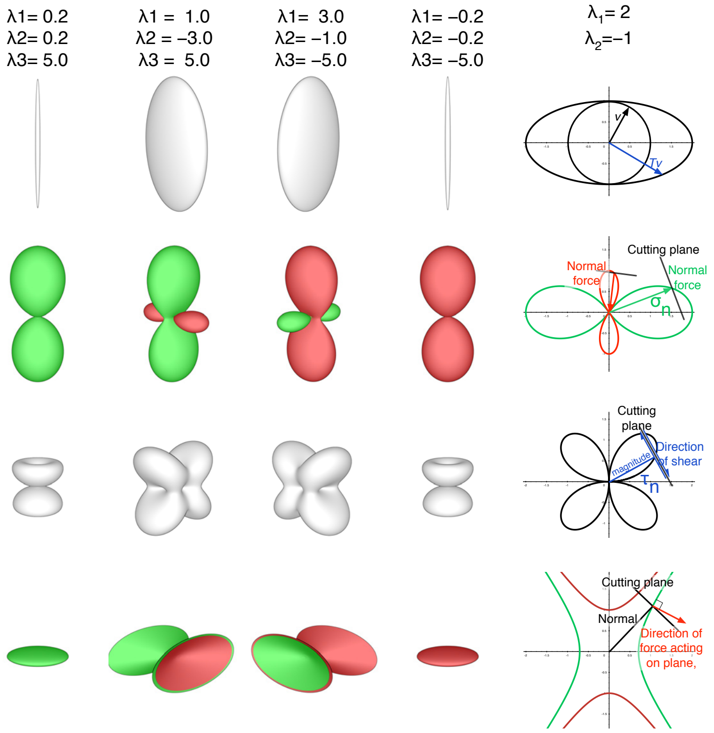


Figure 3.10: Overview of basic stress glyph shapes. From top to bottom: ellipsoid (Lamé's stress ellipsoid), Reynold's glyph displaying the normal forces, HWY glyph displaying the magnitude of shear forces, and quadric surfaces. Whenever it is meaningful to distinguish compressive (red) and tensile (green) forces, they are colored respectively. The right column illustrates the interpretation of the various glyph types for the 2D case.

In the context of mesh generation, Shimada et al. [1996] presented a particle-based approach for the generation of adaptive anisotropic meshes. Their approach relies on an explicitly defined tensor field to specify an anisotropic spacing between the nodes. These spacings then determine inter-particle forces: Repelling forces avoid overlaps and attracting forces close holes.

For diffusion tensors, Kindlmann and Westin [2006] build upon the work of Shimada et al. [1996]. In their work, the positive-definite tensor field is mapped to a potential energy field that determines inter-particle forces. Hlawitschka et al. [2007] extended their work focusing on improved initial sampling and interactivity through the use of an isotropic Delaunay triangulation. Thus, forces are only computed between neighboring particles. For initial sampling, they propose interleaved and stratified sampling.

3.3.7 Discussion

Design A variety of glyph types have been presented; each has its strengths and limitations (e.g., Figure 3.10). Glyphs alone are rarely self-contained and seldom provide *intuitive* insight into a physical process. Nevertheless, glyphs are indispensable for tensor field visualization, because they still are the only way to visualize the whole tensor. Until now, there is no *universal* glyph that is capable to meet the diverse requirements imposed by the various application fields of tensors. Helpful guidelines for glyph design have been formulated by Schultz and Kindlmann [2010]. A first question that always should be imposed is whether to use a glyph in combination with probing or in a global overview visualization. Glyph design will not be covered in this thesis. Instead we focus on placement algorithms of ellipses and ellipsoids in Chapter 5.

Rendering Domain-specific glyphs (Section 3.3.5) contain a rich amount of valuable information but they were not designed with respect to perceptional issues, which appear during rendering. To visualize a whole tensor field, superquadric tensor glyphs (Section 3.3.4) are more appropriate. Moreover, they represent an efficient and elegant way to create meaningful glyph shapes via parameterization of some shape space.

Placement In the last years, several algorithms have been presented for distributing glyphs. They are mainly categorized into methods that rely on particle simulations [Shimada et al., 1996; Kindlmann and Westin, 2006; Hlawitschka et al., 2007] and those that are based on Lloyd relaxation [Feng et al., 2008]. Apart from [Shimada et al., 1996; Kindlmann and Westin, 2006], all methods have only been presented for 2D tensor fields. In Chapter 5 we will present two methods to create anisotropic sample distributions for the distribution of glyphs. We propose a particle-based methods (Section 5.4) that is applicable to 2D tensor fields, and we propose a more general geometry-based method (Section 5.5) that also works in two-manifold domains.

3.4 Global Tensor Visualization Methods

Local methods have the potential to depict the whole tensor information locally. Being used together with placement algorithms, they can even convey continuous structures of the underlying tensor field. Even more appropriate in providing an overview of the data and to emphasize regional coherence and global structures are continuous methods. Available methods depict scalar- and direction-related quantities that are derived from the underlying tensor field.

3.4.1 Challenges

A major challenge in visualizing tensor fields is the reduction of clutter and occlusion. In this context, many difficulties arise that concern both: scalar visualization methods as well as vector visualization methods.

Design Continuous tensor visualization methods mostly depict only a subset of the information that is contained in a tensor field. Hence, they rely on *data reduction* (see also Section 1.3). Similar to local methods, a first step, therefore, includes the decision *which* quantities (scalar or vector) are appropriate to reach a given visualization goal or analysis task and *how* to depict them efficiently.

Transfer function design For methods based on scalar-valued features, transfer functions need to be designed that have the potential to reveal these features in an intuitive way.

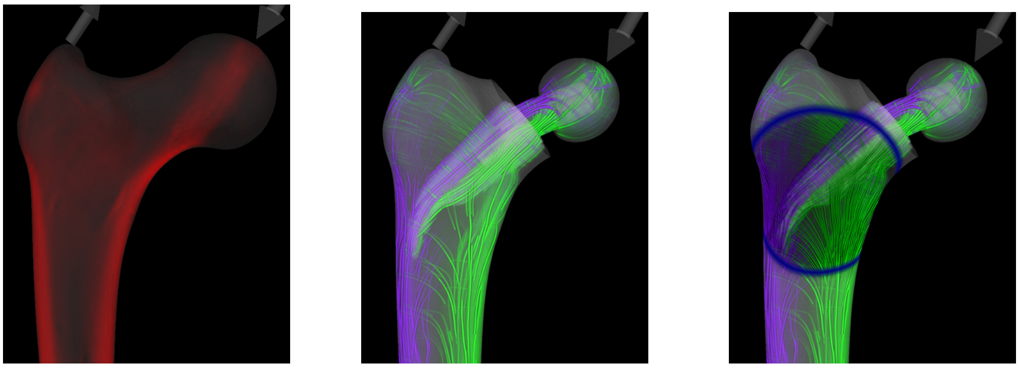
Seeding For methods based on vector-valued features, the question is *where* to depict lines (seeding). The placement of seed points has a strong influence on the quality of the final result in the sense how good vector-valued features can be revealed. In contrast to vector fields, for tensor fields usually multiple directions are of interest, which makes the problem even more difficult.

Generation of application-specific noise Vector visualization methods that adapt LIC textures to depict directional features of tensor fields rely on an input noise texture. Here, the challenge is to generate input noise textures that are specifically designed to depict tensor fields.

3.4.2 Scalar Visualization Methods

Scalar visualization techniques for tensors provide a quick overview of scalar quantities that are derived from the tensor field (e.g., von Mises stress for stress tensors). As part of a hybrid rendering technique (e.g., [Dick et al., 2009]) they can provide context. The following papers use ray-casting as rendering technique.

Kindlmann and Weinstein [1999] propose a volume rendering approach, which builds on three components: (1) Hue-balls using a 2D colormap to encode direction. (2) Triangular barycentric opacity maps that are parameterized by anisotropy using the metrics introduced by [Westin et al., 1997]. (3) Lit-tensors providing a lighting model



(a) von Mises stress (b) Hybrid visualization (c) Focus and context

Figure 3.11: 3D simulated stress tensor field for a human femur under load. Volume rendering of a derived scalar quantity (here: von Mises stress) neglecting directional information (a). Hybrid rendering encoding the sign and magnitude of the three eigenvalues as well as directional information (b). Focus+context visualization that allows for a detailed stress analysis in interactively defined focus regions (c). Image courtesy [Dick et al., 2009].

further enhancing the linear or planar character of the tensor, respectively.

Hlawitschka et al. [2009] focus on directional information for transfer function design. The components of the normalized major eigenvector are mapped to RGB color values, respectively. Transparency is chosen on the basis of scalar-valued features such as fractional anisotropy (FA) or mode. To emphasize fiber bundle boundaries, additional directional information is used.

Dick et al. [2009] presented a color-mapping for stress tensors that distinguishes compressive and tensile forces. Every eigenvalue is separately assigned a color according to its sign. The magnitude of the eigenvalues guides an opacity transfer function: it emphasizes high stresses and fades out low stresses. A compositing scheme summarizes and weights the RGB and α values of the single eigenvalues to one final color per sample. In their work, volume rendering of the eigenvalues serves as context visualization. It is combined with tensorlines, which are described in Section 3.4.3, to validate simulations of physiological stress inside a bone due to an implant (Figure 3.11). The visualization is part of a larger framework that allows the comparative visualization before and after implantation to avoid stress shielding.

3.4.3 Vector Visualization Methods

Many vector visualization methods have been adapted to visualize prevalent directions in tensor fields. In general, one out of the two/three eigenvector fields is chosen. While such generalizations are partially successful there are two aspects that should always be kept in mind. First, when applying methods from vector visualization to depict eigenvector fields, the bi-directionality of eigenvectors has to be considered. This means, for example, that integrators need to be adapted in a way that the direction is checked during integration. Second, the physical meaning of the methods in the context of vector visualization may be lost in the context of tensor fields.

Tensorlines

The analogue to streamlines for vector fields are tensorlines for tensor fields. They are defined as lines that are tangential to one chosen eigenvector field (major, intermediate, minor). For the computation of tensorlines, all integrators for streamlines can be adapted using an additional direction check [Weinstein et al., 1999]. Fiber tracking in diffusion tensor imaging is an example focusing on the major eigenvector assuming that it defines the orientation of nerve fibers in areas of linear anisotropy [Vilanova et al., 2005]. Tensorlines only capture one eigenvector field. An extension are *hyperstreamlines*, which incorporate eigenvalues and the other principal directions in the cross section along the tensorline [Delmarcelle and Hesselink, 1993]. Hyperstreamlines have been utilized in a geomechanical context [Scheuermann et al., 2001]. Even though this method visualizes more information, the method is limited to low resolution due to clutter. Shen and Pang [Shen and Pang, 2004] propose a method called *hyperseed* that tackles the problem of seeding for the computation of tensorlines. The seeding is controlled by the anisotropy of the field and has been used for DTI data.

Wilson and Brannon [2005] have introduced *stress nets*, which represent the directions of maximum shear for 2D fields (Figure 3.12). To achieve a uniform distribution of integral lines, the authors build upon a uniform seeding strategy [Jobard and Lefer, 1997]. In addition, the image background is used to represent scalar-valued features, for example, the magnitude of the deviator (Equation (2.25)).

3.4.4 Texture-Based Methods

In the following, texture-based methods to visualize tensor fields are presented.

Methods based on LIC Several adaptations of the dense line integral convolution (LIC) [Cabral and Leedom, 1993] to tensor fields have been presented. As for most

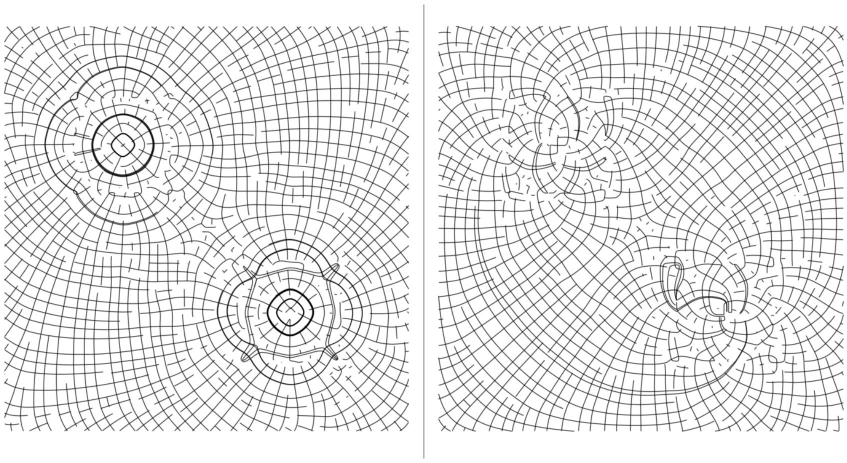


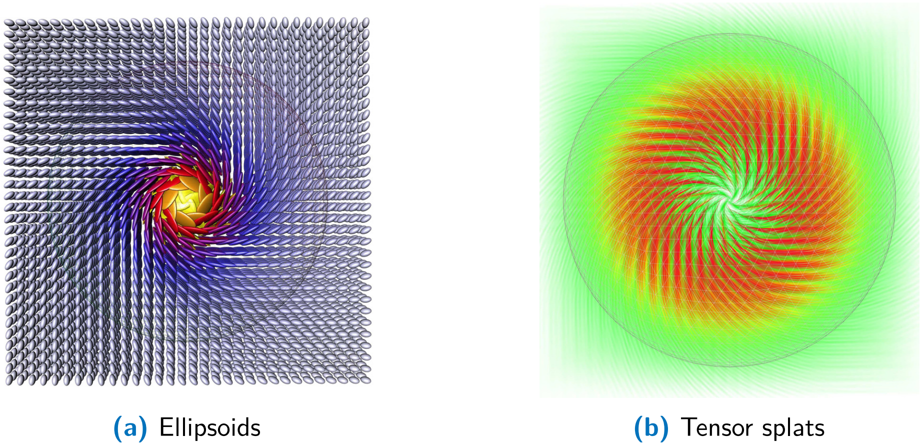
Figure 3.12: Stress nets for the two-point-load dataset. The left image depicts the stress net for the principal directions. The right image depicts the stress net for the same slice and the maximum shear directions.

texture-based methods, they are mainly restricted to 2D surfaces and not suited for 3D tensor fields.

Zheng and Pang [2003] introduced HyperLIC where the one-dimensional filter kernel used in LIC is replaced by a 2D kernel whose shape is determined by the tensor. HyperLIC was designed to emphasize the major eigendirection in anisotropic regions, and to deemphasize isotropic regions without preferred directions. This method is especially suited for positive-definite tensors. For indefinite tensors, this method should be applied carefully. As HyperLIC only considers the magnitude of the eigenvalues, tensors that have eigenvalues with opposite sign are illustrated as isotropic, which can lead to misinterpretations.

Hotz et al. [2004] presented *fabric textures* for the visualization of stress tensor fields. They introduce a mapping of the indefinite stress tensor to a positive-definite metric and use the positive eigenvalues to define spot-noise textures as input for LIC. Then, a LIC-like texture for every eigenvector field is created and the two resulting textures are blended, which results in images that resemble a fabric. Thinner fibers correspond to compression and thicker fibers correspond to expansion.

Reaction diffusion Textures resembling ellipsoid packing (Section 3.3.6) can also be obtained via reaction-diffusion textures. Reaction diffusion of two chemicals can be described by a set of two nonlinear partial differential equations describing the concentrations of two chemicals as a function of time. The change of the concentration



(a) Ellipsoids

(b) Tensor splats

Figure 3.13: Tensor ellipsoids (a) and tensor splats (b). Image courtesy [Benger and Hege, 2004].

is determined by two terms representing the diffusion, encoding the tensor, and the reaction of the chemicals. The method has been introduced by Turing. Similar concepts have been used for tensor field visualization [Kindlmann et al., 2000; Diewald and Rumpf, 2000]. While this approach is capable of generating high quality textures, it suffers from two problems: the difficulty in selecting the right parameters and its high computational costs. Sanderson et al. [2006] proposed an advanced model to overcome these problems.

Tensor splats The idea of tensor splats [Bhalerao and Westin, 2003; Benger, 2004] is to map the tensor's shape onto a geometric primitive and *splat* it onto the image plane (Figure 3.13). The splats are assigned with an opacity value and composited in front-to-back order. Hence, tensor splats are suitable for 3D tensor field visualization. If they are applied to diffusion tensors, anisotropy is used to determine opacity values, for example, regions of high fractional anisotropy are emphasized. Color is assigned with respect to the eigenvalues. Thus, visual clutter is reduced and regions of interest are emphasized. If other tensor invariants/feature descriptors are used, tensor splats can also be applied to other tensor types.

Tensor field design An alternative approach that was presented in the context of graphics and painterly rendering is the design of tensor fields [Zhang et al., 2007] to generate textures. Here, the fields are designed with the goal to achieve specific visual effects.

3.4.5 Discussion

It can be observed that global visualization methods for tensor fields mainly were adapted from the more elaborate fields, namely scalar- and vector visualization and that global methods specifically designed for the depiction of tensor fields are rare.

Design Global methods that were specifically designed for tensor fields are tensor splats [Bhalerao and Westin, 2003; Benger, 2004], HyperLIC [Zheng and Pang, 2003] and fabric textures [Hotz et al., 2004]. HyperLIC is suited for positive-definite tensors for which it has been designed. Better suited to depict indefinite tensors are fabric textures. The fabric metaphor is well suited to illustrate the central properties of 2D stress tensor fields: compression and expansion. We will present examples and results that use this visualization technique in Chapter 4 and in Chapter 5. The limitation of texture-based methods is that they can only be used efficiently for 2D visualizations. Tensor splats are appropriate to visualize 3D fields but have been paid very little attention in the past.

Transfer function design Scalar visualization methods in the context of tensor field visualization are suited to provide a quick overview if an application or dataset is given for which a specific scalar quantity is of interest. We believe that scalar visualization methods should only be applied on scalar-valued features that were derived from the tensor field such as in the work of Dick et al. [2009]. They should not be applied on directional features because in these cases the interpretation is difficult.

Seeding If methods from vector visualization are adapted, effective seeding strategies are needed. Here, mainly seeding methodologies were used that have already been applied in vector visualization (e.g. [Verma et al., 2000]). To depict two eigenvector fields in a three-dimensional visualization, Dick et al. [2009] propose to place seed points only on the mesh surface. Thus, they are able to depict two directions simultaneously in a three-dimensional visualization without introducing occlusion. In Chapter 4 we show how meaningful seed points for stress tensor fields can be determined via brushing in shape-space scatterplots.

Generation of application-specific noise Hotz et al. [2004] have shown how the generation of specific noise textures can be used to tune LIC-like visualizations for the depiction of tensor fields. Zheng and Pang [2003] have adapted the filter kernel to use LIC for the depiction of 2D diffusion tensor fields. In Chapter 5, we show how the quality of fabric textures is improved by combining fabric textures and anisotropic sampling.

3.5 Multiple Views for Tensor Visualization

Gaining useful insights via object-space rendering only (especially in 3D) can be difficult due to the complexity of tensor fields. In recent years, a few papers were concerned with multiple views for tensors, which allow the inspection and visual exploration of tensor properties in detail and at large. Those methods can improve an intuition for the data and help to make more precise statements about data properties than with object-space renderings only.

3.5.1 Challenges

Design When designing multiple view systems for tensor visualization, a crucial question is *which* views should be used and *how* they should be combined.

Interactivity Systems with the goal of visual data exploration must be interactive. Hence, efficient implementations are mandatory.

3.5.2 Fiber Exploration

In the context of DTI, Chen et al. [2009] and Jianu et al. [2009] have both presented multiple-view systems for interactive fiber exploration. They combine spatial fiber representations with 2D diagrams and statistical views [Chen et al., 2009]. Of particular interest is the *multi-dimensional scaling* approach used by Chen et al. [2009] to achieve a low-dimensional embedding of the high-dimensional fibers. Thus, a fiber in object space is narrowed down to a point in 2D space, where similarities are expressed via proximity. Consequently, exploration of fibers becomes much easier than in 3D. Both approaches further apply clustering. Whereas Chen *et al.* [Chen et al., 2009] use k-means clustering applied on the 2D space, Jianu et al. [2009] apply hierarchical clustering. In the latter work, the cluster result is depicted via dendrograms.

3.5.3 Stress Exploration

Crossno et al. [2005] present an approach that links Mohr diagrams and a wire-frame depiction of a finite element model. Their focus is on the analysis of three-dimensional stress tensor fields in a geomechanical context. The overall goal of their work is to understand force-interactions between salt diapirs and “surrounding sediments” to identify “drilling trajectories” that minimize the risk of drilling failures.

3.5.4 Discussion

Design Multiple views for tensor visualization are a relatively new research topic. The presented methods have mainly focused on views that are established for the analysis of multi-variate data such as scatter plots. In Chapter 4 we will present a framework that combines shape-space scatter plots, Mohr diagrams, parallel coordinates and directional histograms, which are linked via feature spaces for tensors.

Interactivity Systems with the goal of visual data exploration must be interactive. This clearly limits the number of approaches that can be used.

3.6 Discussion

We have reviewed and classified methods for analyzing and visualizing second-order tensors focusing on tensors that are not symmetric positive definite. First of all, it can be observed - maybe even more than in scalar or vector field visualizations - that there are no universal solutions. Decisions for the development of new tensor analysis and visualization algorithms are highly application specific. Despite this fact, the following general observations are possible.

Topology extraction methods allow one to compute a topological skeleton consisting of degenerate points and separatrices. Especially in combination with simplification and tracking algorithms, the topological skeleton depicts the complex input data in a simplified manner. Existing methods for topology extraction are of high theoretical and structural interest. However, in their pure form, such skeletons are difficult to interpret (particularly in 3D) and are often not considered as an intuitive data visualization from the perspective of domain experts. Topology extraction, therefore, should be considered as a pre-processing step. Taking this as a basis, more intuitive visualization methods can be developed. Topological methods are not considered in this thesis.

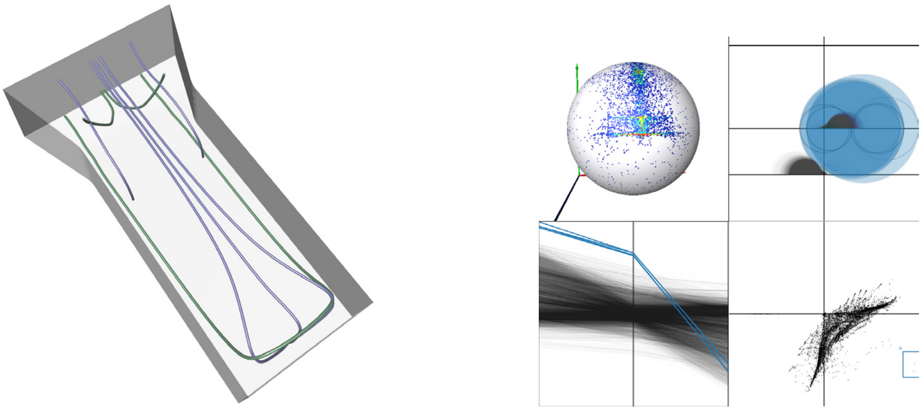
Local tensor visualization methods depict the *whole* tensor information locally. For local tensor visualization, most often glyphs are used. While glyphs have the power to represent all tensor properties, their interpretation is often difficult and needs a lot of expertise. Nevertheless, glyphs are a powerful tool and until now remain the only visualization technique that is capable of visualizing the whole tensor. While complex domain-specific glyphs are often not suited for the visualization of a whole tensor field, superquadric glyphs that account for perceptional issues are more appropriate for this task. In this work, we contribute to this area with two methods that automatically compute meaningful placements for glyphs (Chapter 5).

Global visualization techniques provide continuous depictions of derived scalar- or vector-related features. Here, techniques from scalar and vector field visualization have been adapted. The problem when adapting methods from vector visualization often is a loss of meaning, since they do not necessarily have a physical interpretation for tensor fields, or at least the meaning is very application-specific. In the context of diffusion tensor imaging, tensorlines are used to represent nerve fibers. But, what is the physical meaning of tensorlines in the context of mechanical engineering? We will face this question in Chapter 4, especially in Section 4.5. Overall, global methods are better suited to provide an overview than for an in-depth data analysis.

Multiple view systems for tensors might represent a powerful approach that helps explore the data and to extract meaningful features. We will present a concept that allows for the adaptation of multiple linked views to tensor fields in Chapter 4.

4

A Multi-Perspective Approach to Visual Analysis of Stress Tensor Fields



(a) Hybrid spatial view

(b) Selection in ordered shape space

Figure 4.1: Link-and-brush for stress tensor fields. Outliers representing a high maximum shear stress are selected in ordered shape space (b). For the selected region, seed points are distributed from which tensorlines are started (a).

4.1 Introduction

In this chapter, we present a new framework for the visual exploration of stress tensor fields. The framework implements the concept of *multiple views* together with *linking and brushing*. That is, various diagram views are presented side-by-side with 3D spatial depictions of the data. Regions of interest can be selected in the diagram views via brushing and the selected regions are highlighted in all other views (Figure 4.1). The major motivation to focus on such an approach was the finding that the analysis of stress tensor fields is a relatively new research topic in engineering and, hence, domain experts in these fields have not yet formulated specific questions towards the data. For this reason, we have identified a strong need for *explorative approaches* in this application context. Note that, although tensor visualization is related to the visualization of multivariate data, the difference is that the individual tensor components are not independent. The tensor components are linked by transformation rules when changing the basis. This is not true for individual components of multivariate data. Hence, we cannot simply use the single tensor components to parameterize the diagrams. With our approach, we first define feature spaces and use these to parameterize the diagrams.

The concept of multiple views is beneficial for tensor fields for two reasons. First, the multivariate nature of tensors is addressed, because tensor properties are distributed in various diagrams and views, instead of being encoded in a single image. Thus, the problem of clutter is reduced. Second, novel and more advanced visualization techniques can be combined with diagrams that are well-known or already established.

The concept of linking-and-brushing is beneficial in the context of tensor field visualization, because it offers the possibility to explore the data. Furthermore, the selection (brushing) of interesting data properties or regions of interest in one of the views, helps reduce the data that is visualized.

Challenges and questions that arise when designing such a framework are the decision which views to use for the exploration of tensor fields, how the concept of multiple views can be adapted to tensor fields, and, finally, interactivity. Specific to our approach is the combination of views. In object space, that is, the domain of the tensor field, we propose 3D hybrid visualizations that effectively combine spatial tensor field visualization methods in a single image. The hybrid visualizations are steered by brushing in the diagram views. Special attention has been paid to provide a variety of views and to be flexible in the kind of views that are used.

In Section 4.5, we show examples from engineering applications. In these applications, users still analyze only scalar measures that are derived from the tensor (for example, the von Mises stress) and neglect looking at the tensor itself. However, we strongly believe, and first experiments have strengthened this, that the analysis of the whole tensor field leads to deeper insights into the simulations themselves and into the phenomenon that is simulated. We show that the presented exploratory approach produces new insights into the data as well as new hypotheses regarding stress tensor fields from engineering applications.

Parts of this chapter are based on the paper *A Visual Approach to Analysis of Stress Tensor Fields* [Kratz et al., 2011b].

Structure This chapter is organized as follows. First, we review relevant previous work (Section 4.2). Subsequently, we describe how we build the feature spaces that are used to parameterize the diagram views and spatial depictions (Section 4.3). In the following section, the whole framework is introduced (Section 4.4) including diagram views (Section 4.4.2) and hybrid views in object space (Section 4.4.3). Then, in Section 4.4.4, we complement the manual brushing approach by investigating the value of clustering and scalar field topology to automatically find characteristic subsets in the tensor data. The visualization methods that were combined in the presented framework are shown in Section 4.4.5 on various example datasets. The chapter ends with the presentation of a project that resulted from the development of our framework. First insights into tensor data from structure simulations are presented in Section 4.5. Finally, in Section 4.6 a thorough discussion concludes the presented concept and results.

4.2 Related Work

The interactive analysis of feature spaces has been proposed for scalar fields [Maciejewski et al., 2009; Dobrev et al., 2011], for vector fields [Daniels et al., 2010] and to analyze flow simulation (for example, [Doleisch et al., 2003]). For tensor fields, this is a sparsely researched field. Two approaches have been presented for fiber exploration in the context of diffusion tensor imaging [Chen et al., 2009; Jianu et al., 2009]. Both approaches use clustering in *fiber feature space*. Chen et al. [2009] use k-means clustering combined with multi-dimensional scaling, and Jianu et al. [2009] use a hierarchical clustering approach visualizing the result via dendrograms. For stress tensors, derived scalar measures often have more relevance in a first analysis step than derived directions. We therefore focus on feature spaces spanned by tensor invariants. The barycentric shape space [Kindlmann and Weinstein, 1999; Kindlmann et al., 2000; Weinstein et al., 1999] that has been used to guide diffusion tensor visualizations is also a scalar-valued feature space. Only few visualization papers are related to using brushing in diagram views for tensors [Crossno et al., 2005; Jianu et al., 2009; Chen et al., 2009]. Despite the work of Crossno et al. [2005], combined views for stress tensors - in the variety and flexibility that we will present in the following - have not been presented previously. Also hybrid visualizations for tensor fields have only gained very little attention [Sigfridsson et al., 2002; Dick et al., 2009] and were not yet combined with linking and brushing.

4.3 Feature Spaces for Tensor Fields

A *feature space*, in general, is a d-dimensional abstract space, in which each point corresponds to a *feature*. The point's coordinates correspond to the values of the feature attributes, and the dimension d is determined by the number of attributes that describe the feature. Consequently, similar features, which describe similar attributes, build a proximity in feature space. Many concepts benefit from this property. For example, in pattern recognition this property is used for the classification of features. The challenge when defining such a feature space is to find a set of feature attributes that are suitable to represent the dataset, or to find a set that is suitable to answer a question that is posed to the dataset. In the following, we describe, how we define feature spaces for symmetric tensors in this work.

A symmetric tensor has six degrees of freedom represented by three direction-related entities, which generally are the principal directions, and three eigenvalues (see also Section 2.5). We, therefore, distinguish scalar-related features expressed in shape space (Section 4.3.1) and direction-related features expressed in direction space (Section 4.3.2). A first step when defining features for tensors consists of the tensor's decomposition into distinctive parts[1]. In this work, we mainly use the decomposition into *shape and orientation*. Alternatively or in addition, a decomposition into an *isotropic and an anisotropic* part is possible. In the following, we introduce the feature spaces and the terminology that is used in this thesis.

4.3.1 Shape Space

This section summarizes a concept for the definition of scalar-valued features building upon *tensor invariants* (Section 2.7) and the work of Bahn [1999]. Bahn [1999] introduced the term *eigenvalue space* for the space whose basis is constructed from the tensor's eigenvalues. In this space, the eigenvalues are considered to be coordinates of a point in Euclidean space. Eigenvalue space has been further explored in Ennis and Kindlmann [2006], who propose orthogonal invariants for the visualization of DTI data. Considering a set of invariants as basis for the analysis of strain tensors has been proposed in the work of Criscione et al. [2000]. In this work, we use the term *shape space* for the vector space whose basis is constructed from the tensor invariants.

Ordered shape space Since a permutation of the eigenvalues describes the same tensor, it is sufficient to restrict the shape space to the ordered subset where $\lambda_1 \geq \lambda_2 \geq \lambda_3$.

[1] See Section 2.9 for a list of decomposition schemes that are most often used for second-order tensors in visualization.

See the work of Bahn [1999] for a closer discussion on the ordered eigenvalue space. The *ordered shape space* builds the foundation for all following shape spaces. It is based on the decomposition of the tensor into shape and orientation. For three-dimensional symmetric tensors of second-order, the dimension d of the ordered shape space is $d = 3$. Then, the whole space reduces to four subsets corresponding to positive-definite tensors, negative-definite tensors and two subsets that cover indefinite tensors. Sometimes it is appropriate to consider further reductions of this space, for example, if we deal with specific tensor properties (Section 2.6), have incomplete information, or only partial interest. According to the property this either leads to a dimension reduction (e.g., subspace of traceless tensors) or to the definition of a subset that has the dimensions of the full shape space (e.g., subset of positive-definite tensors).

Shape descriptors A point in shape space is uniquely determined by the three eigenvalues. Equivalently, three coordinates according to any other reference frame of the shape space can be used. We call these coordinates *shape descriptors*. We present examples for stress tensors in Section 4.3.3.

Reference frame An invariant $I = f(\lambda_1, \lambda_2, \lambda_3)$ defines a family of surfaces in shape space. Each set of three independent invariants

$$\nabla_\lambda I_1(\nabla_\lambda I_2 \times \nabla_\lambda I_3) \neq 0 \qquad \text{with} \qquad \nabla_\lambda = (\partial/\partial\lambda_1, \partial/\partial\lambda_2, \partial/\partial\lambda_3), \tag{4.1}$$

defines a local basis of the shape space. A set of invariants qualifies as global basis if I_i is defined everywhere and $\nabla_\lambda I_i \neq 0$ for $i = 1, 2, 3$. From a mathematical point of view, a desirable additional property is orthogonality of the tensor invariants [Bahn, 1999; Criscione et al., 2000; Ennis and Kindlmann, 2006]

$$\nabla_\lambda I_i \cdot \nabla_\lambda I_j = 0 \text{ for all } i, j \in \{1, 2, 3\}. \tag{4.2}$$

Orthogonal invariants exhibit maximum independence of the shape descriptors by isolating changes of one invariant from variations of the others. However, in some cases the physical and domain-specific significance of invariants can predominate mathematical properties. For this reason, in this thesis, we also allow shape spaces that do not have an orthonormal basis.

In summary, an appropriate choice of reference frame depends on the application and its research questions. Each reference frame, that is a *set of invariants*, yields its own shape descriptors and, hence, provides its own view onto the tensor. Examples are given in Section 4.3.3.

Shape-space scaling In order to achieve a better distribution of the shape descriptors, especially for their *visual* analysis, a scaling of the coordinate axes is beneficial. Several scalings and mappings are possible and depend on the diagrams or visualization techniques that are applied to the shape space. For the diagrams (Section 4.4.2), we often use logarithmic mappings. For spatial depictions (Section 4.4.3), sign-preserving mappings [Hotz et al., 2004; Kirby et al., 1999] are a good choice. The idea of sign-preserving mappings is to find a mapping into the positive domain such that the tensor's physical meaning is visually expressed. Hotz et al. [2004], for example, consider stress tensors as generators of a deformation. With their mapping, ellipsoidal glyphs that describe compressive forces have a smaller volume than glyphs that describe expansive stresses.

4.3.2 Direction Space

We use the term *direction space* for the feature space that consists of directions. The full direction information is represented as a triple of points. Because eigenvectors are normalized, no additional scaling is needed and all points lie on the surface of the unit sphere. In general, we are only interested in a single direction or in two selected directions (see also Section 4.3.3). For a single direction, the direction space, is a 2D feature space with a spherical basis. Due to the unoriented nature of the eigenvectors, the space further reduces to a hemisphere.

4.3.3 Example Feature Spaces for Stress Tensors

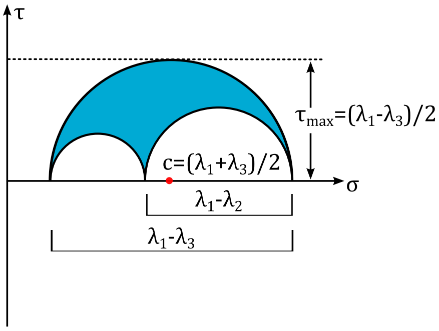


Figure 4.2: Mohr's circle.

Shape descriptors corresponding to the Coulomb failure criterion are [Lund, 2000]:

$$\begin{aligned} \tau_{max} &= \frac{\lambda_1 - \lambda_3}{2} \\ c &= \frac{\lambda_1 + \lambda_3}{2} \\ R &= \frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3} \end{aligned} \tag{4.3}$$

In terms of Mohr's circle (Figure 4.2), c represents the center and τ_{max} the radius of the Mohr circle. The maximum shear stress τ_{max} can also be considered as anisotropy measure. Another common anisotropy measure that considers all three principal stresses is the von Mises stress (Equation (2.30)). The shape factor $R \in [0, 1]$ reveals

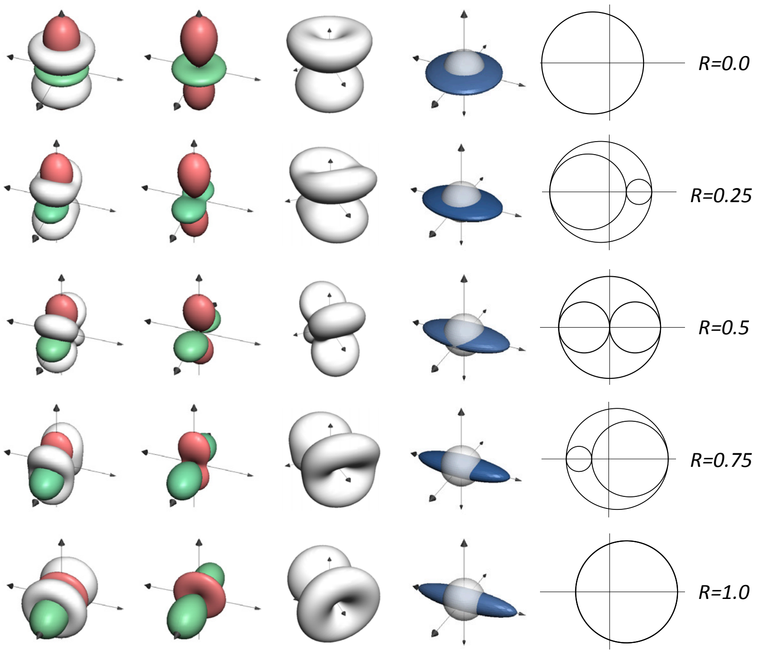


Figure 4.3: Comparison of glyph shapes that are used for stress tensors. The describing invariants as well as the directions are constant; the shape factor R varies. From left to right: Reynolds and HWY glyphs combined, Reynolds glyphs, HWY glyphs, deformation ellipsoid, Mohr's circle.

the kind of anisotropy. Similar to the terminology used in DTI, stresses with $R = 0$ are called *perfectly planar* and stresses with $R = 1$ are called *perfectly linear* in this work. Although the terminology of linearity and planarity has no physical relevance in the context of stress tensor fields, some categorization of basic properties of a tensor field is needed. Furthermore, the terminology is widely used in the field of DTI. Figure 3.9 in Section 3.3.5 shows how ellipsoids and Mohr circles are related to each other. Note that the shape factor is a relative value and undefined for small values of τ_{max} (isotropic stresses). The shape descriptors (τ_{max}, c, R) build no orthogonal coordinate frame (Equation (4.2)).

Shear-shape-factor space An example for a 2D space is the space whose axes are described by the maximum shear stress τ_{max} and the shape factor R that were introduced in Equation (4.3). This space only considers the tensor's deviator (Equation (2.25)), that is, it does not contain any information about the isotropic part which relates to uniform pressure. It is related to the idea of distinguishing tensors in terms of linearity and planarity but this interpretation has no justification for zero-trace tensors such as the deviator. Instead, it gives insight into the distribution of the directions related to maximum shear (see also Figure 4.3). For $R = 0.5$, there are exactly two distinguished

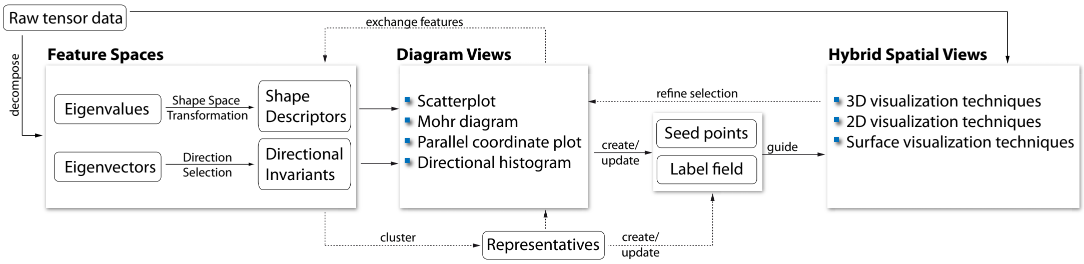


Figure 4.4: Reference model for exploratory visualization of tensor fields. Core of the model are feature spaces. Selected shape descriptors and directions are visualized and analyzed via feature-space plots. In object space, various tensor visualization techniques are combined in a feature-driven hybrid visualization. Optionally, clustering can be applied to automatically compute cluster representatives that have the potential to guide feature-space and object-space visualizations.

normal directions. For $R = 0.0$ and $R = 1.0$, the corresponding normal directions span two cones. The shape factor is nicely represented by the normalized HWY glyphs (Section 3.3.5).

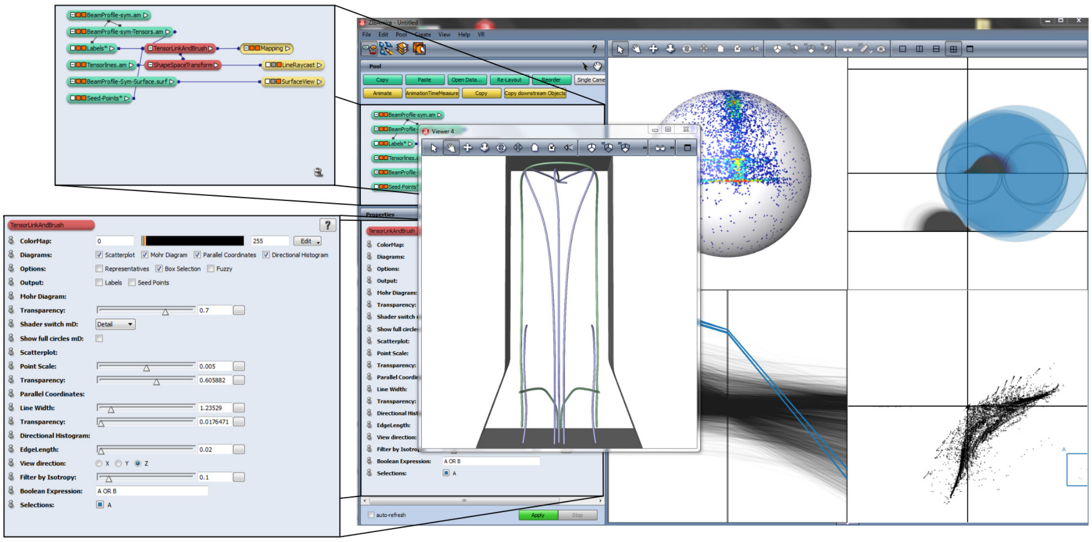
Center-shape-factor space Another example, which is related to the representation of stresses using Mohr circles (Section 3.3.5) is the space spanned by the Mohr center c and the shape factor R (Equation (4.3)).

Directions For stress tensors, which are indefinite, the major and minor eigenvector are of equal importance. The decision, whether both vectors need to be considered or only a single one depends on the underlying application and the question that is posed to the visualization. Another direction of interest for stress tensors is the direction of maximum shear (see, for example, [Wilson and Brannon, 2005]). For three-dimensional tensors, the direction of maximum shear is the bisector between the major $\mathbf{e}_1$ and minor $\mathbf{e}_3$ eigenvectors (Equation (2.6))

$$\mathbf{e}_\tau = \frac{\mathbf{e}_1 - \mathbf{e}_3}{2.0}. \tag{4.4}$$

4.4 A Framework for the Exploration of Tensor Fields

This section introduces the link-and-brush framework (Figure 4.4). It is implemented as prototype in the visualization software Amira [Stalling et al., 2005] (Figure 4.5). Currently, up to five simultaneous views are supported. For the datasets that are described in Appendix A, these views can be interactively exchanged during runtime. This substantially supports data exploration.



TensorLinkAndBrush

ColorMap: 0 255 Edit

Diagrams: ☑ Scatterplot ☑ Mohr Diagram ☑ Parallel Coordinates ☑ Directional Histogram

Options: □ Representatives ☑ Box Selection □ Fuzzy

Output: □ Labels □ Seed Points

Mohr Diagram:

Transparency: 0.7

Shader switch mD: Detail

Show full circles mD: □

Scatterplot:

Point Scale: 0.005

Transparency: 0.605882

Parallel Coordinates:

Line Width: 1.23529

Transparency: 0.0176471

Directional Histogram:

EdgeLength: 0.02

View direction: ○ X ○ Y ● Z

Filter by Isotropy: 0.1

Boolean Expression: A OR B

Selections: ☑ A

Figure 4.5: Screenshot of the brush-and-link framework, which is implemented as prototype into the visualization software Amira. The main window is separated into four diagram views: directional histogram for the major eigenvector field (top left), Mohr diagrams (top right), parallel coordinate plot (bottom left), scatterplots (bottom right). The area that is selected in the scatterplot is highlighted in the Mohr diagram and the parallel coordinate plot. It represents high mixed stresses and complies with the seed points that were used for the computation of the tensorlines. Hybrid object space rendering is displayed in a separate 3D viewer. In the left, the user interface is displayed. The diagrams can be interactively enabled or disabled. Transparencies, colors, and size of the diagram primitives can be adjusted.

4.4.1 Overview

The proposed framework implements the concepts of *multiple views* and *linking-and-brushing* in order to enable the visual exploration of 3D stress tensor fields. Input is a 3D or 2D tensor field of second order that is given on a uniform or tetrahedral (triangulated) grid.

Feature space For the definition of a feature space (Section 4.3), the input tensor data is decomposed into two distinctive parts that separate shape and orientation. Basic input for the scalar-valued *shape space* are the tensor's eigenvalues, which refer to *tensor shape* [Bahn, 1999]. Input for the *direction space*, in general, is a single direction of interest.

Diagram views and hybrid spatial views In order to analyze tensor properties in detail and at large, we separate the visualization into diagram views (Section 4.4.2) and 3D hybrid spatial views (Section 4.4.3).

Label field Selections in the diagram views are linked with the hybrid spatial depictions by a 3D label field of the same grid type and size as the input data. The label field is created and modified depending on what data properties and regions of interest are selected in the plots. The label field determines in *which* regions *what* visualization technique is applied and combined into a hybrid visualization.

Seed points Additionally to the label field, *seed points*, which are the starting points that are used for the integration of tensorlines, can be randomly distributed in labeled regions.

4.4.2 Diagram Views

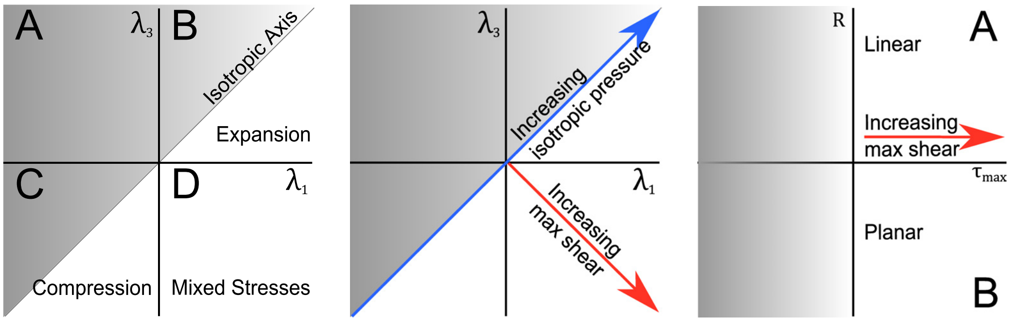
These diagram views abstract from the tensor field's spatial representation and give insight into the (statistical) distribution of tensor properties. All views are linked and used side-by-side. In the current implementation, we have decided on the following diagrams:

- **Scatterplots** deliver insight into the distribution of tensor properties. Brushing within the scatterplots creates and updates a label field and initiates the distribution of seed points that are used for the computation of tensorlines.
- **Parallel Coordinates** are used in addition to the scatterplot because they offer the possibility to represent more than two invariants.
- **Mohr diagrams** represent important invariants for stress tensors. Moreover, they are a common tool in engineering, and therefore familiar to a large group of potential users.
- **Directional histograms** quantify selected directions, for example major eigenvectors or other directions that are derived from the eigenvectors like the direction of maximum shear.

The diagram views are parameterized by:

- Shape descriptors (Section 4.3.1)
- Shape-space scaling (Section 4.3.1)
- Directions (Section 4.3.2)

In the following, the diagrams are presented in detail.



(a) (λ_1, λ_3)-Plot **(b)** (λ_1, λ_3)-Plot **(c)** (τ_{max}, R)

Figure 4.6: Basic patterns in shape-space scatterplot.

Shape-Space Scatterplot

A scatterplot is a 2D diagram that consists of two axes; each axis represents the values of one of the two scalar variables. Input, thus, is a tuple of scalar values. For each such tuple, a point is plotted in the diagram. In this work, a scatterplot is a direct visualization of a 2D shape space (Section 4.3.1). Different scalar-related attributes can be used as input. Usually, scatterplots are used to depict the correlation of two scalar variables. In this work, we are interested in the *pattern* (Figure 4.6) that arises when the points are plotted against each other. These patterns allow a quick visual extraction of basic properties of the input tensor field (see also Section 2.6), such as definiteness, isotropy, anisotropy as well as linearity and planarity. In the following, we illustrate this idea by means of two example configurations. Other configurations are possible.

(λ_1, λ_3)-Scatterplot Input of this plot are the major and minor eigenvalue from the ordered shape space (Section 4.3.1). Using these tensor invariants as input for the scatterplot, basic properties of the input tensor field such as definiteness and isotropy/anisotropy can be determined visually. For stress tensor fields, such a (λ_1, λ_3)-scatterplot is useful to identify normal (compressive or tensile) and shear stresses. See Figure 4.6a and 4.6b for a schematic depiction of this type of scatterplot. We divide the scatterplot into four quadrants (A, B, C, D). Due to the ordering of the eigenvalues (Equation (2.8)), the upper left quadrant (A) is empty. Points in the upper right quadrant (B) correspond to positive-definite tensors (Equation (2.14)). For stress tensors, points in quadrant B represent tensors with tensile stresses. Accordingly, points in the lower left quadrant (C) correspond to negative-definite tensors (Equation (2.15)). For stress tensors, points in quadrant C represent tensors with compressive stresses. Finally, the lower right quadrant (D) contains points that represent indefinite tensors, that is, eigenvalues have mixed sign. For stress tensors, points that are contained in D represent tensors with tensile and compressive stresses. To summarize, we can deduce the following characteristics of stress tensor fields from the (λ_1, λ_3)-scatterplot:

- Points in quadrant B correspond to expansive stresses.
- Points in quadrant C correspond to compressive stresses.
- Points in quadrant D correspond to mixed stresses.
- Points that have a large distance to the isotropic axis exhibit a high level of shear (Figure 4.6b).
- Points that are located near the isotropic axis exhibit no shear at all; they describe tensors with isotropic behavior (Figure 4.6b).

For scaling, we often use a logarithmic mapping for a better distinction of the properties in the scatterplot, and to avoid a cluttering around the origin. As a standard logarithmic mapping has a singularity in zero, we use

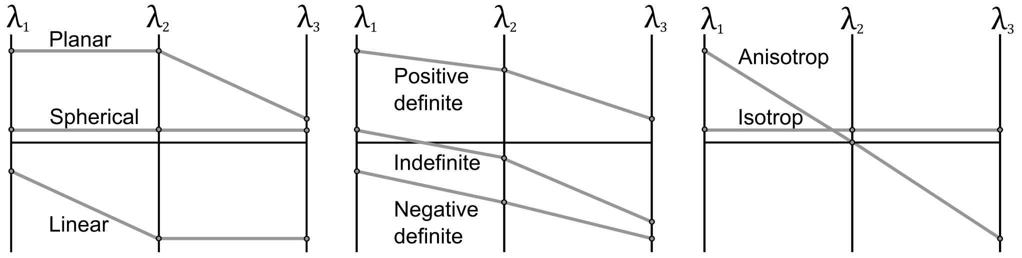
$$f(\lambda_i) = \begin{cases} \log(\lambda_i + 1), & \text{for } \lambda_i \geq 0 \\ -\log(1 - \lambda_i), & \text{for } \lambda_i < 0. \end{cases} \tag{4.5}$$

$(\tau_{\mathbf{max}}, \mathbf{R})$-Scatterplot Input of this plot is the shear-shape-factor space (Section 4.3.3). In contrast to the ordered shape space, the shear-shape-factor space does not distinguish tensors corresponding to tension and compression. Instead, it separates regions of high shear, which differ with respect to shape factor. We can extract the following properties from this plot (Figure 4.6c):

- Points that are in quadrant A correspond to tensors that have linear shape.
- Points that are in quadrant B correspond to tensors that have planar shape.
- The distance to the vertical axis determines the value of shear stress.

Parallel Coordinate Plots

A parallel coordinate plot (PCP) is a 2D diagram that consists of n axes; each axis represents one feature attribute. In a PCP, a n-dimensional feature is transferred to a polyline that intersects the values on each horizontal or vertical axis that represents a specific attribute. The advantage of PCPs, in contrast to scatter plots, is that they are able to depict high-dimensional features. Scatterplots are restricted to only 2D features. For higher-dimensional features, scatterplots are generally combined with multi-dimensional scaling. See for example the work of Chen et al. [2009], where this



(a) Degenerate cases **(b)** Definiteness **(c)** Anisotropy

Figure 4.7: Basic patterns in parallel coordinate plot.

approach has been used for the extraction of fibers from diffusion tensor fields.

Originally, parallel coordinate plots were invented in the context of multi-variate data analysis. In this work, we use PCPs in addition to scatterplots. The PCP is parameterized by the ordered shape space. That is, each vertical axis of the PCP represents one of the three eigenvalues. A triple of ordered eigenvalues, then, results in one polyline. As for the scatterplot (Section 4.4.2), we are interested in the resulting patterns, which enable a quick visual extraction of basic properties of the input tensor field (see also Section 2.6). In the PCP, we can observe the following patterns (see Figure 4.7):

- A straight horizontal line that runs through all three axes, represents an isotropic tensor. In terms of shape, this is a spherically shaped tensor.
- A straight line in the upper sector represents a degenerate tensor with two equal eigenvalues ($\lambda_1 = \lambda_2$), that is, a tensor that has planar shape.
- A straight line in the lower sector represents a degenerate tensor with two equal eigenvalues ($\lambda_2 = \lambda_3$), that is, a tensor that has linear shape.
- The more a line differs from a straight line, the higher is the anisotropy.
- Lines that are located only in the upper (lower) sector correspond to positive-definite (negative-definite) tensors. Lines that cross both sectors correspond to indefinite tensors.

Mohr Diagram

Mohr's circle is a common tool in material mechanics that is used to compute coordinate transformations. In visualization, it has been applied to diffusion tensors to depict the tensor's diffusivity [Bilgen et al., 2002] as well as to stress tensors [Crossno et al., 2005]. Being a known technique for domain experts, Mohr diagrams can ease the access to

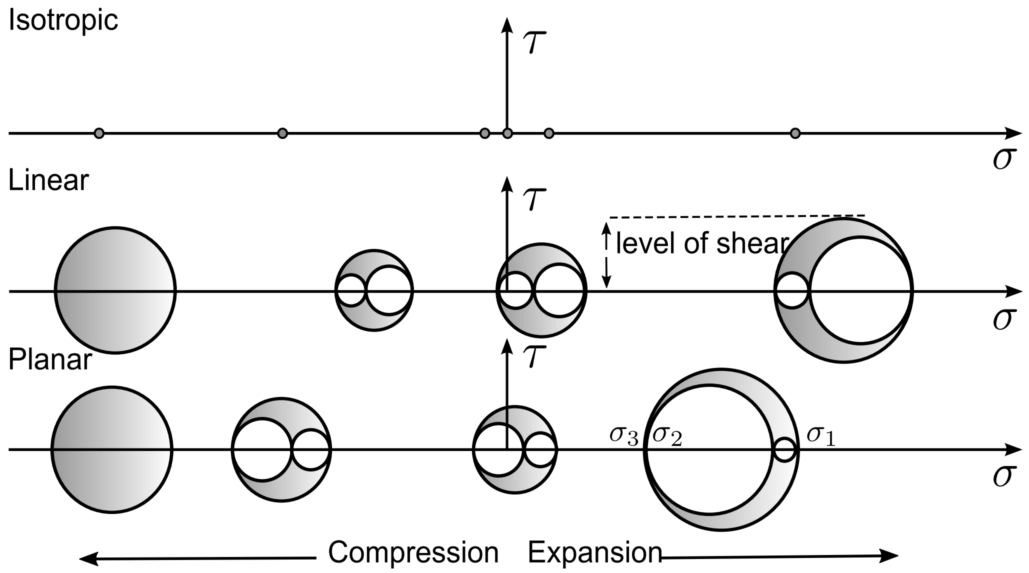


Figure 4.8: The circle's position on the x-axis represents whether the corresponding tensor is in compression (left) or tension (right). The vertical position corresponds to their anisotropic behavior and shape.

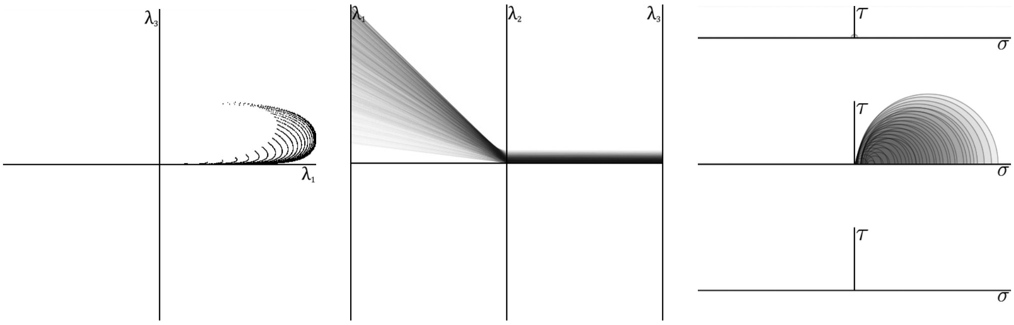
novel visualization methods. See Section 3.3.5 for more details on the basic idea of Mohr's circle.

Figure 4.8 illustrates the Mohr diagram which is one of the default diagrams within the proposed framework. It consists of Mohr circles, which give an impression of the relationship between the three eigenvalues and their relative strength. For a three-dimensional tensor, a single Mohr circle consists of three circles. The radii and position of the circles on the x-axis are determined by the three eigenvalues. Assuming a descending order of the eigenvalues (Equation (2.8)), tensors that describe compressive stresses are positioned on the left and tensors that describe tensile stresses are positioned on the right of the origin. Tensors that describe mixed stresses are centered around the origin. In the original diagram [Crossno et al., 2005], all circles are positioned on a single axis. In this work, we further separate the circles according to their shape (linear, planar, isotropic). For this classification, we divide the Mohr diagram into three separate diagrams (Figure 4.8). To achieve this, we use the shape factor R (see Equation (4.3)). A circle representing a tensor with $R < 0.5$ is drawn on the lower axis, and a circle representing a tensor with $R >= 0.5$ is drawn on the middle axis. Although R is a continuous quantity, this categorization enables the distinction of basic tensor shapes in the diagram. For isotropic tensors, R is not defined. Therefore, we handle this case separately. Circles that represent isotropic tensors are identified by checking whether $\tau_{max} < \epsilon$ (see Equation (4.3)). They are drawn as single points on the upper axis. For the example presented in this chapter, the threshold was $\epsilon = 0.001$. However, the value can be interactively adjusted. To summarize, we can deduce the following characteristics of stress tensor fields from the Mohr diagram:

- Circles that are located left from the origin correspond to compressive stresses.
- Circles that are located right from the origin correspond to expansive stresses.
- Circles around the origin exhibit both compressive and tensile forces.
- The greater the circle's radius, the higher the level of shear.
- Circles degenerating to a single point exhibit no shear at all; they describe tensors with isotropic behavior.
- Circles on the three categorization axes represent isotropic, linear and planar tensors, respectively.

We achieve a further reduction of Mohr circles to be displayed through filtering. Similar to the work of [Crossno et al., 2005], we use the Euclidean distance between two points in shape space for filtering. A more advanced clustering applied to the shape space (Section 4.4.4) results in diagrams that only show circles that reflect the very basic pattern of the diagram.

Figure 4.9 shows all three shape-space diagrams side-by-side for the rotating-star dataset.



(a) (λ_1, λ_3)-Plot **(b)** Parallel coordinate plot **(c)** Mohr diagram

Figure 4.9: Rotating star (Section A.3). All diagrams reveal that the tensorfield is positive definite. From the PCP and the Mohr diagram, it can further be deduced that the stresses exhibit only perfectly linear behavior, i.e., the shape factor is $R = 1$.

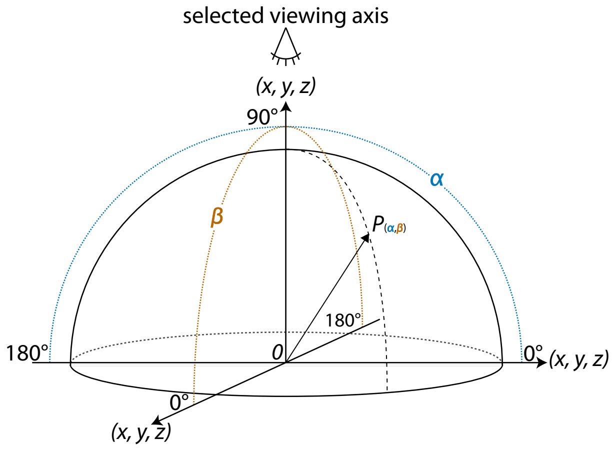


Figure 4.10: The input directions for directional histogram and directional scatterplot are projected on a hemisphere, flipping all vectors to the half space defined by the selected viewing axis. Each point $P_{(\alpha,\beta)}$ on the hemisphere represents all vectors spanning the angles α and β.

Directional Histogram

Directional histograms have been used to visualize the distribution of fiber orientations in sprayed concrete [Fritz et al., 2009] and for diffusion tensors in terms of rose diagrams and 3D scatterplots of the major eigenvector angles [Wu et al., 2004].

Figure 4.10 illustrates the directional histogram, which is used to analyze the distribution of the tensor's directional properties. To illustrate and quantify, for example, the major eigenvector field, the intersection between all major eigenvector fields and the unit sphere needs to be computed. Due to the non-oriented nature of eigenvectors only half of the sphere's surface needs to be considered. Therefore, all vectors are flipped to the positive half space of a pre-selected axis (x, y, z). To create the histogram, we use histogram binning. The number of intersections between vectors and a given surface patch on the sphere are counted, thus performing a region-dependent binning. For accurate results, a uniform subdivision of the surface into equal-sized bins is crucial. Triangle binning results in a discrete visualization of the counted frequencies, where each triangle is colored uniformly. The interpretation of the final plot depends on the selected viewing direction. In the 2D plot, the diagram's center corresponds to all vectors that are collinear with this viewing direction. An arbitrary point on the sphere's surface represents all vectors that span the angles α and β with respect to the two axes orthogonal to the selected viewing direction (Figure 4.10).

4.4.3 Hybrid Views

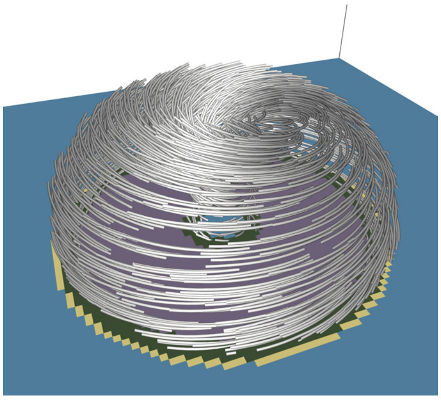


Figure 4.11: Label field and tensorlines started within the grey labeled region.

Spatial views represent the tensor field in its original 3D coordinates. The most basic method to display tensors in a spatial context is to use glyphs (see Section 3.3 for an overview of different glyph types). Although glyphs have the potential to show the entire tensor information, they fail to give a global continuous view of the tensor field (see Section 3.4 for an overview of global tensor visualization methods). Volume rendering methods, for example, provide a quick overview of the tensor field. However, in general, they only work on derived scalar values and do not contain directional information. Tensorlines depict eigenvectors in 2D and 3D. If two directions need to be depicted simultaneously, however, the resulting images soon get cluttered. Texture-based methods, finally, can only be applied to 2D slices or surfaces but not to the whole 3D data domain.

In this work, we therefore propose hybrid views and combine state-of-the-art visualization methods that are suited for stress tensor fields. The visualization techniques that we combine in our framework are listed below. The hybrid visualization is interactively steered by the label field (Figure 4.11) that is created and updated through selections in the diagram views, and by seed points that are randomly distributed in selected regions.

3D Visualization Techniques

The framework that was developed within the scope of this thesis supports the following 3D visualization techniques for tensors.

Tensor glyphs To create the images that are presented in this thesis, we used ellipsoids together with an antisymmetric mapping (see Equation (3.2) with $f = \arctan$). The GPU-based implementation is founded on the work Sigg et al. [2006].

Tensorlines Tensorlines (Section 3.4.3) are used to add directional information. To create hybrid visualizations, seeds are randomly placed inside regions that correspond to a specific label. Starting at these seed points, the line is integrated using a fourth-order Runge-Kutta scheme. The integration is stopped as soon as the line runs into an isotropic region, for example if $\tau_{max} < \epsilon$, or if the line would run outside the domain. The value for the threshold ϵ can be adjusted. For the depiction of the tensorlines, we use an existing *display module* of the visualization software Amira. The line rendering is based on a GPU-based implementation that is founded on the work Sigg et al. [2006].

Volume rendering Volume rendering serves as context view and to provide a quick overview of the tensor data. We implemented standard GPU ray-casting of scalar invariants, for example, the von Mises stress. To distinguish between compressive and expansive areas, we integrated the colormapping proposed by [Dick et al., 2009] into our volume ray-caster. The ray casting implementation is founded on the work of Kruger and Westermann [2003]. The eigenvalues are computed in the fragment shader during ray-casting.

2D Visualization Techniques

In addition to the 3D visualization techniques, we have integrated the following 2D visualization methods.

Texture-based methods To visualize 2D slices, we implemented LIC for tensor fields. Because LIC textures can only depict a single direction, we also implemented fabric textures [Hotz et al., 2004] to display two directions simultaneously. See Section 3.4.3 for more details and a discussion of the techniques. LIC as well as fabric textures were implemented on the GPU to allow for interactive slicing through 3D datasets.

Stress nets In addition to fabric textures, stress nets [Wilson and Brannon, 2005] also have the potential to display two directions simultaneously (Section 3.4.3). In contrast to fabric textures, stress nets are not based on an input noise texture. Hence, the stress net can be combined with the depiction of a scalar value.

Surface Visualization Techniques

To also visualize surface tensor fields, we developed novel visualization methods that are presented in Chapter 5 and Chapter 6. The Voronoi-based visualization (Chapter 6) offers the possibility to use textures to encode tensor properties.

4.4.4 Shape-Space Clustering

Diagram views as presented in Section 4.4.2 can support the identification of basic data characteristics of the underlying tensor data. Via brushing in the diagram views, representative subsets can be found, which then guide structured 3D visualizations depicting only the most relevant information that is contained in the data (Section 4.4.3). Sometimes, however, already the diagrams are cluttered making their manual exploration difficult. Under this perspective we analyzed the feasibility of nonparametric clustering to reduce the data to their basic structure.

For clustering, we decided on the mean-shift algorithm [Fukunaga and Hostetler, 1975]. Mean shift was first presented in the context of pattern recognition [Fukunaga and Hostetler, 1975], and later adapted for computer vision [Cheng, 1995]. A new trend of applying mean-shift on color feature spaces for image segmentation and filtering was initiated by the work of Comaniciu and Meer [2002]. Since then, it is a well-accepted and widely used technique in the field of image analysis. In contrast to k-means clustering, mean shift enables the detection of arbitrarily shaped clusters and no choice needs to be made for k, the number of clusters to be extracted. As one of our goals is to provide a framework that requires no prior knowledge of the datasets and their describing feature spaces, this property is crucial.

Choice of Shape Space

First, a shape space (Section 4.3) is selected for clustering. In this space, the tensor's shape is represented by a point, whose coordinates are called shape descriptors. To prepare the shape space for clustering, the shape descriptors are normalized so that the axes of the space are in the same range. Depending on the tensor field's definiteness, we normalize the values into the $[0, 1]$- or $[-1, 1]$ range. For some datasets, an additional logarithmic mapping is beneficial.

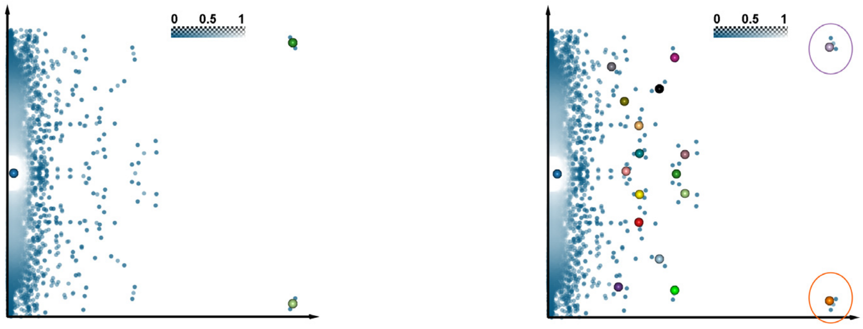
Mean-Shift Clustering

The basic idea is to consider the points corresponding to shape descriptors as representatives of a density function. The central part of the mean-shift algorithm is the estimation of the gradient of this density function and to find its maxima, without computing the density explicitly [Comaniciu and Meer, 2002]. Feature points that converge to the same maximum are assigned to the same cluster.

Given n feature points $\{f_j\}$ with $j = 1 \dots n$ and a seed point $\widetilde{x} \in \{f_j\}$ (the initial seed is determined randomly), the algorithm iteratively computes the mean-shift vector $m(x)$ by first defining a kernel K (i.e., a weighted neighborhood) around $\widetilde{x}$, and then computing the mean of the points within K, defined by

$$m(x) = \frac{\sum_{j=1}^{n} K(\widetilde{x} - x_j)x_j}{\sum_{j=1}^{n} K(\widetilde{x} - x_j)} - \widetilde{x}. \tag{4.6}$$

Finally, the kernel window is translated into the direction of maximum increase in the density, the mean shift vector $m(x)$. These steps are repeated until all points have converged to their corresponding mode. We use a Gaussian kernel, which is sufficient for our purposes since we do not need to support feature spaces with dimension higher than



(a) Adaptive kernel

(b) MAD kernel

Figure 4.12: Comparison of mean-shift clustering with two different kernels. Using a k-nearest neighbor approach, mean shift finds the prevalent maxima. Using the MAD to approximate the kernel radius r, outliers are found, too. The computation is based on the shear-shape-factor space of the two-point-load dataset.

$d = 3$. For higher-dimensional feature spaces, the use of anisotropic kernels (e.g. [Sixing et al., 2008]) is preferable as feature spaces become sparser the higher d is.

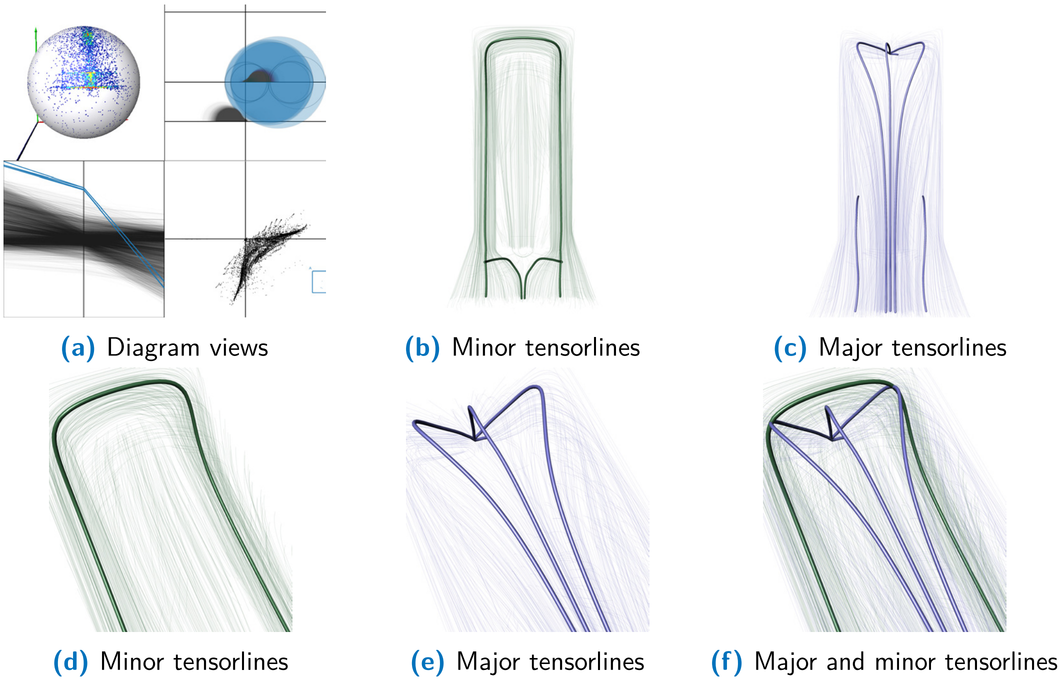
Although mean-shift is a nonparametric clustering algorithm, its accuracy is sensitive with respect to the kernel parameters. Since we do not require any prior knowledge of the density distribution, the choice of the radius r is difficult. We implemented two approaches to deal with this problem: a k-nearest neighbor approach and an estimation of r using the *median absolute deviation* (MAD). See Figure 4.12 for the different results of both approaches. The adaptive approach via k-nearest neighbors identifies the major maxima of the point distribution. Outliers, however, are missed. To also catch outliers and to achieve similar results as with topological methods, we set the kernel radius to a constant value according to the median absolute deviation (see, for example, [Elgammal et al., 2000]): $\mathrm{MAD} = \mathrm{median}|x_i - \widehat{x}|$. This measure has been proven to be stable with respect to outliers. Therefore, we first compute the median $\widehat{x}$ of the whole point distribution for every dimension of the feature space. We then have a MAD for every dimension. The bandwidth is finally set to $r = \min(\mathrm{MAD})$.

Comparison with scalar-field topology The mean-shift algorithm converges to the maxima of an underlying density function without computing this density explicitly. Thus, it decomposes a point distribution into several modes and classifies the points according to the mode to which they converge. For this reason, the mean-shift algorithm is related to scalar-field topology and can also be interpreted as topological decomposition of some feature space [Paris and Durand, 2007]. Discrete Morse theory [Günther et al., 2011] provides a way to extract critical points (maxima, minima, saddles) of a scalar function. The advantage of using discrete Morse theory to extract prevalent features is that a *consistent* hierarchy can be computed using the notion of *persistence* [Edelsbrunner and Harer, 2007]. It quantifies the stability of a topological feature. That is, a critical point is said to be persistent if it is separated from two other critical points by low saddles.

Representative Points from Clusters

Once the clusters have been computed, the next step is to find a meaningful representative for each cluster for the depiction in the diagram views. The most natural choice is to use the modes (or maxima) directly as representative for an entire cluster. However, sometimes the modes do not reflect the overall feature behavior of the cluster appropriately. In this work, we also used the nearest point to the weighted centroid.

4.4.5 Results



(a) Diagram views **(b)** Minor tensorlines **(c)** Major tensorlines

(d) Minor tensorlines **(e)** Major tensorlines **(f)** Major and minor tensorlines

Figure 4.13: Beam profile (Section A.6). Example for a selection in ordered shape space (a). Lines that were started at the selected seed points are bold (b-f). Line bundles are depicted additionally for a comparison of the representative lines and the bundle that they represent. The color violet is used for lines in tensile regions, and the color green is used for lines in compressive regions.

To the best of our knowledge there is no comparable system for visualization and analysis of stress tensor fields to which we could have compared our methods. Therefore, we will present several examples that demonstrate how our system works. Results with a specific application context are given in Section 4.5.

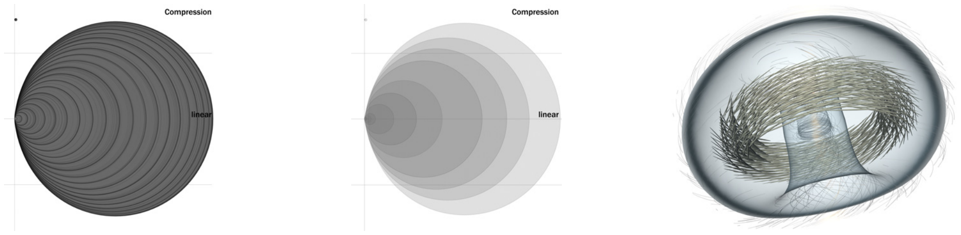
Figure 4.13 shows an example of how our system is used to find meaningful positions for the placement of seed points, which are used as starting points for the integration of tensorlines. We could observe that the points with the highest maximum shear are a good choice for this purpose. To validate this, we depict the tensorlines that were computed on the basis of our selection in ordered shape space (bold lines) as well as tensorlines that were computed on the basis of randomly distributed seed points (thin line bundles). It can be seen that the basic pattern is already revealed with very few tensorlines that were started at the selected seed points.

Figure 4.14 shows that shape-space clustering significantly reduces clutter in the Mohr diagram. The clustering also enables automatic classification of shape-related features. Hence, it provides an alternative to manual brushing to create hybrid visualizations of 3D stress tensor fields.

Figure 4.15 shows an example for a directional histogram binning directions of maximum shear (Equation (4.4). At the beginning of the simulation (Figure 4.15b), all shear directions exhibit a specific angle, which is depicted by a single circle in the diagram. In later time steps, the shear directions become more scattered and the strongly expressed direction splits into two maxima rings (Figure 4.15c). With further increasing time, these two maxima merge again resulting in one dominant ring (Figure 4.15d). According to our domain experts, the splitting is not physical. The assumption is that the visualization using the directional histogram reveals discretization artifacts.

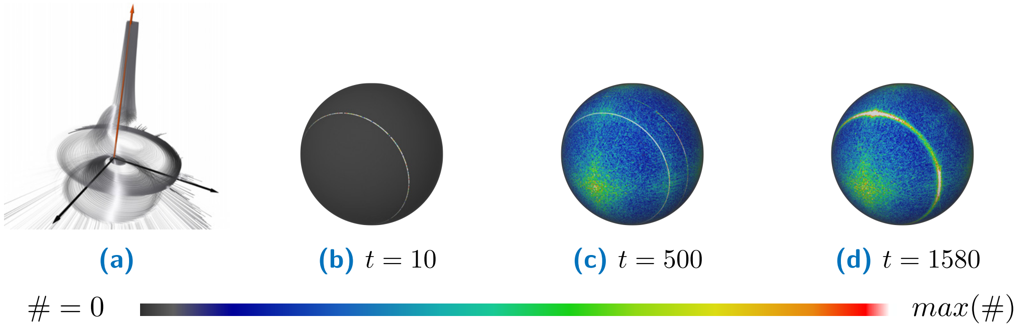
Figures 4.16 and 4.17 compare clustering results using the mean-shift algorithm and discrete Morse theory. In Figure 4.16, the result of mean-shift clustering is shown by the example of a Mohr diagram. In Figure 4.17, we computed the major and minor tensorlines on the basis of automatically computed clusters using mean-shift clustering of the shear-shape-factor space. The clusters correspond to regions of highest maximum shear.

Figures 4.18, 4.19 and 4.20 show the patterns that arise in the shape-space diagrams. The pattern of elastic materials is very similar in all plots, always revealing compressive, tensile and mixed stressed. The star dataset reveals a completely different behavior. Domain experts have shown strong interest in these patterns, because they allow the user to quickly evaluate basic properties from the data, for example, deviations from expected symmetry.



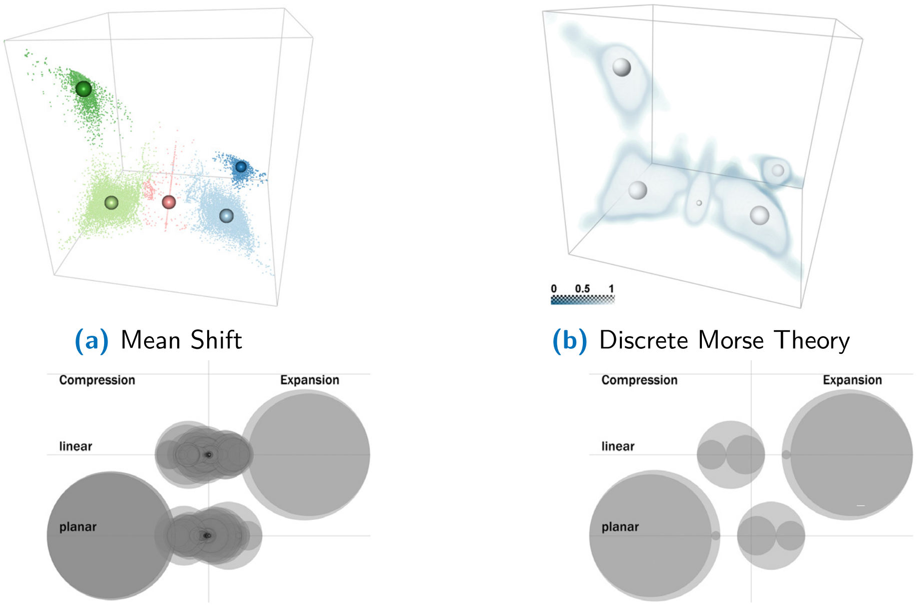
(a) Mohr diagram (unfiltered) (b) Mohr diagram (filtered) (c) Hybrid visualization

Figure 4.14: Rotating star (Section A.3). Due to the high gravitational forces inside the star, only compressive stresses occur, which is revealed clearly in the filtered Mohr diagram (b). It is further shown that only linear behavior, i.e., $\lambda_1 > \lambda_2 \approx \lambda_3$, happens. A few isotropic tensors (small circles at the highest axis) represent the boundary. The hybrid visualization (c) consists of a transparent isosurfaces for the maximum shear stress, glyphs depicted in regions of highest shear and tensorlines to depict the rotational behavior of the neutron star.



(a) (b) $t = 10$ (c) $t = 500$ (d) $t = 1580$

Figure 4.15: Rotating star (Section A.3). Directional histograms (b-d) for the shear vectors for three time steps. The colored triangles represent the number of data points (#) exhibiting a maximum shear direction falling into the triangle. The selected viewing axis for the directional histograms is the z-axis, which is highlighted in orange in (a).



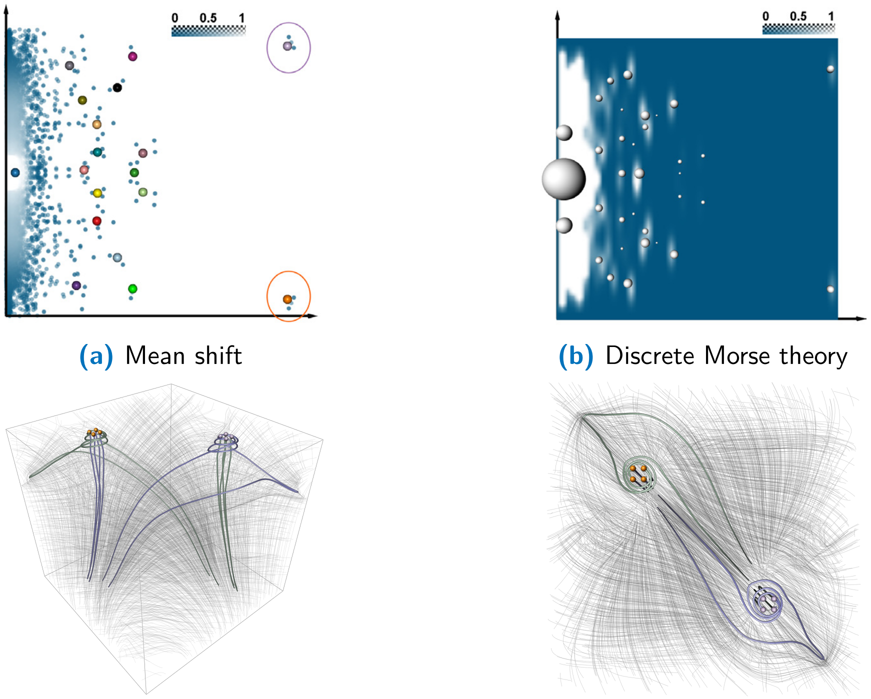
(a) Mean Shift

(b) Discrete Morse Theory

(c) Mohr diagram (unfiltered)

(d) Mohr diagram (filtered)

Figure 4.16: Two-point load (Section A.1). Clustering of the 3D ordered shape space. Maxima extracted by the mean-shift algorithm (a) and using discrete Morse theory (b). The radius of the spheres in (b) is scaled by persistence. The filtered Mohr diagram (d) only displays circles for extracted representatives. In this example we used the nearest point to the weighted centroid of each cluster.



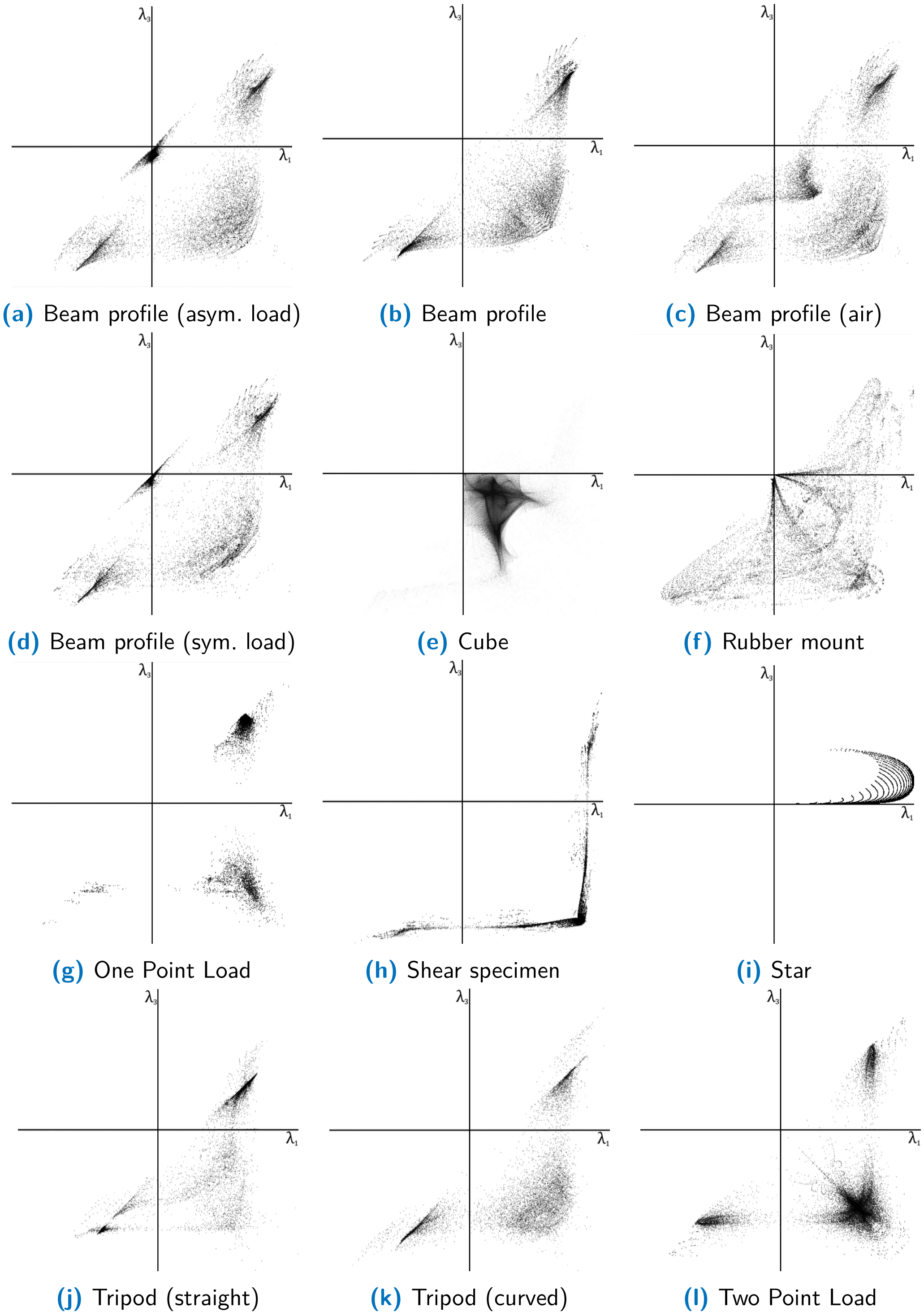
(a) Mean shift

(b) Discrete Morse theory

(c) Tensorlines

(d) Tensorlines

Figure 4.17: Two-point load (Section A.1). Clustering of the 2D shear-shape-factor space. Maxima extracted by the mean-shift algorithm (a) and using discrete Morse theory (b). The radius of the spheres in (b) is scaled by persistence. Tensorlines (c, d) started from two specific clusters are highlighted. Lines integrating the minor eigenvector are colored green and lines integrating the major eigenvector are colored violet. Seed points are colored according to their clusters, which are highlighted in (a).



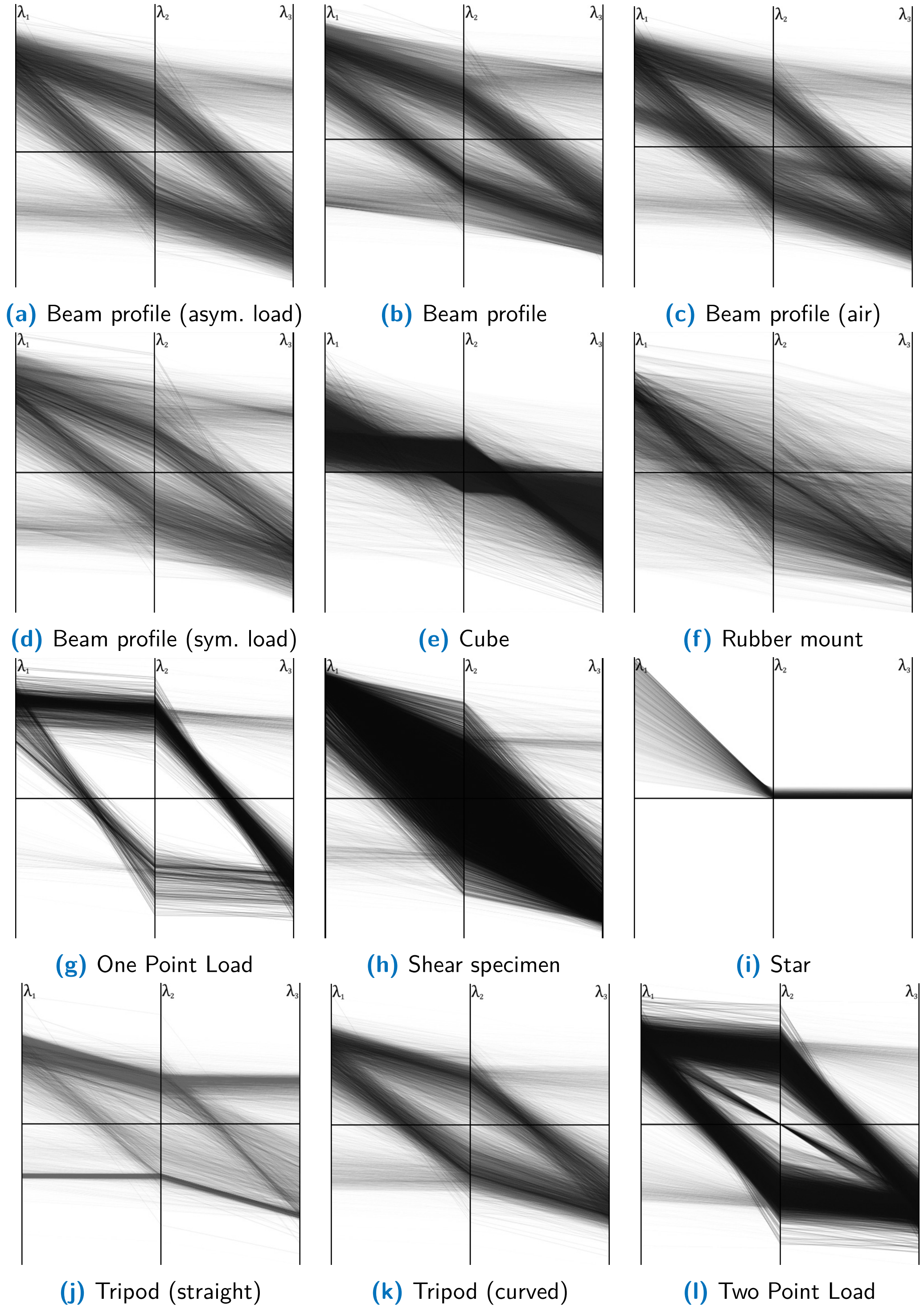
(a) Beam profile (asym. load) **(b)** Beam profile **(c)** Beam profile (air)

(d) Beam profile (sym. load) **(e)** Cube **(f)** Rubber mount

(g) One Point Load **(h)** Shear specimen **(i)** Star

(j) Tripod (straight) **(k)** Tripod (curved) **(l)** Two Point Load

Figure 4.18: Shape-space scatterplot. Patterns for various datasets.



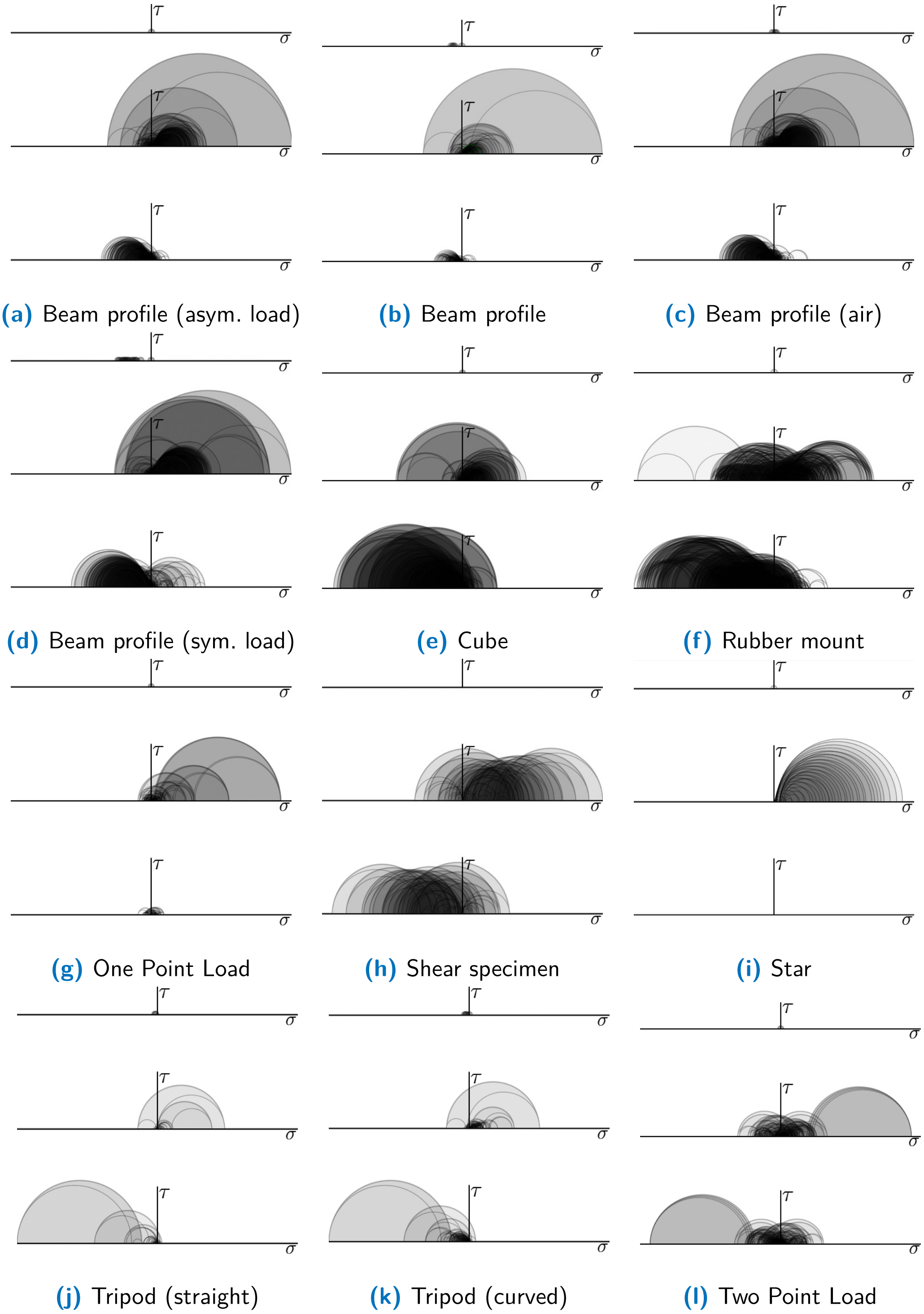
(a) Beam profile (asym. load) **(b)** Beam profile **(c)** Beam profile (air)

(d) Beam profile (sym. load) **(e)** Cube **(f)** Rubber mount

(g) One Point Load **(h)** Shear specimen **(i)** Star

(j) Tripod (straight) **(k)** Tripod (curved) **(l)** Two Point Load

Figure 4.19: Parallel coordinate plot. Patterns for various datasets.



(a) Beam profile (asym. load)

(b) Beam profile

(c) Beam profile (air)

(d) Beam profile (sym. load)

(e) Cube

(f) Rubber mount

(g) One Point Load

(h) Shear specimen

(i) Star

(j) Tripod (straight)

(k) Tripod (curved)

(l) Two Point Load

Figure 4.20: Mohr diagram. Patterns for various datasets.

4.5 Example Application: Visualization and Analysis of 3D Tensor Data from Structure Simulations

Within the last two years, the basic research results that have been presented in Section 4.4.5 have raised interest in the field of engineering. An interdisciplinary project was started using our framework to explore the value of tensor analysis in this field.

4.5.1 Background

Today, product development processes in mechanical engineering are almost entirely carried out via computer-aided simulations. That is, new technical systems are first tested and optimized at the computer. In the focus of this project are structure simulations, which compute forces and deformations. The results of these finite-element method (FEM) simulations are required to ensure the functionality and stability of single components of a technical system as well as of the technical system as a whole. Actual output of such simulations are the local deformations and stresses, which are described by stress and strain tensors. See Section 2.11 for an introduction of these types of tensors. The analysis of the simulation result is driven by the following questions:

- Does the construction element resist the applied load?
- Is the capacity of the construction element fully used?
- If the capacity is not fully used, can it be improved? For example with respect to weight and the amount of material that is needed?

Especially the growing demand for lighter materials and constructions requires a precise and efficient analysis of *material yielding*. A goal, for example, can be to develop a component (of a technical system) that has less weight, needs less material and is more stable with respect to a specific load on the basis of the analysis results.

4.5.2 Common Practice

To date, the basis of the analysis of structure simulations are scalar values that are derived from the tensor data, for example, the von Mises stress. Such a scalar value is then used to compare it with a characteristic threshold value given by a specific failure model (see also Section 2.11). In case, this analysis shows that the external forces exceed certain limits of material strength, there is a risk for structural failure of the component. However, by considering only derived scalar data a lot of additional information that is contained in the tensors is not used to analyze the yielding of a material. In the following, we would like to investigate the importance of tensors within this analysis process. The major question therefore is: Can the consideration of the complete tensor data add value to the analysis and interpretation of structure simulations?

4.5.3 Insights

> *A measure of an effective visualization can also be its ability to generate unpredicted new insights, beyond predefined data analysis tasks. After all, visualization should not only enable biologists to find answers but also to find questions that identify new hypotheses.* [Saraiya et al., 2005]

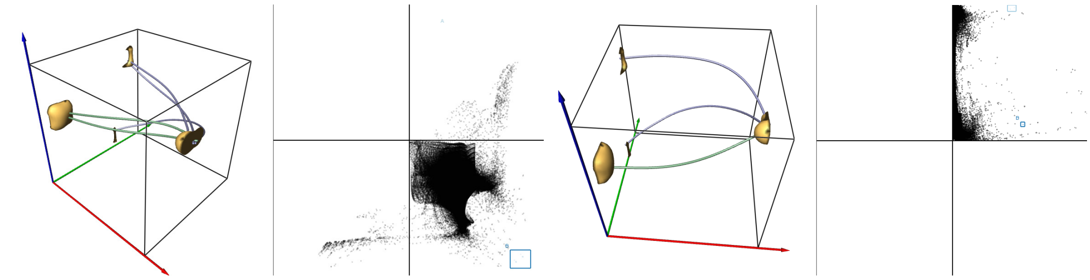
The following results are presented in terms of *insights* [Saraiya et al., 2005] that we gained when applying our methods to various datasets. All datasets are the result of FEM simulations. The data was provided by Prof. Dr. Markus Stommel and Marc Schöneich from Saarland University. See Appendix A for a description of the datasets. The following results relate to the stress tensor fields produced by the simulations.

Saraiya et al. [2005] define an insight to be “an individual observation about the data by the participant”. They name eight “quantifiable characteristics of each insight” that can be encoded in an analysis to evaluate different bioinformatic visualization systems. We use the following three characteristics from their list as a selection that is useful for our purposes:

- **Observation:** Findings about the data.
- **Hypothesis:** Hypotheses lead to new questions and new experiments that influence the design of the input data (from the engineering side) as well as the design of novel visualization algorithms (from the algorithmic side).
- **Directed versus Unexpected:** Directed insights are related to a specific analysis question or visualization task. Unexpected insights reflect the explorative nature of our approach (exploratory discoveries).

In contrast to the work of [Saraiya et al., 2005], we do not use these characteristics for quantification or comparison with other visualization systems for tensor fields. Instead, we use these characteristics for a classification of insights and to improve the structure of the evaluation.

The following list is a recording of discussions that were stimulated by visualization results generated within the framework that was presented in Section 4.4. The following persons participated in the discussions: Markus Stommel, Gerik Scheuermann, Marc Schöneich, Bernhard Burgeth and Ingrid Hotz.



(a) Selection in ordered shape space **(b)** Selection in shear-shape-factor-space

Figure 4.21: Cube (Section A.5). Selection of representative lines via explorative approach. The seed points to compute the tensorlines were determined via brushing in the ordered shape space (a) and in the shear-shape-factor-space (b). The force transmission point and the three stationary points are marked by isosurfaces.

Cube

See Appendix A.5 for a description of the cube dataset.

Observation First images (Figure 4.21) that were presented to the domain experts lead to the assumption that the route of the tensorlines reflects the way of the forces through the dataset.

Unexpected insights An unexpected insight that was triggered from Figure 4.21 was that the tensorlines are curved and that they progress from the force transmission point to the three stationary points. The domain experts assumed that they are straight.

Hypotheses The fact that the tensorlines directly connect the force transmission point and the three stationary points lead to the following hypothesis: With the knowledge of the tensorlines and, hence, the progression of forces that act within a construction element, a better stiffening of construction elements can be achieved.

Tripod

To validate the hypotheses that a better stiffening of construction elements can be achieved with the additional knowledge of the tensorlines, the domain experts developed two new tripod-geometries (Figure 4.23):

- A tripod geometry with straight connections between the force transmission point and the stationary points (Figure 4.23a).

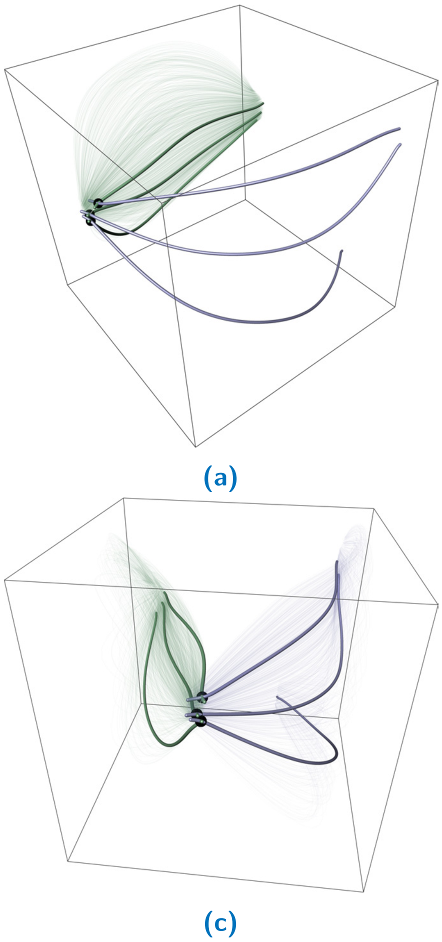
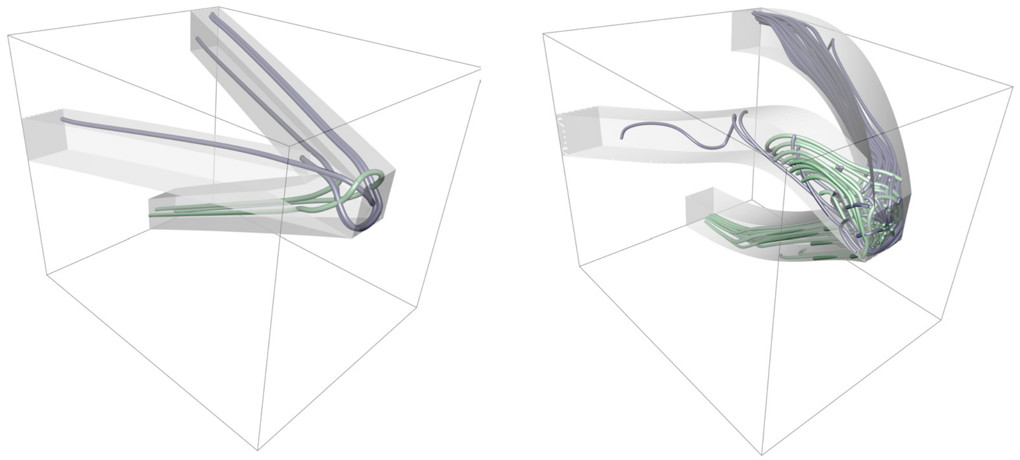


Figure 4.22: Cube (Section A.5). Manual selection of representative lines via explorative approach. The seed points were determined via brushing in shear-shape-factor-space. Figures (a) to (d) depict the same selection from various perspectives. Lines that were considered as good representatives are bold. Line bundles are depicted additionally for a comparison of the representative lines and the bundle that they represent. The color violet is used for lines in tensile regions, and the color green is used for lines in compressive regions.



(a) Tripod (straight legs)

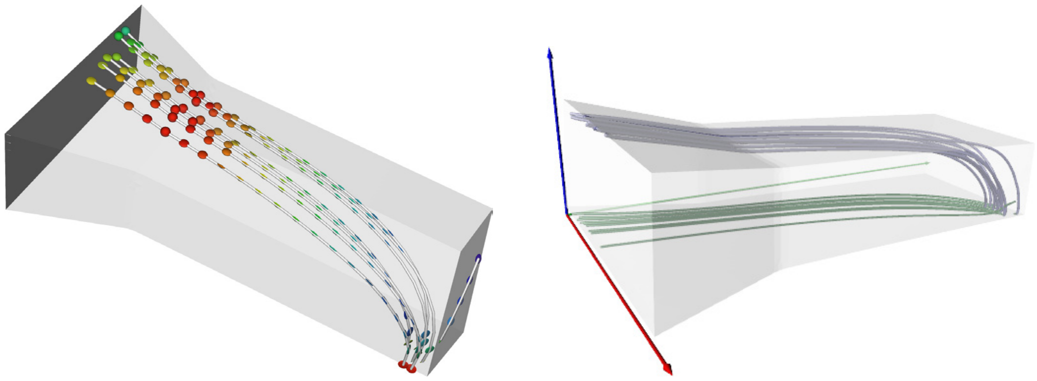
(b) Tripod (curved legs)

Figure 4.23: Cube (Section A.5). Two different tripod geometries. In (a), the force transmission point and the three stationary points are connected in a straight fashion. In (b), the force transmission point and the three stationary points are connected in a curved fashion. The color violet is used for lines in tensile regions, and the color green is used for lines in compressive regions.

- A tripod geometry with curved connections that were modeled on the basis of three representative tensorlines (Figure 4.23b). These tensorlines were determined in an explorative manner for the cube dataset as shown in Figure 4.22.

Observation For the tripod with straight legs, the progression of the tensorlines is straight too. This is, what the domain experts expected. In this case, the simulated deformation complies with the applied load. For the tripod with curved legs, two legs behave as expected: the tensorlines are curved and connect the force transmission point and the stationary points. In the third leg, however, a swirl of the tensorlines occurs. In the simulated deformation, a torsion of the element happens. It can be deduced that the representative tensorlines that are depicted in Figure 4.22 were chosen badly, because a torsion of the element needs to be avoided. However, the negative results lead to exciting new hypotheses.

Hypotheses Although we could not yet prove for the cube dataset that the knowledge about the tensorlines leads to more stiffness, an interesting observation was the swirl of the tensorlines. Since we could observe a torsion in the simulation, this lead to the hypothesis that the tensorlines may be an indicator for the kind of deformation that the construction element undergoes. If this is the case, tensorlines are a valuable tool for validation purposes.



(a) Major tensorlines and ellipsoids

(b) Major and minor tensorlines

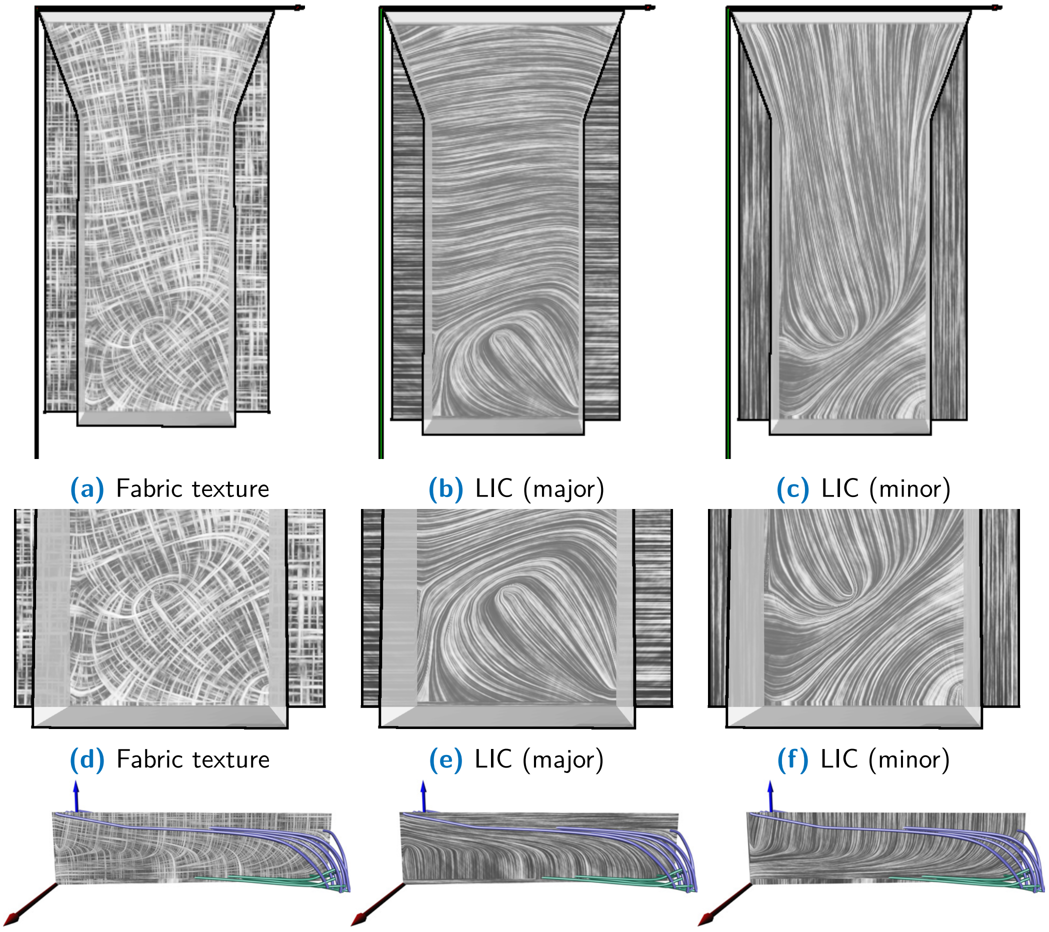
Figure 4.24: Beam profile (Section A.6). The seed points for the computation of the tensorlines were determined via brushing in the shape-space scatterplot. The figures show that the lines connect the force transmission point in the lower right with the part of the geometry that was fixed. In (a), tensorlines integrated along the major eigenvector field are enhanced with ellipsoidal glyphs. In (b), major (violet) and minor (green) tensorlines are depicted and colored accordingly.

Beam Profile

We applied our methods also to a geometry that represents a more realistic example for a construction element. See Appendix A.6 for a description of the dataset.

Observation Considering our concept of providing various views on a single dataset, the experts mentioned that especially the multitude of visualization techniques is important. This observation came up when looking at the results for the beam profile dataset. The reason is that for this dataset, the LIC textures (Figure 4.25) in the yz-plane show more interesting structures than the tensorlines (Figure 4.24). We could again observe that the progression of the tensorlines comply with the real deformation the material undergoes, which strengthens our hypothesis that the tensorlines describe the progression of the forces that act within a material.

Hypotheses The depiction of tensorlines superimposed with glyphs (Figure 4.24a) was rated as interesting. A new hypotheses from the visualization side is that the additional usage of glyphs can support the decision which tensorlines are good representatives. A hypotheses from the engineering side was that the LIC textures (Figure 4.25) can guide the design of rip structures that lead to stiffer and lighter construction elements.



(a) Fabric texture (b) LIC (major) (c) LIC (minor)

(d) Fabric texture (e) LIC (major) (f) LIC (minor)

(g) Fabric texture (h) LIC (major) (i) LIC (minor)

Figure 4.25: Beam profile. Figures (a) to (f) show fabric and LIC textures for a slice in the yz-plane. Figures (g) to (i) show fabric and LIC textures for a slice in the xz-plane superimposed with major and minor tensorlines. In all examples, the force transmission point is in the lower right.

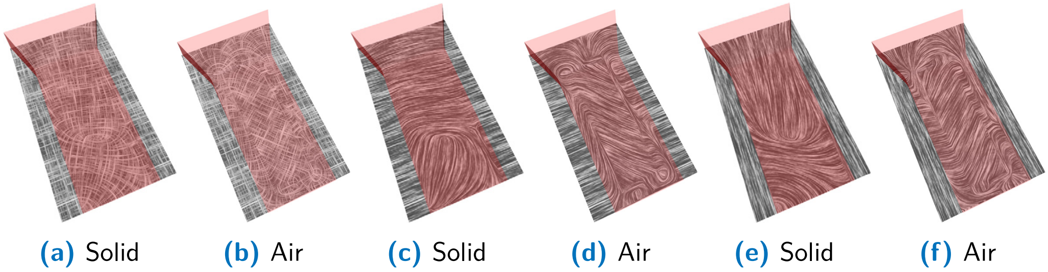


Figure 4.26: Beam profile vs. profile partially filled with air.LIC textures in the yz-plane: fabric texture (a, b), LIC textures integrating the major (c, d) and minor (e, f) eigenvector fields, respectively.

Beam Profile (partially filled with air)

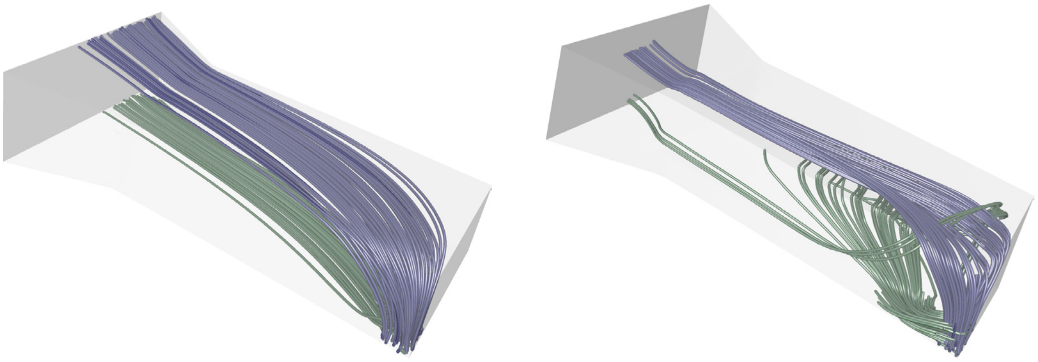
To validate the hypotheses that the LIC textures provide a hint for the design of better rip structures, a new dataset was generated on the basis of the beam profile. In this new simulation, the construction element was partially filled with a material that resembles air to create an example that has no stiffness support.

Observation When observing the LIC and fabric textures (Figure 4.26), a first observation was that there is a significant difference between the solid profile and the profile that is partially filled with *air*. We can observe a sharp transition between the different media and a completely different pattern for the construction element filled with air. The tensorlines for the beam profile shown in Figure 4.27a again comply to the expected progression. The tensorlines for the profile filled with air depicted in Figure 4.27b differ. The lines that integrate the minor eigenvector field (green) again show a swirl which strengthens the hypotheses that the behavior of tensorlines indicates what kind of deformation a material undergoes. For this example dataset, which has no stiffness support, the simulated deformation results in a torsion.

Hypotheses The progression of the tensorlines shown in Figure 4.28 gives a hint for a well-designed rip structure that leads to a stiffer material. That is, for this example, one would design a rip structure with an approximately 45 degree angle. A course of maximum shear within a 45 degree angle is what we would have expected, because we have a torsion of the material.

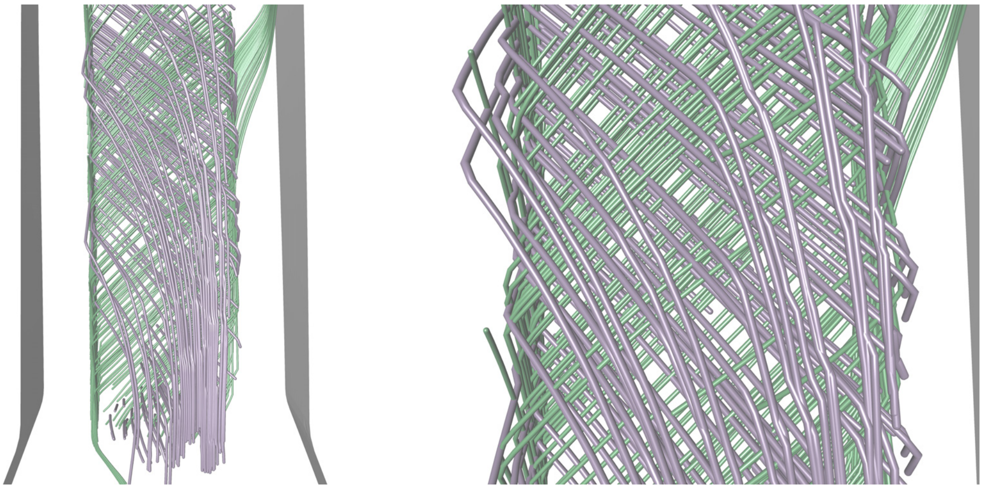
Shear Specimen

For completeness, we further analyzed a construction element for which we know that it is well designed. See Appendix A.7 for a description of the dataset. The question



(a) Tensorlines **(b)** Tensorlines

Figure 4.27: Beam profile (a) and profile filled with air (b).

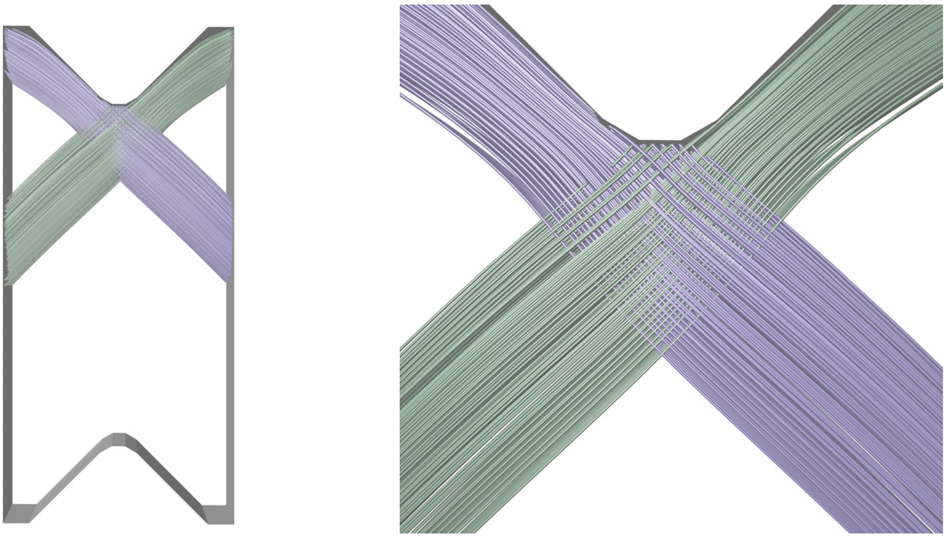


(a) Tensorlines **(b)** Tensorlines

Figure 4.28: Profile filled with air. The progression of the tensorlines (a, b) might give a hint for a well-designed rip structure that might lead to a stiffer material.

was whether the tensorlines can validate that this example constitutes a good design of a construction element.

Directed insight Figure 4.29 confirms the desired load case, which is a progression of the forces in a 45 degree angle. This again strengthens the hypotheses that tensorlines describe the route of the forces.



(a) Tensorlines

(b) Tensorlines

Figure 4.29: Shear specimen. Figures (a, b) show the major and minor tensorlines started in the region of highest shear.

4.6 Discussion

Interactive visual data analysis is widely used in visualization. However, for stress tensor fields it has received very little attention in the past (see also Section 3.6). In this chapter, we have presented a concept that adapts the idea of multiple linked views to stress tensor fields. To achieve this, we interpreted the shape- and direction space as feature spaces that link the abstract tensor data and their depiction in 2D diagrams and 3D hybrid views. Interactivity is crucial when aiming for an explorative approach. Therefore, all visualization methods were designed to run on the GPU. We also investigated mean-shift clustering and scalar field topology to automatically identify relevant properties of tensor fields. We proposed to first define a feature space which is subsequently classified. Finally, we presented insights that were gained by applying our methods to stress tensor fields resulting from structure simulations. We have shown that the presented visualization and analysis tools open up new perspectives on the stress tensor fields, thereby generating new questions and hypotheses about the data.

One of the main challenges for the visualization of three-dimensional tensor fields is that it is almost infeasible to squeeze all the information in one spatial representation. We have demonstrated that this problem can be tackled by distributing all degrees of freedom of a tensor in various views (diagrams and spatial views) instead of trying to integrate all the information into a single visualization. Furthermore, brushing in the diagram views helps reduce the data that is visualized. In combination with hybrid visualizations, regions of interest are emphasized and the complex data is structured. By having feature spaces in the core of the presented framework, specific tensor properties can be extracted on-the-fly. We discovered that especially outliers in the diagrams are of interest to determine meaningful positions for the placement of seed points. A novel assumption that arose during many analysis sessions is that very large parts of the tensor fields do not contain any important information. This could be exploited in the future to create even more meaningful stress tensor visualizations and to alleviate clutter.

Our results and the discussion with domain experts have confirmed the need for powerful visual exploration and analysis tools for tensor fields. The experts confirmed that they did not consider looking at the tensor data because appropriate tools were missing. In addition, we identified the following reasons and bottlenecks, why the tensor data has been neglected in the past:

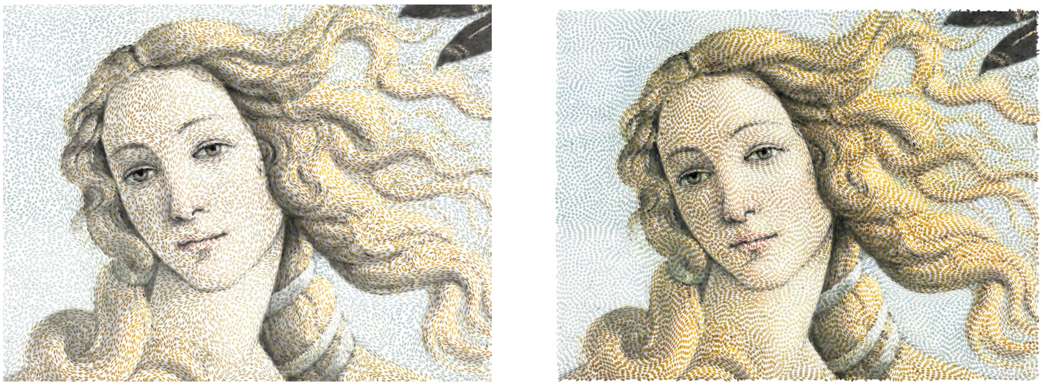
- The use of mathematical concepts and failure models that rely on the original tensor is not common in engineering applications, because failure models that only rely on scalar values are much simpler and it is faster to evaluate them.
- Engineers have strict timing constraints and must quickly decide whether a material or a whole technical system meets the requirements that are posed. It is not feasible for them to analyze all quantities that arise in a simulation.
- Visualization and analysis tools that were specifically designed for the analysis of tensors in engineering applications were not available.

Considering the first point in this list, we could not yet prove that the analysis of stress tensor fields adds value to the analysis of material yielding or that it is more efficient than using only a scalar value that was derived from the tensor data. However, our results allow the assumption that the presented visualizations can indicate failures in simulations. In the context of engineering, we believe that tensor visualization can also give hints that a construction element was designed badly. Overall, the results have motivated us and our cooperation partners to continue investigating, whether it is useful to consider stress tensor field visualizations in addition to traditional failure models in the future. The second point in this list, the fact that engineers have strict timing constraints, should be kept in mind when designing new visualization methods. Our experience is that domain experts favor simple visualization techniques like scatterplots. In object space, a sparse usage of lines and glyphs at specific locations is preferred, which motivates the use of a label field to determine focus and context regions. Moreover, the domain experts rated the brushing-and-linking as extremely helpful to get used to the data.

Although we show the applicability of our approach by the example of stress tensor fields from engineering applications, we believe that the basic idea of the presented concept is applicable to other types of tensors in other application contexts, too.

5

Anisotropic Sampling in Planar and Two-Manifold Domains



(a) Initial anisotropic sampling

(b) Final anisotropic sampling

Figure 5.1: Anisotropic sampling. First, an initial sample distribution is computed (a). Second, the initial distribution is iteratively refined (b).

5.1 Introduction

For glyph-based visualizations (Section 3.3), a basic question is *where* to place the glyphs to provide all details as well as to convey the evolution of tensors across the field. If glyphs are placed at the grid positions, the pattern of this grid becomes visible in the final visualization. This disturbs the perception of the field's continuous behavior and circumvents the discovery of regions with similar behavior. Therefore, the goal of the methods presented in this chapter is to find positions for glyphs that are well distributed and that are independent from the grid positions. A good placement is found if the glyphs do not overlap and significant gaps between them are avoided (Figure 5.1).

Another visualization method for tensors is related to line integral convolution (LIC). Such texture-based visualization techniques (Section 3.4.4) are based on noise images as input. The quality of these input textures is crucial for the final result. Just as for glyph placement, patterns and holes in theses textures result in visible artifacts.

The questions, where to place glyphs and how to generate high quality input noise textures, are strongly related to questions in the field of *blue-noise sampling*. Since the 1980s, aperiodic point distributions with varying density and blue-noise properties are a central research topic in computer graphics [Lagae et al., 2008]. Methods range from simple dart throwing techniques [Cook, 1986] to real-time tile-based methods [Kopf et al., 2006; Lagae and Dutré, 2008]; applications range from sampling theory to mesh generation and illustrative rendering. In general, these methods focus on isotropic samples and are hard to extend to work with anisotropic samples. However, when visualizing tensor fields using glyph- or texture-based approaches, the needed samplings are always anisotropic.

This chapter presents two approaches to generate unstructured distributions of elliptical samples that follow a Poisson-disk distribution avoiding overlaps and holes between the samples as well as periodic patterns. We believe that glyph- and texture-based visualizations only work effective in 2D domains. Therefore, the following methods only deal with slices and triangulated surfaces.

This chapter is based on the following two papers: *Particle-Based Anisotropic Sampling for Two-Dimensional Tensor Field Visualization* [Kratz et al., 2011a] and *Anisotropic Sampling of Planar and Two-Manifold Domains for Texture Generation and Glyph Distribution* [Kratz et al., 2013b].

Structure This chapter is organized as follows. We first discuss the most relevant previous work in Section 5.2. Section 5.3 introduces the concepts and basic ideas which underlie the two approaches that are presented in this chapter. Subsequently, the two approaches for generating anisotropic sample distributions are presented. Both approaches start with an initial sampling distribution that is iteratively refined. The main difference of both approaches is the kind of refinement. The first algorithm is a particle-based approach that is restricted to planar domains (Section 5.4). Such particle-based approaches interpret the samples as moving particles that exert attracting and repelling forces. The second algorithm is a geometry-based approach (Section 5.5). It relies on the definition of an anisotropic triangulation. Here, we propose a gravitational-centered relaxation to refine the initial sample distribution.

5.2 Related Work

The generation of sample distributions with specific properties is a central research topic in computer graphics with applications in rendering, visualization and geometry processing. A huge amount of related work exists, most of which focuses on very specific requirements and applications. The algorithms that are presented in this chapter are also related to mesh generation and remeshing. Since an exhaustive review of related methods from sampling theory and geometry processing is out of scope of this work, we refer the reader to the following survey articles. For a deeper insight into remeshing approaches, we refer to the article by [Alliez et al., 2005]. And for an overview about the creation of aperiodic isotropic sample distributions, we refer to the articles by [Lagae and Dutré, 2008; Lagae et al., 2008, 2010]. In the following, we focus on work that is related to the approaches presented in this chapter.

Many previous work in generating sample distributions focuses on the generation of Poisson-disk distributions [Lagae et al., 2008]. Distributions with this property cover the sample domain densely while maintaining a minimum separation given by a specified radius, the Poisson-disk radius. Common basic algorithms to create such distributions are dart throwing [Cook, 1986] and Lloyd relaxation [Lloyd, 1982]. These algorithms are expensive and have a high memory requirement. Thus, several enhanced methods for generating Poisson-disk distributions have been presented. Among these, tile-based methods [Kopf et al., 2006; Lagae and Dutré, 2008] have gained special interest. Tile-based methods generate a small set of tiles *once*, where each tile has the Poisson-disk characteristics. The tiles can then be used in interactive settings to generate point distributions with arbitrary density. Thus, tile-based methods have a low memory requirement and are suited for real-time applications. Unfortunately, these methods cannot be easily generalized to the anisotropic case, where the samples vary in size, shape and orientation and, therefore, would require a much larger set of tiles and the handling of many special cases.

The objective of mesh generation most often is to find a mesh that is as coarse as possible and as fine as needed to represent all properties of the shape of the underlying object. In this context, anisotropic meshes are beneficial for simulating functions with a strong directional character. The strength of such meshes is that the shape of the mesh elements can be controlled by an underlying anisotropy function (metric tensor field). Various approaches to tackle this problem have been presented [Bossen and Heckbert, 1996; Shimada et al., 1996; Li et al., 1999]. In [Shimada et al., 1996], for example, anisotropy is incorporated in the common *circumcircle test* to compute anisotropic Delaunay meshes. Other solutions build on the construction of an anisotropic Voronoi diagram, for example, [Labelle and Shewchuk, 2003; Boissonnat et al., 2008]. In [Labelle and Shewchuk, 2003; Du and Wang, 2005], the triangle mesh is then defined as the dual of the Voronoi diagram. These approaches are mathematically sound but computationally expensive and hard to generalize to the two-manifold case.

In contrast to mesh generation, remeshing algorithms have the goal to improve a given mesh with respect to specific criteria. Most advanced remeshing approaches rely on surface parameterizations. Similarly, parameterizations are used for anisotropic surface sampling [Li et al., 2010] and meshing of curved surfaces [Shimada et al., 1996]. While for remeshing algorithms an accurate representation of surface details is of high importance, this is not the case for our purpose. This simplifies the sampling process in many ways. Hence, our approach does not require an expensive parameterization.

In visualization, anisotropic sample distributions have been computed to guide the positioning of glyphs [Kindlmann and Westin, 2006; Hlawitschka et al., 2007; Feng et al., 2008]. The more a sample distribution reflects the continuous behavior of an input vector- or tensor field, the more informative the final glyph-based visualization becomes. Examples for applications within the context of tensor field visualization have been presented in [Chen et al., 2011; Goldau et al., 2011]. Building on the particle-based method of [Shimada et al., 1996], Kindlmann and Westin [2006] present an approach for planar domains and volumes. Here, the input diffusion tensor field is mapped to a potential energy field determining inter-particle forces. For the planar case, this approach was extended in [Hlawitschka et al., 2007], which focuses on an improved initial sampling and interactivity by using an isotropic Delaunay triangulation. In this chapter, we show that more meaningful neighbor computations are achieved with anisotropic triangulations. A limitation of these particle force-based algorithms is that they often suffer from many non-intuitive parameters, and that they are sensitive to parameter choices. Their strength is that they are easier to extend to 3D domains.

An alternative to particle-based approaches are geometry-based algorithms. These are often based on Lloyd relaxation [Lloyd, 1982; Du et al., 1999]. For example, Feng et al. [2008] use Lloyd relaxation to generate anisotropic samplings of planar domains. As Lloyd relaxation relies on Voronoi diagrams, this approach requires the computation of an *explicit* generalized Voronoi diagram including the handling of orphans. One advantage of such a relaxation-based method over particle-based methods is that it is almost parameter-free. However, the computation of an explicit generalized Voronoi diagram on surfaces including the handling of orphans is computationally too expensive for our purposes. Building upon ideas of [Feng et al., 2008], Li et al. [2010] extends dart throwing and relaxation to anisotropic sampling. For this purpose, they replace the

Euclidean distance metric by a geodesic one. In their work, relaxation is only applied on planar domains, while surface sampling is restricted to dart throwing in combination with surface parameterizations. The triangle-based approach presented in Section 5.5 avoids the need of surface parameterizations. Moreover, it is time-efficient and needs only few parameters. It also uses relaxation in the two-manifold domain to equalize distances.

5.3 Underlying Concepts and Basic Ideas

This section provides the basic concepts and ideas underlying the two algorithms that are presented in this chapter. Furthermore, it summarizes the relevant prerequisites and assumptions.

Input of both algorithms is:

- A geometric domain Ω that can be either a planar domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$, or a two-manifold domain $\Omega \subset \mathbb{R}^3$ with or without boundary. Note that the particle-based approach presented in Section 5.4 is restricted to planar domains.
- A 2D or 3D input field given on a uniform, triangulated or tetrahedral grid. The input field can be either scalar-, vector- or tensor-valued.
- Optionally, a spatially varying importance function to create adaptive sample distributions.

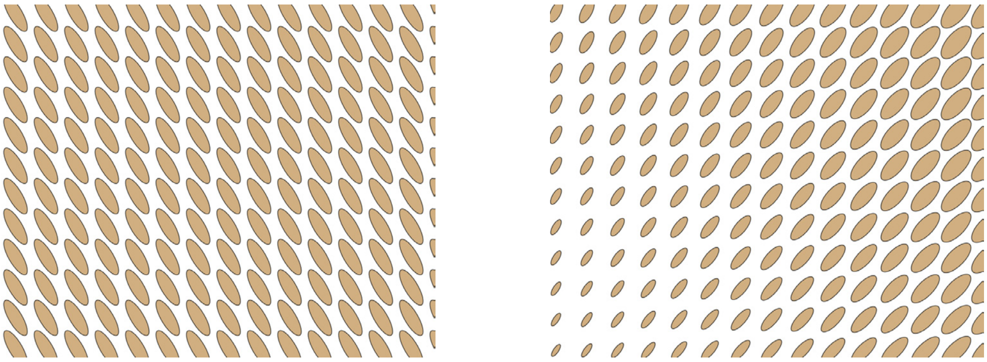
Output of both algorithms is a point distribution that fulfills the Poisson-disk characteristics with respect to the underlying metric tensor field.

5.3.1 Notations

In this chapter the following notations are used: A general second-order tensor is denoted as $\mathbf{T}$; the metric tensor is denoted as $\mathbf{M}$. Bold lower-case letters describe sample positions, for example, $\mathbf{p}$ and $\mathbf{q}$ and vectors $\mathbf{v}$ and $\mathbf{w}$. A sample set is S. The Delaunay triangulation is denoted by D and a surface mesh by $\mathcal{X}$. Small Greek letters, such as α and β, denote angles.

5.3.2 Metric Tensor Fields

Regardless of the input field (scalar, vector, or tensor), the algorithms that are presented in Section 5.4 and in Section 5.5 work on a metric tensor field, which is either a direct



(a) $\mathbf{M} = \mathbf{M}(\mathbf{p}) = const$

(b) $\mathbf{M} = \mathbf{M}(\mathbf{p})$

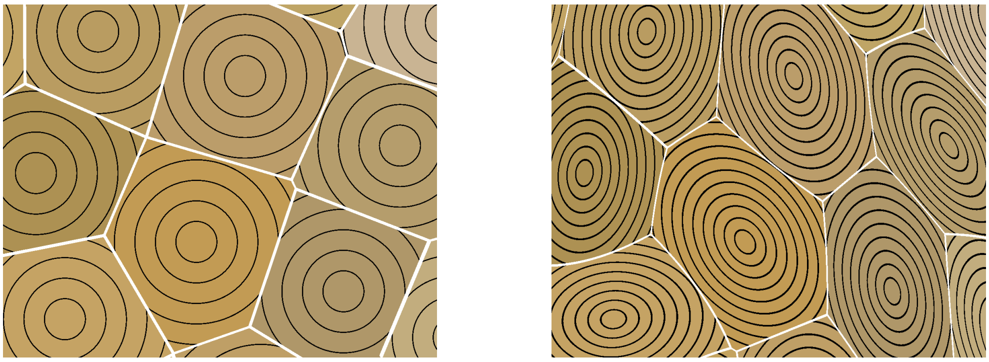
Figure 5.2: Elliptical samples. Given a constant metric tensor field, all samples have the same shape, size and orientation (a). Given a spatially varying metric tensor field, size, shape and orientation of the samples depend on their position (b).

mapping of the input field or is derived from it (see Section 5.3.4). The size, shape and orientation of the samples depend on the sample positions $\mathbf{p} \in \mathbb{R}^d$ and the local metric $\mathbf{M}(\mathbf{p}) \in \mathbb{R}^{n \times n}$ at position $\mathbf{p}$ (Figure 5.2). Here, d and n are 2 in the planar case and 3 in the two-manifold case. In either case, the metric tensor $\mathbf{M}$ is represented by an $n \times n$ symmetric positive-definite matrix. To reconstruct $\mathbf{M}$ at arbitrary sample positions in the planar domain, we use component-wise linear respectively bilinear interpolation. On surfaces, the tensors are interpolated on a per-triangle basis using barycentric coordinates.

The metric tensor describes anisotropic distances between sample positions $\mathbf{p} \in \Omega$. It can be imagined as an ellipse or ellipsoid, respectively, which is scaled according to the reciprocal eigenvalues and oriented according to the eigenvectors of $\mathbf{M}$ [Feng et al., 2008]. Consequently, ellipses and ellipsoids build our basic sample shapes.

We assume that the metric tensor field does not vary strongly within a small *local* environment. The idea behind this assumption is that reasonable visualization results can only be achieved if the variation of the tensor field, compared to the size of the samples, is relatively small. Only with this assumption, the samples are valid representatives for the part of the metric tensor field which they cover (see also [Feng et al., 2008]). Globally, in contrast, sample sizes and shapes can differ significantly (Figure 5.2b).

In the following, we use the term *sample* for elliptical or ellipsoidal sample, and *metric* for a 2D or 3D metric tensor.



(a) Isotropic space

(b) Anisotropic space

Figure 5.3: Undistorted Euclidean space (a) and distorted metric space (b). The black lines are the isolines corresponding to the local metrics.

5.3.3 Isotropic vs. Anisotropic Space

Before we go into the details and present the sampling strategies that were developed in this thesis, we would like to point out the differences of isotropic and anisotropic space.

In this work, we map an arbitrary type of tensor to a *metric tensor* (Section 5.3.4) that describes distances between sample positions $\mathbf{p} \in \Omega$. It can be considered as distortion of a circle into an ellipse. Hence, we distinguish between the *undistorted* Euclidean or isotropic space (Figure 5.3a) and the *distorted* metric or anisotropic space (Figure 5.3b).

A special case arises when the metric tensor field does not vary across the domain Ω, that is, $\mathbf{M} = \mathbf{M}(\mathbf{p}) = \mathit{const}$. For this case, we can transform the anisotropic space into the isotropic space via linear transformation using the inverse metric $\mathbf{M}^{-1}$. Common methods for isotropic point distributions could be used in this space.

In this work, we require our technique to handle spatially varying anisotropies $\mathbf{M} = \mathbf{M}(\mathbf{p})$. Therefore, a global transformation of the complete space is not possible.

5.3.4 Anisotropy Design

In this section, we explain how metric tensor fields are generated in this work. As said in Section 2.5, a tensor

$$\mathbf{M} = \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{U}^T \tag{5.1}$$

can be described by its eigenvalues and eigenvectors, where $\mathbf{U}$ is a rotational matrix whose columns are the eigenvectors of $\mathbf{M}$, and $\mathbf{V}$ is a diagonal matrix whose diagonal elements are the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of $\mathbf{M}$.

If the input data are scalar or vector fields, we first determine $\mathbf{U}$ and $\mathbf{V}$ and then compose them again to get $\mathbf{M}$ (Equation (5.1)). In more detail, we derive the information about the three eigenvectors and their scaling (represented by the eigenvalues) from the input data. If tensor fields are given as input, we first decompose the tensor into $\mathbf{U}$ and $\mathbf{V}$. Then, we have the possibility to manipulate (map, scale) the eigenvalues to finally compose the parts again to get $\mathbf{M}$.

Scalar fields To derive metric tensor fields from color images, we compute the image's gradient as minor eigenvector. The vector field orthogonal to the gradient field represents the major eigenvector field aligned with the edges of the input image. Anisotropy design mainly subsumes the scaling and/or mapping of the eigenvalues.

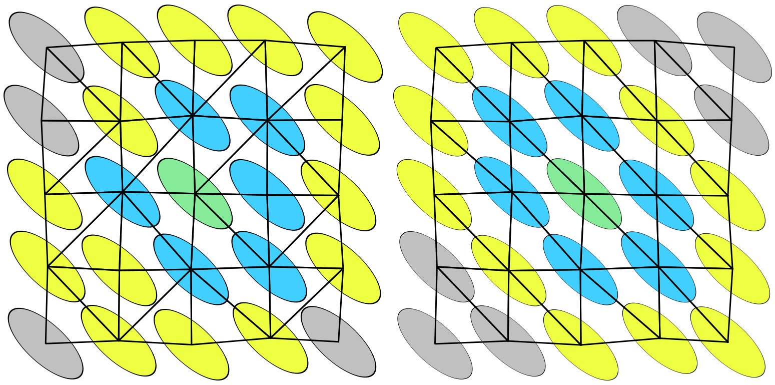
Vector fields If the input data is a vector field, the vector field itself becomes the major eigenvector field and the vector field orthogonal to the input field serves as medium eigenvector field. If *surface* vector fields are given as input, a 3D tensor is generated. Then, the minor eigenvector field is represented by the surface normal where the corresponding eigenvalue is set to $\lambda_3 = \epsilon$. The value that is chosen for ϵ must be a small value but larger than zero due to numerical reasons.

Tensor fields Finally, arbitrary tensor fields are possible input data. In that case, we first decompose the input tensor $\mathbf{T} = \mathbf{U} \cdot \mathbf{V} \cdot \mathbf{U}^T$ into its eigenvalues and eigenvectors. Then, the eigenvalues are scaled through eigenvalue mapping [Hotz et al., 2004].

5.3.5 Anisotropic Triangulations

Since Delaunay triangulations provide a useful notion of neighborhood relations and region of influence, they play a central role in the presented approaches. In the following, a variant of the Delaunay triangulation that we call *anisotropic triangulation* will be defined. It builds the core of both sampling approaches.

Delaunay triangulations and Voronoi diagrams Given a planar Euclidean domain Ω and a point set $P = \{\mathbf{p}_i, i = 1 \ldots n\} \subset \Omega$, a Voronoi diagram is defined as the set of n Voronoi cells Ω_i for which holds that all points that lie within Ω_i are at least as close to $\mathbf{p}_i$ as to any other point in P. The Delaunay triangulation D of Ω with respect to P can be defined by a couple of equivalent properties:



(a) Isotropic neighbors (b) Anisotropic neighbors

Figure 5.4: Neighbor computation of the current sample (green) for an isotropic (a) and anisotropic (b) triangulation of a constant field. In this example, the isotropic Delaunay triangulation and the usage of a one-edge distance (blue samples) can lead to wrong neighbors. Using a two-edge distance (yellow samples) does not provide a valid solution, either (a). Given an anisotropic triangulation, a one-edge distance is sufficient and already results in meaningful neighbors (b).

- D is the dual graph of the Voronoi diagram.
- D maximizes the sum of the minimum angles.
- D guarantees that the circumcircle of each triangle does not contain any other point of P.

In general, Delaunay triangulation algorithms focus on these properties and Voronoi diagrams are computed as dual of the Delaunay triangulation. When generalizing these ideas to non-Euclidean spaces, utilizing a spatially varying metric, the above mentioned properties of the Delaunay triangulation are not equivalent anymore. In general, an anisotropic Delaunay triangulation is defined based on the duality property. Unfortunately, it is not guaranteed that the dual of an anisotropic Voronoi diagram actually results in a valid triangulation (e.g. [Labelle and Shewchuk, 2003; Du and Wang, 2005]). This is already the case for planar domains and even worse for the manifold case. To avoid this problem, several heuristics have been proposed (e.g. [Bossen and Heckbert, 1996; Shimada et al., 1996; Li et al., 1999]). Among the algorithmic challenges are an efficient geodesic distance computation and the fact that Voronoi cells are no longer bounded by straight lines. For time-efficient solutions, approximate definitions and distance measures are useful. We propose to define a triangulation that represents a meaningful neighborhood structure (Figure 5.4), independently from the Voronoi cells. In the following, we introduce our definitions to generate anisotropic triangulations.

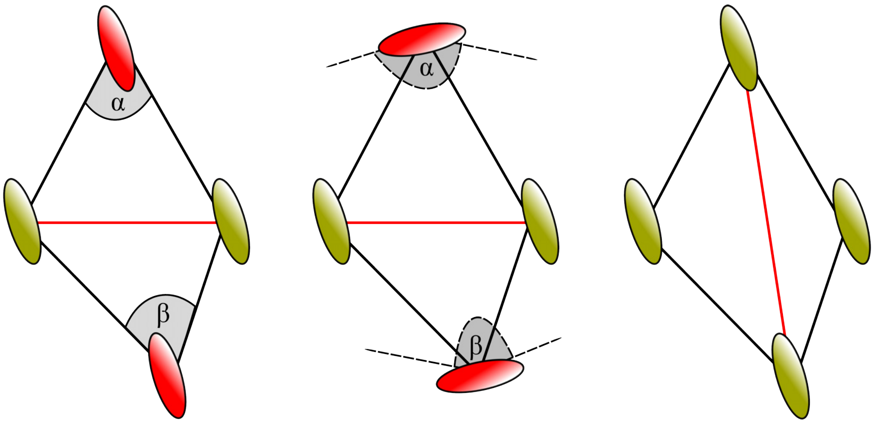


Figure 5.5: Modified edge flip. Left: Configuration for which the angle condition is tested (red edge). Here, the Euclidean angles are depicted. Middle: To validate the angle condition, we analyze the angles opposite to this edge with respect to the inverse metric (swapped ellipses). In this configuration, the angle condition is not fulfilled. Right: Valid configuration.

Modified edge flip To compute a Delaunay triangulation for non-Euclidean metrics of parameterized surfaces, Shimada et al. [1996] propose an edge-flip algorithm based on a generalized circumcircle test in parametric space. Since we deal with triangulated surfaces without given parameterization, we propose to adapt the *Delaunay angle condition* (see also [Navarro et al., 2011]), also applying an edge-flip algorithm. The angle condition is easy to generalize to the anisotropic case and does not require a parameterization. Note that the edge-flip algorithm only concerns the neighborhood relation of the samples and does not change the original surface. We check for each edge of a given triangulation if the sum of its opposite angles α and β satisfies the following condition:

$$\alpha + \beta \leq \pi. \tag{5.2}$$

Therefore, we solely need to adapt the dot product for computing the angles α and β. For two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^3$ and a local metric $\mathbf{M}$, the scalar product becomes

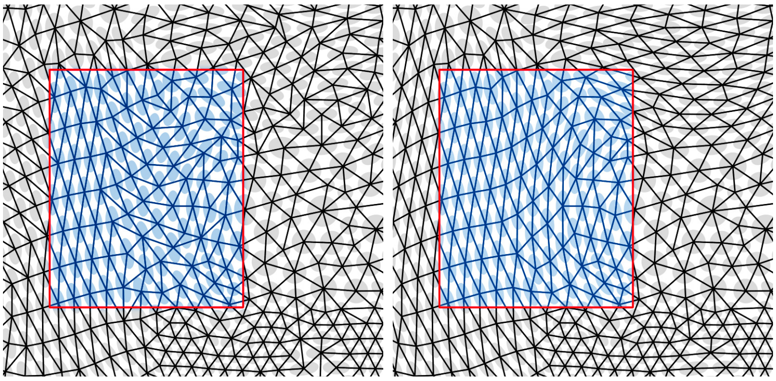
$$\langle \mathbf{v}, \mathbf{w} \rangle_{\mathbf{M}} = \mathbf{v}^T \cdot \mathbf{M} \cdot \mathbf{w}. \tag{5.3}$$

Accordingly, the length of the vector $\mathbf{v}$ is defined as

$$\|\mathbf{v}\|_{\mathbf{M}} = \sqrt{\mathbf{v}^T \cdot \mathbf{M} \cdot \mathbf{v}}. \tag{5.4}$$

If $\mathbf{v}, \mathbf{w}$ are the two vectors that enclose the angle $\angle(\mathbf{v}, \mathbf{w})$ and share the point $\mathbf{p}$ we have

$$\angle(\mathbf{v}, \mathbf{w})_{\mathbf{M}(\mathbf{p})} := \arccos\left(\frac{\langle \mathbf{v}, \mathbf{w} \rangle_{\mathbf{M}(\mathbf{p})}}{\|\mathbf{v}\|_{\mathbf{M}(\mathbf{p})} \|\mathbf{w}\|_{\mathbf{M}(\mathbf{p})}}\right). \tag{5.5}$$



(a) Isotropic mesh **(b)** Anisotropic mesh

Figure 5.6: The figures show an isotropic mesh (a) and an anisotropic mesh with respect to a given metric tensor field (b). The anisotropic mesh aligns naturally with the major eigenvector field.

We evaluate the angle condition using the inverse metric (Figure 5.5) and call the resulting triangulation *anisotropic triangulation.*

Properties Such a triangulation, which is generated with respect to a given metric tensor field, has the following properties (see Figure 5.6 and Figure 5.8):

- Its triangles are stretched according to the metric field and the edges are oriented along the direction of the major eigenvector field [Shimada et al., 2000].
- It naturally has an adaptive character, which supports our requirement of good visual quality.
- It leads to more meaningful neighbor computations in metric space than isotropic triangulations do.

Algorithm (Planar domains) For creating the triangulation in the planar domain, we use an incremental algorithm starting with a *supertriangle* that covers the whole sample domain. We successively refine the triangulation by adding new samples in sparsely populated areas (Sections 5.4.2, 5.5.2) thereby following the approach by Sloan [1987]. Whenever a new sample is added, we check if the angle condition (Equation (5.2)) is still fulfilled. Thus, the triangulation is built concurrently with the initial sample distribution. Figure 5.7 shows an initial sampling and the corresponding anisotropic mesh.

Algorithm (Two-manifold domains) For creating the triangulation in the two-manifold domain, we propose an edge-flip algorithm on the basis of the triangulated input surface.

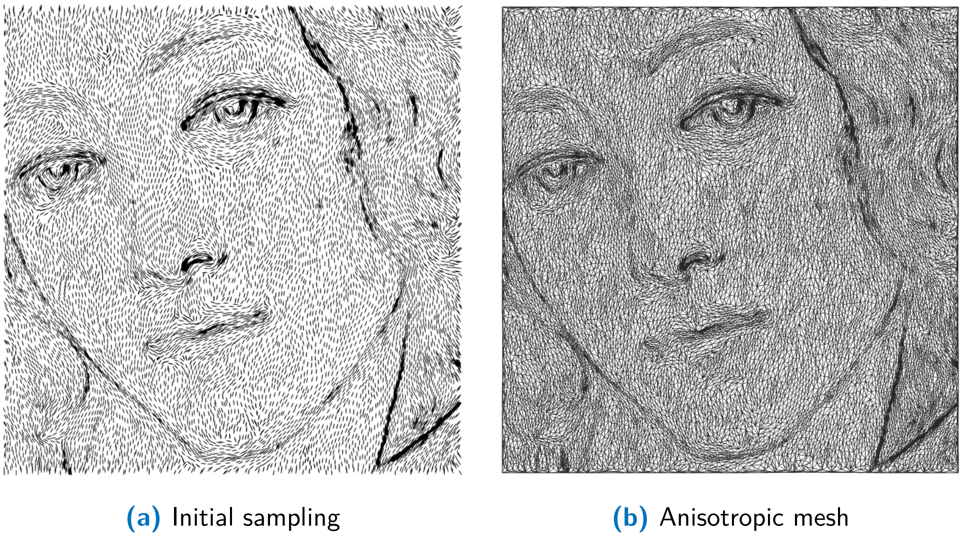


Figure 5.7: Anisotropic triangulation in the planar domain (b) with respect to a non-uniform anisotropic tensor field (a). The anisotropic triangulation aligns with the major eigenvector field, which in this example is tangential to the image edges.

For computing edge flips in parallel, we distinguish four steps (similar to [Navarro et al., 2011]):

1. Starting from the input triangulation, we check for each edge whether its adjacent triangles fulfill the angle condition (Equation (5.2)). If it is not fulfilled, the edge needs to be flipped and, therefore, is labeled $l = 1$. Otherwise and if it is a boundary edge, it is labeled $l = 0$.

2. Next, a subset of edges with label $l = 1$ needs to be found that can be flipped in parallel. This is the case if the two triangles that are adjacent to the current edge do not have another edge that is labeled.

3. Now, the actual edge flip is performed for all labeled edges in parallel.

4. Finally, the triangulation and all neighbor information is updated.

These four steps are repeated until a user-specified number of edges fulfills the angle condition (Equation (5.2)) or a maximum number of iterations was reached. In general, this leads to an anisotropic triangulation as defined in Section 5.3.5. Depending on the underlying metric field, however, dead-locks are possible. These occur if flipping one edge results in a configuration where another edge becomes invalid and vice versa, and it occurs if the two triangles adjacent to the current edge have another edge that needs to be flipped and this configuration cannot be eliminated over several iterations. For

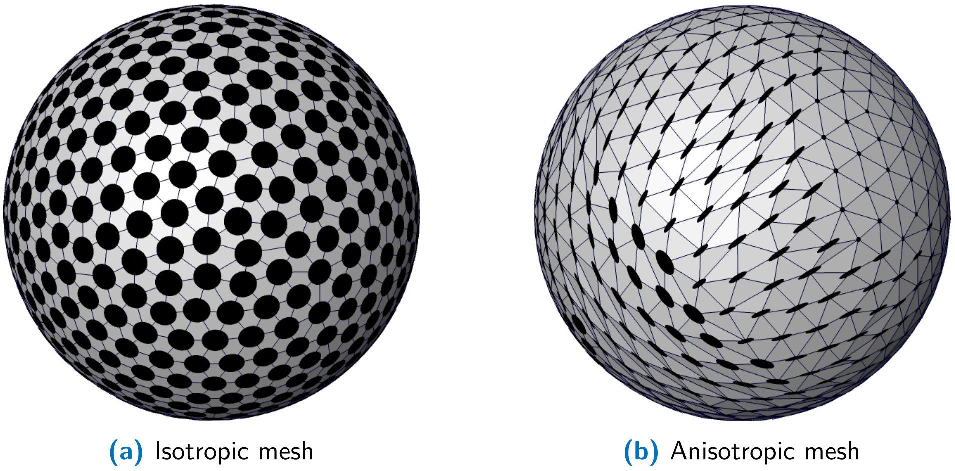


Figure 5.8: Delaunay triangulation in the two-manifold domain: with respect to a uniform isotropic tensor field (a) and a non-uniform anisotropic tensor field (b). The anisotropic triangulation (b) aligns with the major eigenvector field and naturally has an adaptive character.

subsequent computations, however, these edges do not lead to problems, because we only need the neighbor relations to compute minimum distances and, therefore, we use second-order neighbors.

5.3.6 Triangle Fillrate

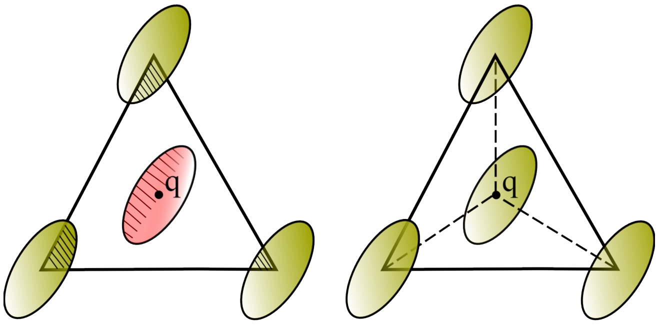


Figure 5.9: To compute the fillrate of a triangle, we use the metric at the triangle's barycenter (red ellipse). If the fillrate falls below a user-defined threshold, a new sample is inserted at the triangle's barycenter (right image) and a retriangulation (dotted lines) is initiated.

In the following, we will present a simple *density measure*, which we use for the creation of an initial sample distribution: the triangle fillrate (Figure 5.9). A triangle's fillrate is the ratio between the subarea $A_\circ$ of the triangle that is covered by the elliptical samples at its corners, and the triangle area $A_\triangle$ itself. To compute $A_\circ$, we use the determinant of the metric at the triangle's barycenter $\mathbf{q} = \frac{1}{3}(\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3)$, where $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are the triangle's vertices. As the sum of all triangle angles is always π, the area $A_\circ$ that is covered by an ellipse/ellipsoid is always half of the area of the ellipse/ellipsoid, that is, $A_\circ = 0.5 \cdot \pi \cdot \det(\mathbf{M}(\mathbf{q}))$. From $A_\circ$ and $A_\triangle$, the fillrate η is computed as

$$\eta = \frac{A_\circ}{A_\triangle}. \tag{5.6}$$

The densest packing of spheres in the 2D domain results in a hexagonal pattern. Assuming such a *hexagonal packing* as gold standard, the optimal fillrate is given as

$$\delta_{fill} = \frac{\pi}{2\sqrt{3}} \approx 90.7\%. \tag{5.7}$$

5.4 Particle-Based Approach

The approach that is presented in the following is a particle-based approach. Here, the samples are interpreted as moving particles. Because the movement is based on forces defined by inter-particle distances, stable distances are achieved when the particle system reaches an equilibrium, that is, when all elliptical samples centered at the particle positions are closely packed without intersecting each other.

On the basis of previous work in the context of meshing [Shimada et al., 2000], anisotropic sampling [Feng et al., 2008] and glyph placement [Kindlmann and Westin, 2006; Hlawitschka et al., 2007], we have developed a method that behaves in a robust way even when the size of the samples varies strongly across the sample domain. This is an important property of our approach which distinguishes the presented method from previous work [Kindlmann and Westin, 2006; Hlawitschka et al., 2007]. To achieve this, we propose the use of an anisotropic triangulation (Section 5.3.5) of particle positions. It leads to more meaningful neighbor computations in metric space than isotropic triangulations do. Furthermore, it provides good means for an explicit and automatic control of prominent holes and overlaps, which otherwise would result in unpleasant visualizations.

5.4.1 Overview

The following steps summarize our method for generating anisotropic sample distributions in planar domains by interpreting the samples as moving particles that exert attracting and repelling forces. Input is a scalar-, vector- or tensor field.

1. **Define a metric tensor field** as described in Section 5.3.4.

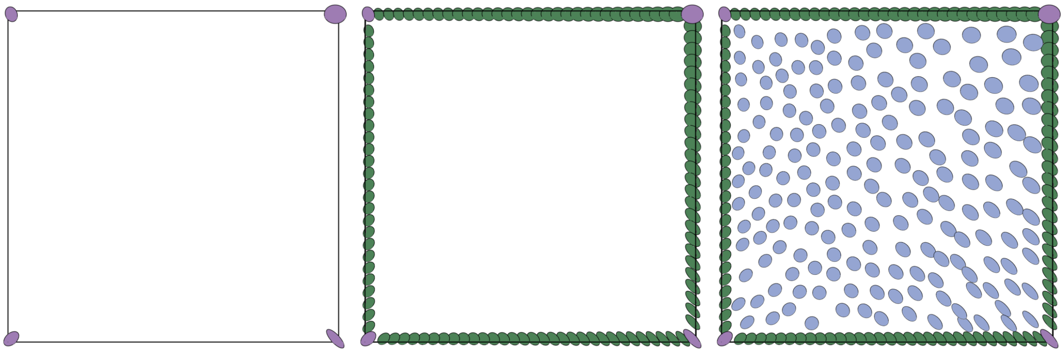

(a) $0D$ (b) $1D$ (c) $2D$

Figure 5.10: To achieve a closed boundary treatment, the domain is sampled in order of dimension. First, samples are placed at the corners (a), then they are distributed along the edges (b) and finally they are inserted into the domain via dart throwing (c).

2. **Generate an initial sample distribution respecting this metric.** To generate such a distribution, we use anisotropic dart throwing (Section 5.4.2).

3. **Iteratively refine the initial sample distribution** via particle movement (Section 5.4.3) until all forces are balanced or a maximum number of iterations is reached. An anisotropic triangulation enables fast and correct neighbor-queries and it is also used for an automatic population control (Section 5.4.4).

5.4.2 Initial Sampling

The initial sample distribution should have the following properties:

- The distances between particles are well-balanced so that significant holes and overlaps are minimized.
- The samples cover the domain densely.
- It is unstructured so that periodic arrangements are reduced.

To generate a sample distribution that fulfills these criteria, we developed a variant of relaxation dart throwing [McCool and Eugene, 1992] that can handle samples with elliptical shape. According to Shimada et al. [2000], we sample the given domain Ω in order of dimension: First, samples are placed at the corners of Ω, which are kept fixed throughout the whole simulation (Figure 5.10a). Second, samples on the edges of Ω are distributed via subdivision (Figure 5.10b). To avoid a structured placement of edge samples, they are repositioned using a one-dimensional particle movement (Section 5.4.3). Now, these positions are kept fixed throughout the whole simulation, which leads to a closed boundary treatment that avoids particles leaving the domain. Furthermore, this boundary treatment eliminates the necessity to define an external force as it was required in previous work [Kindlmann and Westin, 2006; Hlawitschka et al., 2007].

To distribute samples inside the domain, we use dart throwing [Cook, 1986], that is, samples are successively *thrown* into Ω (Figure 5.10c). They are rejected if they would intersect with any previously accepted ellipse. To generalize this algorithm to work with anisotropic samples, we use the same distance measure (Equation (5.10)) as for the force function. The original dart throwing algorithm terminates if the rejection-rate exceeds some threshold. In this work, we apply relaxation dart throwing. Starting with larger samples, their size is successively reduced if the rejection-rate becomes too large, that is, if the ratio of the number of accepted against the number of rejected samples falls below a user-specified threshold value ϵ. To create the images in this section, we used $\epsilon = 0.01$. To determine ellipse sizes, we use the relative Poisson disk radius r given as $r = \rho r_{max}$, with $\rho \in [0, 1]$ [Lagae and Dutré, 2005]. In our case r_{max} is implicitly given by the local metric. For relaxation dart throwing we start with a value of $\rho = 0.9$ which is reduced by 0.05 whenever the rejection rate becomes too large. We stop reducing sample sizes when $\rho = 0.75$. As said in [Lagae et al., 2008], ρ needs to be large but not too large to avoid regular configurations ($\rho = 1$ corresponds to a hexagonal arrangement). An advantage of relaxation dart throwing is that it terminates when a desired number of samples has been accepted, which is one of our requirements. In order to estimate this number N, we compute an average value of the determinant $\det_{avg}$ of $\mathbf{M}$, which provides a measure for the average size of elliptical samples. Given the area A of the domain Ω, an appropriate number N of samples then is

$$N = \frac{A}{\det_{avg}}. \tag{5.8}$$

5.4.3 Particle Movement

In the following, we describe how the initial sample distribution is refined via particle movement. The idea of this step is to equalize the distances between the samples so that they become more uniform. To achieve this, a force function is defined that controls the movement. The force function is a function of particle distances exerting attracting and repelling forces depending on the inter-particle distances.

Distance measure Previous work approximate geodesic distances [Feng et al., 2008; Li et al., 2010] via distance computation in the distorted space. These measures, however, are not symmetric, which would require an additional conflict check [Li et al., 2010]. According to Shimada et al. [2000], we define the optimal distance d_{opt} between sample

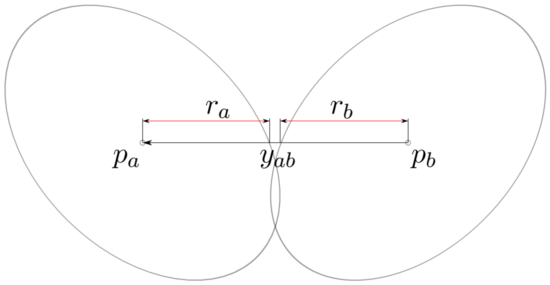


Figure 5.11: Approximated distance computation between the neighboring samples p_a and p_b.

positions $\mathbf{p}_a$ and $\mathbf{p}_b$ to be the sum of the two lengths r_a and r_b to the ellipses' surface in the direction of the connecting line $\mathbf{y}_{ab} = \mathbf{p}_a - \mathbf{p}_b$ (Figure 5.11)

$$d_{opt} = \underbrace{\frac{\|\mathbf{M}(\mathbf{p}_a) \cdot \mathbf{y}_{ab}\|}{\|\mathbf{y}_{ab}\|}}_{r_a} + \underbrace{\frac{\|\mathbf{M}(\mathbf{p}_b) \cdot \mathbf{y}_{ab}\|}{\|\mathbf{y}_{ab}\|}}_{r_b}. \tag{5.9}$$

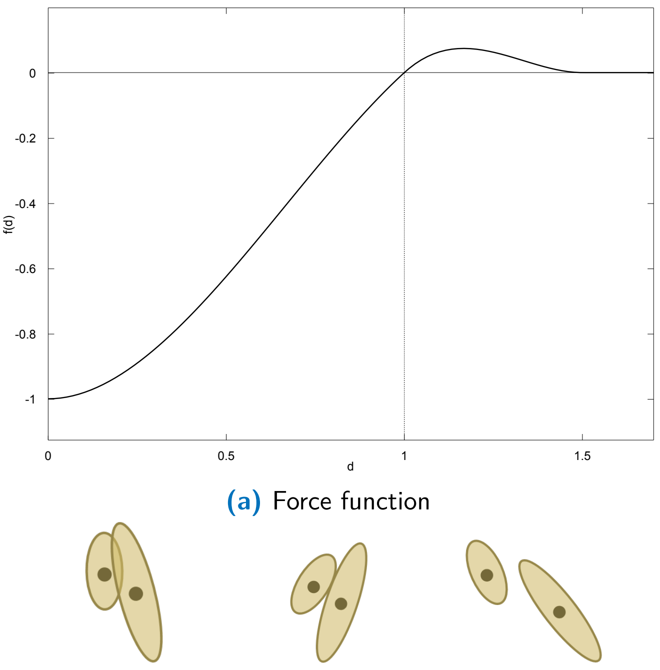
Thus, the optimal distance with respect to the force function is always $d_{opt} = 1$ independent from the local metric. The force function (Section 5.4.3) then is defined over the relation d of the inter-particle distance y_{ab} and d_{opt}

$$d = \frac{\|\mathbf{y}_{ab}\|}{d_{opt}}. \tag{5.10}$$

Figure 5.11 shows that this approximation does not avoid all kinds of intersections. However, we find that this measure is a good compromise with respect to the generation of holes and overlaps. Preventing, for example, the two samples shown in Figure 5.11 from intersection would result in a bigger hole at the top.

Force function The force function f, being defined over the distance d (Equation (5.10)), controls the particle movement. The goal is to compute stable particle distances that are characterized by elliptical samples that are closely packed and non-intersecting. Therefore, we define a piecewise cubic spline $f(d)$ with the following properties (see also Figure 5.12):

- It has a zero crossing at $d = 1$ so that particles with optimal distance do not move.
- If $]0 < d < 1[$, repelling forces are exerted pushing particles apart.
- If $]1 < d < (1+\gamma)[$, attracting forces are exerted pulling particles together. The parameter γ controls the range of attracting forces. As we only consider particle



(a) Force function

(b) $d < 1$ **(c)** $d \approx 1$ **(d)** $d > 1$

Figure 5.12: Force function. Within the range $(0 < d < 1)$ repelling forces are exerted. Forces become zero if $d = 1$ and within the range $(1 < d < (1 + \gamma))$ attracting forces are exerted.

interactions of neighboring samples, excluding long-distance interactions, our method is not very sensitive with respect to γ. It can be set on a fixed value of $\gamma = 0.5$.

The force F_a, which acts on particle a, is determined from the sum of the forces from the neighboring particles $F_a = \sum_{b,b \neq a} f_{ab}$. The equations of motion [Kindlmann and Westin, 2006; Hlawitschka et al., 2007]

$$\frac{d^2p}{dt^2} = F_a + C_{drag}\frac{d_p}{d_t} \tag{5.11}$$

are solved numerically via Euler integration using a step size of $\Delta t = 0.2$. Since the force scales with sample size, this leads to small steps for small particles and larger steps for large samples. The drag parameter C_{drag} counteracts the particle motion to avoid oscillations and to guarantee numerical stability [Kindlmann and Westin, 2006].

N	Initial Seeding	100 Iterations	1000 Iterations
1000	7s	0.3s	3s
10000	90s	4s	40s

Table 5.1: Timing statistics in seconds measured on an 2.6 GHz Intel Xeon Quad Core. The performance of the particle-based approach mainly depends on the desired sampling resolution N and the time needed for initial seeding. The particle movement is very fast.

5.4.4 Automatic Insert and Delete Operations

The estimation of the number of particles n for the initial placement (Section 5.4.2) is challenging. Especially considering our requirement of a stable particle system that can handle samples of varying size, n is only a rough estimate. Therefore, we include automatic delete and insert operations that adjust the number of samples during simulation. This further enables an explicit control of holes and overlaps. According to Shimada and Gossard [1995], we identify over- and underpopulated regions via analyzing inter-particle forces. As soon as F exceeds (falls below) a given threshold, particles are removed (inserted). Contrary to geometry-based approaches, which only depend on attractive forces, particle-based methods are more sensitive to holes. Thus, overpopulated regions appear only rarely.

To automatically insert samples in underpopulated regions, we exploit the anisotropic triangulation (Section 5.3.5). Therefore, for each triangle, the fillrate η, being the ratio of the covered area and the triangle's area, is computed (Section 5.3.6). Assuming a hexagonal packing as gold standard, the optimal fillrate is $\delta_{fill} \approx 0.9$ (Equation (5.7)). If η is significantly below δ_{fill}, a new sample is inserted at $\mathbf{q}$. For the images that are depicted in this section, we inserted a new sample whenever $\eta \leq 0.6$.

5.4.5 Results and Applications

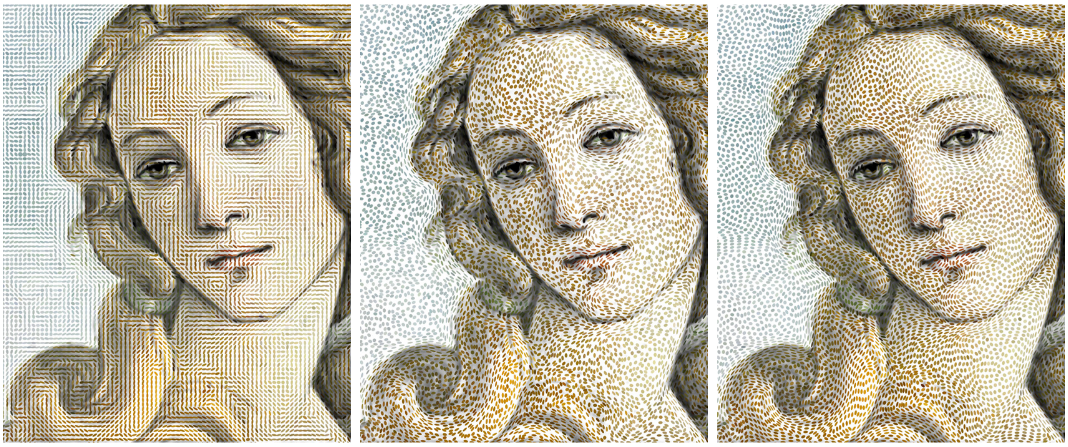
We have presented a particle-based approach to generate unstructured distributions of non-intersecting elliptical samples in planar domains. In the following, we evaluate this method with respect to its performance and show examples for its application for glyph placement and texture generation. Therefore, we applied the proposed methods to a slice of the two-point-load dataset (see Appendix A.1) (Figure 5.14) and to a color image (Figure 5.13) In the latter case, we computed the image's gradient- and its orthogonal vector field as described in Section 5.3.4 to generate a metric tensor field. For both examples, we used an anti-symmetric mapping (Equation (3.2)).

All images and timing results were obtained on an Intel Xeon X5550 2.67 GHz system with 8 cores and an NVIDIA GeForce GTX680 graphics card.

Performance

The time needed for generation mainly depends on the desired sampling resolution and the number of iterations needed until a stable placement has been reached (Table 5.1). To create the images shown in Figure 5.13 and Figure 5.14, an update of the Delaunay triangulation every eighth iteration was sufficient. The frequency of updates is independent from the number of samples. Instead it depends on the variation of sample size, that is, more frequent updates are required if the sample size across Ω varies strongly. In our examples, we found a stable configuration after 100 iterations. For comparison, Kindlmann and Westin [2006] computed 2000 iterations to create the 2D results in their work, which took about 8 minutes. Usable results appeared after 400 iterations. Unfortunately, they do not say anything about the number of glyphs. Hlawitschka et al. [2007] mention that their packing becomes stable after "several hundred steps". For $N = 1000$ (10000) glyphs and 100 iterations they need $2s$ ($55s$). Because all of these measures were carried out on different PCs and in different years, the times should be considered only as a rough comparison value. In fact, more interesting are the number of iterations.

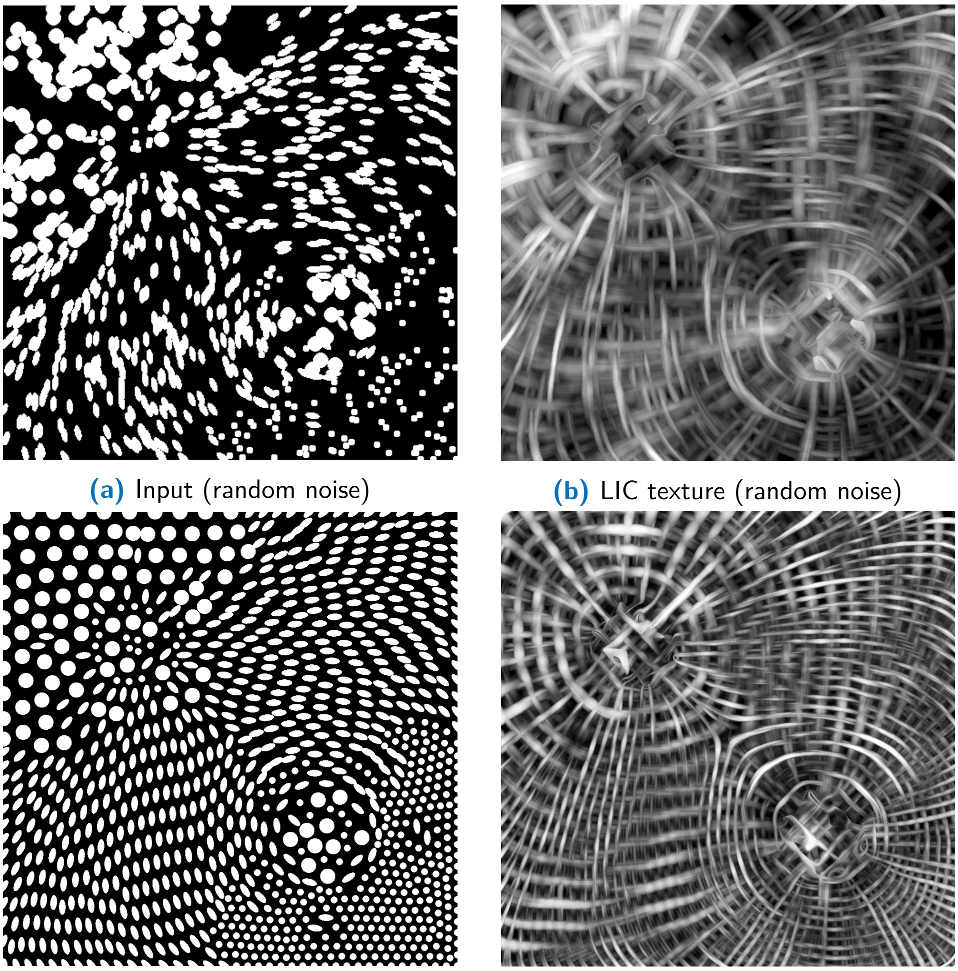
Applications



(a) Uniform sampling **(b)** Initial anisotropic sampling **(c)** Final anisotropic sampling

Figure 5.13: In these images, size and shape of elliptical samples are determined from a metric based on the image gradient and its orthogonal vector field. To compute the sampling distribution for these images, 1000 iterations were needed. The images show the samples placed at the grid positions (a), before particle movement (b) and the result of the particle-based approach (c).

In the following, we show that the distribution is well suited for the placement of glyphs and as input noise texture for the creation of *fabric textures*.



(a) Input (random noise)

(b) LIC texture (random noise)

(c) Input (our method)

(d) LIC texture (our method)

Figure 5.14: These images show a slice from a stress tensor dataset from a finite-element simulation that describes the elastic behavior of a block on which a pushing and a pulling force have been applied. To compute the sampling distribution (c) used as input to create a fabric texture (d), 100 iterations were needed. Holes/clutter in the random noise input (a) result in darker/brighter regions in the final image (b), which gives the impression of a 3D field. In (d), thinner fibers in compressive regions can be distinguished clearly from thicker fibers in tensile regions.

Glyph Positioning For the placement of tensor glyphs a good distribution should have the following properties:

- Dense, to provide all relevant details and the evolution of tensors across the field.
- Unstructured, because the human visual system is sensitive to the recognition of patterns.
- Evenly, minimizing prominent holes and overlaps.

Figures 5.13c and 3.5b demonstrate the superior quality of glyph-based depictions when the glyphs are centered at the particle positions of the stable configuration. The images become more informative and visually pleasant. If glyphs are placed at discrete grid positions, the pattern of the underlying grid becomes visible and disturbs the perception of the field's continuous behavior. Figure 5.13b further shows the good quality of our initial sample distribution. As our technique provides an automatic control of overlaps, holes in the initial distribution are filled during particle movement. However, the quality of the initial distribution influences the algorithm's convergence time.

Texture Generation Hotz et al. [2004] have presented a global texture-based method that is especially designed to represent the central features of stress tensors, namely compression and expansion. Therefore, a LIC-like texture for both eigenvector fields is created and then the two resulting textures are blended, which results in images that resemble a fabric: thinner fibers indicate compression and thicker fibers indicate expansion. For this method, sparse noise textures (Figures 5.14b, 5.14d) are preferred as input. However, especially sparse noise textures can suffer from holes which are visible in the final image (Figures 5.14a, 5.14b). A good distribution that is used as input noise texture should have the following properties:

- Even, to ensure a uniform brightness across the field. Otherwise, brighter regions would be emphasized and, thus, might be perceived as more important.
- Unstructured, to avoid patterns to become visible in the final visualization.

Figure 5.14 compares the results using a random noise texture as input (Figures 5.14a, 5.14b) and one computed with our technique (Figures 5.14c, 5.14d). Using a noise distribution computed with our technique results in a visualization where contrasts are more balanced and the distances between the fibers are more uniform. Furthermore, the visualization that is based on our technique has less the appearance of being three-dimensional, which otherwise could lead to misinterpretations.

5.4.6 Discussion

In this section, we presented a particle-based approach to generate anisotropic sample distributions. We extended relaxation dart throwing to work with anisotropic samples.

Therefore, we use a distance measure that identifies intersections between elliptical samples. The iterative refinement of the initial distribution via particle movement was extended by introducing the use of an anisotropic triangulation. Through this approach, the computation of forces becomes more stable due to improved neighbor computations in comparison to isotropic triangulations. As a consequence, our method is more stable with respect to samples that strongly vary in size across Ω. Since particle-based approaches are very sensitive to holes in the initial distribution, automatic insert and remove operations were presented. These are able to detect over- and underpopulated regions during iterative refinement. Due to this automatic control, holes in the initial distribution are filled during simulation. This is an important step, because the quality of the initial distribution influences the final distribution's quality as well as the algorithm's convergence time. In general, 100 iterations are sufficient to create a stable particle configuration. In all our test cases, we never needed more than 1000 iterations.

We have shown the applicability of the resulting sample distribution to improve glyph-based depictions and as input texture for the creation of fabric textures. A major bottleneck in the presented particle-based approach is the time needed to create the initial sample distribution. To optimize this step, the use of the anisotropic triangulation would be beneficial so that conflict checks are only computed between neighboring samples. However, we have identified even more potential in the anisotropic triangulation. Therefore, we will present a method to create initial sample distributions in Section 5.5 that does not need any conflict checks. With this method, we are able to create up to $10,000$ samples per second. Another disadvantage of the particle-based approach is that it is very sensitive to parameter choices. Although, we already identified default values for the needed parameters and also got rid of parameters that were needed in previous approaches, there is still room for improvement. Finally, the number of iterations needed for the refinement of the initial distribution could be minimized. In Section 5.5, we present an algorithm that exploits the anisotropic triangulation in various ways, which improves the performance significantly.

A generalization of the particle- approach to work with 3D tensor fields would be possible and was shown in previous work [Kindlmann and Westin, 2006]. In this work, we have not considered an extension of the method to work with 3D tensor fields, because we believe that 3D glyph visualizations do not result in very meaningful visualizations. Instead, we focus on a more general approach that works in planar as well as in two-manifold domains in the next section. With this method, we lose the easy generalization to 3D domains. However, we will show that it comes with many advantages and is highly time efficient.

5.5 Triangle-Based Approach

This section presents a time-efficient approach for the generation of anisotropic sample distributions that only depends on intuitive design parameters. The presented approach works on planar as well as on two-manifold domains. Similar to Section 5.4, we employ an anisotropic triangulation. In this work, however, we further exploit the properties of such a triangulation for the fast creation of high-quality initial sample distributions as well as for gravitational-centered relaxation. Thus, our approach is not only an extension to two-manifold domains. It also significantly speeds up the generation of anisotropic sample distributions for the planar case compared to the approach presented in Section 5.4. Due to this speed-up, the triangle-based approach also enables interactive slicing through 3D datasets.

The extension of the planar case to two-manifold domains implicates several additional challenges. To achieve interactive results there is always a compromise between an approach with a solid theoretical basis and a more practical/approximative solution. The design of our solution is guided by the following requirements:

1. The sample generation should be time-efficient.
2. It should depend on very few intuitive parameters.
3. The sample distribution should have a random character (no noticeable pattern) and the sample domain should be covered densely (avoid large empty areas as well as cluttered areas which both would be visually distracting).

The most critical part for the generalization to surfaces is the computation of distances and the generalization of distance measures. Our technique avoids costly distance computations whenever possible. Where distance computations are needed, we use fast and easy to compute approximations which are sufficient for our purposes and result in sample sets that fulfill our requirement of good *visual quality* (see Requirement (3)).

5.5.1 Overview

The following steps summarize our method for generating anisotropic sample distributions on planar and two-manifold domains. Input is a scalar-, vector- or tensor field.

1. **Define a metric tensor field** as defined in Section 5.3.4.
2. **Generate an initial sample distribution respecting this metric.** To generate such a distribution, we employ anisotropic triangulations (Section 5.3.5)

and compute triangle fillrates to find areas that are sparsely populated. In these areas, new samples are added. Furthermore, two additional design options are provided: (1) the overall density is steered by the fillrate η. (2) A spatially varying importance function facilitates a local density adaptation.

3. **Apply a relaxation process (Section 5.5.3)** that equalizes triangle sizes with respect to the underlying metric tensor field. Thus, more uniform sample distances with respect to the underlying metric tensor field are achieved. The relaxation is based on the anisotropic triangulation.

5.5.2 Initial Sampling

The initial sampling procedure computes a set of samples $S := \{\mathbf{p}_i | \mathbf{p}_i \in \Omega, i = 1 \dots n\}$, where n is the number of samples. We require our initial sample distribution to have the same properties as they were discussed in Section 5.4: the samples should cover the domain densely (no holes) while a minimum separation between them is maintained (no overlaps). The major bottleneck of methods to generate such distributions, for example, dart throwing, are costly distance computations [Feng et al., 2008]. In the following, we present a method that exploits the triangulation's properties (Section 5.3.5) to generate an initial sampling with high visual quality. Thus, we do not require distance computations to identify sample intersections. The properties of the triangulation in combination with the assumption that the metric tensor field does not vary strongly within a small local environment (Section 5.3.2), and finally the fillrate, allows us to pass on explicit sample intersection tests.

Algorithm Starting point of initial sampling is a *coarse* anisotropic triangulation of the input domain. For identifying areas where it is beneficial to insert a new sample, we use the fillrate of a triangle as measurement (Section 5.3.6). In [Kratz et al., 2011b], the triangle fillrate was used for population control in order to insert or remove samples during the refinement procedure. We use the triangle fillrate as guidance for the initial sampling procedure. To guarantee interactivity, we propose the following computations.

We can guide the distribution's density by changing the target value for η. Smaller values result in less dense initial samplings and higher values in denser samplings. To add adaptivity, an additional importance function can be used that influences the scale of the samples and, thus, also the fillrate. For example, an importance function that is guided by the magnitude of the image gradient creates initial sample distributions that are denser at the edges of the input image (see also Figure 5.17).

In the planar case, we start by randomly sampling the boundary $\partial\Omega$ of the planar sample domain and adding a few random samples within Ω.

1. For these samples, an anisotropic triangulation is computed that results in a set of triangles $T := \{t_k | t_k \in \Omega, k = 1 \dots m\}$, where m is the number of triangles.

2. For each triangle t_k, its fillrate is determined. If the fillrate is below a user-defined threshold, a new sample is added at the triangle's barycenter.

3. Then, the triangulation is updated and the procedure is started again with step (1).

4. Steps (1)-(3) are repeated until a desired density, specified by η, has been reached.

In the two-manifold case, we assume a given triangulated surface $\mathcal{X}$.

1. In a first step, $\mathcal{X}$ is simplified using an edge-contraction algorithm [Garland and Heckbert, 1997]. This procedure yields a coarse sample mesh $\widehat{\mathcal{X}}$ and an associated set of triangles $T := \{t_k | t_k \in \Omega, k = 1 \dots m\}$, where m is the number of triangles.

2. The second step is similar to the planar case with the difference that new sample positions need to be *projected* onto the original surface as it will be described in Section 5.5.4.

3. The triangulation is updated and the procedure is started again with step (2) until a desired sample density is reached.

5.5.3 Gravitational-Centered Relaxation

The initial sample distribution already fulfills Poisson-disk characteristics. To equalize sample distances, we propose a gravitational-centered relaxation on the basis of the anisotropic triangulation. It equalizes triangle sizes with respect to the metric tensor field so that sample distances become more uniform (e.g., Figure 5.17(d)). This is desirable for the rendering of anisotropic Voronoi diagrams, which we will introduce in Chapter 6.

Centroid of tensor-weighted star For each sample $\mathbf{p}_i$, we consider its one-ring neighborhood $t_j \in N(\mathbf{p}_i)$ (Figure 5.15). This *star* consists of n triangles that all share the sample $\mathbf{p}_i$ and have a barycenter $\mathbf{q}_j$. The centroid c_i of the star with respect to the metric is defined in analogy to the center of mass, which will be regained when using the Euclidean metric

$$c_i = \mathbf{M}_i^{-1} \cdot \sum_{j=1}^{n} A_{\triangle j} \cdot (\mathbf{M}(\mathbf{q}_j) \cdot \mathbf{p}_i), \tag{5.12}$$

whereby $\mathbf{M}_i = \sum_{j=1}^{n} A_{\triangle j} \cdot \mathbf{M}(\mathbf{q}_j)$.

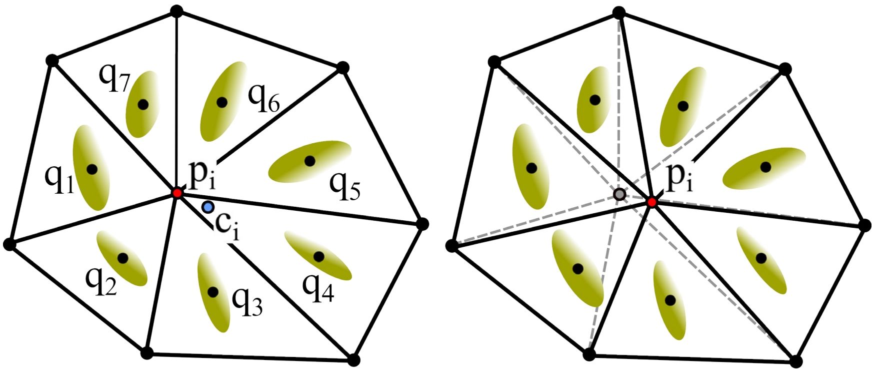


Figure 5.15: For gravitational-centered relaxation, the star around each sample is considered. To compute the centroid of this star, the metric tensors (green ellipses) at the triangles' barycenters are used as weights (left). Once the star's centroid is computed, the sample position and its one-ring neighborhood are updated (right).

Point relocation The point relocation comprises an update of the current sample's one-ring neighborhood. For planar domains, the sample $\mathbf{p}_i$ is simply translated towards the centroid $\mathbf{c}_i$ with

$$\mathbf{p}_i \leftarrow \mathbf{p}_i + (\mathbf{c}_i - \mathbf{p}_i). \tag{5.13}$$

For the two-manifold case, centroids that are computed with Equation (5.13) in general do not lie on $\mathcal{X}$. Therefore, the translated points need to be projected back onto $\mathcal{X}$. The point relocation for surfaces, thus, becomes

$$\begin{aligned} \mathbf{p}_i &\leftarrow P(\mathbf{p}_i + (\mathbf{c}_i - \mathbf{p}_i), \mathbf{n}_i) \\ \mathbf{p}_i &\leftarrow P(\mathbf{p}_i + (\mathbf{c}_i - \mathbf{p}_i), -\mathbf{n}_i), \end{aligned} \tag{5.14}$$

where P is the projection of the translated point onto $\mathcal{X}$ (Section 5.5.4), either in the positive normal direction $\mathbf{n}_i$ or in the negative normal direction $-\mathbf{n}_i$.

Algorithm The gravitational-centered relaxation is based on the initial sample set $S := \{\mathbf{p}_i | \mathbf{p}_i \in \Omega, i = 1 \ldots n\}$, where n is the number of samples.

1. For each sample $\mathbf{p}_i$, its one-ring neighborhood is found.
2. Then, for each one-ring neighborhood, its centroid is computed.
3. Next, the sample $\mathbf{p}_i$ is translated towards this centroid $\mathbf{c}_i$.
4. Finally, in the two-manifold domain, the translated point is further projected back onto $\mathcal{X}$.

5. The procedure starts again with step (1) until a stable configuration has been found, that is, when all samples lie in the centroid of their surrounding star, or if a desired number of iterations has been reached.

5.5.4 Back-Projection

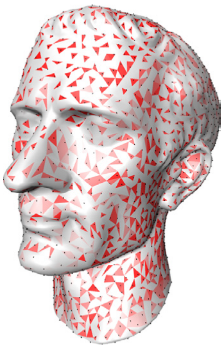


Figure 5.16: Link between samples and original surface.

For initial sampling and relaxation in the two-manifold domain, we need to maintain the link between the original mesh and the sample mesh. That is, for each sample, we need to know which triangle of $\mathcal{X}$ it corresponds to (Figure 5.16). To do this efficiently, we use a *triangle octree* (see, e.g., [Samet, 1988]) to store all triangles of the input mesh $\mathcal{X}$. As projection of a point $\mathbf{p}_i$, we take the closest intersection point of the ray that starts in $\mathbf{p}_i$ and goes into the direction of $\mathbf{n}_i$ and $-\mathbf{n}_i$, respectively, with $\mathcal{X}$. With the octree data structure, the triangles that need to be checked for intersection are identified in an efficient way. The intersection test is done in a hierarchical fashion starting with the root node that encloses all triangles and traversing the octree until a node is found that is not subdivided anymore. Then, its elements, i.e. its triangles, are checked for intersection with the ray.

5.5.5 Volume Slicing

Slicing enables the inspection of 3D input data. These can be volume data but also 2D animated scenes or time-dependent data. To provide a *smooth* transition between the visualizations (Voronoi diagrams or glyphs) of single slices, we adapt our anisotropic sampling for planar domains in the following way: Initial sampling as described in Section 5.5.2 is done once for the first slice. For this slice, a gravitational-centered relaxation (Section 5.5.3) is computed until a stable configuration is achieved. For subsequent slices, we use the result of the previous slice as initial sample distribution. Before relaxation, we compute the fillrate for each triangle of the previous result and insert or remove samples if needed. Due to spatial coherence between the slices, these operations are rarely required. Finally, very few relaxation steps (in our examples a maximum of 5 steps were sufficient) are needed until a stable sample configuration is achieved, which guarantees interactivity while inspecting the 3D volume or depicting 2D animated scenes.

5.5.6 Results and Applications

In the following, we evaluate the presented algorithm with respect to its quality and its performance. For this, we have applied the proposed methods to various datasets that are presented in Appendix A. In addition, we created the following examples:

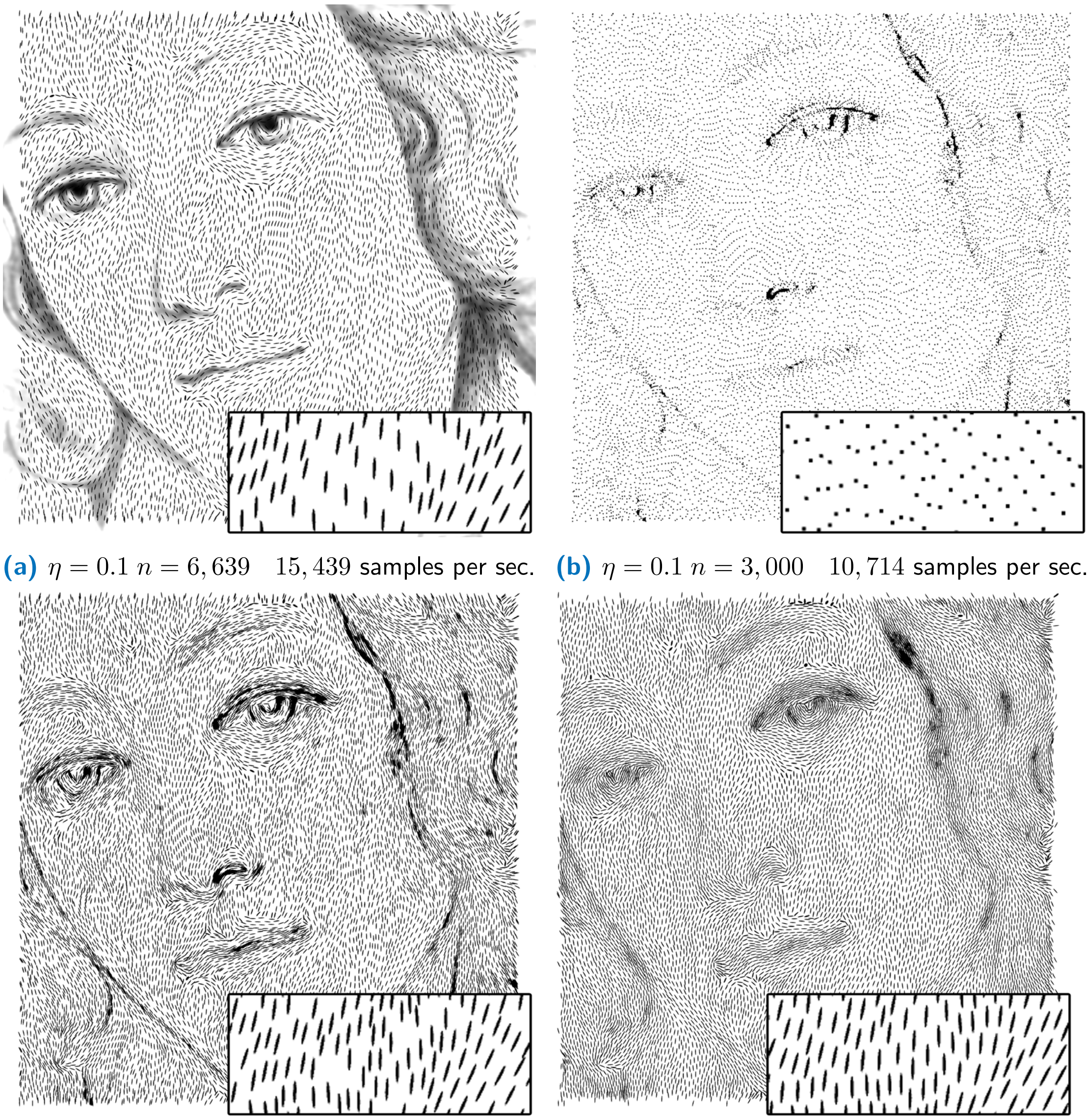
- **Color image:** For the planar case, we used the same color image as input as in Section 5.4.5. Results for this input dataset are shown in Figure 5.17.
- **Analytical surfaces and metric tensor fields:** To analyze the quality of the generated sample distributions (Figure 5.18) and to determine performance numbers (Figure 5.19), we generated tensor fields with specific anisotropy behaviors. These are: uniform isotropic, uniform anisotropic and non-uniform anisotropic.

All images and timing results were obtained on an Intel Xeon X5550 2.67 GHz system with 8 cores and an NVIDIA GeForce GTX680 graphics card.

Planar Domain

Figure 5.17 shows results of initial sampling in the planar domain. The images vary by the three design parameters for the sampling: global density steered by the fillrate η, an additional spatially varying importance function that is guided by the magnitude of the image gradient, and the metric tensor field that was derived from the input image. Depending on the input data and these parameters, our initial sampling strategy generates approximately $10,000$ samples per second. On average, the generation of the images depicted in Figure 5.17 took about 0.5 seconds. For comparison, anisotropic dart throwing as presented in [Li et al., 2010] generates about 200 samples per second and anisotropic dart throwing as presented in Section 5.4 generates about 100 samples per second. The images show that the initial sampling strategy efficiently generates sample distributions that avoid holes and clutter. It reaches already a high quality even without the relaxation process and may be sufficient for many applications. The uniformity of the sample distribution depends on the underlying metric tensor field. See also Figure 5.18 for an analysis of the quality of the sample distribution.

In Figure 5.17d, a relaxation result after 10 iterations is shown. The relaxed sample distribution is much more uniform. Focusing on the sample images, the initial sampling is more pleasing to the eye due to the formation of regular patterns during the relaxation process. But the relaxed version results in less artifacts in the Voronoi cell rendering, which we will present and discuss in Chapter 6.



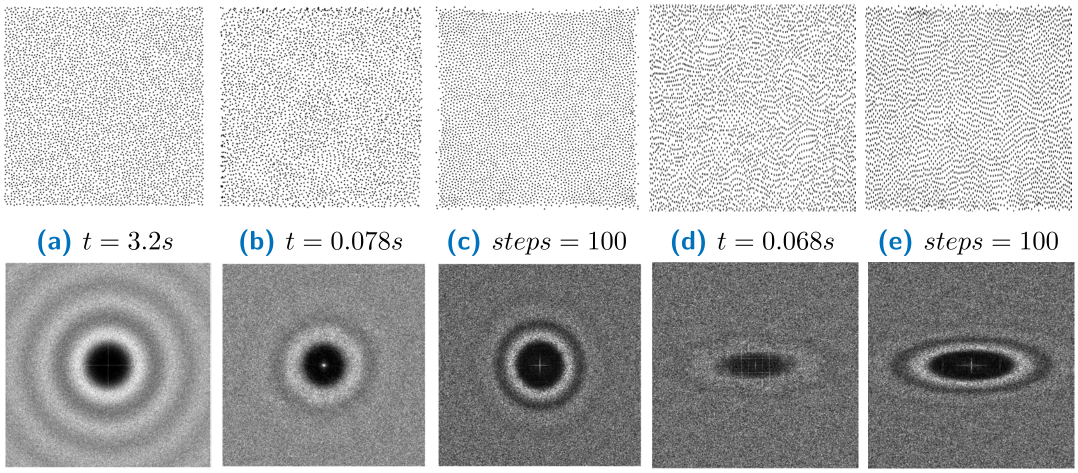
(a) $\eta = 0.1$ $n = 6,639$ $15,439$ samples per sec.

(b) $\eta = 0.1$ $n = 3,000$ $10,714$ samples per sec.

(c) $\eta = 0.1$ $n = 3,000$ $10,714$ samples per sec.

(d) Relaxed (c), 10 steps

Figure 5.17: Results for initial sampling (a-c) and relaxation (d). The global density is controlled via the fillrate η. Local density is controlled via an additional importance function based on the image's gradient (c). Anisotropy is steered by the choice of metric: In (b) an isotropic metric was used; the other examples (a, c, d) show an anisotropic metric.



(a) $t = 3.2s$ **(b)** $t = 0.078s$ **(c)** $steps = 100$ **(d)** $t = 0.068s$ **(e)** $steps = 100$

(f) Dart throwing **(g)** Initial **(h)** Relaxed **(i)** Initial **(j)** Relaxed

Figure 5.18: Isotropic point sets (a-c), anisotropic point sets (d,e) and their corresponding power spectra (f-j). The number of samples in all examples was $n = 4000$. The power spectra all have the characteristics of a blue noise spectrum. That is, a zero region for low frequencies and a relatively constant high-frequency region. Also only few repetition and grid artifacts are visible. Overlaps at the borders arise because of the random initialization at the borders and because our method does not compute any sample intersections. For the analysis, we have used the point set analysis tool presented in [Schlömer and Deussen, 2011].

Metric	Original Mesh	Initial Mesh	Sample Mesh	Initial (ms)	Relaxation (ms)
Uniform isotrop (a)	6077 (Sphere)	217	430 + 217	80	29
Uniform anisotrop (b)	6077 (Sphere)	217	413 + 217	75	26
Non-uniform anisotrop (c)	6077 (Sphere)	217	11587 + 217	716	38
Uniform isotrop (d)	1422 (Calypso)	400	2744 + 400	70	71
Uniform anisotrop (e)	1422 (Calypso)	400	1591 + 400	51	5
Non-uniform anisotrop (f)	1422 (Calypso)	400	236 + 400	24	13
Non-uniform anisotrop	112088 (Aneurysma)	1127	10535 + 1127	1600	900

Figure 5.19: Timings for the sample generation in the two-manifold domain for several analytic surfaces (see also Figure 6.10 in Chapter 6). Timings for the relaxation are given for a single iteration. To create the figures that are presented in this chapter, we have computed 100 relaxation steps. A stable sample configuration, however, is generally achieved after 10 to 20 relaxation steps.

Two-Manifold Domain

Figure 5.20 depicts results for the sample distribution and the corresponding anisotropic meshes in the two-manifold domain. The metric tensor field is designed on basis of the wall shear stress of a blood-flow simulation (see Appendix A.4). In this example, the fillrate was set to $\eta = 1.0$. That is, as many samples as possible were distributed on the surface. The resulting anisotropic triangulation represents the anisotropy of the underlying metric. The close-ups show the triangulation before and after relaxation. It can be seen that the triangle sizes equalize well with respect to the underlying metric tensor field. To demonstrate that our method can deal with highly anisotropic and varying metrics, we have exaggerated the anisotropy in this example.

Depending on the complexity of the original surface, our initial sampling strategy generates around $6,000$ samples per second in the two-manifold domain for this example. The most time-consuming steps are the projection of new samples onto the original surfaces and the re-triangulation of the sample mesh. The table in Figure 5.19 provides an overview of times that were needed to generate an initial sample distribution for two analytic examples (sphere and calypso) and the aneurysm dataset. It can be observed that the time needed for initial sampling also depends on the size of the original mesh. That is, adding a single sample took about ≈ 0.06 ms for the sphere dataset, ≈ 0.03 ms for the calypso mesh and ≈ 0.08 ms for the aneurysm dataset. Overall, the time needed for initial sampling depends on many aspects: the coarseness of the initial mesh, the variance of the metric field, the size of the metric tensors, and how many samples can be added in a single step. In this sense, the timings given in Figure 5.19 should be treated only as reference.

The table in Figure 5.19 provides an overview of times that were needed for a single relaxation step for the examples presented in this section. The times that are needed for relaxation mainly depend on the size of the sample mesh but also on the quality of the initial sampling, because it influences how many samples are moved in a single step. The major bottleneck here is the back-projection of relocated samples onto the original surface. A single relaxation step for the aneurysm dataset takes about 0.9 seconds.

5.5.7 Discussion

In this section, we have presented a method for the generation of anisotropic sample distributions in the planar domain as well as in the two-manifold domain.

The sampling approach consists of two main steps, the initial sampling and the subsequent relaxation. The goal of initial sampling is to generate sample distributions that cover the underlying domain densely while significant holes and cluttered areas are avoided. We have shown that this can be achieved efficiently through the use of a *density measure* that we call *triangle fillrate* in combination with anisotropic triangulations. This combination of an intuitive measure and a data structure that reflects the underlying data, enables an efficient identification of areas where inserting new samples is beneficial. The resulting method can be considered as a kind of *guided dart throwing*, where costly intersection tests as well as conflict checks are not needed. This makes

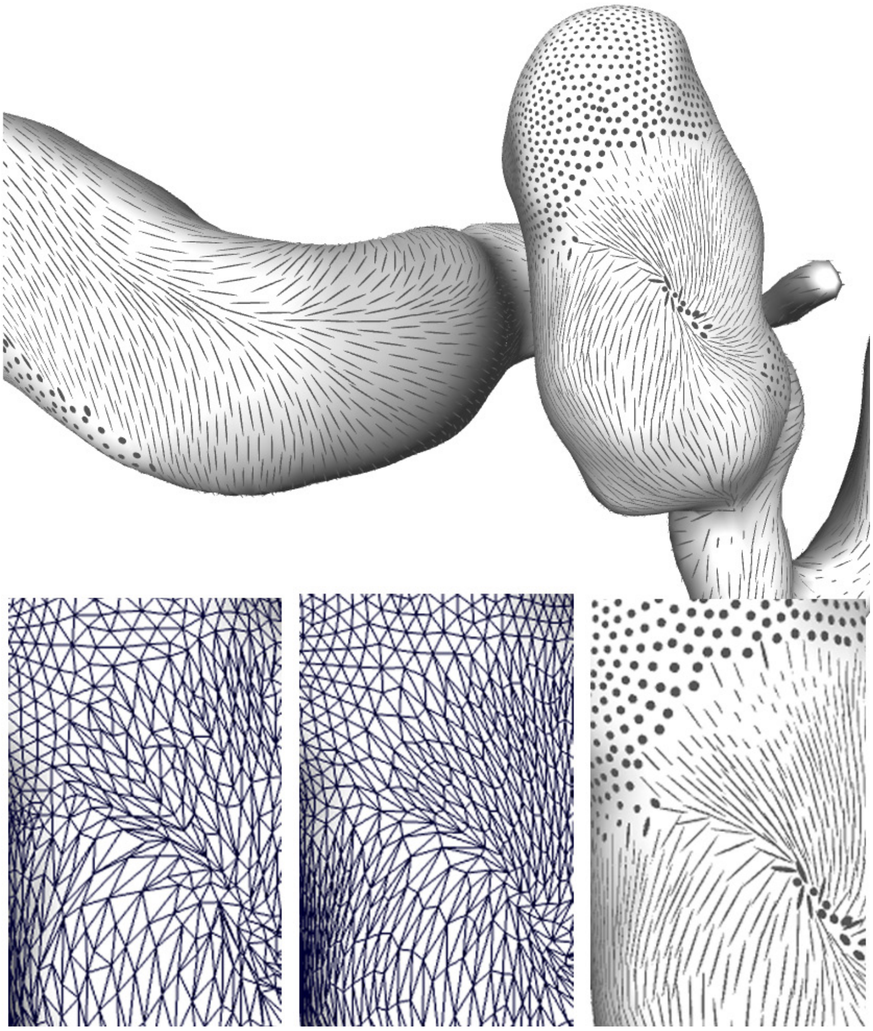


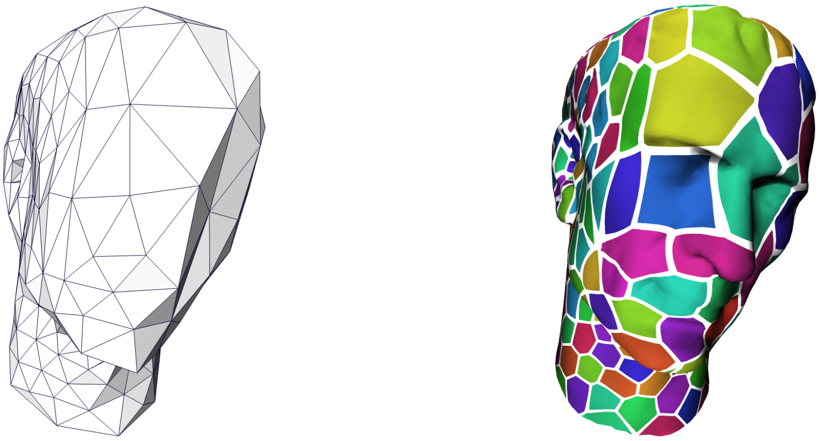
Figure 5.20: Results for the generation of anisotropic sample distributions in the two-manifold domain ($\eta = 1.0, n = 10,535, t = 1.6s$). The zooms show the anisotropic mesh before (left) and after (middle) 100 relaxation steps, and the resulting glyph distribution on the basis of the relaxed sample set (right).

the triangle-based approach much faster compared to the particle-based sampling that was presented in Section 5.4. Furthermore, in contrast to the particle-based approach, only one parameter is required to generate a sample set: the fillrate that controls the distribution's density. Additionally, an importance function can be used to generate adaptive sample distributions.

Motivated by Lloyd relaxation, which is commonly used in remeshing and sampling approaches, we have proposed a gravitational-centered relaxation to equalize triangle sizes with respect to the metric tensor field. Gravitational-centered relaxation has the advantage that it is solely triangle-based and does not require an explicit representation of an anisotropic Voronoi diagram. This makes it a stable and time-efficient method to generate more uniform sample distributions as they are needed for the visualization of the anisotropic Voronoi diagram. For applications such as stippling, however, the relaxed sample set is too uniform. Here, the initial sampling result might be preferable over the relaxed version. In contrast to the triangle-based approach, much less iterations are needed for the refinement based on gravitational-centered relaxation. Moreover, no additional insert or remove operations are required.

The most time-consuming step during initial sampling and relaxation in the two-manifold domain is the back-projection, because we need to maintain the correspondence between sample mesh and original mesh. For the timings in this section, the back-projection was done every time a new sample was inserted or every time a sample was moved. Here, a speed-up can be achieved if the back-projection is only done once after adding all samples. Then, a breadth-first search starting from the last intersected triangle might be sufficient.

6 Anisotropic Voronoi Cell Rendering


(a) Sample mesh

(b) Voronoi cells

Figure 6.1: On the basis of a well distributed point set, an anisotropic sample mesh is generated (a). This mesh is the basis to compute anisotropic Voronoi cells (b) that are used as base elements for texture generation.

6.1 Introduction

This chapter presents a novel tensor visualization method that works on two-manifold domains. The basic idea is to compute an anisotropic Voronoi diagram which consists of Voronoi cells that are distorted with respect to a local metric (Figure 6.1). These anisotropic Voronoi cells then are used as base elements for texture mapping. We show that such a diagram can be computed efficiently in the fragment shader if a well-distributed sample set as presented in Chapter 5 is given as input. Main design parameters of the visualization approach are the metric tensor field (see also Section 5.3.4) and textures that are mapped onto Voronoi cells. We show that textures provide a variety of possibilities to design the final image. A hatch pattern, for example, creates images that have a painterly character; using a single line as input texture produces images that reveal similar information like LIC textures (see Section 3.4.4); using a single dot as input texture results in a visualization that is similar to glyph-based depictions.

This chapter is based on the paper *Anisotropic Sampling of Planar and Two-Manifold Domains for Texture Generation and Glyph Distribution* [Kratz et al., 2013b].

Structure This chapter is organized as follows. First, we introduce our definition of an anisotropic Voronoi diagram (Section 6.2). Then we present a GPU-based implementation for rendering and texturing of such a diagram in Section 6.3. In Section 6.4 we show the applicability of the presented visualization method on various examples. In particular, we visualize the formation of endothelia cells of a blood vessel in accordance with a blood simulation. Since endothelia cells naturally have shapes that resemble anisotropic Voronoi cells, this is an application that directly benefits from our approach.

6.2 Anisotropic Voronoi Diagrams

We compute anisotropic Voronoi diagrams on the basis of a well-distributed sample set, for example, by using one of the methods presented in Chapter 5. Given such a sample set, we are able to compute the anisotropic Voronoi diagram using a simplified distance measure and without requiring a surface parameterization.

Let S be a set of well-defined sample positions $\mathbf{p}_i$. In our case, a generalized Voronoi diagram partitions the domain Ω into n Voronoi cells Ω_i, where each region corresponds to a *Voronoi site* that is centered at $\mathbf{p}_i$. In this work, a site has elliptical or ellipsoidal shape and is described by a metric tensor $\mathbf{M}$ (Section 5.3.2). A Voronoi cell Ω_i of a site centered in $\mathbf{p}_i$ then is defined as the set of all points $P \subset \Omega$ that are at least as close to the site in $\mathbf{p}_i$ than to any other site in $\mathbf{p}_j \in \Omega$ with $j = 1 \dots n$ and $i \neq j$

$$\Omega_i = \{P \in \Omega | d_{\mathbf{M}(\mathbf{p}_i)}(\mathbf{p}_i, P) \leq d_{\mathbf{M}(\mathbf{p}_j)}(\mathbf{p}_j, P)\}, \tag{6.1}$$

with $d_{\mathbf{M}(\mathbf{p})}$ (Equation (6.2)) using the 3D coordinates of $\mathbf{p}_i$.

In metric space, Voronoi cells are not bounded by straight lines but by curves. Furthermore, they might be neither convex nor connected. In this case, *orphans* can appear,

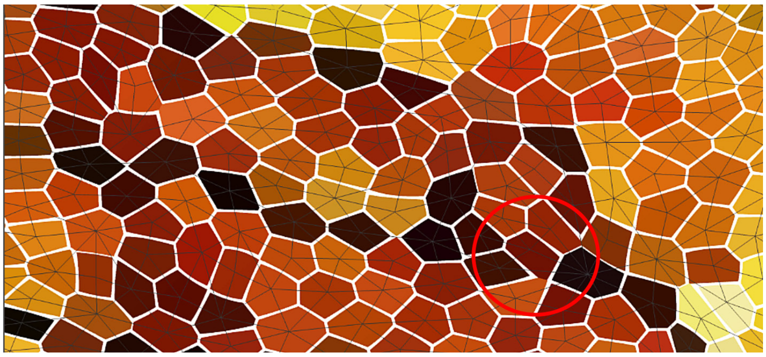


Figure 6.2: Visual artifacts appear only rarely. They are visible if the borders of the Voronoi cells are displayed. The triangulation and the Voronoi diagram were computed independently in this example, i.e., there is no duality between both graphs.

which are part of a Voronoi region that do not necessarily contain the region's barycenter (see, for example, the work of [Feng et al., 2008]). The basic idea of our approach is to compute the Voronoi diagram on the basis of a sample set that has the property that all samples are equally distributed across the input domain Ω with respect to an underlying metric tensor field. For such well-defined sample distributions, orphans generally do not appear and the Voronoi diagram can be computed via a simplified distance measure.

The most natural generalization of the Voronoi diagram is to use a geodesic distance measure, which, however, is computationally expensive. To ensure a time-efficient solution, we apply a distance measure that has already been successfully used in previous work [Labelle and Shewchuk, 2003; Feng et al., 2008] to generalize Voronoi diagrams. Assuming a local metric $\mathbf{M_p}$ and two points $\mathbf{p}$ and $\mathbf{q}$, the simplified distance measure is defined as

$$d_{\mathbf{M_p}}(\mathbf{p},\mathbf{q}) = \|\mathbf{q}-\mathbf{p}\|_{\mathbf{M_p}} = \sqrt{(\mathbf{q}-\mathbf{p})^T \cdot \mathbf{M_p} \cdot (\mathbf{q}-\mathbf{p})}. \tag{6.2}$$

This distance measure simulates a piecewise constant metric. Thus, it fits very well to the idea of elliptic glyphs, which represent the region of the tensor field that they cover, and it is also generalizable to two-manifold domains.

Our method does not guarantee the absence of orphans in areas where the anisotropy of the metric tensor field varies strongly. However, as we use the Voronoi diagram for visualization purposes only, the rare appearance of orphans is negligible. Visual artifacts only occur when the borders of the Voronoi cells are displayed (Figure 6.2). They are not visible if the cells are texturized. Furthermore, we require that the input tensor fields locally do not have large differences in anisotropy, because only then the Voronoi cells are valid representatives for the part of the tensor field that they cover (see also Section 5.3.2). In contrast, for tensor fields that have a high *local* variation of anisotropy, the appearance of orphans would be more likely.

6.3 Rendering and Texturing

To guarantee interactivity, which is one of our most important requirements, we present a GPU implementation that computes the Voronoi cells in the fragment shader. In contrast to previous GPU implementations of generalized Voronoi diagrams in the planar domain (e.g., [Yuan et al., 2011]) or centroidal Voronoi diagrams in the two-manifold domain (e.g., [Rong et al., 2011]), we do not need an explicit representation of the diagram. For mapping textures into Voronoi regions, we solely need to know the region's centroid and the local metric at that point.

The following algorithm was implemented using the Open Graphics Library (OpenGL) and shader programs of the OpenGL Shading Language (GLSL).

6.3.1 Pre-Processing and Data Structures

This section introduces the data structures and the pre-processing necessary to compute anisotropic Voronoi cells in the fragment shader. In the following, we assume that a sample set and its anisotropic triangulation, as described in Chapter 5, are given.

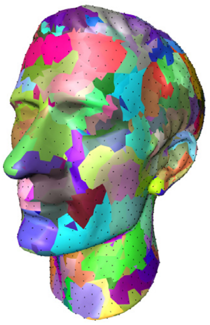


Figure 6.3: Enriched surface

Enriched original surface To compute Voronoi cells, we require a data structure that provides information about the sample's local neighborhood. Therefore, input of the rendering step is the original high-quality mesh $\mathcal{X}$ that additionally contains the information about the 3D sample positions and their corresponding site ids. Since this *meta data* is encoded in the sample mesh $\widehat{\mathcal{X}}$, we *splat* the site ids of $\widehat{\mathcal{X}}$ onto the triangles of $\mathcal{X}$. That is, the id of a sample is projected onto $\mathcal{X}$ and the information is spread into its neighborhood (Figure 6.3). To achieve this, we use the following approach.

Initially, all triangles of $\mathcal{X}$ are labeled with $i = -1$. Then, all vertices of $\widehat{\mathcal{X}}$ are inspected and their ids are splatted onto $\mathcal{X}$ via a breadth-first search. First, the current site is projected onto $\mathcal{X}$ (see also Section 5.5.3). Starting from this intersected triangle, we assign each neighboring triangle to the current site if the triangle's id is $i < 0$, i.e., no id was assigned to this triangle so far. If $i > 0$, we compute the Euclidean distance in 3D space to the current site and to the previously assigned site. If the distance to the current site is smaller, the triangle's site id is updated. The size of the neighborhood that needs to be considered depends on the relation between the number of triangles of the original surface and the number of samples. Each sample of $\mathcal{X}$ needs to know which triangles of $\mathcal{X}$ are in its vicinity. The correspondence between $\widehat{\mathcal{X}}$ and $\mathcal{X}$ does not need to be precise, because we use second-order neighbors for the computation of the Voronoi cells. We only need one triangle that is in the vicinity of the current sample. Then, we can determine the correct neighbors using this sample as starting point.

GPU Data Structures

The information that we need to compute anisotropic Voronoi cells in the fragment shader and to evaluate the distance function given in Equation (6.2) are the metric tensors, the site ids and the coordinates of (at least) the current sample's one-ring neighborhood.

The basic idea is to consider the samples $\mathbf{p}_i$ of S as Voronoi sites around which the regions Ω_i are generated. Each site is described by:

- **Id:** A unique id $i = 1 \ldots n$
- **Shape:** Coordinates $\mathbf{p}_i \in \mathbb{R}^3$ and metric $\mathbf{M}(\mathbf{p}_i)$

In order to upload this information to the GPU, where the rendering is performed, we encode this information into two textures (see also Figure 6.4):

- **Id texture:** The id texture stores the site ids and the information about the local one-ring neighborhood of a site.
- **Attribute texture:** The attribute texture stores the coordinates and the metric tensor of a site.

Id texture The id texture stores the information about the local one-ring neighborhood of a site in the luminance channel of a 3D floating point texture. To build this texture, we traverse all sites of the sample mesh. The site's id determines the position i, j in the 3D texture where then its local neighborhood is stored in z-direction. If n is the number of samples or Voronoi sites, we set the texture size to $\lceil \sqrt{n}\, \rceil \times \lceil \sqrt{n}\, \rceil \times 16$. The depth of 16 is a fixed value and ensures enough memory for the whole one-ring neighborhood. Actually, the texture's depth depends on the degree (or valence) of the vertex v that represents the current Voronoi site. Consequently, a few texels will not contain any information and, thus, are superfluous. They are labeled as invalid, that is, $i = -1$. When traversing the neighbors of a site, hence, an id of $i = -1$ indicates that all neighbors of the current site have been considered.

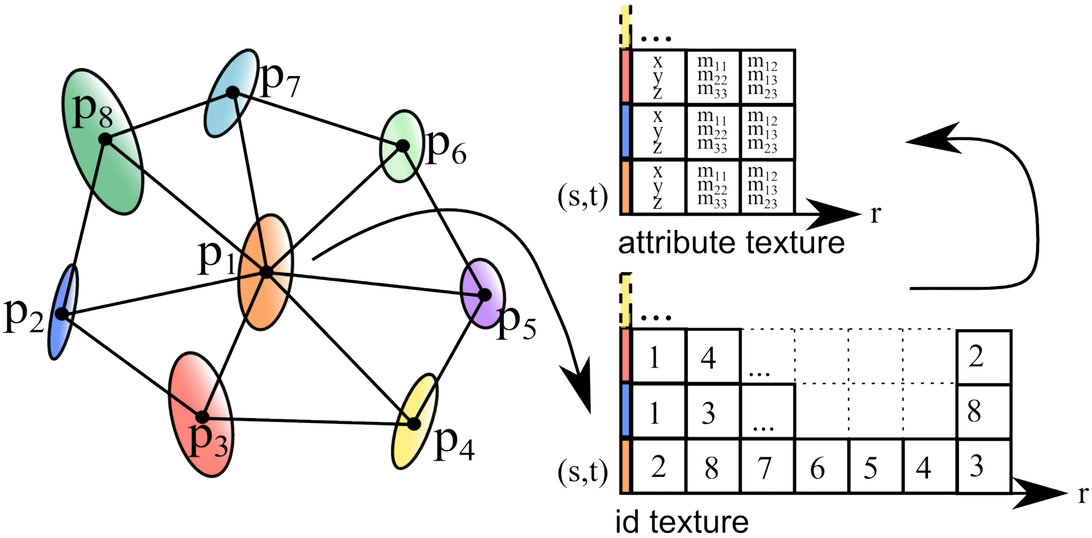


Figure 6.4: GPU data structures for the computation of Voronoi cells. The id texture stores the ids of the current sample's one-ring neighborhood. The attribute texture stores the corresponding 3D coordinates and tensors. (r, s, t) denote texture coordinates.

Attribute texture The attribute texture stores the coordinates and the metric tensor of a site in the *RGB*-channels of a 3D floating point texture. The first texture layer stores the coordinates, where $R_1 = \mathbf{x}, G_1 = \mathbf{y}, B_1 = \mathbf{z}$. The second and third texture layers store the tensor components, where $R_2 = m_{11}, G_2 = m_{22}, B_2 = m_{33}$ and $R_3 = m_{12}, G_3 = m_{13}, B_3 = m_{23}$. We set the size of the metric texture to $\lceil\sqrt{n}\,\rceil \times \lceil\sqrt{n}\,\rceil \times 3$, for 3D metric tensors, and $\lceil\sqrt{n}\,\rceil \times \lceil\sqrt{n}\,\rceil \times 2$, for 2D metric tensors.

6.3.2 Voronoi Cell Computation

Given all the data needed (Sec. 6.3.1) to draw the Voronoi cells on the original surface, the cells are computed in the vertex and fragment stages. All geometry information (vertex positions, normals, site ids) of the enriched original mesh is uploaded in a vertex buffer object. Surface rendering then is initiated by rendering $\mathcal{X}$.

The vertex stage mainly processes the information of the enriched original mesh. Besides the regular operations that are performed in the vertex stage, the interpolated vertex coordinates and the *flat* site ids are passed to the fragment shader.

The fragment stage then computes the Voronoi cells on a per-fragment basis. In addition to the interpolated vertex coordinates and the site id of the current vertex, the id texture and the attribute texture are provided as input. With this information, we can determine - for the current fragment - which site the interpolated vertex position is nearest to using the distance measure given in Equation (6.2). Depending on this assignment, the fragment is colored or texturized.

6.3.3 Texturing of Voronoi Cells

To compute the texture coordinates of the current Voronoi cell, we need the interpolated vertex coordinate $\widehat{\mathbf{p}}_i$, the coordinates of the current site $\mathbf{p}_i$ and the metric tensor at the current site $\mathbf{M}(\mathbf{p}_i)$. The computation of the texture coordinates is done in the fragment shader.

The coordinates of the current vertex position are transformed into the local coordinate system of the current Voronoi cell Ω_i, i.e., the coordinate system that is distorted by the local metric and with $\mathbf{p}_i$ lying in the center

$$\mathbf{p}_{\Omega_i} = (\mathbf{M}(\mathbf{p}_i)^{-1} \cdot (\widehat{\mathbf{p}}_i - \mathbf{p}_i)) + \frac{w}{2}. \tag{6.3}$$

Here, w is the width of the input texture and $w = h$ is assumed. Up to this stage, all computations were performed in $\mathbb{R}^3$. To get the 2D texture coordinates $\mathbf{p}_{uv} \in [0, 1]$, $\mathbf{p}_{\Omega_i}$ is projected into the 2D parameter space of the cell and normalized to the $[0, 1]$-range

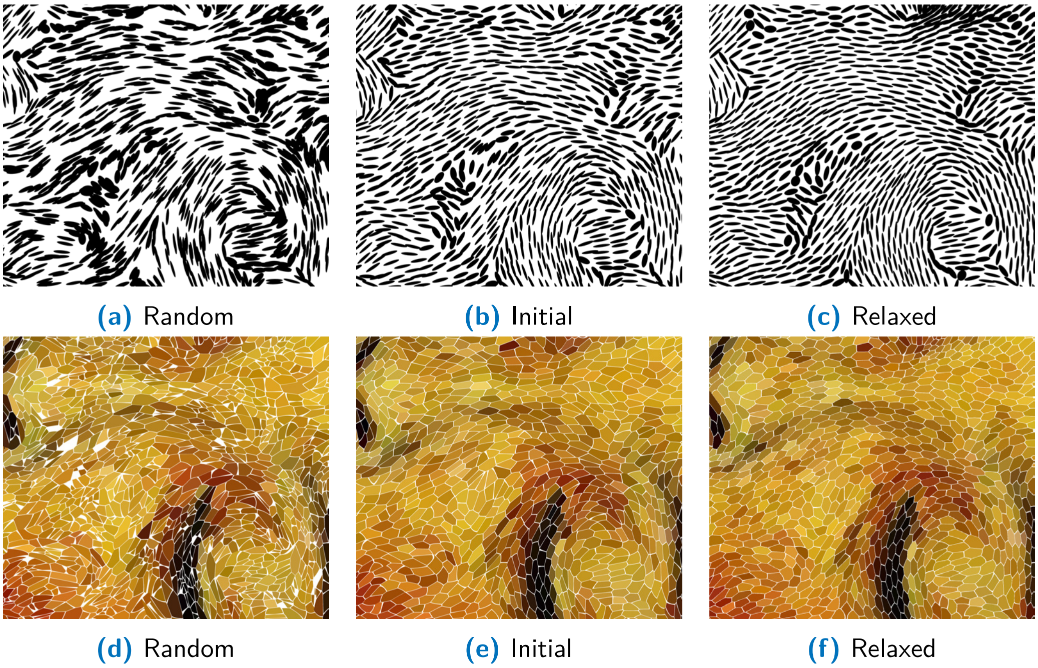
$$\mathbf{p}_u = \frac{\langle \mathbf{p}_{\Omega_i}, \mathbf{e}_1 \rangle}{w}; \qquad \mathbf{p}_v = \frac{\langle \mathbf{p}_{\Omega_i}, \mathbf{e}_2 \rangle}{w}. \tag{6.4}$$

Here, the projection vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ are the major and medium eigenvectors of the projected metric tensor $\widehat{\mathbf{M}}$ (Equation (2.27)).

6.4 Results and Discussion

To demonstrate the potential of our approach, we present Voronoi-based visualizations of several examples for the planar as well as the two-manifold case. To show results in the planar domain, we used color images as input from which we derived a metric tensor field. See Section 5.3.4 for further details on how the metric tensor field is generated. We also show Voronoi-based visualizations of slices that were extracted from 3D tensor fields (see Appendix A).

Considering the performance, anisotropic Voronoi-cell rendering is achieved at interactive frame rates once a valid sample set is computed. In the current implementation, we have used quadratic textures as GPU data structures, which results in some redundant storage and, thus, a higher memory consumption than actually needed. In the future



(a) Random **(b)** Initial **(c)** Relaxed

(d) Random **(e)** Initial **(f)** Relaxed

Figure 6.5: Input sample sets (a), (b), (c) and the corresponding Voronoi cell rendering (d), (e), (f). The zooms show that holes and overlaps in the sample distribution result in artifacts and erroneous cells in the Voronoi visualization. For well-defined sample distributions such as in (e) and (f) also the anisotropic Voronoi diagram is of high quality.

we will improve this through the use of independent texture fetches and index buffer objects. However, for the examples presented in this work, memory consumption was not an issue. For good visualization results, a lower number of samples is preferable over a higher number, since many samples result in many small Voronoi cells that are difficult to perceive by a human observer.

Figure 6.5 demonstrates the effect of the input sample distribution on the final image. Using a random sample distribution as input, artifacts occur in the Voronoi-based visualization that are caused by holes and clutter in the initial sample distribution. For a good approximation of anisotropic Voronoi diagrams, a well-distributed sample set is required as input. While the initial sample set already produces good results, an even higher quality is achieved with the relaxed sample set. For the generation of Figure 6.5 and Figure 6.6, the input sample sets were generated via the triangle-based approach that was presented in Section 5.5. To create the relaxed sample set that was used as input for Figure 6.6, 100 relaxation steps were carried out in 1.5 seconds. That is, a single relaxation step in the planar domain took 15 ms for this example.

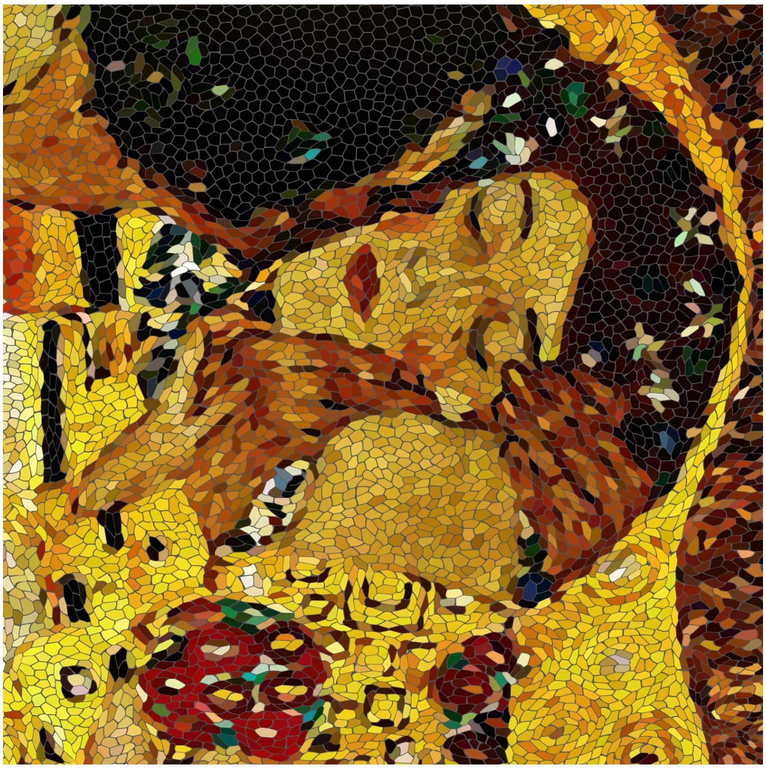


Figure 6.6: Mosaic-like image generated with anisotropic Voronoi cell rendering.

Figure 6.7 shows example renderings using different input textures. Drawing the borders of the Voronoi cells results in mosaic-like images (Figure 6.7a). If an input texture with a single point is used as input, the resulting image is similar to a glyph-based visualization (Figure 6.7c). Mapping a hatch pattern onto the anisotropic Voronoi cells creates an image that has a painterly character emphasizing the major eigenvector field (Figure 6.7d). The input sample set was generated via the particle-based approach presented in Section 5.4. To create the relaxed sample set, 1000 iterations were carried out.

Figure 6.8 shows example renderings using different input textures for an analytic tensor field that has one degenerate point in the center. The images show texturized Voronoi cells using two different line textures as input (Figures 6.8b, 6.8c). The color is more transparent in isotropic regions where directions are not defined. The LIC texture (Figure 6.8a) is shown for a comparison. The input sample set was generated via the

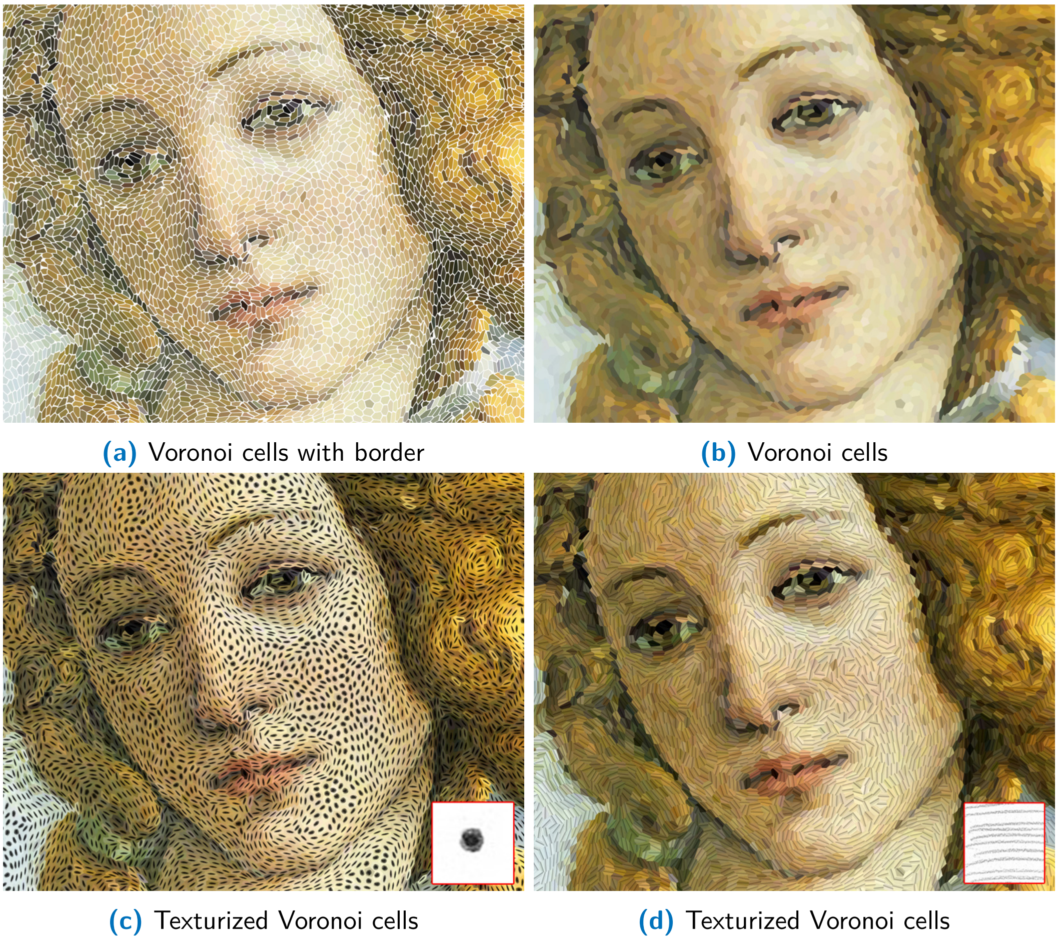
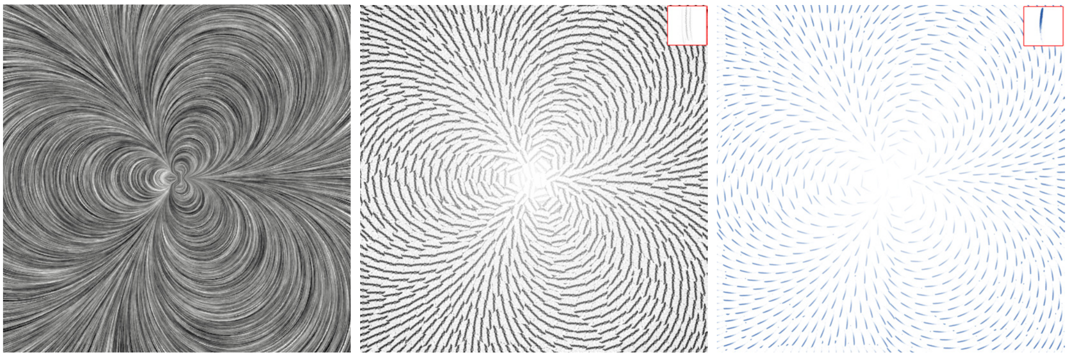

(a) Voronoi cells with border
(b) Voronoi cells
(c) Texturized Voronoi cells
(d) Texturized Voronoi cells

Figure 6.7: Anisotropic Voronoi cells with and with no border (a, b). A point input texture (c) and a hatch pattern (d) mapped to anisotropic Voronoi cells.

particle-based approach presented in Section 5.4. To create the relaxed sample set, 100 iterations were needed.

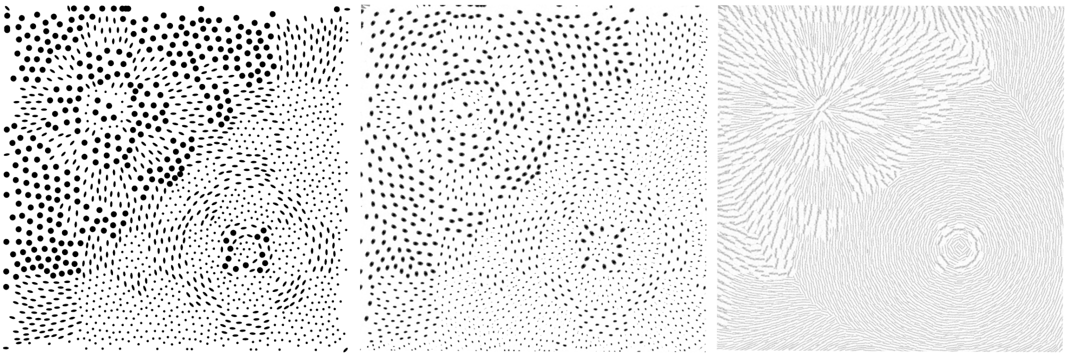
Figure 6.9 shows renderings for a slice extracted from the two-point-load dataset (see Appendix A.1). Using a point as input texture, the resulting image (Figure 6.9b) is similar to a glyph-based rendering (Figure 6.9a). Using a line as input texture, an image is created that reveals similar information like LIC textures (Figure 6.9c).

To show results in the two-manifold domain, we applied our method to several analytic surfaces (Figure 6.10). In these examples, the metric tensor fields were generated randomly. We further visualize the formation of endothelia cells of a blood vessel in



(a) LIC texture (b) Texturized Voronoi cells (c) Texturized Voronoi cells

Figure 6.8: Two different hatch patterns applied to anisotropic Voronoi cells (b, c) and a LIC texture (a) for comparison.



(a) Glyph rendering (b) Texturized Voronoi cells (c) Texturized Voronoi cells

Figure 6.9: Glyph rendering of a slice of the two-point-load dataset (a). A point (b) and a hatch pattern (c) applied to anisotropic Voronoi cells extracted on the same slice.

accordance with the simulation of blood flow in an aneurysm (Figure 6.11). Since endothelia cells naturally have shapes that are similar to anisotropic Voronoi cells (Figure 6.12), this is an application that directly benefits from our approach.

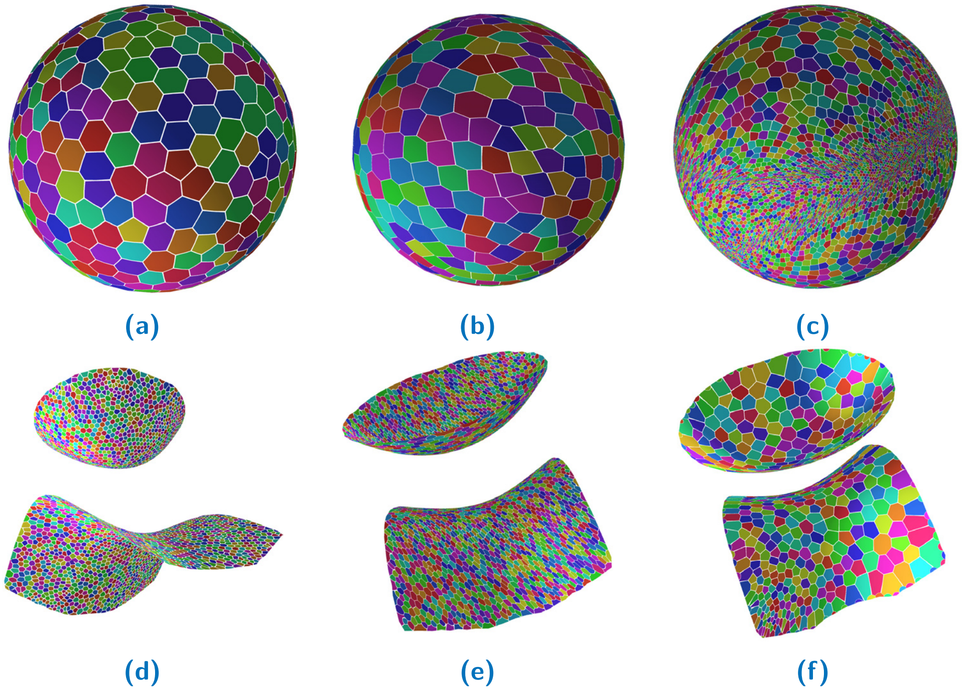


Figure 6.10: Anisotropic Voronoi diagrams for several analytic surfaces (sphere (a-c) and calypso (d-f)). The metric tensor fields in these examples were generated randomly. See Figure 5.19 in Chapter 5 for the timings that were needed for the sample generation. Rendering of these surfaces performs at interactive rates (≈ 100 fps).

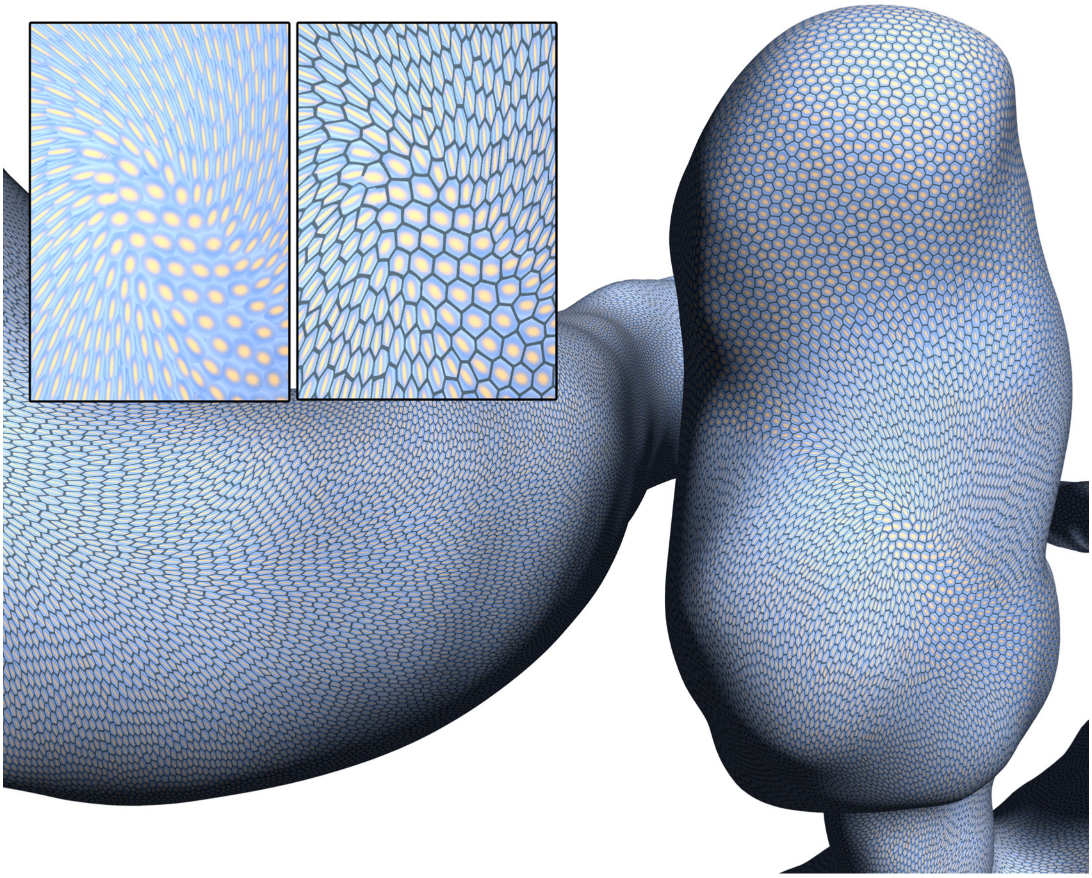


Figure 6.11: Visualizations generated via anisotropic Voronoi cell rendering. Voronoi cells were generated on the basis of a surface vector field that represents the wall shear stress. The input texture was designed so that it resembles a schematic depiction of the endothelia cells of a blood vessel (see Figure 6.12 below). The Voronoi cells align with the blood flow.

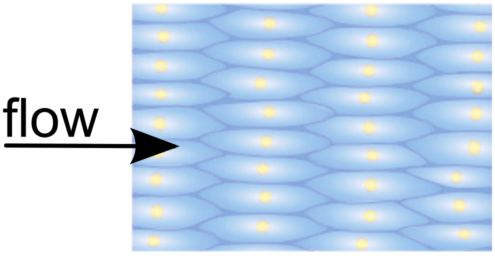


Figure 6.12: Schematic depiction of endothelia cells motivated by a depiction in the work of Malek et al. [1999].

7 Conclusion and Outlook

In this chapter, the thesis concludes with a summary of the major contributions (Section 7.1), an outlook into possible future work (Section 7.2), and some final remarks (Section 7.3).

7.1 Summary

The major motivation of this thesis was to develop visualization and analysis methods to investigate 3D indefinite tensor fields. The major goals that were revealed in Section 1.3 were the development of an undirected visualization concept, the possibility to extract specific tensor properties on-the-fly and the development of algorithms to generate anisotropic sample distributions on two-manifold domains. To achieve this, the thesis presented algorithmic as well as conceptual solutions that build upon ideas from information visualization, scientific visualization, data analysis and computer graphics.

Categorization and analysis of previous work Besides conceptual and technical contributions, the thesis also faced the following question: What are the challenges for the visualization of tensors that are not positive definite? In Chapter 3 and with the state-of-the-art report [Kratz et al., 2013a], we highlighted the major challenges for tensors that are not symmetric and positive definite. Moreover, we presented and classified existing research work.

Visual data exploration In Chapter 4, we presented a concept that adapts the idea of multiple linked views to stress tensor fields. With the resulting framework, we presented a tool to visualize and analyze stress tensor fields. By combining a wealth of visualization methods, we were able to generate new insights into the data. In close collaboration with domain experts from the field of engineering, we could also generate novel, promising hypotheses concerning stress tensor fields. Various discussions with our cooperation partners led to the impression that the *directional information* contained in stress tensors might be very valuable. Although looking at the major eigenvector field is common in the field of DTI, this is a new finding for stress tensor fields from engineering. We believe that with the simultaneous depiction of the major and the minor eigenvector fields of stress tensors from structure simulations, badly designed construction elements can be identified. As a consequence, a new hypothesis is that the visualization of directional information might also have the potential to guide an *optimal* design of construction elements. Hence, the visualizations that were presented in this thesis have created new hypotheses about the data. Now, these need more investigation on the engineering side.

Anisotropic sampling In Chapter 5, we presented two approaches for the generation of anisotropic sample distributions in 2D domains. Especially with the triangle-based approach (Section 5.5), we contribute with an algorithm that is easy-to-implement, time efficient and leads to stable results even if sample sizes vary strongly across the given domain Ω. We demonstrated that such sample distributions guide an optimal placement of glyphs to improve the quality of fabric textures and to compute anisotropic Voronoi cells for texturization. Furthermore, the triangle-based approach can be used in conjunction with slicing, which enables interactive exploration of three-dimensional tensor fields.

Texturing of anisotropic Voronoi cells In Chapter 6, we introduced a novel visualization algorithm for tensor fields. Once a sample set has been generated with one of the two methods presented in Chapter 5, the rendering and texturing of anisotropic Voronoi cells works at interactive frame rates. With this approach, the usage of textures to encode the six degrees of freedom of a 3D second-order tensor becomes possible.

7.2 Future Work

Despite the contributions that are summarized in Section 7.1, many fundamental questions and technical problems remain or evolved within the scope of this thesis.

Visual data exploration Although we could create new insights into the data and were able to reveal interesting structures in stress tensor fields from engineering, feature definition for (stress) tensors remains an open problem. Compared to scalar, vector and flow visualization, rarely any feature definitions exist. For example, in most flow fields, regardless of the application field, the extraction of vortices is of interest. For tensor fields, there is no comparable structure. This becomes even worse when looking at tensor fields in diverse application areas. As long as there is a lack of clear feature definitions, visual data exploration is a necessity. Here, multiple view visualization systems can significantly help, because they offer the possibility to combine 2D plots and diagrams - which users are familiar with - with 3D visualizations and advanced glyph representations. Hence, they enable the identification of interesting features. Once new features of interest have been identified, advanced visualization techniques can be developed specifically for these features.

Texture-based approaches for tensor fields Anisotropic Voronoi cell rendering is very flexible and can be applied to many more applications than those that were presented in this work. We believe that the use of textures within the field of tensor visualization is very powerful. Future work to extend the presented anisotropic Voronoi cell rendering may include the design of more specific textures and the development of texture synthesis methods for the visualization of tensor fields.

7.3 Concluding Remarks

The major course of this thesis was to investigate 3D tensor fields through interactive visual exploration, either by combining a multitude of visualization methods that are interactively steered and exchanged, by various textures that highlight specific tensor properties, or via slicing through 3D tensor fields. Keeping in mind the tensor's six degrees of freedom and the difficultly to interpret them, interactive methods to explore indefinite tensor fields are a promising field.

A
Datasets

The following list introduces the datasets that were used in this thesis.

A.1 Two-Point Load

This dataset resulted from a FEM simulation of a block on which loads in two directions were applied that cause tensile and compressive stresses. To produce the results that are presented in this thesis, the data was resampled on a uniform grid with a resolution of $30 \times 30 \times 30$. Despite this very low resolution, the dataset describes the most typical characteristics of stress tensor fields and, therefore, is well suited to understand very basic properties of these fields. We thank Boris Jeremic from the University of California, Davis, for providing this dataset.

A.2 One-Point Load

This dataset is similar to the two-point load (Section A.1) and also the result of a FEM simulation. In this case, a load in one direction was applied that causes tensile stresses. To produce the results that are presented in this thesis, the data was resampled on a uniform grid with a resolution of $30 \times 30 \times 30$. We thank Boris Jeremic from the University of California, Davis, for providing this dataset.

A.3 Rotating Neutron Star

This time-dependent dataset resulted from an astrophysical simulation of a rotating neutron star's dynamics. Analyzing the evolution of such systems plays a major role for the understanding of the fundamental processes involved in core collapse supernovae and gravitational wave production. The simulation results consist of (complex) scalar-, vector- and tensor fields. In this work, we focus on the resulting tensor fields. The 3D data is initially given on a grid with spatially varying resolution (AMR). To produce the results that are presented in this thesis, the data was resampled on a uniform grid with a resolution of 128 × 128 × 128 samples.

Note that, due to the high gravitational forces inside the star, only compressive stresses occur. According to our sign convention (Equation (2.8)) this means that all stresses are negative. In such a case it is common in the respective application areas to consider only the absolute value of the stresses. Hence, the principal stresses are ordered according to their magnitude, i.e., $|\sigma_1| \geq |\sigma_2| \geq |\sigma_3|$. As a consequence, the dataset reveals positive-definite behavior. We thank Luca Baiotti from the Albert Einstein Institute (AEI), Potsdam, for providing this dataset.

A.4 Aneurysm

The aneurysm dataset resulted from a blood flow simulation. The images that are presented in Chapter 5 and in Chapter 6 were computed on the basis of a metric tensor field. This metric tensor field was generated on the basis of an input surface vector field (Figure A.1) that represents the *wall shear stress*. The data is given on the nodes of a triangulated surface field with 112088 nodes. See Section 5.3.4 for the details how the metric tensor field is derived. The dataset was provided by Leonid Goubergrits and Jens Schaller from Charité Berlin.

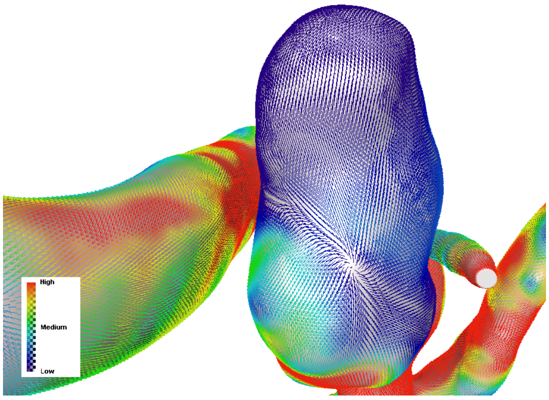
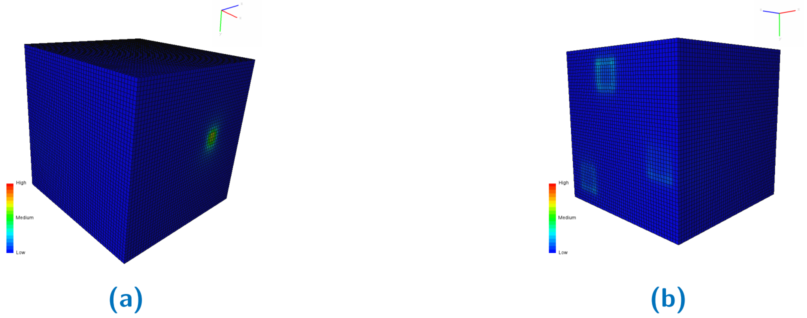


Figure A.1: Aneurysm dataset. The surface vector field is color-coded according to the magnitude of the wall shear stress.

A.5 Cube

This dataset is the result of a FEM simulation of a cube on which loads in two directions were applied. The magnitude of the implemented loads was $5N$ in x-direction and $10N$ in y-direction. Opposite to this local forces, the cube was fixed at exactly three stationary points. Due to the different directions of load, tension and pressure are created in this cubic structure. To apply the presented tensor visualization methods to this dataset, the data was resampled on a uniform grid of size $64 \times 64 \times 64$. We thank Prof. Dr. Markus Stommel from Saarland University for providing this dataset.



(a) (b)

Figure A.2: Cube dataset. Color-coding according to the von Mises stress.

A.6 Beam Profile

This dataset is the result of a FEM simulation of a construction element. The material has the properties of a polymer. We have considered three variants of this simulation, which are described in the following:

Asymmetric force In this simulation, the force that acts on one side of this material is a surface tension. On the opposite vertical surface, the element was fixed. Besides an expected bending in z-direction, the construction element gets twisted by the resulting torsion forces due to the asymmetric force transmission in this example.

Symmetric forces In this simulation, the force that acts on one side of the element is a surface tension, too, but it is applied directly in the middle. On the opposite vertical surface, the element was fixed. Here, only a bending in z-direction occurs due to the symmetric force transmission in this example.

Filled with air An alternative simulation used the same forces but parts of the construction element were modeled to resemble the behavior of *air*. The data is given on the nodes of a tetrahedral mesh with 14643 nodes. We thank Marc Schöneich from Saarland University for providing this dataset.

High
Medium
Low
High
Medium
Low

(a)

(b)

Figure A.3: Beam profile. Color-coding according to the von Mises stress.

A.7 Shear Specimen

This dataset is the result of a FEM simulation. It shows a *part* of a construction element in which shear happens. Similar to the beam profile described in Section A.6, one vertical side was fixed. On the opposite vertical surface, a surface load was applied. This type of load most closely corresponds to *real* experiments in which both sides are clamped and, hence, are fixed. The geometry has two notches, which are the interesting parts of this dataset. Here, the generated shear forces can be evaluated. The data is given on the nodes of a tetrahedral mesh with 20770 nodes. We thank Marc Schöneich from Saarland University for providing this dataset.

High
Medium
Low
High
Medium
Low

(a)

(b)

Figure A.4: Shear specimen. Color-coding according to the von Mises stress.

Bibliography

R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis and Applications*. Springer, New York, 1988.

S. Aja-Fernández, R. de Luis García, D. Tao, and X. Li, editors. *Tensors in Image Processing and Computer Vision*. Springer London, 2009.

P. Alliez, D. Cohen-Steiner, O. Devillers, B. Levy, and M. Desbrun. Anisotropic Polygonal Remeshing. *ACM Transactions on Graphics (SIGGRAPH '03)*, 22(3): 485–493, 2003.

P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. Recent Advances in Remeshing of Surfaces. State-of-the-Art Report, AIM@SHAPE EU Network, 2005.

C. Auer, J. Sreevalsan–Nair, V. Zobel, and I. Hotz. 2D Tensor Field Segmentation. In *Scientific Visualization: Interactions, Features, Metaphors*, volume 2 of *Dagstuhl Follow-Ups*, pages 17–35. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2011.

C. Auer, C. Stripf, A. Kratz, and I. Hotz. Glyph- and Texture-based Visualization of Segmented Tensor Fields. In *International Conference on Information Visualization Theory and Applications (IVAPP'12)*, pages 670–677, Rome, Italy, 2012.

M. M. Bahn. Invariant and Orthonormal Scalar Measures Derived from Magnetic Resonance Diffusion Tensor Imaging. *Journal of Magnetic Resonance*, 141(1):68–77, 1999.

A. H. Barr. Superquadrics and Angle-Preserving Transformations. *IEEE Comput. Graph. Appl.*, 1(1):11–23, 1981.

W. Benger. *Visualization of General Relativistic Tensor Fields via a Fiber Bundle Data Model*. Ph.d. thesis, Free University Berlin, 2004.

W. Benger and H.-C. Hege. Tensor Splats. In *Visualization and Data Analysis '04*, volume 5295, pages 151–162, San Jose, USA, 2004. Proceedings of SPIE.

D. Bergeron. Visualization reference model (panel session position statement). In *Proceedings of the Conference on Visualization (Vis'93)*, pages 337–342, 1993.

A. Bhalerao and C.-F. Westin. Tensor Splats: Visualising Tensor Fields by Texture Mapped Volume Rendering. In *MICCAI'03*, pages 294–302, Montreal, Canada, 2003.

M. Bilgen and P. A. Narayana. Mohr Diagram Interpretation of Anisotropic Diffusion Indices in MRI. *Magn Reson Med*, 31:567–572, 2003.

M. Bilgen, I. Elshafiey, and P. A. Narayana. Mohr Diagram Representation of Anisotropic Diffusion Tensor in MRI. *Magn Reson Med*, 47:823–827, 2002.

J.-D. Boissonnat, C. Wormser, and M. Yvinec. Locally Uniform Anisotropic Meshing. In *Proceedings of the annual symposium on computational geometry*, pages 270–277, 2008.

F. J. Bossen and P. S. Heckbert. A Pliant Method for Anisotropic Mesh Generation. In *5th Intl. Meshing Roundtable*, pages 63–74, 1996.

R. M. Brannon. *Functional and Structured Tensor Analysis for Engineers*. The University of Utah 2003, Utah, USA, 2003.

B. Cabral and L. Leedom. Imaging Vector Fields Using Line Integral Convolution. In J. T. Kajiya, editor, *Proceedings of SIGGRAPH'93*, volume 27, pages 263–272, New York, USA, 1993. ACM.

L. Cammoun, C. Castaño-Moraga, E. Muñoz-Moreno, D. Sosa-Cabrera, B. Acar, M. Rodriguez-Florido, A. Brun, H. Knutsson, and J. P. Thiran. A Review of Tensors and Tensor Signal Processing. In *Tensors in Image Processing and Computer Vision.* Springer, London, UK, 2009.

G. Chen, G. Esch, P. Wonka, P. Müller, and E. Zhang. Interactive Procedural Street Modeling. *ACM Transactions on Graphics (SIGGRAPH '08)*, 27(3):1–10, 2008.

G. Chen, D. Palke, Z. Lin, H. Yeh, P. Vincent, R. Laramee, and E. Zhang. Asymmetric Tensor Field Visualization for Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1979–1988, 2011.

W. Chen, Z. Ding, S. Zhang, A. MacKay-Brandt, S. Correia, H. Qu, J. A. Crow, D. F. Tate, Z. Yan, and Q. Peng. A Novel Interface for Interactive Exploration of DTI Fibers. *IEEE Transactions on Visualization and Computer Graphics*, 15:1433–1440, 2009.

Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:790–799, 1995.

D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5): 603–619, 2002. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.1000236.

R. L. Cook. Stochastic Sampling in Computer Graphics. *ACM Transactions on Graphics*, 5(1):51–72, 1986.

J. C. Criscione, J. D. Humphrey, A. S. Douglas, and W. C. Hunter. An invariant basis for natural strain which yields orthogonal stress response terms in isotropic hyperelasticity. *Journal of the Mechanics and Physics of Solids*, 48:2445–2465, 2000.

P. Crossno, D. H. Rogers, R. M. Brannon, and D. Coblentz. Visualization of Salt-Induced Stress Perturbations. In *Proceedings of the Conference on Visualization (Vis'04)*, pages 369–376, Washington, USA, 2004.

P. Crossno, D. H. Rogers, R. M. Brannon, D. Coblentz, and J. T. Fredrich. Visualization of Geologic Stress Perturbations Using Mohr Diagrams. *IEEE Transactions on Visualization and Computer Graphics*, 11(5):508–518, 2005.

J. Daniels, E. W. Anderson, L. G. Nonato, and C. T. Silva. Interactive vector field feature identification. *IEEE Transactions on Visualization and Computer Graphics (Vis'10)*, 16(6):1560–1569, 2010.

D. A. Danielson. *Vectors and Tensors in Engineering and Physics.* Addison-Wesley Publishing Company, Massachusetts, USA, 2nd edition, 1997.

R. de Luis-García, C. Alberola-López, and C.-F. Westin. Segmentation of Tensor Fields: Recent Advances and Perspectives. In *Tensors in Image Processing and Computer Vision.* Springer, London, UK, 2009.

T. Delmarcelle. *The Visualization of Second-Order Tensor Fields* . PhD thesis, Stanford University, December 1994.

T. Delmarcelle and L. Hesselink. Visualizing Second-Order Tensor Fields with Hyperstreamlines. *IEEE Comput. Graph. Appl.*, 13(4):25–33, 1993.

T. Delmarcelle and L. Hesselink. The Topology of Second-Order Tensor Fields. In *Proceedings of the conference on Visualization (Vis'94)*, pages 140–148, Washington, USA, 1994.

C. Dick, J. Georgii, R. Burgkart, and R. Westermann. Stress Tensor Field Visualization for Implant Planning in Orthopedics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1399–1406, 2009.

U. Diewald and M. Rumpf. Visualization of Principal Curvature Directions by Anisotropic Diffusion. In *Proceedings of Vision, Modeling, and Visualization (VMV'00)*, pages 293–302, Saarbrücken, Germany, 2000. Aka GmbH.

P. Dobrev, T. V. Long, and L. Linsen. A Cluster Hierarchy-based Volume Rendering Approach for Interactive Visual Exploration of Multi-variate Volume Data. In *VMV*, pages 137–144, 2011.

H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, pages 239–248. Eurographics Association, 2003.

Q. Du and D. Wang. Anisotropic Centroidal Voronoi Tessellations and Their Applications. *SIAM J. Sci. Comput.*, 26(3):737–761, Mar. 2005.

Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi Tessellations: Applications and Algorithms. *SIAM Review*, 41:637–676, 1999.

H. Edelsbrunner and J. Harer. Persistent homology — a survey. *In Twenty Years After, Contemporary Mathematics, AMS, Surveys on Discrete and Computational Geometry: Twenty Years Later*, pages 257–282, 2007.

A. M. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *Proceedings of the 6th European Conference on Computer Vision-Part II*, ECCV '00, pages 751–767, London, UK, UK, 2000. Springer-Verlag.

D. B. Ennis and G. Kindlmann. Orthogonal Tensor Invariants and the Analysis of Diffusion Tensor Magnetic Resonance Images. *Magn Reson Med*, 55(1):136–146, 2006.

C. Feddern, J. Weickert, and B. Burgeth. Level-Set Methods for Tensor-Valued Images. In *Proc. IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pages 65–72, Nice, France, 2003.

L. Feng, I. Hotz, B. Hamann, and K. Joy. Anisotropic Noise Samples. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):342–354, 2008.

P. T. Fletcher and S. Joshi. Riemannian Geometry for the Statistical Analysis of Diffusion Tensor Data. *Signal Process.*, 87(2):250–262, 2007.

L. Fritz, M. Hadwiger, G. Geier, G. Pittino, and M. E. Gröller. A visual approach to efficient analysis and quantification of ductile iron and reinforced sprayed concrete. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1343–1350, Oct. 2009.

K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, January 1975.

M. Garland and P. S. Heckbert. Surface Simplification Using Quadric Error Metrics. In *Proceedings of the conference on computer graphics and interactive techniques*, SIGGRAPH '97, pages 209–216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

M. Goldau, A. Wiebel, N. S. Gorbach, C. Melzer, M. Hlawitschka, G. Scheuermann, and M. Tittgemeyer. Fiber Stippling: An Illustrative Rendering for Probabilistic Diffusion Tractography. In *IEEE BioVis Proceedings*, pages 23–30, 2011.

D. Günther, J. Reininghaus, H. Wagner, and I. Hotz. Memory efficient computation of persistent homology for 3D image data using discrete Morse theory. In *Sibgrapi 2011 - Technical Papers*, Maceio, Alagoas, Brazil, August 2011.

R. B. Haber. Visualization Techniques for Engineering Mechanics. *Comput. Syst. Educ.*, 1(1):37–50, 1990.

H. Hagen and C. Garth. An Introduction to Tensors. In *Visualization and Processing of Tensor Fields*, Mathematics and Visualization. Springer, Berlin, Heidelberg, 2006.

Y. M. A. Hashash, J. I.-C. Yao, and D. 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.

L. Hesselink, Y. Levy, and Y. Lavin. The Topology of Symmetric, Second-Order 3D Tensor Fields. *IEEE Transactions on Visualization and Computer Graphics*, 3(1): 1–11, 1997.

M. Hlawitschka, G. Scheuermann, and B. Hamann. Interactive Glyph Placement for Tensor Fields. In *ISVC (1)*, pages 331–340, Berlin, Heidelberg, 2007. Springer.

M. Hlawitschka, S. Eichelbaum, and G. Scheuermann. Fast and Memory Efficient GPU–based Rendering of Tensor Data. In *Proceedings of the IADIS International Conference on Computer Graphics and Visualization '08*, Amsterdam, The Netherlands, 2008.

M. Hlawitschka, G. Scheuermann, G. H. Weber, O. T. Carmichael, B. Hamann, and A. Anwander. Interactive Volume Rendering of Diffusion Tensor Data. In *Visualization and Processing of Tensor Fields: Advances and Perspectives*, Mathematics and Visualization. Springer, Berlin, Heidelberg, 2009.

I. Hotz, L. Feng, H. Hagen, B. Hamann, K. Joy, and B. Jeremic. Physically-Based Methods for Tensor Field Visualization. In *Proceedings of the Conference on Visualization (Vis'04)*, pages 123–130, Washington, USA, 2004. IEEE Computer Society.

I. Hotz, J. Sreevalsan–Nair, H. Hagen, and B. Hamann. Tensor Field Reconstruction Based on Eigenvector and Eigenvalue Interpolation. In *Scientific Visualization: Advanced Concepts*, volume 1 of *Dagstuhl Follow-Ups*, pages 110–123. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2010.

T. Jankun-Kelly and K. 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.

R. Jianu, C. Demiralp, and D. Laidlaw. Exploring 3D DTI Fiber Tracts with Linked 2D Representations. *IEEE Transactions on Visualization and Computer Graphics*, 15:1449–1456, 2009.

B. Jobard and W. Lefer. Creating evenly-spaced streamlines of arbitrary density. In *8th Eurographics Workshop on Visualization in Scientific Computing*, pages 45–55, Boulogne-sur-Mer, France, April 1997. Springer Verlag.

G. Kindlmann. Superquadric Tensor Glyphs. In *Proceedings of IEEE / EG Symposium on Visualization (VisSim'04)*, pages 147–154, 2004.

G. Kindlmann and D. Weinstein. Hue-Balls and Lit-Tensors for Direct Volume Rendering of Diffusion Tensor Fields. In *Proceedings of the Conference on Visualization (Vis'99)*, pages 183–189, 1999.

G. Kindlmann and C.-F. Westin. Diffusion Tensor Visualization with Glyph Packing. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1329–1336, 2006.

G. Kindlmann, D. Weinstein, and D. Hart. Strategies for Direct Volume Rendering of Diffusion Tensor Fields. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):124–138, 2000.

G. Kindlmann, T. Schultz, X. Tricoche, M. A. O. Vasilescu, A. Vilanova, and E. Zhang. Tutorial on Tensors in Visualization at IEEE VisWeek '10, October 2010.

R. M. Kirby, H. Marmanis, and D. H. Laidlaw. Visualizing multivalued data from 2D incompressible flows using concepts from painting. In *Proc. Visualization*, pages 333–340, 1999.

J. Kopf, D. Cohen-Or, O. Deussen, and D. Lischinski. Recursive Wang Tiles for Real-Time Blue Noise. *ACM Transactions on Graphics (SIGGRAPH '06)*, 25(3): 509–518, 2006.

A. Kratz, N. Kettlitz, and I. Hotz. Particle-Based Anisotropic Sampling for Two-Dimensional Tensor Field Visualization. In *Proceedings of Vision, Modeling, and Visualization (VMV'11)*, pages 145–152, 2011a.

A. Kratz, B. Meyer, and I. Hotz. A Visual Approach to Analysis of Stress Tensor Fields. In *Scientific Visualization: Interactions, Features, Metaphors*, volume 2 of *Dagstuhl Follow-Ups*, pages 188–211. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011b.

A. Kratz, C. Auer, M. Stommel, and I. Hotz. Visualization and Analysis of Second-Order Tensors: Moving Beyond the Symmetric Positive-Definite Case. *Computer Graphics Forum*, 1:49–74, 2013a.

A. Kratz, D. Baum, and I. Hotz. Anisotropic Sampling of Planar and Two-Manifold Domains for Texture Generation and Glyph Distribution (accepted for publication). *IEEE Transactions on Visualization and Computer Graphics*, 2013b.

R. Kriz, M. Yaman, M. Harting, and A. Ray. Visualization of Zeroth, Second, Fourth, Higher Order Tensors, and Invariance of Tensor Equations. Summer Research Project 2003, 2005.

J. Kruger and R. Westermann. Acceleration techniques for gpu-based volume rendering. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pages 38–, Washington, DC, USA, 2003. IEEE Computer Society.

F. Labelle and J. R. Shewchuk. Anisotropic Voronoi Diagrams and Guaranteed-quality Anisotropic Mesh Generation. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 191–200, 2003.

A. Lagae and P. Dutré. A Comparison of Methods for Generating Poisson Disk Distributions. *Computer Graphics Forum*, 27(1):114–129, 2008.

A. Lagae and P. Dutré. A Procedural Object Distribution Function. *ACM Transactions on Graphics*, 24:1442–1461, 2005. ISSN 0730-0301.

A. Lagae, C. S. Kaplan, C.-W. Fu, V. Ostromoukhov, J. Kopf, and O. Deussen. Tile-Based Methods for Interactive Applications. SIGGRAPH 2008 Class, SIGGRAPH 2008, Los Angeles, USA, August 2008.

A. Lagae, S. Lefebvre, R. Cook, T. DeRose, G. Drettakis, D. S. Ebert, J. P. Lewis, K. Perlin, and M. Zwicker. State of the Art in Procedural Noise Functions. In *EG 2010 - State of the Art Reports*, 2010.

D. Laidlaw and J. Weickert, editors. *Visualization and Processing of Tensor Field*. Springer, 2008.

D. H. Laidlaw, E. T. Ahrens, D. Kremers, M. J. Avalos, R. E. Jacobs, and C. Readhead. Visualizing Diffusion Tensor Images of the Mouse Spinal Cord. In *Proceedings of the Conference on Visualization (Vis'98)*, pages 127–134, 1998.

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

Y. Lavin, R. Batra, L. Hesselink, and Y. Levy. The Topology of Symmetric Tensor Fields. *AIAA 13th Computational Fluid Dynamics Conference*, page 2084, 1997a.

Y. Lavin, Y. Levy, and L. Hesselink. Singularities in Nonuniform Tensor Fields. In *Proceedings of the 8th Annual IEEE Conference on Visualization (VISU-97)*, pages 59–66, 1997b.

G. Lavoue, F. Dupont, and A. Baskurt. A New CAD Mesh Segmentation Method, Based on Curvature Tensor Analysis. *Computer-Aided Design*, 37(10):975–987, 2005.

C. Lenglet, M. Rousson, R. Deriche, and l. 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, 2006.

H. Li, L.-Y. Wei, P. V. Sander, and C.-W. Fu. Anisotropic Blue Noise Sampling. *ACM Transactions on Graphics*, 29(6):1–12, 2010.

X.-Y. Li, S.-H. Teng, and A. Üngör. Biting Ellipses to Generate Anisotropic Mesh. In *IMR'99*, pages 97–108, 1999.

S. P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

B. Lund. *Crustal Stress Studies Using Microearthquakes and Boreholes*. PhD thesis, Department of Earth Sciences, Uppsala University, 2000.

R. Maciejewski, I. Woo, W. Chen, and D. Ebert. Structuring Feature Space: A Non-Parametric Method for Volumetric Transfer Function Generation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1473–1480, 2009.

A. M. Malek, S. L. Alper, and S. Izumo. Hemodynamic Shear Stress and Its Role in Atherosclerosis. *Journal of the American Medical Association*, 282(21):2035–2042, 1999.

M. McCool and F. Eugene. Hierarchical Poisson Disk Sampling Distributions. In *Proceedings of the conference on Graphics interface*, pages 94–105, 1992.

M. Moakher and P. G. Batchelor. Symmetric Positive-Definite Matrices. In *Visualization and Processing of Tensor Fields*, Mathematics and Visualization, pages 285–297. Springer, 2006.

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.

T. Munzner, C. Johnson, R. Moorhead, H. Pfister, P. Rheingans, and T. S. Yoo. NIH-NSF Visualization Research Challenges Report Summary. *IEEE Comput. Graph. Appl.*, 26(2):20–24, 2006.

C. Navarro, N. Hitschfeld-Kahler, and E. Scheihing. A Parallel GPU-based Algorithm For Delaunay Edge-flips. In M. Hoffmann, editor, *Proceedings of the 27th European Workshop on Computational Geometry*, pages 75–78, 2011.

A. Neeman, B. Jeremic, and A. Pang. Visualizing Tensor Fields in Geomechanics. In *Proceedings of the Conference on Visualization (Vis'05)*, page 5, Minneapolis, USA, 2005. IEEE Computer Society.

S. Paris and F. Durand. A Topological Approach to Hierarchical Segmentation using Mean Shift. In *CVPR*, 2007.

X. Pennec, P. Fillard, and N. Ayache. A Riemannian Framework for Tensor Computing. *Int. J. Comput. Vision*, 66:41–66, 2006.

G. Rong, Y. Liu, W. Wang, X. Yin, D. Gu, and X. Guo. GPU-Assisted Computation of Centroidal Voronoi Tessellation. *IEEE Transactions on Visualization and Computer Graphics*, 17(3):345–356, May 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2010.53.

H. Samet. An Overview of Quadtrees, Octrees, and Related Hierarchical Data Structures. In R. A. Earnshaw, editor, *Theoretical Foundations of Computer Graphics and CAD*, pages 51–68. Springer, 1988.

A. R. Sanderson, M. Kirby, C. R. Johnson, and L. Yang. Advanced Reaction-Diffusion Models for Texture Synthesis. *Journal of Graphics Tools*, 11(3):47–71, 2006.

P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):443–456, July 2005. ISSN 1077-2626.

G. Scheuermann, J. Frey, H. Hagen, B. Hamann, B. Jeremic, and K. I. Joy. Visualization of Seismic Soils Structure Interaction Simulations. In *Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing (VIIP 2001)*, pages 78–83, Marbella, Spain, 2001. ACTA Press.

T. Schlömer and O. Deussen. Accurate Spectral Analysis of Two-Dimensional Point Sets. *Journal of Graphics, GPU, and Game Tools*, 15(3):152–160, 2011.

T. Schultz and G. Kindlmann. Superquadric Glyphs for Symmetric Second-Order Tensors. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1595–1604, 2010.

C. D. Shaw, J. A. Hall, C. Blahut, D. S. Ebert, and D. A. Roberts. Using Shape to Visualize Multivariate Data. In *Proceedings of the Workshop on New Paradigms in Information Visualization and Manipulation (NPIVM '99)*, pages 17–20, New York, USA, 1999. ACM.

H.-W. Shen and A. Pang. Anisotropy-Based Seeding for Hyperstreamlines. In *Proceedings of IASTED Computer Graphics and Imaging (CGIM)*, Kauai, Hawaii, USA, 2004.

K. Shimada and D. C. Gossard. Bubble Mesh: Automated Triangular Meshing of Non-Manifold Geometry by Sphere Packing. In *SMA '95: Proceedings of the Third Symposium on Solid Modeling and Applications*, 1995.

K. Shimada, A. Yamada, and T. Itoh. Anisotropic Triangular Meshing of Parametric Surfaces via Close Packing of Ellipsoidal Bubbles. In *In 6th International Meshing Roundtable*, pages 375–390, Park City, Utah, USA, 1996. IMR Proceedings.

K. Shimada, A. Yamada, and T. Itoh. Anisotropic Triangulation of Parametric Surfaces via Close Packing of Ellipsoids. *Int. J. Comput. Geometry Appl.*, 10:417–440, 2000.

A. Sigfridsson, T. Ebbers, E. Heiberg, and L. Wigström. Tensor field visualisation using adaptive filtering of noise fields combined with glyph rendering. In *VIS '02: Proceedings of the conference on Visualization '02*, pages 371–378. IEEE Computer Society, 2002. ISBN 0-7803-7498-3.

C. Sigg, T. Weyrich, M. Botsch, and M. Gross. GPU-Based Ray-Casting of Quadratic Surfaces. In *IEEE/EG Symposium on Volume and Point-Based Graphics*, pages 59–66, Boston, USA, 2006.

X. Sixing, H. Jingxin, and X. Xiaozhu. An algorithm of mean-shift template update based on mixture gaussian model. In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Volume 02*, pages 1040–1044. IEEE Computer Society, 2008.

S. W. Sloan. A Fast Algorithm for Constructing Delaunay Triangulations in the Plane. *Advances in Engineering Software*, 9:35–55, 1987.

J. Sreevalsan–Nair, C. Auer, B. Hamann, and I. Hotz. Eigenvector-based Interpolation and Segmentation of 2D Tensor Fields. In *Topological Methods in Visualization (TopoInVis'09)*, Mathematics and Visualization. Springer, Berlin Heidelberg, 2010.

D. Stalling, M. Westerhoff, and H.-C. Hege. Amira: a Highly Interactive System for Visual Data Analysis. *The Visualization Handbook*, pages 749–767, 2005.

X. Tricoche, G. Scheuermann, H. Hagen, and S. Clauss. Vector and Tensor Field Topology Simplification on Irregular Grids. In *Proceedings of the Symposium on Data Visualization (VisSym '01)*, pages 107–116, Vienna, Austria, 2001. Springer.

X. Tricoche, G. L. Kindlmann, and C.-F. Westin. Invariant Crease Lines for Topological and Structural Analysis of Tensor Fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1627–1634, 2008.

V. Verma, D. Kao, and A. Pang. A Flow-Guided Streamline Seeding Strategy. In *Proceedings of the Conference on Visualization (Vis'00)*, pages 163–170, Los Alamitos, USA, 2000. IEEE Computer Society.

A. Vilanova, S. Zhang, G. Kindlmann, and D. H. Laidlaw. An Introduction to Visualization of Diffusion Tensor Imaging and its Applications. In *Visualization and Image Processing of Tensor Fields*, pages 121–153. Springer, Berlin, Heidelberg, 2005.

Z. Wang. *Diffusion Tensor Field Restoration and Segmentation.* PhD thesis, University of Florida, 2004.

Z. Wang and B. C. Vemuri. An Affine Invariant Tensor Dissimilarity Measure and its Applications to Tensor-valued Image Segmentation. In *Proceedings of Computer Vision and Pattern Recognition*, Washington, USA, 2004. IEEE Computer Society.

J. Weickert and H. Hagen, editors. *Visualization and Processing of Tensor Fields.* Mathematics and Visualization. Springer, Berlin, Heidelberg, 2005.

D. Weinstein, G. Kindlmann, and E. Lundberg. Tensorlines: Advection-Diffusion based Propagation through Diffusion Tensor Fields. In *Proceedings of the Conference on Visualization (Vis'99)*, pages 249–253, Los Alamitos, USA, 1999. IEEE Computer Society.

Y. Weldeselassie and G. Hamarneh. DT-MRI Segmentation Using Graph Cuts. In *SPIE Medical Imaging*, pages 1–9, San Diego, USA, 2007.

C.-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz. Geometrical Diffusion Measures for MRI from Tensor Basis Analysis. In *Proceedings of the International Society for Magnetic Resonance Medicine (ISMRM)*, page 1742, Vancouver, Canada, 1997.

A. Wilson and R. Brannon. Exploring 2D Tensor Fields Using Stress Nets. In *Proceedings of the Conference on Visualization (Vis'05)*, Minneapolis, USA, 2005.

Y.-C. Wu, A. S. Field, M. K. Chung, B. Badie, and A. L. Alex. Quantitative analysis of diffusion tensor orientation: Theoretical framework. In *Magnetic Resonance in Medicine*, pages 1146–1155, 2004.

Z. Yuan, G. Rong, X. Guo, and W. Wang. Generalized Voronoi Diagram Computation on GPU. In *Proceedings of the 2011 Eighth International Symposium on Voronoi Diagrams in Science*, pages 75–82, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4483-0.

E. Zhang, J. Hays, and G. Turk. Interactive Tensor Field Design and Visualization on Surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 13(1): 94–107, 2007.

E. Zhang, H. Yeh, Z. Lin, and R. Lamaree. Asymmetric Tensor Analysis for Flow Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15: 106–122, 2008.

S. Zhang, C. Demiralp, and D. H. Laidlaw. Visualizing Diffusion Tensor MR Images Using Streamtubes and Streamsurfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(4):454–462, 2003.

X. Zheng and A. Pang. HyperLIC. In *Proceedings of the Conference on Visualization (Vis'03)*, pages 249–256, Washington, USA, 2003. IEEE Computer Society.

X. Zheng and A. Pang. Topological Lines in 3D Tensor Fields. In *Proceedings of the Conference on Visualization (Vis'04)*. IEEE Computer Society, 2004.

X. Zheng and A. Pang. 2D Asymmetric Tensor Analysis. In *Proceedings of the Conference on Visualization (Vis'05)*, pages 3–10, 2005.

X. Zheng, B. N. Parlett, and A. Pang. Topological Lines in 3D Tensor Fields and Discriminant Hessian Factorization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):395–407, 2005.

X. Zheng, X. Tricoche, and A. Pang. Degenerate 3D Tensors. In *Visualization and Processing of Tensor Fields, Mathematics + Visualization*, pages 241–256. Springer, Berlin, Heidelberg, 2006.

L. Zhukov, K. Museth, D. Breen, R. Whitaker, and A. Barr. Level Set Modeling and Segmentation of DT-MRI Brain Data. *Journal Electronic Imaging*, 12(1):125–133, 2003.

U. Ziyan, D. Tuch, and C.-F. Westin. Segmentation of Thalamic Nuclei from DTI using Spectral Clustering. In *MICCAI'06*, pages 807–814, Berlin, Heidelberg, 2006. Springer.

Declaration

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, 2013

Andrea Kratz