TIM STEFAN GERRITS

VISUALIZATION OF SECOND-ORDER TENSOR DATA AND VECTOR FIELD ENSEMBLES

DISSERTATION

Visualization of Second-Order Tensor Data and Vector Field Ensembles

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik der
Otto-von-Guericke-Universität Magdeburg

von Tim Stefan Gerrits
geb. am 10.12.1990 in Waiblingen

Gutachterinnen/Gutachter

Prof. Dr.-Ing. Holger Theisel
Prof. Dr. Ingrid Hotz
Prof. Dr.-Ing. Thomas Schultz

Magdeburg, den 30. April 2021

ABSTRACT

In scientific disciplines such as mechanics, medical imaging, or fluid dynamics, simulations and measurements are used to gain an understanding of real-world phenomena. Working with and analyzing such data allows scientists to derive better insights, improve current models that try to describe real-world processes, or make better predictions about future events. Increasing computing power as well as advances in algorithms and measuring techniques allow for larger amounts as well as for more accurate data. Additionally, to make data more reliable and trustworthy, the uncertainty of the system or the measurements can also be recorded, and the information made available. This, however, also makes it harder to process, structure, and understand the resulting information. Visualizing high-dimensional, uncertain, or multivariate data has therefore grown to be a major challenge in the field of scientific visualization.

The fields of our contributions are twofold:
First, we introduce novel visualization techniques for second-order tensors. These mathematical objects are used to describe physical quantities in a variety of applications, from diffusion tensor imaging (DTI) to describing derivatives of vector fields. Glyphs have proven to be a valuable visualization for domain experts. While most of the known glyph techniques are limited to symmetric second-order tensors only, we introduce a new glyph design capable of encoding any given 2D or 3D second-order tensor following a set of design principles. We further extend the new construction which then allows the glyphs to represent Jacobian matrices of unsteady vector fields. The final contribution in this field is a novel extension for a variety of tensor glyphs to represent uncertain symmetric second-order tensors.

Secondly, we deal with vector field ensembles. A single vector field often describes the movement of liquids like wind or water. A vector field ensemble is a collection of such fields over the same domain that might either be the results of simulations with varying parameters or a set of measurements of the same phenomenon. As such, ensembles provide a way to have uncertainty represented with a collection of different representations of the same experiment. Domain experts need appropriate visualizations for finding trends, differences, and similarities within the ensemble members. Side-by-side comparisons prove unsuitable for that task when the number of fields is too high. A better strategy is to find features that are able to represent or describe the whole ensemble. We, therefore, introduce a new operator, called the Approximate Parallel Vectors Operator. It finds all locations within an ensemble vector field,

where the vectors of all ensemble members are approximately parallel and thus very similar to each other.

All approaches and visualization techniques are applied to synthetic and real-world data sets and therefore provide a number of novel visualization tools for the investigation of scientific data.

ZUSAMMENFASSUNG

Simulationen und Messungen helfen in vielen wissenschaftlichen Disziplinen wie der Mechanik, der Medizinischen Bildgebung oder der Strömungsmechanik, ein besseres Verständnis für auftretende Phänomene zu erhalten. Die Arbeit mit und die Analyse solcher Daten erlaubt es Wissenschaftler*innen, bessere Einblicke zu erhalten, derzeit gängige Modelle zu verbessern, die versuchen reelle Prozesse zu beschreiben oder bessere Voraussagen über die Zukunft zu treffen. Der Anstieg der Rechenleistung sowie die Fortschritte bei Algorithmen und Messtechniken sorgen für mehr Datenmengen und genauere Daten. Zusätzlich können die Unsicherheiten von Systemen oder Messungen aufgezeichnet und bereitgestellt werden, um Daten zu erlangen, die vertrauenswürdiger und zuverlässiger sind. Dies sorgt allerdings dafür, dass das Verarbeiten, Strukturieren und Verstehen der gesammelten Informationen komplizierter wird. Daher ist das Visualisieren von hochdimensionalen, unsicheren und multivariaten Daten zu einer massiven Herausforderung im Bereich der wissenschaftlichen Visualisierung geworden.

Die Ansätze, die hier präsentiert werden können in zwei Bereiche aufgeteilt werden:
Zuerst werden neue Techniken zur Visualisierung von Tensoren zweiter Ordnung präsentiert. Diese mathematischen Objekte werden dazu genutzt, physikalische Größen in verschiedensten Bereichen zu beschreiben. Dies reicht von der Diffusionstensorbildgebung (DTI) bis hin zur Ableitung von Vektorfeldern. Dabei haben sich Glyphen als wertvolles Visualisierungswerkzeug für Fachexpert*innen erwiesen. Während die meisten bekannten Konstruktionen von Glyphen allerdings auf den symmetrischen Fall von Tensoren zweiter Ordnung limitiert sind, stellen wir eine neue Konstruktion vor, welche einer Sammlung an Designgrundsätzen folgt und es ermöglicht, jeglichen 2D oder 3D Tensor zweiter Ordnung zu repräsentieren. Außerdem erweitern wir diese Konstruktion, sodass es ebenfalls möglich wird, Jacobimatrizen von zeitabhängigen Vektorfeldern darzustellen. Der letzte Beitrag in diesem Bereich ist eine neuartige Erweiterung für eine Vielzahl von Tensorglyphen, um ebenfalls unsichere symmetrische Tensoren zweiter Ordnung darzustellen.
Der zweite Teil setzt sich mit Ensemblen von Vektofeldern auseinander. Ein einzelnes Vektorfeld beschreibt oftmals die Bewegung von Flüssigkeiten wie Wind oder Wasser. Ein Vektorfeldensemble ist eine Sammlung solcher Felder, die denselben Bereich beschreiben und beispielsweise entweder das Ergebnis von Simulationen mit sich ändernden Parametern, oder eine Sammlung an Messungen desselben Phänomens darstellt. Es ist daher eine Möglichkeit, Unsicherheit durch eine Sammlung an verschiedenen Realisierungen desselben Experiments zu repräsentieren.

Fachexpert*innen benötigen passende Visualisierungen, um Tendenzen, Unterschiede und Ähnlichkeiten innerhalb der Ensemblemitglieder zu finden. Wenn die Zahl der Felder zu hoch ist, stellen sich direkte Gegenüberstellungen für diese Aufgaben als unpassend heraus. Eine bessere Strategie stellt das Finden von Merkmalen dar, die das gesamte Ensemble beschreiben oder repräsentieren können. Wir stellen daher einen neuen Operator vor, den wir den Approximate Parallel Vectors Operator nennen. Dieser findet alle Orte innerhalb eines Ensemblevektorfeldes, an denen die Vektoren aller Ensmemblemitglieder annähernd parallel und somit sehr ähnlich zueinander sind.

Alle Ansätze und Visualisierungstechniken werden auf künstlich erstellten sowie echtweltlichen Datensätzen angewandt und stellen somit eine Reihe von neuartigen Visualisierungswerkzeugen für die Untersuchung von wissenschaftlichen Daten dar.

CONTENTS

Part I

PRELIMINARIES

1 INTRODUCTION

Scientific computing aims to model, simulate, and analyze phenomena to help understand the world around and within us. By means of simulations or measurements, domain experts produce large quantities of numerical data in a variety of disciplines. This includes data such as *Computational Fluid Dynamic* (CFD) simulations, that are used to model complex processes like combustion or fluid motion, as well as data obtained from *Diffusion Tensor Magnetic Resonance Imaging* (DT-MRI) representing brain tissue structures. The resulting data is manifold not only in its meaning but can also be high in its dimensionality. Understanding the acquired data is crucial to verify assumptions, derive new information, or just explore a given set of data.
Scientific visualization aims to support these processes by transforming abstract data into simpler representations, such as images or image sequences, which then allow for faster and better understanding. This utilizes the fact, that humans strongly rely on their visual perception to interpret and perceive the world around them. From everyday applications such as maps indicating weather forecast data, to specific and complex analysis tools to survey a patient's blood flow: visualization can help to drastically reduce the cognitive effort needed to comprehensively study data.

Due to the ever-increasing computational power, growing available storage as well as improving methods, not only the amount of data that is produced and needs to be stored has raised significantly. Additionally, several mappings into spaces of different dimensions, such as scalar, vector, or tensor data, describing different aspects of a simulation or measurement can be extracted at the same time.
In weather simulations, for instance, pressure or temperature are often recorded as scalar fields, movement of particles or fluids, on the other hand, can be described by vectorial data. Second-order tensors are commonly used to represent quantities that describe even more complex phenomena like mechanical stresses or diffusion processes such as those described by the diffusion tensor within tissue. Further, the quality of each of these data types can be evaluated in terms of how reliable a given quantity is. Especially in applications where the consequences of a wrong conclusion are connected with high risks, such as medical applications, it is crucial that experts can assess the uncertainty of the data. This has motivated scientists and engineers alike to collect and save even more and higher-dimensional data. It seems only logical, that

collecting all this information allows for a better and deeper understanding of the overall processes. It may also lead to new knowledge about correlations within the different quantities that were not clear before. There is, however, one severe drawback that arises: processing all this data mentally, connecting its meaning, and making sense of it becomes increasingly difficult. This directly applies to scientific visualization as well: present visualization techniques are often not able to cope with the high amount of data, either in terms of their data management and efficiency or in the resulting images, that might be too complex or cluttered to serve its original purpose. Metaphors used before to communicate core insights, might not be applicable anymore. Adapting and improving known approaches has the advantage, that users presented with these visualizations might be able to understand and use them with ease. For some types of data, however, there do not even exist any known visualization approaches. Each data type has its own set of difficulties and possibilities that each span research fields on their own.

Providing simple yet meaningful visualizations in the face of these challenges has led to numerous research contributions over the last few years. Within these, especially visualization of second-order tensors has gained a lot of momentum. These mathematical objects are a standard choice to represent complex physical quantities such as the diffusion within the human brain matter or the curvature of a smooth geometric surface. This does, however, also mean that interpreting and working with such tensor data is also complex and generally strongly application dependent. That is why new mathematical concepts and frameworks were developed over the years that allow for easier handling, processing, and understanding of tensor objects. By decomposing tensors or deriving tensor properties, these can be mapped to scalar or vectorial data that is easier to understand and can be visualized with known techniques to allow domain experts to gain insights into such fields. Specific tensor properties can be used to classify tensors to then find structures in tensor fields such as topological features. This does however omit parts of the data encoded by the tensors. For deeper insight and analysis, a complete and direct encoding of such data is often desirable. One possible way to do this is by the use of glyphs: the depiction of tensor data at sampled locations by means of simple and comprehensible geometric primitives has become a well-known and widely accepted technique, known as *tensor glyph visualization*. Mapping multiple tensor values to different aspects like size, shape, orientation, and color, provides a powerful tool for investigating the underlying data. Most contributions have however mostly been focused on restricted cases, demanding specific tensor properties such as symmetries or specific definiteness. This does exclude data from various applications such as computational fluid dynamics, where tensors are often found to be non-symmetric. Further, tensor data is often treated as “certain”. Measured or simulated tensor fields, just as most data, often incorporate uncertainties that might be of importance for domain experts especially in the context of medical applications. Treating the data as uncertain increases the complexity of these already high-dimensional objects even further, which most existing glyph constructions cannot incorporate.

Similar to second-order tensor fields, dealing with uncertainty in vector fields has also received a lot of attention in recent years. Powerful models exist that are used to describe meteorological phenomena like wind movement or behavior of flow like blood or water. They can also be used to predict such events by running numerical simulations. As the complexity of involved and intercorrelated physical influences and laws can only be modeled up to a certain degree, such models can only provide results up to a limited accuracy. To account for this, uncertainty can be estimated using not only one result. Multiple measurements can be taken, or simulations are run several times with varying simulation parameters or alternative models leading to a collection of data known as an *ensemble*. Organizations that provide highly accurate weather predictions, such as the European Centre for Medium-Range Weather Forecasts (ECMWF), base their predictions on such ensemble forecasts. When dealing with ensemble vector fields, each ensemble member is a vector field defined on the same spatial and temporal domain. Here, too, several works have been proposed over the last years that deal with *ensemble visualization*. Often, known techniques from vector field visualization are applied to each member independently leading to a collection of images. If the number of members is small enough, they can simply be but in juxtaposition. For large ensembles, this might, however, be inefficient or lead to cluttering, making it difficult to explore the data. To get a better understanding of the entire ensemble, statistical measures such as mean or standard deviation can be derived and rendered in a single image. This might however lead to a loss of important features and structures within the ensemble members.

This thesis is the result of the effort to analyze and tackle the problems mentioned above. This includes reviewing current state-of-the-art visualization approaches and further the development of new techniques for the visual analysis of high-dimensional and uncertain data. We do not limit the research on certain applications but want to provide general ideas that can be applied to a variety of problems. We, therefore, focus on glyphs for second-order tensors as well as vector field ensembles and not only wish to extend the list of tools available to domain experts to explore, analyze and visualize such data, but further hope that this research contributes to a better understanding of the data itself as well as the development of suitable visualization techniques in the future.

The thesis is structured as follows:
Chapter 2 serves as a general introduction to scientific visualization and offers background to key concepts of visualization and their mathematical properties used throughout this thesis. As the fields of research contributions in this dissertation are twofold, Part ii introduces novel glyph-based visualization techniques for second-order tensor data.

- First, we give suitable definitions of tensors and list not only important tensor properties but also give several applications in which such data is used and how it can be visualized.
- We then take a closer look at tensor glyphs as means of tensor field visualization and discuss existing glyph techniques.
- We introduce and justify a set of strict design principles in Chapter 5, which we set as a basis for further glyph design. Building upon this list, we present a new glyph that is capable of representing any given 2D or 3D second-order tensor without further requirements on tensor properties like symmetry or definiteness.
- Based upon this work, Chapter 6 shows, how the glyphs can further be extended to also include time-dependent 2D or 3D Jacobian matrices, which form a special case of second-order tensors.
- Finally, in Chapter 7 we propose a novel technique that allows different classes of symmetric tensor glyphs to be extended with an offset surface that represents uncertainty given by a covariance matrix.

Glyphs are computed for a variety of data sets, including measured diffusion tensor data, simulated stress tensors, and the derivative of flow simulations.

In Part iii, we investigate the visualization of vector field ensembles.

- First, we introduce vector field ensemble data and challenges in its visualization. We especially focus on the extraction and visualization of line-type features such as vortex core lines.
- We introduce a new operator in Chapter 10, called the *Approximate Parallel Vectors Operator* which introduces new line-type features that are able to represent or describe the ensemble data as a whole instead of looking at each ensemble member separately. It is therefore a generalization of the well-known parallel vectors operator. This, too, is applied to a variety of datasets.

The thesis is concluded by a summary as well as a section on future work in Part iv.

1.1 LIST OF PUBLICATIONS

The following articles have been published in peer-reviewed international conferences as a result of this thesis:

- T. Gerrits, C. Rössl, and H. Theisel
  Glyphs for General Second-Order 2D and 3D Tensors
  IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Scientific Visualization 2016), 2017

- T. Gerrits, C. Rössl, and H. Theisel
  Glyphs for Space-Time Jacobians of Time-Dependent Vector Fields
  Journal of WSCG, 2017

- T. Gerrits, C. Rössl, and H. Theisel
  An Approximate Parallel Vectors Operator for Multiple Vector Fields
  Computer Graphics Forum (Proc. EuroVis), 2018

- T. Oster, A. Abdelsamie, M. Motejat, T. Gerrits, C. Rössl, D. Thévenin and H. Theisel
  On-The-Fly Tracking of Flame Surfaces for the Visual Analysis of Combustion Processes
  Computer Graphics Forum, 2018

- T. Gerrits, C. Rössl, and H. Theisel
  Towards Glyphs for Uncertain Symmetric Second-Order Tensors
  Computer Graphics Forum (Proc. EuroVis), 2019

2 SCIENTIFIC VISUALIZATION

The term *visualization* refers to a multitude of techniques and a common definition does not exist [24]. All of them do, however, share a similar and simple goal: efficient communication of messages that can be extracted from the given data by the use of images. Scientific visualization does indeed share this goal, but deals, as the name indicates, with scientific data. Whereas the term was first coined by McCormick et al. [119] more than thirty years ago, the research areas it covers as well as the challenges arising from new technological developments have grown steadily over the years [62], [178], [206]. While implementations may vary greatly, depending on the application, data type, and visualization goal, they can be summarized by the following pipeline [59]:
First, scientific data, either taken from simulations or measured in real-world situations is analyzed, followed by filtering, such that only relevant data is used. Then, the data is mapped to a geometric model which is then rendered to an image. Especially filtering and mapping are the processes that allow for the most control over the expressiveness and efficiency of the visualization. Depending on the data type and application, it might be of interest to either try and find *features* within a given set of data, that are able to describe a certain behavior of the underlying phenomenon or using a *direct* visualization method to explore what might be hidden within the data. We give a formal introduction to the most relevant data types appearing in this thesis. To serve as a background for the following parts, we further introduce several visualization concepts or techniques relevant to these types.

2.1 DATA TYPES AND VISUALIZATION

When considering data from scientific research, especially in an engineering context, we often have to deal with sampled continuous data defined in a spatial context. This means that measurements or simulations collect information for distinct locations in space. Associating each location with a piece of data is known as fields. Formally, a field is given by a function

$$f : \mathrm{D} \to \mathrm{C} \tag{2.1}$$

over the domain $\mathrm{D} \subset \mathbb{R}^n$ which maps each location in an n-dimensional space to a value in $\mathrm{C} \subset \mathbb{R}^c$. These are thus c-dimensional values, which characterize the type of data, the visualization has to deal with. When, for instance, the temperature within a region is measured as a scalar value ($c = 1$) at each location, it can be described as a scalar field. The same applies to vectors and tensors, resulting in vector fields and tensor fields, respectively. In the following, scalar and vector fields are introduced as well as common visualization techniques. Further, second-order tensors as a special case of tensors are presented with several examples to illustrate their usefulness. A more detailed discussion of tensor concepts and their properties will be given in Section 3.1. Finally, the general concepts of uncertainty and ensembles are introduced.

2.1.1 *Notation*

Unless stated differently, the notation throughout this work sticks to the following convention:
Scalar fields and scalar-valued functions are denoted by lower-case letters, e. g. s and $s(\mathbf{x}),\ \mathbf{x} \in \mathbb{R}^n$, whereas bold lower-case letters indicate points, vectors, and vector-valued functions, e. g. $\mathbf{v}$ and $\mathbf{v}(\mathbf{x})$.

A set is written using upper-case letters, e. g. S, while matrices, second-order tensors as well as tensor-valued functions are depicted as bold upper-case letters, e. g. $\mathbf{T}$ and $\mathbf{T}(\mathbf{x})$.

In the context of this work, $\mathbf{v}(\mathbf{x})$ represents steady or time-independent vector fields, while $\mathbf{v}(\mathbf{x}, t)$ represents unsteady or time-dependent vector fields. In a similar fashion, symbols accented with a tilde denote data including temporal information, e. g. $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{T}}$. Symbols accented with a bar denote a mean value, e. g. $\bar{\mathbf{v}}$ and $\bar{\mathbf{T}}$ while $\overline{z}$ is the complex conjugate of a complex number z.

Further, some common operations will reappear throughout the thesis. Using the transpose of a real vector or a real matrix is written as $\mathbf{v}^{\mathrm{T}}$ and $\mathbf{T}^{\mathrm{T}}$, the Hermitian transpose as $\mathbf{T}^{*}$ and multiplication by a scalar s as $s \cdot \mathbf{v}$ or $s \cdot \mathbf{T}$. Ofte, we omit the operator and write $s\,\mathbf{T}$. The total derivative of a function f with respect to an argument x is given by df/dx and the partial derivative by $\partial f/\partial x$. We also use the shorthand notation f_x. The inverse of a matrix is noted as $\mathbf{T}^{-1}$. While a Cartesian product is denoted by the symbol $\times$, the tensor product is indicated by $\otimes$.

Common symbols to appear are the nabla operator ∇, especially in the context of the vector field gradient as $\mathbf{T} = \nabla \mathbf{v}$. It is defined as a vector containing the partial derivative symbols with respect to the given spatial dimensions $\nabla = (\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_n})$. $\mathbf{0}$ is the zero matrix and $\mathbf{I}$ the Identity matrix.
Decomposing matrices into eigenvalues and eigenvectors or singular values and vectors is explained in greater detail in Section 3.1.1. Eigen-

values and singular values of a matrix $\mathbf{T}$ are named $\lambda_i(\mathbf{T})$ and $\sigma_i(\mathbf{T})$ while i indicates the i^{th} value. We assume eigenvalue subscripts to be applied according to their order, such that $\lambda_1(\mathbf{T}) \geq \cdots \geq \lambda_n(\mathbf{T})$ for n eigenvalues. The eigenvector $\mathbf{e}_i(\mathbf{T})$ and the associated i^{th} eigenvalue share the same index.

Whenever the context is clear, we omit the explicit reference to the matrix and write, e. g., λ_i instead of $\lambda_i(\mathbf{T})$, which also applies to derived matrix values such as γ mentioned in Section 3.2.4 instead of $\gamma(\mathbf{T})$.

2.1.2 *Scalars and Scalar Fields*

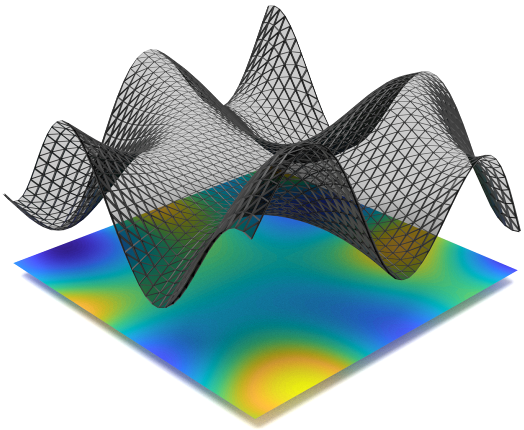


Figure 2.1: A scalar field on a 2D domain visualized with a wireframe to represent the scalar value as height and an additional projection of the scalar value to a color map.

In scalar fields, a scalar value $s(\mathbf{x}, t)$ is assigned to each position $\mathbf{x} \in \mathbb{R}^n$ in an n-dimensional domain, where t denotes the time. When the scalar field is steady, the parameter t can be omitted, resulting in $s(\mathbf{x})$. The value itself fully characterizes the quantity.
Due to their low dimensionality, scalar fields can be found in numerous applications and can be considered part of a humans' everyday life. Temperature and pressure are typical results from meteorological measurements or simulations. Computer Tomography (CT) and Magnetic Resonance Imaging (MRI) both provide scalar values that allow for the analysis of tissue properties. Scalars, however, can also be used to describe derived properties of higher-order data types like vectors or tensors.

Bars, charts, and line graphs are basic tools to visualize scalar values in a one-dimensional space. Even users with little experience can communicate messages as such tools are included in standard office software and available for free. Techniques that visualize the scalar value at each position of the domain directly are known as *image-based methods*. In a

two-dimensional domain, scalars can also be mapped to the intensity values of a grey value image or color channels, which allows for a simple and efficient visualization. This *color mapping* produces heatmaps, which are, for instance, used in weather forecasts, where scalar values representing temperature are mapped to color values as seen in Figure 2.1. Alternatively, and also shown in Figure 2.1, *height mapping* can be used to represent the scalar value as an actual height value of a 3D surface. Scalar fields in 3D domains pose to be more challenging to visualize due to visual cluttering. Color and opacity values have to be chosen carefully resulting in suitable transfer functions that reveal only the structures a user is interested in. This is known as *volume rendering* and still contributes to a lot of scientific publications today. Browsing through the volume with a slice view, while each slice shows only a two-dimensional subset is a well-known technique in medical applications to deal with this issue.
Alternatively, instead of visualizing the scalar values directly, one might try to represent a whole field by only showing meaningful features. Such approaches are known as *geometry-based methods*. Showing local and global extrema by extracting critical points or ridge and valley lines helps to understand the overall structure of a scalar field, without observing every single data value. Contour plots or isolines and isosurfaces allow to find structures, that share the same value and can be found in several applications from the well-known height lines in geographic maps to the tracking of flame surfaces in combustion data.

2.1.3 *Vectors and Vector Fields*

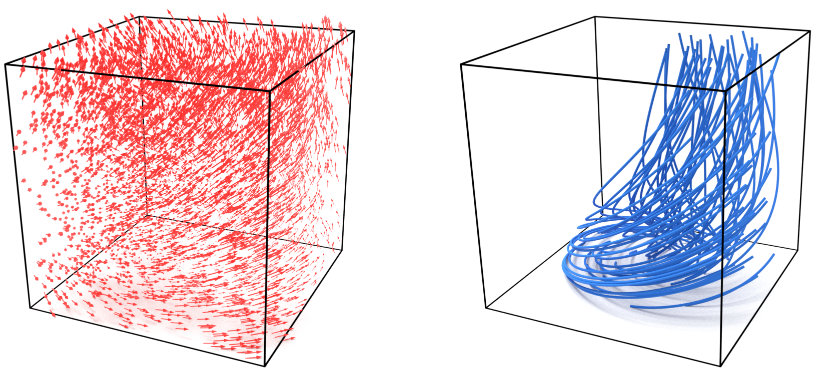


Figure 2.2: A 3D rotational flow visualized with arrow glyphs at uniformly sampled locations (left) and streamlines (right).

In a similar manner, vector fields are fields that assign a c-dimensional vector value $\mathbf{v}(\mathbf{x}, t)$ or $\mathbf{v}(\mathbf{x})$ where $\mathbf{v}(\mathbf{x}, t) = (v_1(\mathbf{x}, t), \ldots, v_c(\mathbf{x}, t))^{\mathrm{T}}$ or $\mathbf{v} = (v_1(\mathbf{x}), \ldots, v_c(\mathbf{x}))^{\mathrm{T}}$ respectively, to each position $\mathbf{x} \in \mathbb{R}^n$ in an n-dimensional domain. One way to interpret this is that vectors represent not only a magnitude but also a direction. They can encode a location in space but also physical quantities like force, direction, velocity, or gradients of scalar fields. When dealing with velocity in a 3D Cartesian coordinate system, the vector components are often given as $\mathbf{v} = (u(\mathbf{x}), v(\mathbf{x}), w(\mathbf{x}))^{\mathrm{T}}$, which represent the velocity in direction of the coordinate axes. A vector field that changes over time is called *unsteady* or *time-dependent* vector field, whereas one that does not change is considered *steady* or *time-independent*. Especially in computational fluid dynamics applications, vectors are used to represent the movement of matter or particles within what is often called a flow field. Improving methods of predicting the flow of liquids like water, gases, or even blood and analyzing its behavior is still an ongoing research topic and has huge implications in our everyday life from predicting weather to building more aerodynamic transportation vehicles.

The scientific discipline concerned with the visualization of such vector or flow fields is aptly called flow visualization or *flowvis* for short. The image-based techniques from scalar fields can be used to visualize the components of vector fields or derived quantities such as the magnitude. In 2D and 3D, simple arrow glyphs can be used to directly represent whole vectors at a given location resulting in so-called arrow plots as shown in Figure 2.2 (left). Yet again, visual cluttering poses a challenge, especially in 3D, which is why several alternatives have been proposed over the years. A well-known texture-based visualization is given by what is called a Line Integral Convolution (LIC) visualization, where the single color values of a noise texture are advected within the vector field and form lines that indicate the flow. This technique was intended to work for steady 2D flows or on surfaces but has been extended to deal with unsteady and 3D vector fields as well. Vector field features are often using metaphors related to flow applications. Many features are therefore based on the movement of particles within a flow and aim to follow the trajectory of such particles over time. Integral lines such as streamlines, pathlines, streaklines, or timelines indicate such paths and are a powerful visualization tool for analyzing the global behavior of the flow as indicated in Figure 2.2 (right). They all describe structures that are generated by starting an integration process along the vectors of a vector field from a seeding location. A detailed description is given in Section 9.1. Similar concepts can be applied to start integration for lines or even surfaces.
Analogous to scalar fields, vector fields can also be represented by only considering their structure by extracting characteristic points or features and visualizing them. This is known as *vector field topology*. Locations where the flow vanishes, i. e., $\mathbf{v} = \mathbf{0}$, are known as critical points and separatrices are lines that connect such points with each other segmenting the field into areas with different flow behavior. The local description of the behavior of the flow can be encoded in a second-order tensor known as the Jacobian matrix. Decomposing a Jacobian matrix

into its eigenvectors and corresponding eigenvalues allows to characterize critical points in flows depending on the behavior of particles in its vicinity.
Locating regions of swirling motion within fluids is a major objective in several applications. It can define the quality of combustion processes by indicating, how well gases are mixed. Rotating air is also known to produce drag which makes planes more inefficient. Therefore, extracting vortices as flow features can help improve designs and increase efficiency. This is why defining, finding, extracting, and visualizing vortices has grown to be a major challenge in flow visualization. The background section of Part iii is dedicated to discussing these challenges.

2.1.4 *Tensors and Tensor Fields*

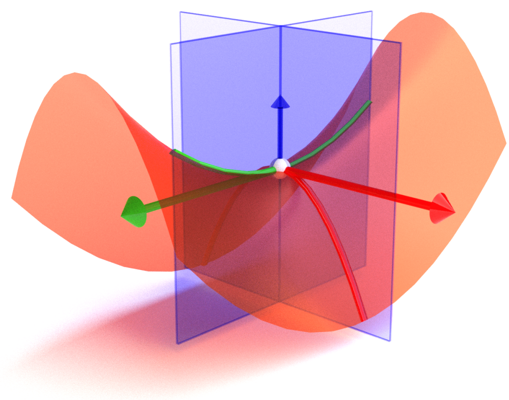


Figure 2.3: Local curvature at a given point of a smooth surface describes how the surface normal changes when moving away from the current location into a given tangential direction. This information can be completely described by a Hessian or curvature tensor. The eigenvectors of the tensor correspond to the directions of the most (red) and least (green) change of the normal (blue).

As phenomena described by scalar and vector data can be observed in numerous everyday life situations, their motivation and use are often straightforward. The concept of tensors on the other hand is more complex. While we want to introduce a certain aspect of tensors here already, Chapter 3 gives a detailed introduction and general discussion of tensors. In the context of this work, we are dealing with second-order tensors, also known as rank-2 tensors, which can be represented by square matrices. Even though scalars and vectors can also represent tensors, the term *tensor* often implies that we are dealing with tensors of that type. Such a tensor field can therefore be defined as a field that assigns an $m \times m$ matrix $\mathbf{T}(\mathbf{x})$ where

$$\mathbf{T}(\mathbf{x}) = \begin{pmatrix} t_{11}(\mathbf{x}) & \cdots & t_{1m}(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ t_{m1}(\mathbf{x}) & \cdots & t_{mm}(\mathbf{x}) \end{pmatrix}$$

to each position $\mathbf{x} \in \mathbb{R}^n$ in an n-dimensional domain. If the tensor changes over time, it is denoted as $\mathbf{T}(\mathbf{x}, t)$ accordingly. Such a tensor can be used to carry more information than scalar or vector quantities. One example is depicted in Figure 2.3. The change of direction of a normal vector on a surface, also known as curvature, is dependent on the direction moved away from the current location. This information can be fully described by collecting the partial derivatives of second-order of the function describing the surface, which is known as the Hessian matrix and in this specific context can also be called curvature tensor. Other examples are discussed in Section 3.2.

While scalars can be directly mapped to points or color and vectors to arrow glyphs, finding an appropriate and easy to understand visualization for tensors proves to be a challenging task. Sure enough, all matrix components can be visualized independently as independent scalar quantities. This would however ignore the intrinsic meaning and structure that is encoded within the tensor. There exist a number of derived scalar and vectorial quantities, which can then be visualized using the techniques mentioned above. A detailed discussion of visualization methods for tensor data is given in Section 3.3.

2.2 UNCERTAIN AND ENSEMBLE DATA

When analyzing visualizations, we often assume that the data that is represented is accurate and free from uncertainty. In the real world, this is seldom the case. A temperature might vary within a region, the predicted path of a storm is often only the most likely route it may take and diffusion tensor measurements in MRI scans are often highly influenced by noise. The exact outcome of an experiment or value of a measurement is not clear, thus, uncertainty can be understood as the lack of information. We do, however, need information to make decisions or derive further knowledge. Especially in critical applications like medicine or meteorology, where decisions based on the data can affect humans' wellbeing, it is crucial that no wrong conclusions are drawn. Therefore, it is desirable to introduce a notion of how trustworthy the displayed data is or how reliable a shown result actually can be reproduced.
The sources of such uncertainty can be manifold. Models used to simulate phenomena are mostly simplifying abstractions and often provide

an incomplete description. Further, small changes in model parameters or initial conditions might lead to strongly varying results. Measurements can be biased, influenced by noise, or might only provide values up to a certain accuracy. Processing the data, interpolating values and even the rendering process itself can also introduce more uncertainty. Due to the improvement of hardware as well as algorithms, such information on uncertainties can now be included in the measuring or simulation processes. This does, however, lead to an increase in data that needs to be processed and displayed. Including such information in the visualization process is what is known as *uncertainty visualization*.

Mathematically, this can be modeled with probabilities. A probability distribution is used to provide information on how likely it is, that an experiment turns out in a specific way. Instead of assigning a certain value to a location, we now describe, how likely it is, that a specific value occurs. Given a field over the domain $\mathbf{D}$ and the c-dimensional data attributes represented as a collection of continuous random variables $\mathbf{v} = [v_1, \ldots, v_c]^{\mathrm{T}}$, the infinitesimal probability of any value can be described by a *probability density function* (PDF) $\rho(\mathbf{x}; \mathbf{v})$, where

- $\mathbf{x} \in \mathbf{D}$ and $\mathbf{v} \in \mathbb{R}^c$
- $\rho(\mathbf{x}; \mathbf{v}) \geq 0$
- $\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \rho(\mathbf{x}; \mathbf{v})\, dv_1 \ldots dv_c = 1$ for all $\mathbf{x} \in \mathbf{D}$.

A *certain* field, as described in the sections before, is therefore just a special case of an uncertain field, where the probability is exactly 1 for a specific value and 0 for all others. The specific PDF for a sample point is typically not known and is often only approximated. Simpler parametric functions, such as the Gaussian distribution, also known as Normal distribution, which can be described by only a few parameters, can be assumed, which drastically reduces the complexity. If common PDFs do not suffice, finite mixtures of them or more sophisticated models might be used and fitted to the data to better represent the actual distribution. Finding an appropriate model is crucial in making assumptions or conclusions from the data.

Instead of describing uncertainty as a field of distributions, *ensemble visualization* follows the idea, that the uncertainty can be represented by a collection, called an *ensemble*, of possible certain fields. For instance, a weather simulation might be run several times with slightly varying simulation parameters, different models, or initial conditions, resulting in a set of results, called *ensemble members*, showing the phenomenon in the same domain. Analyzing the similarities and differences of the members and their features allows for an understanding of the variability of the data. In this work, we limit all ensemble members to have the same data type. Therefore, an ensemble field with m members assigns

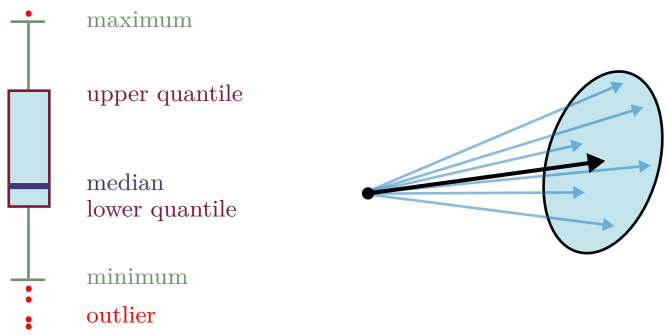


Figure 2.4: Left: 1D boxplot visualization of uncertain unimodal scalar data. Right: 2D uncertain vector represented by a mean vector in black and uncertainty represented by an ellipse that indicates how the vector might deviate from the mean.

m distinct c-dimensional attributes to each position $\mathbf{x} \in \mathbb{R}^n$ in the n-dimensional domain. Dealing with such data and how to find appropriate visualizations is further elaborated in Chapter 9.

As seen in the previous section, each data type comes with its own challenges for a suitable visualization. Adding uncertainty to the process adds more parameters that need to be included and mapped onto free visualization parameters like color or geometry. If suitable, existing techniques might be enhanced or completely new approaches need to be developed, depending on how the uncertainty is characterized. Uncertainties in values, such as temperature or measured diffusion tensors, are treated differently than location uncertainties, such as the location of vortex core lines within a flow or isocontours in scalar fields. A well-known approach to showing uncertain scalar data values is the boxplot, as depicted in Figure 2.4 (left), which is a summary of the distribution by displaying the minimum and maximum values, upper and lower quartiles as well as the median. Numerous extensions, like the range plot, vase plot, or violin plot, as well as evaluations, have been proposed over the years and are still one of the most used tools to compare data sets. When dealing with vector-valued data as directional information, uncertainty may appear within direction and magnitude, as indicated in Figure 2.4 (right). Using glyphs like the uncertainty glyphs by Wittenbrink et al. [205] or texture-based methods allows for local analysis of the uncertainty. Uncertainty visualization can also be introduced to extracted features of vector fields such as integral lines or surfaces, which is further discussed in Section 9.1. The high complexity of second- or higher-order tensor data is even more challenging, as is discussed in Chapter 7.

When dealing with ensembles as an indicator of uncertainty, there exist two major strategies of visualizing the uncertainty. A pipeline

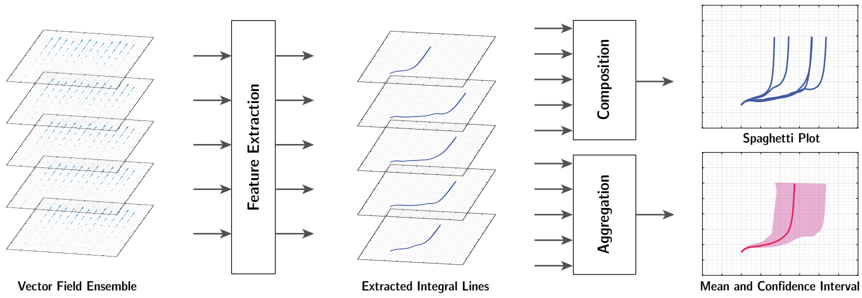


Figure 2.5: Possible vector field ensemble visualization pipeline: A streamline is extracted from each vector field ensemble member resulting in a collection of curves. Different visualization approaches can now be applied. The curves can be composited in a combined view such as the spaghetti plot on the top right. Alternatively, information about the curves such as mean and standard deviation can be aggregated and used for uncertainty visualizations like at the bottom right.

showing this process with an ensemble of vector fields is shown in Figure 2.5. One strategy is to aggregate information of all members and display the results similar to the approaches that were listed above. The bottom right image in Figure 2.5 shows the mean trajectory in red computed from all input trajectories and an uncertainty visualization. The other strategy is to treat the distinct members separately first and combine the results. Features such as streamlines might be extracted and visualized for each field and then put in a juxtaposition to compare. As all members are describing the same domain, another option is to display all the extracted features within a combined visualization such as a spaghetti plot which is shown in the top right image in Figure 2.5. When the number of ensemble members is high, this can lead to a list of problems including visual cluttering, which leads to the demand for more advanced visualization techniques. In Part iii, we focus on vector field ensembles and discuss existing and new visualization techniques for such data.

Part II

VISUALIZATION OF SECOND-ORDER TENSOR DATA

3 INTRODUCTION TO TENSOR DATA

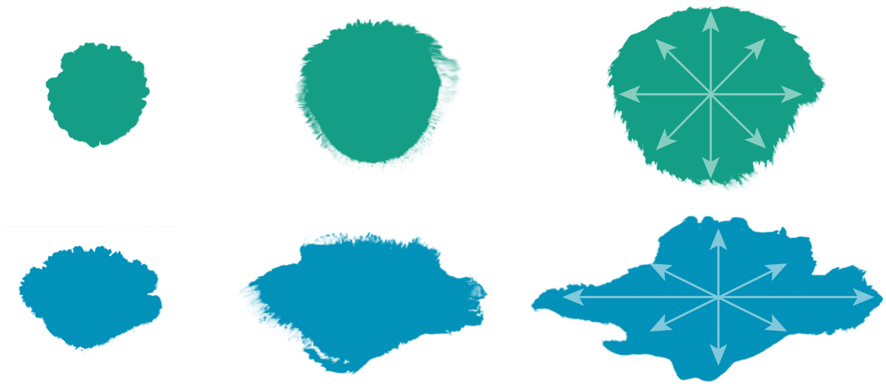


Figure 3.1: When a drop of ink is placed on material such as absorbent paper, the color spreads over time based on the structure of the material. This process is called diffusion and can be captured by a diffusion tensor. This figure shows two different examples where three successive instances in time from left to right are shown. The upper drop of ink diffuses similarly in all directions, which is known as isotropic diffusion, whereas the lower one has a tendency to move stronger into a certain direction, known as anisotropic diffusion.

Second-order tensors can describe more complex aspects of physical phenomena and we briefly introduced the general idea of second-order tensors and tensor fields in Section 2.1.2. As several contributions within this thesis deal with the visualization of such data especially by using so-called tensor glyphs, we discuss tensors in this chapter in greater detail. To do so, we start by offering more detailed descriptions of these objects and their properties in the background Section 3.1. We focus on aspects that are relevant to either the contributions that we introduce in this thesis or to existing related work, with the aim to understand the rationale behind the used techniques. Further, we give some tangible examples for second-order tensors from the scientific and engineering context in Section 3.2 and introduce concepts related to analyzing them. This will help us to further group and understand different existing visualization approaches which are discussed in Section 3.3. A special focus lies on tensor glyphs (Chapter 4) as a powerful direct visualization tool. Listing and explaining related works, we point out existing shortcomings and limitations, that we chose to address within this work.

3.1 BACKGROUND

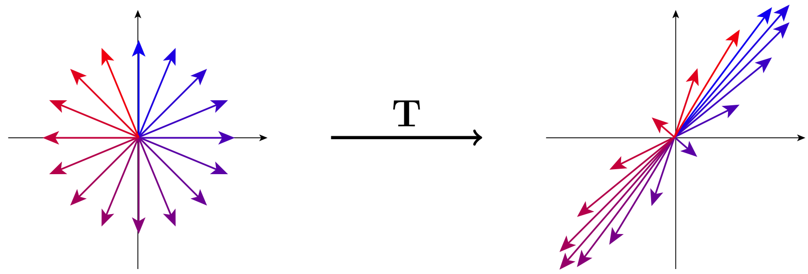


Figure 3.2: Tensors as linear operators: a tensor acts as a linear operator that maps a set of vectors (left) to new vectors (right).

There exist several different definitions of what a tensor is, depending on application or generalization. They do however all describe the same geometric concept and can be transformed into one another. Overly simplified, tensors are mathematical objects that describe quantities independently from their frame of reference, which is a desirable property, especially in physical applications. It takes in a number of input vectors and produces an output that is invariant under a change of basis. Further, its components transform in specific and predictable ways. Thus, one way to define a tensor $\mathbf{T}$ is as a scalar-valued multilinear map

$$\mathbf{T} : \underbrace{V^* \times \cdots \times V^*}_{n} \times \underbrace{V \times \cdots \times V}_{m} \to \mathbb{R}$$

where V is a vector space, V^* its corresponding dual space of covectors and the mapping is linear in all of its arguments. Similarly, a tensor can be described as an element of the tensor product of vector spaces:

$$\mathbf{T} \in \underbrace{V \otimes \cdots \otimes V}_{n} \otimes \underbrace{V^* \otimes \cdots \otimes V^*}_{m}$$

The pair given by the number of vectors n and covectors m defines the type of a tensor, namely an (n, m)-tensor, whereas the sum $n+m$ defines the rank or order of it. In the scope of this work, we are using the Cartesian orthonormal coordinate system, which means that V and V^* behave identically and are interchangeable. This allows a further interpretation, which is especially useful in the context of this work: a tensor of second-order can be seen as a linear operator that maps a vector $\mathbf{v} \in V$ to another vector in the same vector space.

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

It describes how it acts on all unit vectors, indicated in Figure 3.2, and the eigenvectors indicate the locations with the strongest deformation.

As we are mainly interested in rank-2 or second-order tensors, the term *tensor* in this thesis will from here on refer to tensors of second-order, if not stated differently. Further, tensors will be represented as matrices in orthonormal bases only, such that only Cartesian tensors are used. Not only is this the standard representation in a computer science context, it further ensures that tensors are uniquely defined by their matrix.

To further be able to analyze and distinguish different tensors and make assumptions about how to deal with them mathematically, we first need to introduce some terms that allow for a more specific description. A tensor $\mathbf{T}$ represented by a square $n \times n$ matrix with components t_{ij}, $i, j \in [1, \dots, n]$, can be described in regard to its symmetries:
$\mathbf{T}$ is considered

- *symmetric* if $t_{ij} = t_{ji}$
- *asymmetric* otherwise.

Asymmetric tensors can further be described as

- *skew-* or *anti-symmetric* if $t_{ij} = -t_{ji}$

Further, for all vectors $\mathbf{v}$ distinct from the zero vector and with an appropriate number of components, $\mathbf{T}$ is

- *positive definite* if $\mathbf{v}^{\mathrm{T}}\mathbf{T}\,\mathbf{v} > 0$
- *negative definite* if $\mathbf{v}^{\mathrm{T}}\mathbf{T}\,\mathbf{v} < 0$
- *positive semi-definite* if $\mathbf{v}^{\mathrm{T}}\mathbf{T}\,\mathbf{v} \geq 0$
- *negative semi-definite* if $\mathbf{v}^{\mathrm{T}}\mathbf{T}\,\mathbf{v} \leq 0$
- *indefinite* otherwise.

The tensor is considered *traceless* if $\mathrm{tr}(\mathbf{T}) = t_{11} + \dots + t_{nn} = 0$.

Several of the properties above allow us to make assumptions about the results of further analysis of a tensor, such as the signs of the eigenvalues of an eigendecomposition.

3.1.1 *Tensor Decomposition*

As we can deal with tensors as linear operators and as such as matrices, we can make use of numerous tools from linear algebra that appear in the context of matrices. For a lot of applications, where this applies, it makes sense to decompose a tensor into factors. This means, that the tensor can be described as a sequence of operations, where each has a distinct meaning. When talking about matrices specifically, this is also known as matrix factorization (see, e. g., [174]). This can lead to easier processing of data or describe a certain underlying behavior in a way that is more tangible. For instance, the major directions of the diffusion described by a diffusion tensor as well as diffusivity strength are captured within the eigenvectors and eigenvalues produced by such a decomposition.

Eigenvalues and Eigenvectors

Any n-dimensional vector $\mathbf{e} \neq \mathbf{0}$ is considered an eigenvector of a square $n \times n$ matrix $\mathbf{T} \in \mathbb{R}$ if $\mathbf{T}\,\mathbf{e} = \lambda\,\mathbf{e}$ and λ is the corresponding eigenvalue. Geometrically, this means, that any linear transformation described by a matrix $\mathbf{T}$, maps an eigenvector $\mathbf{e}$ onto a scaled version of itself, where the scaling factor is given by λ. The equation above is known as the eigenvalue equation and allows us to derive what is known as the characteristic equation or characteristic polynomial of a matrix

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

This polynomial of degree n in λ has exactly n roots, which means that for an $n \times n$ matrix, there exist n, neither necessarily real-valued nor distinct eigenvalues $\lambda_1, \dots, \lambda_n$.

Eigendecomposition

Eigendecomposition, which is also known as spectral decomposition, is a matrix factorization that expresses a matrix $\mathbf{T}$ uniquely in terms of its eigenvalues and eigenvectors such that $\mathbf{T} = \mathbf{X}\,\Lambda\,\mathbf{X}^{-1}$. $\mathbf{X}$ is a square $n \times n$ matrix where the ith column is the ith eigenvector $\mathbf{e}_i$ of $\mathbf{T}$ and Λ is a diagonal matrix, where the diagonal elements are the corresponding eigenvalues $\Lambda_{ii} = \lambda_i$. The ordering is often given by the eigenvalues such that $\lambda_1 \geq \dots \geq \lambda_n$. A matrix can only be decomposed like this, if it is diagonalizable, i. e., there exist n linearly independent eigenvectors. For symmetric matrices, where $\mathbf{T} = \mathbf{T}^{\mathrm{T}}$, the eigenvalues are always real and eigenvectors are orthogonal, i. e., $\mathbf{X}^{-1} = \mathbf{X}^{\mathrm{T}}$ and the equation simplifies to

$$\mathbf{T} = \mathbf{X}\,\Lambda\,\mathbf{X}^{\mathrm{T}} = \begin{pmatrix} | & | & & | \\ \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_n \\ | & | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{pmatrix} \begin{pmatrix} — & \mathbf{e}_1 & — \\ — & \mathbf{e}_2 & — \\ & \vdots & \\ — & \mathbf{e}_n & — \end{pmatrix}.$$

Further, within this work, the plane that is spanned by two eigenvectors is referred to as an *eigenplane*.

Singular Value Decomposition (SVD)

Any real-valued $m \times n$ matrix $\mathbf{T}$, which is not necessarily square, can be decomposed into $\mathbf{T} = \mathbf{U}\,\Sigma\,\mathbf{V}^{\mathrm{T}}$ where $\mathbf{U}$ and $\mathbf{V}^{\mathrm{T}}$ are real orthogonal matrices and Σ is a diagonal matrix. The orthonormal columns of $\mathbf{U}$ are also referred to as the left singular vectors, those of $\mathbf{V}^{\mathrm{T}}$ right singular vectors accordingly and the elements on the diagonal of Σ are the non-negative real singular values $\Sigma_{ii} = \sigma_i$. Geometrically, this means that $\mathbf{T}$ maps the ith basis vector of $\mathbf{V}$ to the ith scaled basis vector of $\mathbf{U}$ where the scaling is given by the ith singular value such that $\mathbf{T}\,\mathbf{V} = \mathbf{U}\,\Sigma$. It can be shown that singular value decomposition and eigendecomposition are related as we can apply the decomposition to $\mathbf{T}^{\mathrm{T}}\mathbf{T}$ such that

$$\mathbf{T}^{\mathrm{T}}\mathbf{T} = \mathbf{V}\,\Sigma^{\mathrm{T}}\,\mathbf{U}^{\mathrm{T}}\mathbf{U}\,\Sigma\,\mathbf{V}^{\mathrm{T}} = \mathbf{V}\,(\Sigma\,\Sigma^{\mathrm{T}})\,\mathbf{V}^{\mathrm{T}}.$$

As $\mathbf{T}^{\mathrm{T}}\mathbf{T}$ is a square symmetric matrix and $(\Sigma\,\Sigma^{\mathrm{T}})$ a diagonal matrix, it resembles a typical form of an eigendecomposition where $\mathbf{V}$ includes the eigenvectors and the non-zero singular values are the square roots of the non-zero eigenvalues. Similarly, it can be shown that the columns of $\mathbf{U}$ are eigenvectors of $\mathbf{T}\,\mathbf{T}^{\mathrm{T}}$. One advantage of the singular value decomposition is, that while eigenvalues of a matrix can turn out complex, singular values are always real-valued.

Further, the singular value decomposition can be used to construct the *polar decomposition*

$$\mathbf{T} \;=\; \mathbf{U}\,\Sigma\,\mathbf{V}^{\mathrm{T}} = (\mathbf{U}\,\mathbf{V}^{\mathrm{T}})\,(\mathbf{V}\,\Sigma\,\mathbf{V}^{\mathrm{T}}) \;=\; \mathbf{Q}\,\mathbf{H}\;,$$

which can be used to factor any square real matrix into an orthogonal matrix $\mathbf{Q}$ and a symmetric positive semidefinite matrix $\mathbf{H}$ with $\lambda_i(\mathbf{H}) = \sigma_i(\mathbf{T})$. Geometrically, $\mathbf{H}$ can be understood as describing scaling along orthogonal axes while $\mathbf{Q}$ resembles a distance preserving transformation, such as a rotation or a reflection, as the columns form an orthonormal basis. If $\det(\mathbf{Q}) = +1$, $\mathbf{Q}$ is known as a special orthogonal matrix and describes a rotation as opposed to reflections. When dealing with matrices in $\mathbb{R}^2$, such a special orthogonal matrix $\mathbf{Q}$ can be parametrized by an angle of rotation γ only, which is given as $\tan\gamma = q_{21}/q_{11}$.

Several tensor norms can be described by using the singular values. The *spectral norm* of $\mathbf{T}$ given by $||\mathbf{T}||_2$ is defined by the maximum singular value such that

$$||\mathbf{T}||_2 = \sigma_1(\mathbf{T}) = \sqrt{\lambda_1(\mathbf{T}^{\mathrm{T}}\,\mathbf{T})}\;.$$

Additionally, the *Frobenius norm* can be described as

$$||\mathbf{T}||_F = \sqrt{\sigma_1(\mathbf{T})^2 + \ldots + \sigma_n(\mathbf{T})^2}$$

or alternatively as

$$||\mathbf{T}||_F = \sqrt{\textstyle\sum_{i=0}^{n}\sum_{j=0}^{n} t_{ij}^2}\;.$$

Further Decompositions

Besides the singular value decomposition and eigendecomposition, which we will frequently refer to throughout this thesis, there exist a number of further tensor decompositions such as a factorization of any tensor into a symmetric and an antisymmetric part:

$$\mathbf{T} = \mathbf{S} + \mathbf{A} = \underbrace{\frac{1}{2}(\mathbf{T} + \mathbf{T}^{\mathrm{T}})}_{S} + \underbrace{\frac{1}{2}(\mathbf{T} - \mathbf{T}^{\mathrm{T}})}_{A} \ .$$

In most physical applications, the antisymmetric part then contains information on rotation while the symmetric part encodes scaling and shear. A symmetric tensor can also be decomposed into an isotropic and deviatoric factor. For 3×3 tensors, it follows

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

These are used frequently within medical and mechanical tensor applications where domain experts are interested in areas in tensor fields where the quantity encoded changes uniformly in all directions or favors certain directions.

3.1.2 *Tensor Invariants*

Tensor invariants are derived properties of a tensor that do not change under a change of the frame of reference. This means that while the components of a tensor in matrix representation do change under changes such as domain rotation, invariants do not and thus are objective descriptions. For instance, the diffusion tensor itself, which is measured in a DT-MRI scan, depends on how a patient is positioned for the scan. Especially scalar invariants are frequently used to analyze and compare the resulting data as they are easily visualized. The significance of a specific invariant and its impact is highly application dependent. These properties can describe different aspects of the tensor data such as the degree to which diffusion is anisotropic. Three of such scalar invariants often used in engineering and medical context that deal with 3D tensors are called the principal invariants, which can be described in terms of the trace and determinant of the tensor:

- $I_1 = \mathrm{tr}(\mathbf{T}) = \lambda_1 + \lambda_2 + \lambda_3$
- $I_2 = \frac{1}{2}((\mathrm{tr}(\mathbf{T}))^2 - \mathrm{tr}(\mathbf{T}^2)) = \lambda_1\lambda_2 + \lambda_1\,\lambda_3 + \lambda_2\,\lambda_3$
- $I_3 = \det(\mathbf{T}) = \lambda_1\,\lambda_2\,\lambda_3$

Further invariants especially developed for diffusion tensor analysis include *relative anisotropy (RA)* and *fractional anisotropy (FA)* as presented by Basser et al. [12], *volume ratio* and *lattice index* by Perpaoli and Basser [145], shape descriptors like the *linear, planar and spherical shape measures* by Westin et al. [201] or the *tensor mode* by Ennis and Kindlmann [40].

3.1.3 *Second-Order Tensors as Vectors*

Symmetric tensors represented as matrices store redundant information. While a 3×3 tensor has 9 entries, there are in fact only 6 distinct values. The Mandel notation allows to represent symmetric tensors as vectors. We define the operator $\mathsf{v}(\cdot)$ that transforms a symmetric second-order $(n \times n)$ tensor $\mathbf{T}$ into a vector. In this work, we deal with tensors with $n = 2, 3, 6$ which gives vectors of dimension $3, 6, 21$, respectively, as

$$\begin{aligned}
\mathsf{v}(\mathbf{T}) &= \left(t_{11}, t_{22}, \sqrt{2}\, t_{12}\right)^{\mathrm{T}} \\
\mathsf{v}(\mathbf{T}) &= \left(t_{11}, t_{22}, t_{33}, \sqrt{2}\, t_{12}, \sqrt{2}\, t_{13}, \sqrt{2}\, t_{23}\right)^{\mathrm{T}} \\
\mathsf{v}(\mathbf{T}) &= \left(t_{11}, \ldots, t_{66}, \sqrt{2}\, t_{12}, \ldots, \sqrt{2}\, t_{16}, \ldots, \sqrt{2}\, t_{56}\right)^{\mathrm{T}}.
\end{aligned}$$

Note that $\mathsf{v}(\cdot)$ describes an isometric embedding of the tensor space into $\mathbb{R}^{3/6/21}$, i. e., scalar products, and hence distances are preserved. In particular, the following holds:

$$\mathbf{r}^{\mathrm{T}}\ \mathbf{T}\ \mathbf{r} = \mathsf{v}(\mathbf{r}\mathbf{r}^{\mathrm{T}})^{\mathrm{T}} \mathsf{v}(\mathbf{T}) \tag{3.1}$$

for a symmetric second-order tensor $\mathbf{T}$ and a vector $\mathbf{r}$. Further, any rotation $\mathbf{R}$ in domain coordinates acting on a tensor $\mathbf{T}$ corresponds to a rotation $\hat{\mathbf{R}}$ acting on $\mathsf{v}(\mathbf{T})$ in the isomorphic vector space such that

$$\mathsf{v}(\mathbf{R}\ \mathbf{T}\ \mathbf{R}^{\mathrm{T}}) = \hat{\mathbf{R}}\ \mathsf{v}(\mathbf{T})\ . \tag{3.2}$$

For details on notation and properties see, e. g., [66].

3.2 TENSOR FIELD ANALYSIS

Tensor data is omnipresent in a variety of applications from medical data such as the aforementioned diffusion data to applications in aerodynamics and hydrodynamics. The analysis of tensor data therefore aims towards a better understanding of these complex phenomena. However, due to their high dimensionality, this poses a challenging task. Visualization aims to support domain experts with that task by emphasizing interesting structures or behaviors within the data or simplifying the fields for the analysis of derived properties. Before we discuss visualization approaches of tensor data, we first introduce different common applications as well as the requirements that arise from them. Table 3.1 offers a short summary of selected tensor examples and their properties.

3.2.1 *Diffusion Tensor*

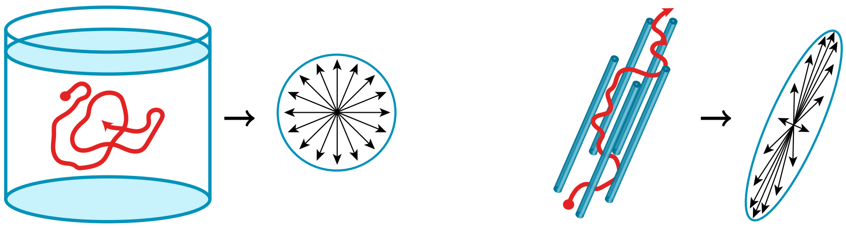


Figure 3.3: Pictorial visualization of diffusion: the trajectory of a massless particle within different media shows different diffusion behavior. Unrestricted isotropic diffusion (left) in free water and restricted anisotropic diffusion (right) along axon fibers in white matter.

Diffusion is the process of particle movement of material through another. While in unrestricted fluid, such diffusion is constant, liquid moving through tissue, however, is more likely to favor certain directions, depending on the tissue structure, which is known as anisotropic diffusion. This behavior can be measured and made visible for particles such as water molecules in brain tissue by using *Diffusion-Weighted Imaging (DWI)*, which is a non-invasive technique that uses magnetic field gradients to measure how restricted particle motion happens at voxels. In particular, *Diffusion Tensor Magnetic Resonance Imaging (DT-MRI)* uses the so-called diffusion tensor, which describes the 3D direction-dependent diffusion using a Gaussian model. It can be acquired by solving a system of equations that use diffusion-weighted measurements in at least six different directions, known as the *apparent diffusion coefficients (ADCs)*, as the results of the Stejskal-Tanner equation [10]. The diffusion tensor can be represented by a 3×3 positive semi-definite symmetric matrix, i. e., there are only 6 independent variables. This further implies that eigenvectors always form an orthonormal basis and eigenvalues are always real and nonnegative values, such that $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ as introduced in Section 3.1.1. The major eigenvector represents the direction of the strongest diffusion and diffusion strength is implied by the corresponding eigenvalue. When two eigenvalues are repeated, all vectors that lay in the plane spanned by the corresponding eigenvectors, are valid eigenvalue choices, which means there are infinitely many possible eigenvectors in that plane. This indicates isotropic diffusion within this plane and is often referred to as a degenerate tensor. When three eigenvalues are repeated, the tensor is purely isotropic, and the diffusion is identical in all directions.

Neuroscience applications are mainly focused on anisotropy measures, as well as the direction of the strongest diffusion as an indicator of neural fibers. As the tube-like geometry of such fibers results in constraint motion of the particles into preferred directions as indicated by Figure 3.3, this information can be used to get an idea of the structure of fiber tissue such as muscles or nerves. Finding such nerve tracts using the tensor data and visualizing it is known as tractography. Analyzing those allows experts to find abnormalities, diagnose diseases, and propose treatments based on the data.

Even though diffusion tensor imaging offers valuable insights into medical applications such as brain network connectivities, it suffers from a few limitations [124]. This is mainly due to the fact, that it represents a rather oversimplified idea of the actual fiber structures with only one main direction and it is not capable of modeling more complex signals. One of the most discussed drawbacks is its poor fit to signals acquired from voxels that exhibit "crossing fibers", which include intertwined, passing, or diverging fiber bundles. Applications that rely on diffusion tensor data such as several tractography approaches will therefore produce wrong or inaccurate results. Other approaches such as *High Angular Resolution Diffusion Imaging (HARDI)* try to tackle these issues by taking a far larger number of measurements which can then be represented by probability functions which in turn also require new visualization techniques.

3.2.2 *Stress and Strain Tensor*

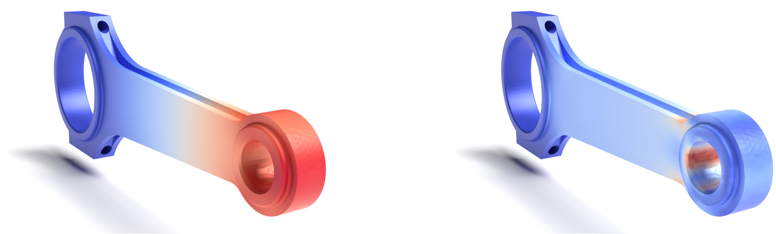


Figure 3.4: A displacement force is applied to a steel connecting rod (left). The color indicates the strength of the force applied. This produces stresses within the material which can be described by a stress tensor field. The derived von Mises stress value (right) indicates where the material is more likely to fail.

In a physics context, stress describes the reactions within materials based on external forces, as particles within the material are forced to move and interact with one another. Similarly, strain describes the deformation of the body based on external stress. Both stress and strain tensors are therefore capable of describing how material is reacting when loads are applied and therefore describe a materials' strength. As stress within a material in 3D is not only defined by the direction of the applied force, but also by the orientation of the surface it acts upon, it can be described by a symmetric indefinite tensor

$$\mathbf{T} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}.$$

The diagonal elements $\sigma_{11}, \sigma_{22}, \sigma_{33}$ are also referred to as the normal stresses, whereas the off-diagonal elements are called shear stresses. The signs of these components indicate if the stress is either compressive or tensile. The eigenvectors are occasionally also referred to as the principal stress axes and the eigenvalues as principal stresses. Unlike the diffusion tensor, decomposing a stress tensor into eigenvalue-eigenvector pairs might yield negative eigenvalues. Mechanical engineering applications often make use of derived tensor quantities such as the maximum shear stress or the von Mises stress to analyze under which circumstances a material might fail.

3.2.3 *Gradient Tensor of Vector Fields*

Fluid flows are most commonly represented by vector fields that describe the instantaneous movement of massless particles at each location with a vector known as velocity. The velocity gradient holds the information, how the velocity changes when moving away from the current location, depending on the direction and it is derived from the first-order Taylor series expansion of the velocity at that point. The matrix containing all the first-order derivatives of $\mathbf{v}$ is known as the Jacobian matrix $\mathbf{T}$:

$$\mathbf{T} = \nabla\mathbf{v} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix} = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{pmatrix}$$

Jacobian matrices are general, thus not necessarily symmetric and indefinite tensors, which means that eigenvectors can be non-orthogonal, and eigenvalues can be complex-valued. In 2D, they appear either as two real or a pair of complex conjugate eigenvalues while in 3D, the eigenvalues are either all real, or there is one real eigenvalue and a complex conjugate pair. Assuming a linear vector field around a given location, the Jacobian matrix can be used to assign a vector to these domain locations $\mathbf{x}$ by $\mathbf{v}' = \mathbf{T}\,\mathbf{x}$. This is displayed in Figure 3.5.

This assumption can be used to analyze the local behavior of vector fields. At critical points, where the velocity is zero, the eigenvectors and eigenvalues of the Jacobian matrix with full rank can be used to classify these first-order features within the flow: when eigenvectors are real,

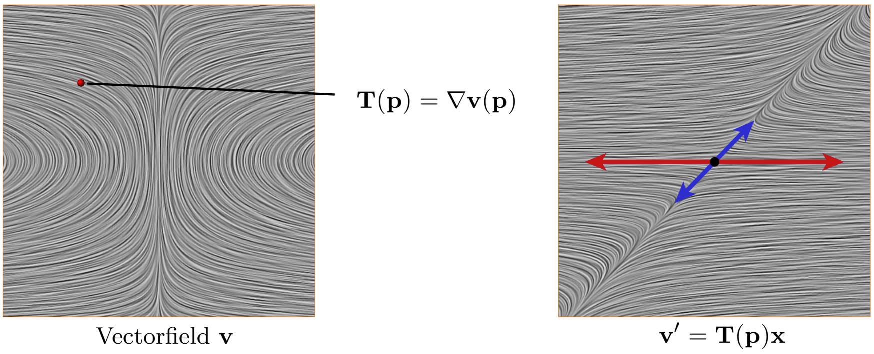


Figure 3.5: Left: LIC texture visualization of a 2D flow field $\mathbf{v}$. The Jacobian matrix $\mathbf{T}$ at a location $\mathbf{p}$ and its eigenvectors can be used to analyze the local flow behavior at that point by assuming a linear vector field (right).

the eigenvectors are tangent to streamlines ending at that point and the eigenvalue signs indicate whether particles are flowing towards or away from the point. When the eigenvalues are complex, this indicates rotational flow behavior. Thus, within the analysis of 2D vector field topology, the eigenvalues $\lambda_i \in \mathbb{C}$ allow classifying the flow behavior around a point as a

- Saddle (Sa), if
  $\mathrm{Re}\,(\lambda_1) < 0, \mathrm{Re}\,(\lambda_2) > 0$ and $\mathrm{Im}\,(\lambda_1) = i_2 = 0$
- Repelling Node (RN), if
  $\mathrm{Re}\,(\lambda_1), \mathrm{Re}\,(\lambda_2) > 0$ and $\mathrm{Im}\,(\lambda_1) = \mathrm{Im}\,(\lambda_2) = 0$
- Attracting Node (AN), if
  $\mathrm{Re}\,(\lambda_1), \mathrm{Re}\,(\lambda_2) < 0$ and $\mathrm{Im}\,(\lambda_1) = \mathrm{Im}\,(\lambda_2) = 0$
- Repelling Focus (RF), if
  $\mathrm{Re}\,(\lambda_1) = \mathrm{Re}\,(\lambda_2) > 0$ and $\mathrm{Im}\,(\lambda_1) = -\,\mathrm{Im}\,(\lambda_2) \neq 0$
- Attracting Focus (AF), if
  $\mathrm{Re}\,(\lambda_1) = \mathrm{Re}\,(\lambda_2) < 0$ and $\mathrm{Im}\,(\lambda_1) = -\,\mathrm{Im}\,(\lambda_2) \neq 0$
- Center (C), if
  $\mathrm{Re}\,(\lambda_1) = \mathrm{Re}\,(\lambda_2) = 0$ and $\mathrm{Im}\,(\lambda_1) = -\,\mathrm{Im}\,(\lambda_2) \neq 0.$

In 3D, these points have an additional in- or outflow component, depending on the eigenvalue sign. The gist of a vector field can therefore be described by finding and showing such locations [67]. The Jacobian matrix is often used to identify even more topological structures such as separatrices [68], attachment and separation lines [91] or interesting features such as vortex core lines [86], [157], [175] which can then be visualized.
While many analysis strategies for symmetric tensors often rely on quantities based on real eigenvalues, they might not be applicable to

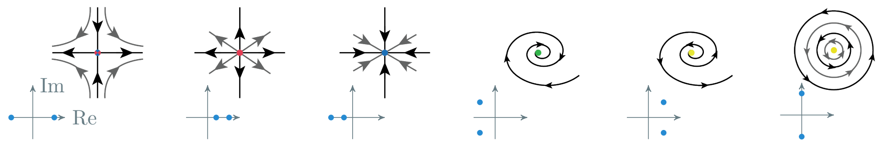


Figure 3.6: Line glyphs indicating particle motion within a linear approximation of the 2D velocity field around a critical point. They are classified from left to right as Saddle, Attracting Node, Repelling Node, Attracting Focus, Repelling Focus, and Center.

asymmetric tensor fields. One strategy proposed by Delmarcelle and Hesselink [32] is to decompose the tensor into its symmetric and antisymmetric parts as described in Section 3.1.1. The symmetric part can then be analyzed with known strategies for symmetric tensors and the antisymmetric part can be represented by a vector field. This does however not capture the geometric meaning encoded by the tensor. New analysis methods for asymmetric tensors have only recently been developed [112], [210], [212] and are still an active research topic. When dealing with such fields, parts of the domain, where eigendecomposition still yields real eigenvalues are considered to be in the real domain, whereas the others are located in the complex domain. It was further shown that the locations where degenerate tensors appear, can then form lines within the field, instead of isolated points, as it was the case for symmetric tensors. In fields that have real and complex domains, degenerate tensors are, in fact, the border between both domains and thus a feature revealing the topological structure.

Tensor	Encoded Quantity	Symmetric	Definiteness
Diffusion	Diffusivity of moving particles depending on direction	yes	positive semi-definite
Stress	Material description of internal forces based on external forces	yes	indefinite
Strain	Local change of a body due to stress	yes	indefinite
Surface Curvature	Local change of the normal of a smooth surface depending on direction	yes	indefinite
Deformation Gradient	Local deformation of volume element	no	indefinite
Velocity Gradient	Local change of velocity depending on direction	no	indefinite

Table 3.1: Several different second-order tensors from real-world applications and their properties.

3.2.4 *The (γ, r) Phase Space*

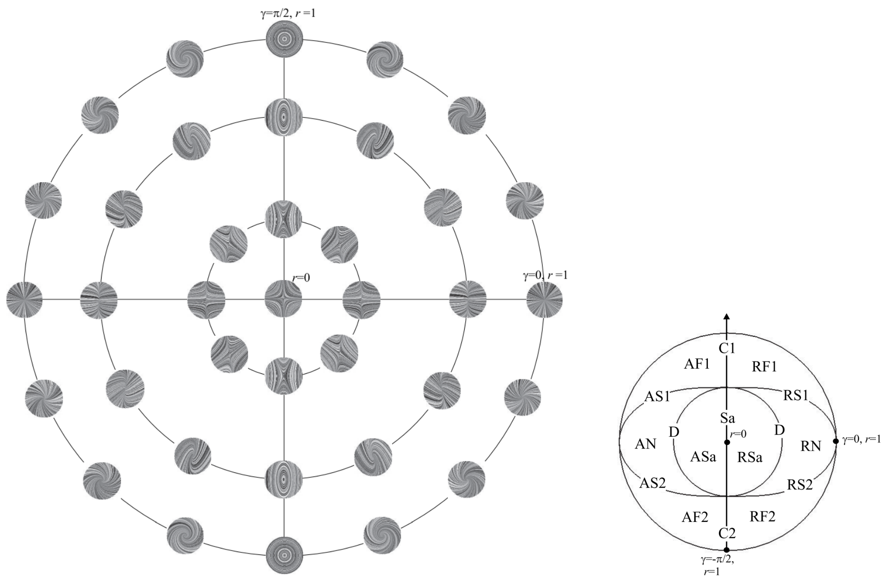


Figure 3.7: Visual representations of the (γ, r) phase space by Theisel and Weinkauf [181]. Left: sampled locations with the related linear vector field visualized with LIC textures based on the represented first-order critical points. Right: classification of first-order critical points within different sections of the space.

By only looking at matrix representations of tensors, it is hard to get an idea of how similar or dissimilar two tensors actually are. It, therefore, makes sense to quantify a difference measure that allows to better describe and distinguish classes of tensors as well as to find and analyze transitions between different cases. Originally intended as a tool for comparing vector fields based on their topology, Lavin et al. [108] introduced the mapping of Jacobian matrices of critical points in 2D vector fields to an (α, β) phase plane and measuring what is known as the earth mover's distance. The main idea is, that this captures the amount of work that needs to be performed in order to transform one critical point into another. This idea was then extended by Theisel and Weinkauf [181] to a new parametrization, which maps a given 2D tensor to the so-called (γ, r) phase plane to allow for a complete classification scheme for first-order critical points.

As shown by Rössl and Theisel in [47], the mapping can be calculated based on a subspace of all 2D tensors where any of the tensors $\mathbf{T}^*$ can be described in terms of their polar decomposition

$$\mathbf{T}^* \;=\; \mathbf{Q}\,\mathbf{H} \;=\; \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix} \begin{pmatrix} 1 & \\ & \sigma_2 \end{pmatrix} \,, \tag{3.3}$$

such that the spectral norm of $\mathbf{T}^*$ is constrained to 1, meaning $||\mathbf{T}^*||_2 = 1$. Further, there is no orthogonal transform in $\mathbf{H}$ (i. e., the right singular vectors $\mathbf{V} = \mathbf{I}$). This means that tensors mapped to the subspace have domain scaling and rotation removed as these factors do not change the behavior of the flow at a given location and thus form a collection of what is called reference critical points. The parameter $\gamma \in [0, 2\pi)$ determines the amount of rotation, while $r \in [0,1]$ determines $\mathbf{T}^*$'s smaller singular value $\sigma_2 = \frac{1-2\sqrt{(1-r)r}}{2r-1} \in [-1,1]$. Additionally, the sign of σ_2 directs the sign of the determinant $\mathrm{sgn}(\sigma_2) = \mathrm{sgn}(\det(\mathbf{T}^*))$. This now allows us to map any *arbitrary* tensor $\mathbf{T}$ to the (γ, r)-plane to get a reference critical point $\mathbf{T}^*$ which describes the same flow behavior. The mapping can be achieved by using the polar decomposition to yield γ and calculate $r = \frac{1}{2} + \det(\mathbf{T}) / ||\mathbf{T}||_F^2 = \frac{1}{2} + \mathrm{sgn}(\det(\mathbf{T}))\, \frac{\sigma_1 \sigma 2}{\sigma_1^2 + \sigma_2^2}$.

By analyzing $\mathbf{T}^*$, we can derive a number of properties for a tensor $\mathbf{T}$ following the statements of Theisel and Weinkauf [181]:

(i) Let $r^\star = \frac{1}{1+\sin^2\gamma}$. Then $r^\star \geq \frac{1}{2}$, and

$$\mathbf{T} \text{ has eigenvalues } \begin{cases} \lambda_1 \neq \lambda_2 \in \mathbb{R} & \text{for } r < r^\star \\ \lambda_1 = \lambda_2 \in \mathbb{R} & \text{for } r = r^\star \\ \lambda_1 = \overline{\lambda_2} \in \mathbb{C} & \text{else} \end{cases} .$$

(ii) $$\det(\mathbf{T}) = \lambda_1 \lambda_2 \begin{cases} < 0 & \text{for } 0 \leq r < \frac{1}{2} \\ = 0 & \text{for } r = \frac{1}{2} \\ > 0 & \text{else} \end{cases} .$$

This is visualized in Figure 3.7, where the space is represented in polar coordinates. Parameter r is the distance from the center and tensors that lie on a ray from the center can be transformed into another by applying a scaling, whereas γ denotes the angle and tensors lying on a circle can be transformed into another by adding rotation. As it describes all classes of tensors, all possible flow behaviors are also found within this space. Figure 3.7 (left) shows LIC textures for sampled locations, while on the right, the whole space is segmented into areas of similar flow behavior. The inner circle describes critical points with saddle (Sa) flow, divided into attracting (ASa) or repelling (RSa) saddle behavior. There further are certain locations for attracting (AN) and repelling (RN) nodes as well as attracting (AF) and repelling (RF) foci. On the outer circle, tensors describe center flow (C) that differs in rotation strength and direction (clockwise and anti-clockwise), whereas the special cases $\gamma = 0$ and $\gamma = \pi$ denote sources and sinks. Tensors on the border of the inner circle itself are called degenerate cases (D). Beware that this "degeneracy" is different from the definition of a degenerate tensor that we introduced earlier where eigenvalues are repeated. In their work, it is used for tensors with a zero determinant such as when one eigenvalue vanishes, and which does not resemble a first-order critical point of a vector field. They further added attracting (AS) and repelling stars (RS) as tensors on the boundary between the real and complex domain, which is represented by the pill-like shape. The change of the tensor while moving within the space can be analyzed in terms of how the eigenvectors and eigenvalues change. While eigenvalues within the inner circle are real-valued and eigenvectors orthogonal, the tensors between the circle and the outer oval boundary have real eigenvalues, but eigenvectors can be non-orthogonal. They do, however, all lie within the real domain. The transition from the real to the complex domain follows a specific pattern: while $\gamma \neq 0$ and $\gamma \neq \pi$, moving from the center outwards by increasing r towards the oval decreases the angle between the eigenvectors until they coalesce. Further, increasing r within the real domain changes the relation of eigenvalues. This subspace allows to study and describe all possible classes of 2D tensors and, more importantly, smooth transitions between them, such that it is a useful basis for designing tensor visualizations that capture all cases.

3.3 VISUALIZATION OF TENSOR FIELDS

As we have seen, tensors do appear in a variety of applications, which makes it important that visualization tools are available that support the analysis of tensor properties in a way to draw conclusions from them. As different kinds of tensors are used to describe different physical phenomena, domain experts might be interested in different aspects of the tensor. In this section, we will go through some general visualization concepts, as well as domain-specific visualizations.

Vector field visualization and tensor field visualization are closely related. Instead of finding visual representations for two or three vector components in 2D or 3D vector fields, tensors are typically represented by 2×2 or 3×3 matrices, which increases the number of variables and makes interpreting as well as finding appropriate visualizations a more challenging task.
A straightforward solution for visualizing a single tensor is to treat each of the components as another scalar field and use scalar field visualization techniques such as mapping the scalar values to color or luminance. Each scalar field can then be rendered separately as seen in Figure 3.8. This technique does however ignore the structural or geometric information, that is encoded by the tensor. A change of the reference frame would also lead to a change in the scalar fields and thus in the visualization, which conflicts with the general idea behind tensors as reference independent descriptions.
Calculating scalar tensor invariants, as described in Section 3.1.2 is therefore a viable option for tensor analysis. Especially in the medical

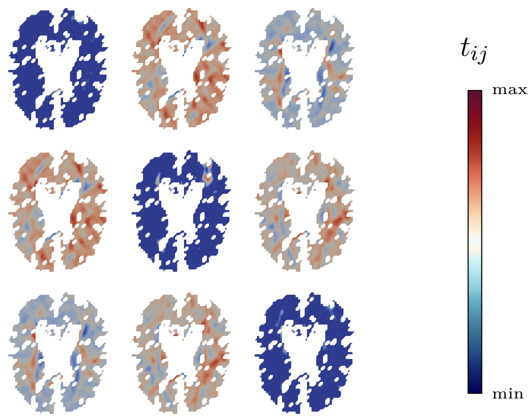


Figure 3.8: Visualization of a 2D slice of 3D Diffusion tensors: each tensor component is visualized independently as a scalar field mapped to color and presented in a combined matrix visualization.

context, it is common to analyze scalar quantities that describe the anisotropy of diffusion tensors [7], [145], [201]. Similar quantities exist for stress tensors [29], [35], [79], [100], which support the analysis of properties like material failures. The resulting scalar fields can then be analyzed using techniques from scalar field visualization such as isosurface extraction [40], [149] or scalar field topology [183].

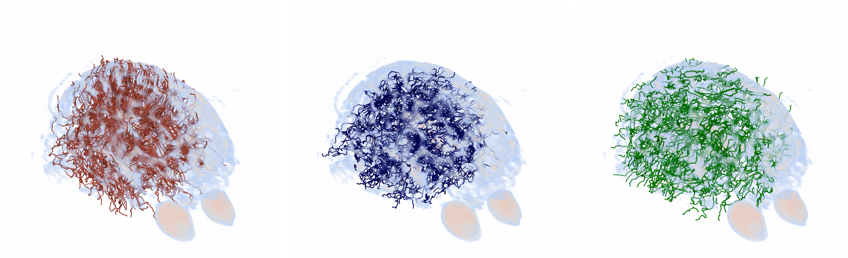


Figure 3.9: Streamline visualizations of the three eigenvector fields derived from diffusion tensor data in a human brain. A transparent volume rendering is added for context information.

Just as the eigenvalues are interesting scalar quantities of a tensor, the eigenvectors do play an important role in tensor visualization. A tensor field can be represented by the two (in 2D) or three (in 3D) eigenvector fields as shown in Figure 3.9, which are similar to vector fields with some differences: eigenvectors do not have an orientation, which means, such an eigenvector field describes a bidirectional flow. Further, as every scaled version of an eigenvector is also an eigenvector, it often makes sense to use unit eigenvectors which means the Euclidean norm of the vectors is 1. This decomposition of tensors to eigenvectors allows using techniques from vector field visualization to visualize the resulting fields. Especially the major eigenvector, which is corresponding to the largest eigenvalue, is of interest. Tractography fiber tracking approaches for instance use the major eigenvector as an indication of fiber direction in regions with strong anisotropy. A simple approach is to map the components of the major eigenvector to the different channels of the RGB color model [138]. In general, tensor field lines [160] or tensor lines in short, are tangent lines to eigenvectors of the tensor and as such similar to streamlines of vector fields. This also means that they use only one of the existing eigenvector fields and ignore the remaining tensor information. Additionally, care needs to be taken such that the line integration does not switch orientation due to sudden changes in eigenvector orientation. Delmarcelle and Hesselink propose a technique that enhances such lines called hyperstreamlines [32], [33]. These line-type features also result from the advection of particles in the direction of the chosen eigenvector but are modified by the remaining information. They are often represented as tubes or ribbons where diameter or width is changed depending on the other eigenvalues. All these line techniques however are relying on a stable ordering of the values. The choice is problematic close to locations of degenerate tensors, where eigenvalues are repeated and can be associated with different eigenvectors. When integrating tensor lines in near-isotropic areas, the order might change abruptly. Weinstein and Kindlmann introduced what they also call Tensorlines [200] to address these issues by trying to preserve the directions of the lines in the vicinity of such locations. The idea of hyperstreamlines can also be extended to form so-called hyperstreamsurfaces [87] by seeding several hyperstreamlines and connecting them.

Zheng and Pang propose an extension to the well-known line integral convolution texture visualization for vector fields [25], [190] called HyperLIC [211]. Instead of accumulating color values along one vector directions, they filter a noise texture within small areas that are defined and deformed by the eigenvectors and eigenvalues resulting in lines in parts of the field, where one eigenvalue is dominant and blurry regions in isotropic parts. Similarly, tensor fabrics [39], [80] produce continuous tensor field visualizations on surfaces.

Tensor field topology, similar to vector field topology, extract locations and features that describe the structure of tensor fields. The definition, extraction, and visualization of topological features have been discussed in a variety of contributions over the last years, especially for the symmetric case (see e. g., [5], [70], [109], [156], [184], [192], [213]). Whereas in vector field topology, critical points are locations, where the vector field vanishes and which can be classified by the vector field gradient, in tensor fields, one is often interested in finding degenerate tensors. In 2D, where tensors can only be isotropic or anisotropic, these locations can be classified by the tensor index into trisectors and different types of wedges and are connected with separatrices. In 3D, tensors can also show full ($\lambda_1 = \lambda_2 = \lambda_3$) and partial ($\lambda_1 = \lambda_2$ or $\lambda_2 = \lambda_3$) isotropy which leads to degenerate lines. Lately, the definition and extraction of

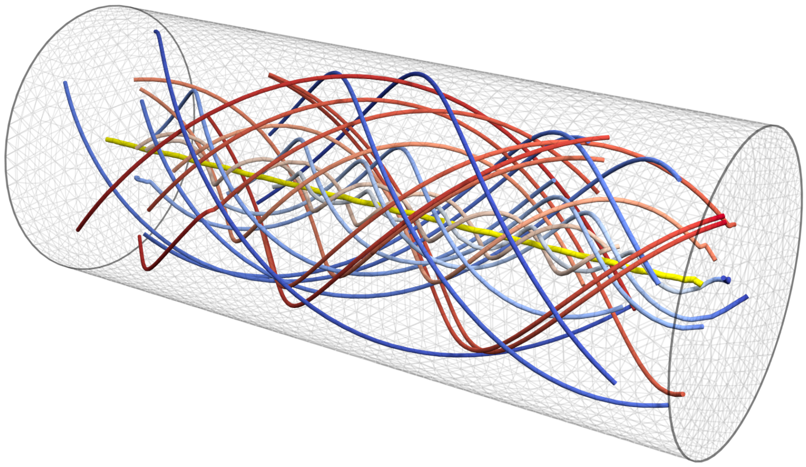


Figure 3.10: The yellow tube represents a core line of a 3D second-order stress tensor field. They are the center of swirling eigenvector trajectories. Image by Oster et al. [130].

feature surfaces such as those indicating locations of traceless tensors, neutral tensors [139], or extremal surfaces [216] have been added to the list of available features. Many of these approaches do, however, rely on real eigenvalues and eigenvectors as shown in Section 3.2 and are therefore not easily extended for asymmetric tensors. The concept of dual eigenvectors [212] and pseudo-eigenvectors [210] allows for a continuous extension of the eigenvector into the complex domain, such that tensor lines and hyperstreamlines can be extended. Further publications that discuss tensor field topology including the asymmetric cases are [65], [107], [112].

Hagen et al. visualize deformation tensor fields [61] by extending the idea of focal surfaces [60] for tensor properties like the maximum eigenvalue. Whereas parallel vectors of velocity fields and derived vector fields, like acceleration, can indicate the existence of features such as vortex core lines [54], Oster et al. [130] propose a similar approach using parallel eigenvectors [129] of a given tensor field, as shown in Figure 3.10. Further ideas that aid the process of tensor field analysis, such as geodesics [38] or linked views for tensor field analysis [27], [88] as well as more detailed discussions on tensor field visualization can be found in a variety of works (see e. g., [16], [18], [71], [99], [100], [102]).

Besides those general visualization approaches, few techniques exist that also investigate the development of tensors over time. In the context of vector fields, pathlines of a finite set of seed points are used to visualize flow in this case. Especially in topology-based visualization techniques, some works have been proposed, where the path of features over time is visualized as presented by Uffinger and Sadlo [188]. Regarding tensor fields in general, Delmarcelle and Hesselink [34] as well as Trichoche et al. [185], [186] introduced tracking techniques for degenerate points in time-dependent 2D symmetric tensor fields. The tracked points form lines while separatrices form surfaces in the spatiotemporal domain.

Similarly, only little work can be found on the visualization of uncertain tensor fields. Visualizing uncertainty is in general a challenging task, as it adds another dimension to the visualization space. There exists a multitude of literature on uncertainty visualization, see, e. g., the reviews by Bonneau et al. [20] and Brodlie et al. [23], but high-dimensional data is only sparsely covered. A notorious example of uncertainty in tensor data is the diffusion tensor, which is obtained, e. g., from diffusion tensor magnetic resonance imaging (DT-MRI). Uncertainty typically stems from the measurement process, which introduces a significant amount of Gaussian noise [11]. An alternative source of uncertainty is the fusion of tensors from members of an ensemble. Several approaches deal with visualizing the uncertainty in tensor fields by considering not the whole tensor and its uncertainty but only derived scalar and vector invariants such as Behrens et al. [14] or the work of Schultz et al. [168], that presents fiber tracking methods for probabilistic tractography to extract fuzzy features.

While techniques visualizing only derived quantities of a tensor are often referred to as indirect visualizations, a very popular direct technique, which shows all the information given by a tensor, is the use of glyphs, which are then referred to as tensor glyphs. They are geometric objects placed at sampled locations within a tensor field to display the local tensor properties by mapping them to geometric variables like shape, orientation, or color. Their construction, interpretation as well as related work will be discussed in the following chapter.

4

TENSOR GLYPHS

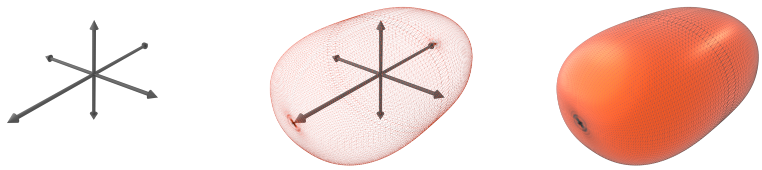


Figure 4.1: The construction of some existing tensor glyphs for symmetric tensors such as the superquadric tensor glyphs [93], [165] is based on the eigendecomposition of the tensor. The scaled eigenvectors (left) are represented by the axes of a geometric object such as a superellipsoid (center). The extend and orientation of the surface fully captures the information encoded by the tensor without explicitly showing arrow glyphs for the eigenvectors (right).

Glyphs are omnipresent in scientific visualization. Whenever multidimensional information is to be visualized at a certain location, glyphs are the standard choice. Essentially, glyphs are geometric objects that represent data at a sampled location within the field. Tensor glyphs are such glyphs that aim to represent a given tensor. When $\mathbf{T}(\mathbf{x})$ is the tensor value at a given sample location $\mathbf{x}$, we define $G(\mathbf{T}(\mathbf{x}))$ to be its corresponding glyph representation. This is typically done for a number of locations, such that several glyphs are placed next to each other. For velocity data in $\mathbb{R}^2$ and $\mathbb{R}^3$ a straightforward choice is to represent it as an arrow glyph encoding direction and magnitude. For tensor data represented by matrices, there is no such easy direct mapping. A common thing to do is therefore to map derived tensor attributes onto geometric properties like shape, orientation, color, or texture. For such multidimensional data of a certain type, there is generally not one best glyph. The design choices are either highly tailored to the application or based on general design rules [21], [111], [196].

Before we discuss such rules applicable to tensor data and desirable properties in Section 5.1, we list and elucidate the development of existing glyph constructions.

4.1 RELATED WORK

One of the first known uses of glyphs representing tensor data was already presented over 25 years ago by de Leeuw and van Wijk [31]. To analyze not only the velocity of fluid flows, but also the local change, they use the vector field Jacobian matrix to produce what they call a flow probe. Curvature, shear, acceleration, torsion, and convergence are mapped to geometric primitives such as vectors, rings, and discs. Besides flow analysis, it was especially diffusion tensor imaging that became the most popular domain for glyph visualization. Therefore, the earliest glyphs for direct tensor visualization arose in the context of medical imaging. The eigenvectors and eigenvalues of a decomposed tensor are capable of uniquely representing all the information that is given by the tensor, thus a tensor is completely represented by a glyph that encodes these attributes. As eigenvectors of symmetric positive definite tensors, such as the diffusion tensor, always appear as mutually orthogonal vectors, they can easily be mapped to the axes of geometric objects. Due to the nature of eigenvectors being bidirectional, these shapes should also exhibit mirror symmetry with respect to the axes. Pierpaoli and Basser [145] used ellipsoid-shaped glyphs where the eigenvector orientations are aligned to the principal axes of the ellipsoid and then scaled according to the eigenvalue. Figure 4.1 illustrates this mapping. The longer part of the ellipsoid is pointing in the direction of the major eigenvector and can thus be used to indicate fiber directions when dealing with diffusion tensors in brain matter data. Placing uniformly distributed glyphs in slices of MRI data allows to analyze the overall fiber distribution but also get the full tensor information at the sampled locations. Laidlaw et al. [104] used normalized versions of such ellipsoids, whereas several other mappings to different shapes such as cubes [215], cylinders [204], and octahedra [197] exist. Rotations and minor changes of these shapes are however hard to distinguish, due to their orthogonality as shown by Parker et al. [141]. Superellipsoids [46], which belong to the class of superquadrics [9], are much better suited for that task [170]. A comparison of different glyphs with varying viewing directions is given in Figure 4.2. Similar to Jankun-Kelly and Mehta [82], Kindlmann

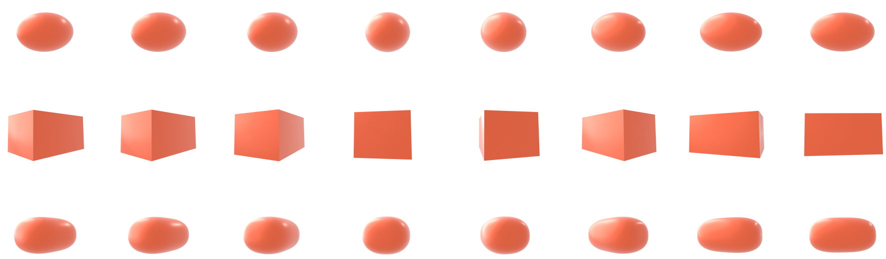


Figure 4.2: Eight different tensors based on rotating the eigenvectors visualized by three different types of tensor glyphs. First row: Ellipsoid glyphs. Second row: Box glyphs. Third row: Superquadric tensor glyphs [93], [165].

proposes to base the glyph construction upon the superellipsoid. These glyphs are known as the Superquadric tensor glyphs [93]. Here, too, the axes of the shape are aligned to the eigenvector directions and scaled based on the eigenvalues. Zhang et al. [209] use a combination of such glyphs to indicate the difference between two tensors. These techniques are however all limited to the symmetric positive-definite case. For general symmetric tensors, such as stress tensors, eigenvectors are still orthogonal, the eigenvalues can however also include negative real numbers. Schultz and Kindlmann [165] made use of the fact, that the elegant mathematical description of superquadrics allows to easily alter the shape smoothly and even go from concave to convex shapes to extend the glyphs. Concave shapes indicate eigenvalues with opposite signs and color was added to indicate, whether the eigenvalue corresponding to the eigenvector was positive or negative. Further, in the case of degenerate tensors, where eigenvalues repeat and an infinite number of eigenvectors are possible, the superquadric forms a circle or a sphere, accordingly, indicating this ambiguity. Their construction is based on a set of carefully chosen guidelines which is discussed in greater detail in Section 5.1. An overview of glyphs specifically designed for diffusion tensor imaging as well as design guidelines is given by Ropinski et al. [151], [152].

There exist several approaches specially tailored to visualizing stress tensors [97], [98], [100] in mechanical engineering. The glyph known as Mohr's circle [30] is a two-dimensional representation based on the maximum shear stresses that occur on any arbitrary cutting plane at a given position, which is helpful for failure analysis of materials. Regardless of the mechanical background, it is capable of visualizing any symmetric tensor. Haber glyphs [58] consist of an ellipsoid that is shaped by the minor and medium eigenvectors and their corresponding eigenvalues, as well as a rod, that highlights the major eigenvector. The advantages, as well as limitations of these and further glyphs such as the Reynolds tensor glyphs [123] and HWY glyphs [64], are discussed by Kriz et al. [101].

Different from the examples listed above, vector field gradients do not necessarily appear as symmetric matrices which poses a big challenge in tensor glyph visualization. Relying on eigendecomposition is difficult, as non-orthogonal eigenvectors can occur and eigenvalues can be complex. Similar to the aforementioned flow probe [31], several works use glyphs to offer additional information within vector fields, such as adding derived tensor information to vector glyphs [96] or placing them along streamlines [110], [162]. Globus et al. [51] propose glyphs placed only at critical points that consist of crossing lines representing the eigenvectors. Similarly, Theisel et al. [57] propose *icons* at such locations. The shape distinguishes between tensors with real eigenvalues and complex ones, while the color indicates eigenvalue signs. The works from Loeffelmann et al. [117] and Wiebel et al. [203] further show the behavior of the surrounding vector field at these locations without restricting themselves to the linear approximation. The first approach places streamlets close to critical points, the latter introduces a spherical surface around the critical point, which is then advected by the flow. Auer et al. [6] place sketch-like representations based on tensor information to indicate trends in the flow. Palke et al. [140] as well as Chen et al. [26] present hybrid visualizations: they use hyperstreamlines for the tensor field visualization and place ellipse glyphs in the complex domain. For the continuity of the lines as well as the construction of the glyphs, they make use of the dual eigenvectors or pseudo-eigenvectors [210], [212]. Seltzer and Kindlmann [169] recently presented glyphs for general – symmetric and asymmetric – second-order 2D tensors, extending the superquadrics with textures that indicate rotational flow behavior. Further approaches to visualizing non-symmetric tensors are discussed by Kratz et al. [99].

The static display of dynamic or time-dependent behavior of data with the help of glyphs is still a challenging task with a lot of open research questions. When the data dimensionality is low enough, the temporal development of quantities can be mapped to free geometric axes of glyphs [182]. Tensors representing unsteady or time-dependent flow fields as well as their gradients are only sparsely covered by glyph-based approaches, as opposed to steady flow fields. The aforementioned glyphs by Wiebel et al. [203] can indicate the temporal development of particles close to critical points. The flow radar glyph by Hlawatsch et al. [72] maps flow properties onto a radial glyph. Flow direction is mapped onto angles while time is mapped onto the radius. Similarly, the pathline glyphs by Hlawatsch et al. [73] represent flow behavior in a static glyph by downscaling pathlines and placing them at certain locations in the flow field thus combining both glyph and streamline visualization. Concerning glyphs for time-dependent tensor fields themselves, Benger and Hege [15] propose the use of tensor splats as a possible glyph for time-dependent symmetric positive-definite tensors. They propose using animation or superimpose transparent ellipsoid tensor glyphs to show temporal development.

Few works are known to encode uncertainty within flow fields, such as the flow-radar glyph by [72] again or Wittenbrink et al. [205]. Jones introduced the *cones of uncertainty* [90] to encode the local variance of eigenvector estimates by a confidence visualization. Schultz et al. [166] provide a new glyph that aims at a more detailed understanding of the distribution of fiber variability from DT-MRI. The construction is based on decomposing the probability measure into a main direction and a residual, combining both into what they call the *HiFiVE* glyph. Jiao et al. [89] compute what they call SIP glyphs from orientation distribution functions and volume rendering of a large number of samples from the distribution. The volume data that results from superimposing these renderings can visualize the shape inclusion probability (SIP, see [118]) gives one possible geometric interpretation of uncertainty. Basser et al. [11], [13] suggest two visualizations: First, they propose visualizing the covariance matrix independently of the mean tensor. Their second visualization proposes showing the mean tensor and its variance as three isosurfaces representing mean and standard deviation. Abbasloo et al. [1] provide another solution: the main idea is to consider a spectral analysis and to visualize the tensor perturbations in the directions of the 6 eigenvectors of the covariance matrix. Then the whole covariance matrix is covered by the simultaneous observation of 6 tensor deformations. It is further discussed in Section 7.6. In a similar approach for ensembles of tensor data, Zhang et al. [208] provide a framework to combine several visualizations to gain a general overview of the whole ensemble, as well as detailed information of distinct tensor properties. They divide uncertainty into three independent parts (scaling, shape, and rotation) and encode each with one variance number.

4.2 CHALLENGES IN TENSOR GLYPH DESIGN

As we have seen, tensor glyphs are a good and often used choice for direct visualization of tensor data and their construction has led to a variety of publications within the last years. They can provide an overview of the entire tensor field but also give a detailed description of the tensor information at a given location. But we have also seen that most of these glyphs are exclusively constructed for symmetric tensors. And among these, the majority is further devoted to positive definite tensors. As general tensors, including asymmetric and indefinite matrices, appear frequently in applications such as computational fluid dynamics and mechanical engineering, as shown in Table 3.1, suitable glyph techniques represent a gap in the literature. The mapping of such tensor data onto geometric glyph properties is a challenging task and needs a thorough mathematical investigation. The same applies to the visualization of time-dependent, as well as uncertain tensor data. In the following chapters, we want to address and analyze these challenges. Based on these observations, we propose possible solutions that allow the use of tensor glyphs to represent such data.

There are, of course, further discussions that arise besides the construction of tensor glyphs, such as placement strategies [76], [95], [153], [195], rendering of glyphs [207] or tensor field interpolation [2], [45], [136], [137], [161], [198] that are important research topics in tensor glyph visualization. We do however focus on the construction of new tensor glyphs for general, time-dependent, and uncertain tensor fields.

5

GLYPHS FOR GENERAL SECOND-ORDER 2D AND 3D TENSORS

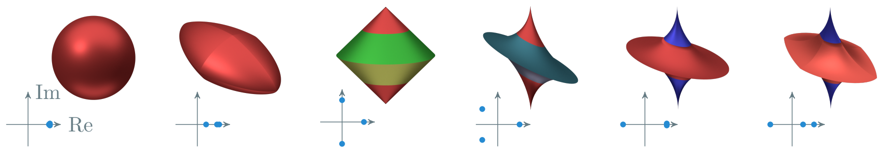


Figure 5.1: Glyphs for different general second-order 3D tensors. Their eigenvalues are plotted in the complex plane.

This chapter is based on the publication:

T. Gerrits, C. Rössl, and H. Theisel
Glyphs for General Second-Order 2D and 3D Tensors
IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Scientific Visualization 2016), 2017

The previous chapter has shown how tensor glyphs can be used as a useful tool for direct visualization of tensor data. Reiterating existing literature did however also indicate, that most techniques are focused on diffusion or stress tensors, which are symmetric. In this chapter, we search for glyphs for *general* second-order tensors in 2D and 3D, i. e., tensors that are *not* necessarily symmetric. Such tensors appear in a variety of applications, e. g., in computational fluid dynamics and flow visualization as the Jacobian matrix of velocity fields. Therefore, introducing new glyphs can extend the set of available tools for investigating and understanding such data.

In this chapter, general second-order tensors are considered as general matrices in $\mathbb{R}^{2\times 2}$ or $\mathbb{R}^{3\times 3}$ without constraints like symmetry, i. e., the space of all such tensors has 4 or 9 dimensions, respectively. The design space of possible tensor glyphs for such tensors is extremely large, which makes coming up with a new glyph a challenging task. It makes sense to reduce the design space in search of a suitable glyph by introducing a list of requirements or *wishes* as we call them, which ensures that the glyph construction follows some specific rules or exhibits wanted behavior in special conditions. This will ultimately limit the possible design choices such as shape, color, or behavior of the glyph. Such properties must be chosen in a way, that the glyph is capable of transporting meaningful and reliable information to the observer, based on mathematical considerations of tensor properties. Therefore, the first section of this chapter is dedicated to finding such rules and formulating them in a mathematically rigorous manner.

5.1 A WISH LIST FOR TENSOR GLYPH DESIGN

Listing desired behavior for tensor glyphs has been applied in previous works on tensor glyphs, such as in the construction of the superquadric tensor glyphs by Schultz and Kindlmann [165]. Incorporating the mathematic structures of the special case of symmetric tensors, the rules they state in their work demand suitable glyphs to follow rules such as *symmetry preservation*, *invariance under scaling*, *disambiguity*, and *continuity*. These are useful restrictions that can also be applied to the construction of general and possibly non-symmetric tensors. They further demand a property called *invariance under eigenplane projections*. This makes sense for symmetric tensors, as their eigenvalues are always real and orthogonal. For general tensors, however, this property is not well-defined as the eigenplanes may not be perpendicular or not even real at all. A straightforward approach to solve this issue could be to decompose a non-symmetric tensor into a sum of the symmetric part and the antisymmetric part as shown in Section 3.1.1 and visualized accordingly. This way, the information about the eigenvalues and eigenvectors of the original tensor $\mathbf{T}$ is lost, and no direct encoding of this information is possible. We do however think, that using the original tensor information is a desired property. Based on these observations, we define our wish list as follows:

Let $\mathbf{T}$ be a general 2D or 3D tensor represented by a – not necessarily symmetric – matrix and let $G(\mathbf{T})$ be its corresponding glyph.

(a) *Invariance under isometric domain transformation.*
Let $\mathbf{Q}$ denote an isometric map, e. g., rotation or reflection, as an orthogonal matrix. Then the domain transformation of the tensor should result in the same transformation of the glyph.

$$G(\mathbf{Q}\,\mathbf{T}\,\mathbf{Q}^{\mathrm{T}}) \;=\; \mathbf{Q}\,G(\mathbf{T})\;. \tag{5.1}$$

(b) *Scaling invariance.*
A uniform scaling of the tensor should result in the same scaling of the glyph, i. e., for any $s > 0 \in \mathbb{R}$

$$G(s\,\mathbf{T}) = s\,G(\mathbf{T})\;.$$

(c) *Direct encoding of real eigenvalues and eigenvectors.*
If $\mathbf{T}$ has real eigenvalues and eigenvectors, they should be directly visible

in the glyph. This is justified by the fact that the eigenvalues and eigenvectors capture all information of the tensor. If they are all real-valued, they provide geometric information that is suitable for direct visualization: direction of eigenvectors and magnitude of eigenvalues. This is the case for symmetric tensors with orthogonal eigenvectors and also for a class of non-symmetric tensors with real-valued but non-orthogonal eigenvectors.

(d) *Uniqueness.*
A tensor $\mathbf{T}$ should result in a unique glyph. We also demand the reverse: a glyph should have a unique tensor. For any two tensors $\mathbf{T}_1, \mathbf{T}_2$ we demand

$$\mathbf{T}_1 \neq \mathbf{T}_2 \quad \Rightarrow \quad G(\mathbf{T}_1) \neq G(\mathbf{T}_2)\ . \tag{5.2}$$

We introduce *weak uniqueness* that requires Equation (5.2) only for tensors of full rank, i. e., two lower-rank tensors may "share" the same glyph.

(e) *Continuity.*
Continuous changes of the tensor must result in continuous changes of the glyph. In particular, there should be no instantaneous change of the glyph appearance for a small change of the tensor. This includes the transition from positive to negative determinant, from orthogonal to non-orthogonal real eigenvectors, from distinct to multiple eigenvalues, and from real to complex eigenvalues.

$$\mathbf{T}_1 \approx \mathbf{T}_2 \quad \Rightarrow \quad G(\mathbf{T}_1) \approx G(\mathbf{T}_2)\ .$$

As stated before, these wishes are similar to the guidelines for the superquadric tensor glyphs. To address the problem that can arise from dealing with non-real and non-orthogonal eigenvectors for eigenplane projection, we replace their requirement for invariance under eigenplane projection by the new property (c) which is stronger in some sense: it generalizes explicitly to the case of real but non-orthogonal eigenvectors, and it makes no assertion for the complex case.

We consider the properties stated as a guideline for glyph design and further claim, that glyph constructions must follow them to be a suitable visualization tool for general tensors. This further allows us to discuss the existing work which was listed in Section 4.1 in terms of how they relate to these properties and if they are suitable to be used for general tensors.

Similar to the superquadric tensor glyphs, most existing techniques have been explicitly developed for symmetric tensors, such that their construction demands real eigenvalues and eigenvectors. For instance, the ellipsoid glyphs by Globus et al. [51] neither consider complex eigenvalues nor do they provide uniqueness for different eigenvalue signs thus

lacking requirement (d). The glyph proposed by De Leeuw and van Wijk [31] contains derived values from the Jacobian matrix, however, the eigenvalues are not directly encoded, and thus lacking (c). Mohr's circles [30] visualize only eigenvalues and are therefore lacking invariance to domain rotations (a) while the Haber glyphs [58] are not continuous (lacking (e)) when the eigenvectors are not well-defined. The icon glyphs for visualization of critical points in flows by Theisel et al. [57] lack uniqueness (c). The hybrid visualization of Palke et al. [140] does not cover the complete space of 2D tensors and they do not provide an extension to 3D. Similarly, the visualization by Zhang et al. [210] incorporates discontinuities whenever eigenvectors are not well defined due to equal eigenvalues. Further approaches to visualizing non-symmetric tensors such as [35], [99], [100] also do not provide glyphs for the complete space of non-symmetric tensors. All these do therefore not provide a suitable starting point for a general glyph construction.
The recent work by Seltzer and Kindlmann [169] introduces glyphs for general second-order tensors in 2D, but it is not able to meet the continuity and rotation invariance requirements. An in-detail discussion of the most relevant contributions as well as a comparison to our work is given at the end of this chapter in Section 5.6.4.

The following table gives an overview of existing relevant work on 3D general tensor glyphs. Existing techniques are evaluated with respect to satisfying conditions (a)-(e) from Section 5.1. In addition, the column (f) indicates if the technique is general, i. e., not restricted to symmetric tensors.

method / satisfies	(a)	(b)	(c)	(d)	(e)	(f)
Kindlmann and Schultz [93], [165]	✓	✓	✓	✓	✓	✗
Seltzer and Kindlmann [169]	✗	✓	✓	✓	✗	✓
tensor decomposition	✓	✓	✗	✓	✓	✓
Globus et al. [51]	✓	✓	✓	✗	✓	✓
de Leeuw and van Wijk [31]	✓	✓	✗	✗	✓	✓
Theisel et al. [57]	✓	✓	✓	✗	✗	✓
Mohr's circle [30]	✗	✗	✓	✗	✓	✓
Haber glyph [58]	✓	✓	✓	✗	✗	✓

After revisiting these works, we state, that we are not aware of existing works, that present a glyph for general tensors – neither 2D nor 3D – that fulfill all conditions (a)-(e).

5.2 GLYPHS FOR 2D TENSORS

In this section, we develop a new construction technique for glyphs for 2D tensors that meets all requirements (a)-(e) postulated in Section 5.1. As the Jacobian matrix of a vector field is a prominent representative of a general tensor, throughout the following sections, we will often make use of the interpretation of a tensor as such. This might further support

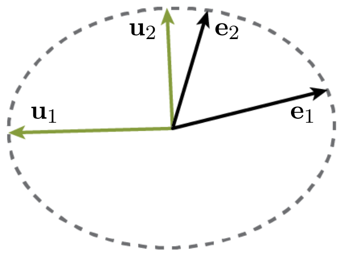


Figure 5.2: The characteristic ellipse interpolates both the endpoints of the scaled real eigenvector $\mathbf{e}_1$, $\mathbf{e}_2$ as well as the scaled left singular vectors $\mathbf{u}_1$, $\mathbf{u}_2$ of a decomposed tensor.

easier comprehensibility of design choices. The application is however not limited to Jacobian matrices.

5.2.1 *Preliminary Consideration*

We start with a general observation that strongly influences the glyph design:

Proposition 5.2.1. *It is impossible to use only shape for defining a glyph that satisfies conditions (a)-(e). At least one more continuous value has to be encoded in a channel different from shape.*

In short, a circle is the only shape that is capable of representing the eigenvector directions of tensors with repeated eigenvalues, due to the fact, that there exist infinite possible eigenvectors as stated in Section 3.2. There are, however, multiple distinct tensors that can have repeated eigenvalues. A proof can be found in Appendix A and a similar proof is given by [169].

In the following, we separate the construction into designing shape and mapping color.

5.2.2 *Shape*

Given is a tensor $\mathbf{T} \in \mathbb{R}^{2\times 2}$ with eigenvalues $\lambda_{1,2}$. The basic geometric primitive for the construction of the shape of the associated glyph is the *characteristic ellipse* of $\mathbf{T}$ that we define as the point set that satisfies

$$\mathbf{x}^{\mathrm{T}}\left(\mathbf{T}\,\mathbf{T}^{\mathrm{T}}\right)^{-1}\mathbf{x} \;=\; 1\;. \tag{5.3}$$

This implicit curve is an ellipse that interpolates for $\lambda_{1,2} \in \mathbb{R}$ the eigenvectors scaled by eigenvalues, i. e., the columns of $\pm\mathbf{X}\Lambda$. Note that in general, the eigenvectors $\mathbf{X}$ are *not* orthogonal. For *complex* eigenvalues, the orthogonal axes of the ellipse are spanned by the left

singular vectors $\mathbf{U}$. This relationship is also illustrated in Figure 5.2. A proof of these properties can be found in Appendix B.
We parameterize the implicitly defined characteristic ellipse as a piecewise rational quadratic Bézier curve [41], where each piece is an arc with the scaled eigenvectors (or left singular vectors if $\lambda_i \in \mathbb{C}$) as endpoints. Each piece is defined by three control points $\mathbf{b}_i$ and weights w_i. Due to endpoint interpolation, $\mathbf{b}_0, \mathbf{b}_2$ are given by the scaled eigenvectors with standard weights $w_0 = w_2 = 1$. The center control point is

$$\mathbf{b}_1 \;=\; \omega\,(\mathbf{b}_0 + \mathbf{b}_2) \quad \text{with weight } w_1 \;=\; \cos{}^{\alpha}\!/_{2}\;,$$

with

$$\omega \;=\; \frac{1}{1+\cos\alpha} \tag{5.4}$$

where α is the angle enclosed by $\mathbf{b}_0, \mathbf{0}, \mathbf{b}_2$ (or two eigenvectors, respectively). This results in two rational pieces for the smaller and larger enclosed angle, and the remaining two pieces can be determined from symmetry. Figure 5.3a shows an example, and Appendix B describes a detailed construction of the parametrization.

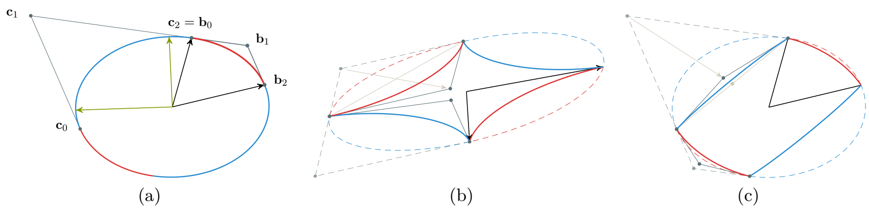


Figure 5.3: (a) Characteristic ellipse of a non-symmetric tensor with real eigenvalues. The black arrows denote the eigenvectors scaled by the eigenvalues ($\mathbf{X}\Lambda$), and the green arrows show the orthogonal left singular vectors scaled by singular values ($\mathbf{U}\Sigma$). The four arcs are parametrized as rational quadratic Bézier curves shown in red ( ) and blue ( ). The control polygons $\mathbf{b}_i, \mathbf{c}_i$ are shown for two arcs (in gray). Joining rational pieces smoothly at $\mathbf{b}_0$ requires that $\mathbf{c}_1, \mathbf{c}_2 = \mathbf{b}_0$, and $\mathbf{b}_1$ are colinear. (b) For a saddle configuration, the center control points, e. g., $\mathbf{b}_1$ of the original characteristic ellipse configuration (dashed) are moved "beyond" $\frac{1}{2}(\mathbf{b}_0 + \mathbf{b}_2)$ ( ) towards the origin to obtain a concave shape (solid). (c) For a positive definite tensor, we introduce sharp bends to indicate the directions of the eigenvectors. The center control points are moved closer towards $\frac{1}{2}(\mathbf{b}_0 + \mathbf{b}_2)$ ( ).

We consider the mapping of $\mathbf{T}$ into the (γ, r)-plane as presented in Section 3.2.4. Further, we make use of the statements by Theisel and Weinkauf [181] about the derived properties of eigenvalues and determinant of the given tensor, such that we can now use γ and r to distinguish between different possible cases. Each case determines a *modification* of the characteristic ellipse, and each modification is defined in a way that guarantees requirements (a)-(d) and in particular (e), the continuous transition between the different cases. We emphasize this by explicitly reviewing the transitions as special cases. All modifications are described for one rational piece $(\mathbf{b}_i, w_i)$, $i = 0, 1, 2$, with $\alpha = \angle(\mathbf{b}_0, \mathbf{0}, \mathbf{b}_2)$. The same modification is applied equally to all pieces.

Let $r^\star = \frac{1}{1+\sin^2\gamma}$.

CASE $0 \le r < \frac{1}{2}$. $\mathbf{T}$ has real eigenvalues and a negative determinant $\lambda_1\lambda_2 < 0$. The glyph for this "saddle" configuration should be a *concave* shape that conveys the directions and magnitude of eigenvectors and eigenvalues, which can be interpreted as "inflow" and "outflow" if the tensor is a Jacobian matrix of a vector field. Each arc of the characteristic ellipse is modified such that the center control point is moved to $\frac{1}{2}\,|\cos\alpha|\;(\mathbf{b}_0 + \mathbf{b}_2)$. This yields a concave shape as $\mathbf{b}_1$ is closer to the origin $\mathbf{0}$ than $\frac{1}{2}(\mathbf{b}_0 + \mathbf{b}_2)$. Figure 5.3b shows an example.

CASE $r = \frac{1}{2}$. At this transition, $\operatorname{rank}(\mathbf{T}) = 1$, and one of the two real eigenvalues vanishes, i. e., $\lambda_1\lambda_2 = 0$. As a consequence, the characteristic ellipse "degenerates" to a line segment. The behavior is continuous as for $h \to 0$, both $r = \frac{1}{2} \pm h$ result in the same glyph.

CASE $\frac{1}{2} < r < r^\star$. This can be seen as the simplest case of a positive definite $(\lambda_{1,2} > 0)$ or negative definite $(\lambda_{1,2} < 0)$ tensor $\mathbf{T}$. The glyph should be *convex* and clearly indicate the directions and magnitude of eigenvectors and eigenvalues. We modify the smooth characteristic ellipse such that there are sharp bends in these directions, i. e., the curve should be only C^0-continuous at the endpoints $\pm\mathbf{X}\Lambda$ of the elliptic arcs. Note that $\mathbf{X}$ is orthogonal only for symmetric $\mathbf{T}$. For general tensors, the indicated directions do *not* coincide with the principal axes of the characteristic ellipse.
For each arc, we move the center control point towards $\frac{1}{2}(\mathbf{b}_0 + \mathbf{b}_2)$ as follows:
Define the ratio $\tau = \lambda_1/\lambda_2$ of eigenvalues and $\tau^\star = \min\{\tau, 1/\tau\} \in [0, 1]$, and let

$$\omega^\star \;=\; (1 - |\sin\gamma|)\,\Big((1-\tau^\star)\,\tfrac{1}{2} \,+\, \tau^\star\,\omega\Big) \,+\, |\sin\gamma|\;\omega\;. \tag{5.5}$$

The new position of the center control point is $\mathbf{b}_1 = \omega^\star\,(\mathbf{b}_0 + \mathbf{b}_2)$. In order to ensure that sharp bends develop more rapidly near the transitions $r = \frac{1}{2}$ and $r = r^\star$, we suggest applying an additional transfer function and to replace $\tau^\star$ in the above formula by $f(\tau^\star)$ with $f(t) = 4\,(t - \frac{1}{2})^3 + \frac{1}{2}$. Figure 5.3c shows an example.
The amount of "sharpening" is maximal at the transition $r = \frac{1}{2}$ – think of the "degenerated stick" as a "diamond" – and it gradually fades out towards the smooth characteristic ellipse as $r \to r^\star$ and the smaller angle enclosed by two eigenvectors vanishes.

CASE $r = r^\star$. At this transition, the eigenvalues are equal $\lambda_1 = \lambda_2$, and the eigenvectors are parallel, i. e., the smaller of the enclosed angles is zero. The shape of the glyph is the characteristic ellipse *without modification*.

CASE $r^\star < r \le 1$. $\mathbf{T}$ has complex eigenvalues $\lambda_1 = \overline{\lambda_2}$. The principal axes of the characteristic ellipse are spanned by the left singular vectors of $\mathbf{T}$. *No modification* is applied.

CASE $r = 1$. In the limit, $\mathbf{T}$ is a rotation matrix $\mathbf{Q}(\gamma)$ as with $r = 1$ its singular values must be equal, $\sigma_1 = \sigma_2$. This follows from the definition of the (γ, r) parametrization using the constrained-norm tensor $\mathbf{T}$ with $||\mathbf{T}||_2 = \sigma_1(\mathbf{T}) = 1$: with $r = 1$ we have also $\sigma_2(\mathbf{T}) = 1$. Then $\mathbf{T}$ is invariant to domain rotation, and also the glyph must be invariant to rotation. The shape of the glyph is just the characteristic ellipse, which is a circle for $\sigma_1 = \sigma_2$. Figure 5.7 shows different shapes in the (γ, r)-plane.

5.2.3 *Color*

We use color to encode the angle γ. For $r \ge \frac{1}{2}$, each glyph is filled with a single, "flat" color. Any continuous color map of the circle is possible. We use the color map shown in Figure 5.4 that maps positive and negative definite ($\gamma = 0$ and $\gamma = \pi$) tensors to red and blue tones, and 90-degree rotations ($\gamma = \pi/2$ and $\gamma = 3\pi/2$) to yellow and green tones, respectively.
The "saddle" case $r < \frac{1}{2}$ is treated differently because we want to distinguish the directions of inflow and outflow and thus use two colors. First, we have to make sure that one color gives a continuous transition along the circle $r = 1/2$ in the (γ, r)-space. Let $\lambda_1 \le 0 \le \lambda_2$, and the corresponding eigenvectors $\mathbf{X}_{\cdot 1}$ and $\mathbf{X}_{\cdot 2}$ oriented such that $\det(\mathbf{X}) < 0$, further let $\alpha \in [0, \pi]$ be the angle enclosed by $\mathbf{X}_{\cdot 1}$ and $\mathbf{X}_{\cdot 2}$. Then we get two γ-values

$$\gamma_1 \;=\; \frac{\pi}{2} + \alpha \quad \text{and} \quad \gamma_2 \;=\; \frac{\pi}{2} - \alpha \;,$$

that are color-coded as described above. The inner circles in Figure 5.4 show the two colors for the respective points in (γ, r)-space.
In addition, a partition of the glyph's geometry is required.
Let $\mathbf{f}_i(t) : [0, 1] \to \mathbb{R}^2$, $i = 0, \dots, 3$, denote the four rational pieces that define the boundary of the glyph. With a circular shift of the "global" parametrization by $1/2$ we obtain

$$\mathbf{g}_i(t) \;=\; \begin{cases} \mathbf{f}_i(t + 1/2) & \text{for } t \le 1/2 \\ \mathbf{f}_{i+1 \bmod 4}(t - 1/2) & \text{for } t > 1/2 \end{cases} \;, \quad i = 0, \dots, 3 \;,$$

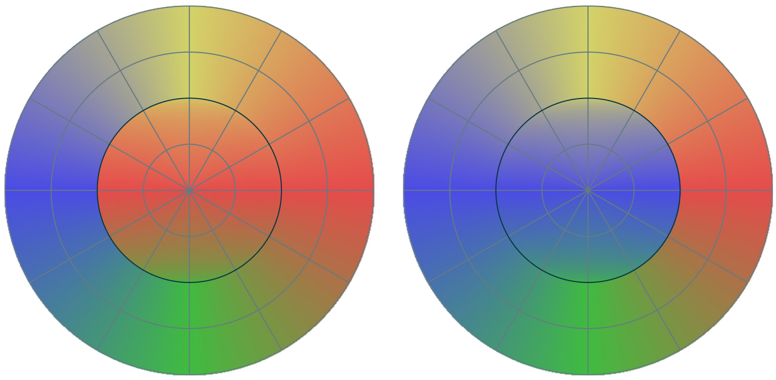


Figure 5.4: The glyph's color is determined by the angle γ. The figure shows the color map used in this chapter as the outer band ($r \geq 1/2$) of the color wheel with a superimposed polar coordinate system (γ, r). Darker circles indicate $r = 1/2$. The color is constant for $1/2 \leq r \leq 1$. Two colors are required for saddles ($r < 1/2$) to indicate "inflow" and "outflow" directions. We use complementary colors shown in the inner circles that also depend on the angle enclosed by the (real) eigenvectors.

such that each image of $\mathbf{g}_i(t) : [0,1] \to \mathbb{R}^2$ consists of two half-arcs that indicate the direction of an eigenvector. This partitions the glyph symmetrically into four patches. Figure 5.7 shows different shapes and colors in the (γ, r)-plane.

5.3 GLYPHS FOR 3D TENSORS

We utilize the 2D construction and in particular the (γ, r) parametrization as much as possible for the 3D setting for general tensors $\mathbf{T} \in \mathbb{R}^{3\times3}$. This leads to two cases depending on the configuration of eigenvectors and eigenvalues. In the first case, there is one distinct pair of eigenvalues either having the opposite sign to the third one or being complex conjugates. In this case, the distinct pair of eigenvectors (for real eigenvalues) or left singular vectors (for complex eigenvalues) span a uniquely defined *base plane*. In the second case, no distinctive eigenvectors exist because all three eigenvalues are real and positive (or all three are real and negative). For both cases, we discuss first shape and then color of the glyph.

5.3.1 *Case 1: A Well-Defined Base Plane Exists.*

SHAPE. A well-defined base plane exists, if either all eigenvalues are real and one differs in sign, or if two eigenvalues are complex conjugates. In the first case ("saddle"), the plane is spanned by the eigenvectors corresponding to the two eigenvalues with the same sign. In the second

case ("swirling"), the plane is spanned by the left singular vectors. It is straightforward, to extend the condition to eigenvalues equal to zero, but it requires suitable solutions to known problems that can arise from dealing with near-zero values, such as rendering issues.
We construct the shape from eight triangular patches. One of them is shown in Figure 5.5a. Suppose that $\mathbf{x}_1$ and $\mathbf{x}_2$ are the scaled eigenvectors spanning the base plane, and $\mathbf{x}_3$ is the remaining scaled eigenvector. Then the intersection of the desired patch with the base plane is exactly the solution of the 2D glyph in the base plane. Furthermore, we use the information from the base plane as well as the ratio between the associated eigenvalues and the remaining eigenvalue to determine the shape outside the base plane. We describe the patch as a rational bi-quadratic patch $\mathbf{f}(u, v)$ with a degeneracy, i. e., an undefined normal, at $\mathbf{x}_3$, where the parameters w_{12}, μ_{12}, and ν_{12} (see Figure 5.5b) are determined as follows. w_{12} and μ_{12} are chosen such that we get the 2D glyph in the base plane: μ_{12} is obtained similarly to Equation (5.4), and w_{12} is obtained similarly to Equation (5.5). The remaining ν_{12} determines the global convexity/concavity of the shape and is chosen as

$$\lambda_{12} \;=\; \frac{\lambda_1 + \lambda_2}{2}\,, \quad \nu_{12} \;=\; \frac{1}{2} + \frac{1}{2}\,\mathrm{sgn}(\lambda_{12}\lambda_3)\left(|\cos\gamma|\,\frac{|2\lambda_{12}\lambda_3|}{\lambda_{12}^2 + \lambda_3^2}\right)^n$$

where γ is from the (γ, r) parametrization of the projection of $\mathbf{T}$ into the base plane and the exponent n controls "sharpness" of the shape near discontinuities. This patch construction is repeated eight times for each combination of $\pm\mathbf{x}_i$ as patch corners to obtain the entire shape. The resulting patches have the following properties.

- $\mathbf{f}_u(0, v) \;=\; h(v)\,\mathbf{f}_u(0, 0)$ for a certain function $h(v)$. This means that the partial derivative of $\mathbf{f}$ w.r.t. u does not change direction along the boundary curve from $\mathbf{x}_1$ to $\mathbf{x}_3$. (A similar statement holds for $\mathbf{f}_u(1, v)$.) As a consequence, in case of the characteristic ellipse as base shape, adjacent patches are G^1-*continuous* along the junction curves that are not in the base plane.

- If $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ build an orthonormal system, $\mathbf{f}(u, v)$ is the octant of a sphere (see [144]).

Details of the patch construction are reviewed in Appendix C.

COLOR. The glyph consists of the following colors: Close to the base plane, we color code the γ value of the 2D case in the projection into the base plane. Note that depending on the side of the base plane, this requires two different colors. If from one side the value γ is encoded, the view from the other side must encode $-\gamma$. This becomes clear when interpreting the information as rotational behavior which changes from clockwise to anticlockwise and vice versa. For the coloring of the regions close to $\mathbf{x}_3$ a binary choice is sufficient: red for $\lambda_3 > 0$, blue else. It remains to define at what v-value the "hard" transition between the two colors takes place. We choose $v = \frac{\sigma_1}{\sigma_1+\lambda_3}$, where $\sigma_1 = \sigma_1(\mathbf{T}_P) = ||\mathbf{T}_P||_2$ is the spectral norm of the projection of $\mathbf{T}$ into the base plane. This makes sure that for $\lambda_3 \to 0$, the color of the whole 3D shape converges to the color of the 2D glyph in the base plane.

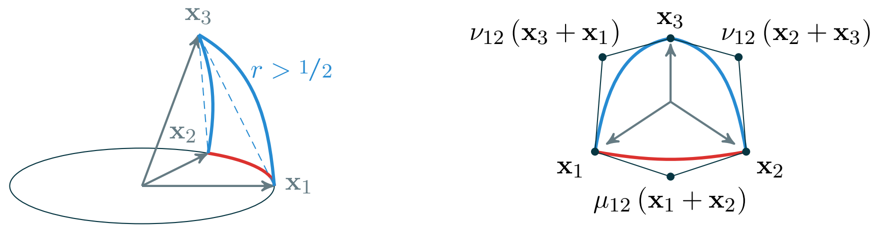
5.3.2 *Case 2: There is No Unique Base Plane.*

SHAPE. In this case, all pairs of eigenvectors can be chosen equally to span a base plane. This means that depending on the particular choice, we have three different patches for each octant of the shape. We propose to blend patches using a weighted average. For this, two problems have to be solved: (1.) The three patches are given in different parametrization, which prohibits a direct blending. (2.) The blend weights must be chosen to ensure a smooth transition of the shape between case 1 and case 2. To solve the first problem, we apply a non-standard reparametrization of the patch from u, v-coordinates to barycentric coordinates $\beta_1 + \beta_2 + \beta_3 = 1$, which is detailed in Appendix C.
To address the second problem, the blend weights for patch evaluation are chosen as

$$\begin{aligned} W_1 &= |(\lambda_3 - \lambda_1)(\lambda_3 - \lambda_2)\lambda_1\lambda_2| \\ W_2 &= |(\lambda_1 - \lambda_2)(\lambda_1 - \lambda_3)\lambda_2\lambda_3| \\ W_3 &= |(\lambda_2 - \lambda_3)(\lambda_2 - \lambda_1)\lambda_3\lambda_1|\,. \end{aligned} \tag{5.6}$$

This ensures that if, e. g., λ_1 and λ_2 get close to each other, W_1 and W_2 get close to 0, meaning that we have the desired smooth transition between case 1 and case 2. The same desired transition takes place for $\lambda_3 \to 0$. If all eigenvalues are identical, the patches are also identical, and all weights would equally evaluate to zero and would lead to a degenerate patch. In this case, all weights are set to an equal, nonzero value.

COLOR. For color, we use the same weighted average as for shape. For every barycentric coordinate $\beta_1, \beta_2, \beta_3$ we have a γ value for each patch (either the γ-value of the base plane or $\gamma = 0$ or $\gamma = \pi$ towards the patch corner away from the base plane). The three γ-values are averaged by the same blend weights W_1, W_2, W_3. Note that this way, one final patch can consist of up to eight different colors. However, in practice, they can hardly be distinguished: all of them are rather red (for outflow) or rather blue (for inflow). This is desired because in this case, all relevant information for uniqueness lies in the shape. We have to apply this seemingly complicated color mapping to ensure continuity between case 1 and case 2.



(a) (b)

Figure 5.5: (a) If a base plane exists, it defines an ellipse, and every rational piece (red) defines a surface patch together with the two other arcs (blue and gray), which use a standard weight $\sqrt{2}/2$. (b) Similar to 2D, the control points are determined as linear combinations of scaled eigenvectors $\mathbf{x}_i$. Note that without a well-defined base plane, all three possible patches are evaluated and “blended”.

Figure 5.6: Flat-shaped glyph without (left) and with eigenstick (right).

5.3.3 *Eigensticks*

Our 3D construction so far does not ensure uniqueness for the case $\mathrm{rank}(\mathbf{T}) = 2$. In this case, no information about the direction of the eigenvector corresponding to the zero eigenvalue is encoded in the glyph. This can be fixed by additionally rendering eigensticks, i. e., carefully scaled real eigenvectors of $\mathbf{T}$ as

$$\begin{aligned} &\pm\,(\lambda_1 - \lambda_2)\,(\lambda_1 - \lambda_3)\,\mathbf{X}_{\cdot 1}\ , \\ &\pm\,(\lambda_2 - \lambda_1)\,(\lambda_2 - \lambda_3)\,\mathbf{X}_{\cdot 2}\ , \\ &\pm\,(\lambda_3 - \lambda_1)\,(\lambda_3 - \lambda_2)\,\mathbf{X}_{\cdot 3}\ , \end{aligned}$$

with $\mathbf{X}_{\cdot i}$ denoting the i-th eigenvector. In most cases, eigensticks are rendered *inside* the shape and therefore not visible. Only in the case of flat shapes, i. e., there is one rather small eigenvalue, they become visible. Figure 5.6 shows an example. Note that eigensticks also fulfill the continuity condition for coinciding eigenvalues the corresponding eigensticks converge to the zero vector.

5.4 RESULTS

Figure 5.7 samples the (γ, r)-plane and shows the corresponding glyphs. Figure 5.8 shows (scaled) glyphs that visualize the Jacobian matrix of a 2D slice of the flow behind a square cylinder. The underlying flow field is visualized by a LIC texture with the superimposed glyphs. The glyphs

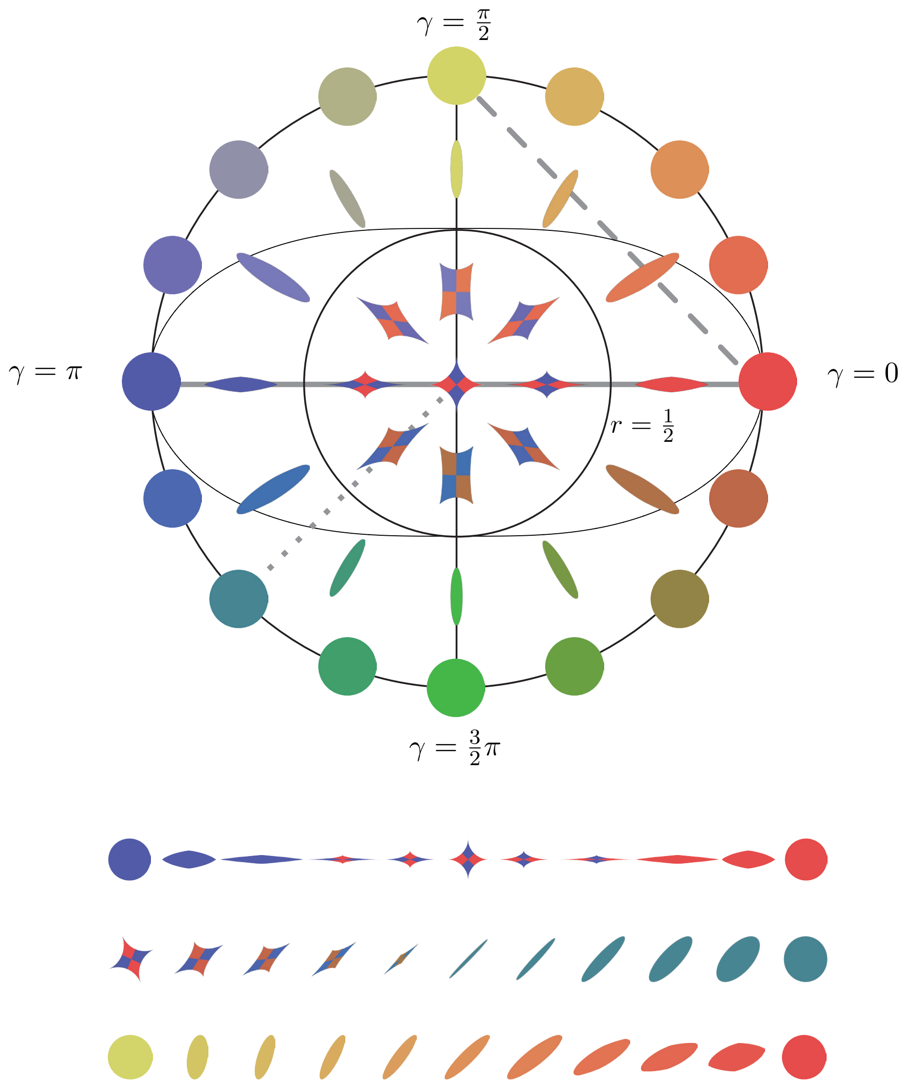


Figure 5.7: Sampling glyphs in the (γ, r)-plane. The circles on the top figure indicate $r = 1/2$ (zero determinant on the transition between saddle and definite) and $r = 1$ (equal eigenvalues), and the oval shape is the curve $r = r^\star$ (equal eigenvectors and eigenvalues on the transition between real and complex). The bottom figure shows sampled glyphs corresponding to the line segments. top row: solid line $\gamma \in \{0, \pi\}$; center row: dotted line $\gamma = 5\pi/4$; bottom row: dashed line $(\gamma, r) = (1-t)\,(0, 1) + t\,(\pi/2, 1)$.

vary significantly in scale, and there are regions of rapid transition between different glyphs. Two closeups zoom into interesting regions. Figure 5.1 shows a selection of different glyphs for 3D tensors together with the eigenvalues in the complex plane. More examples are given in Figure 5.9 which includes two cases with one of the three real eigenvectors getting close to zero: the corresponding glyphs have a flat shape (not seen from the chosen perspective), and the eigensticks become

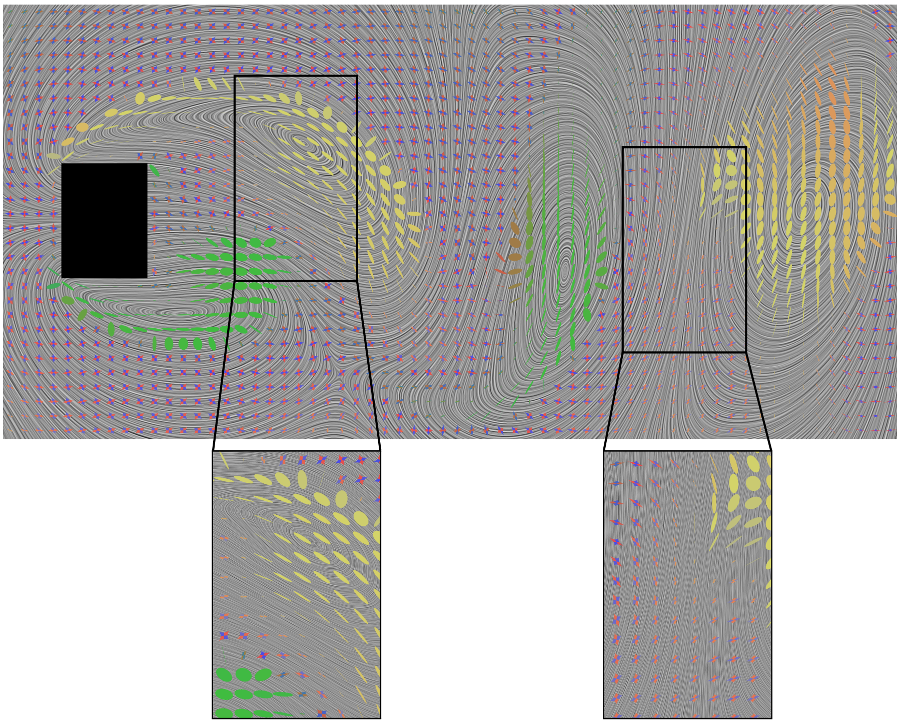


Figure 5.8: Glyphs visualizing the Jacobian matrix of the flow around a square cylinder with two closeups (rectangles). The underlying LIC image visualizes the flow.

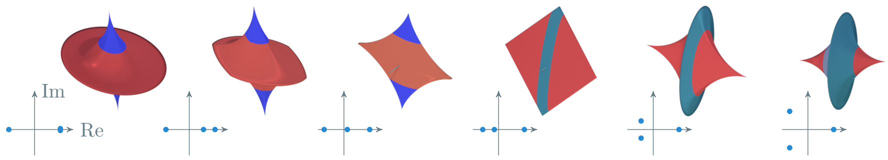


Figure 5.9: Selection of glyphs for 3D tensors with their eigenvalues plotted in the complex plane. Note that the two center glyphs feature visible eigensticks due to one of the three eigenvalues getting close to zero.

visible and convey the direction of the eigenvector associated with the near-zero eigenvalue.

Figure 5.10 shows 3D glyphs in the Jacobian matrix field of a flow that stems from a simulation of a Rayleigh-Bénard convection. The underlying flow field is illustrated with a few illuminated streamlines.

3D glyphs that are close to rank 2 appear flat, which makes it difficult to recognize the direction of the eigenvector corresponding to the near-zero eigenvalue. Eigensticks as described in the previous Section 5.3.3 remedy this deficiency. Figure 5.6 explicitly compares the same glyph with and without rendering an eigenstick that emphasizes the direction of the eigenvector.

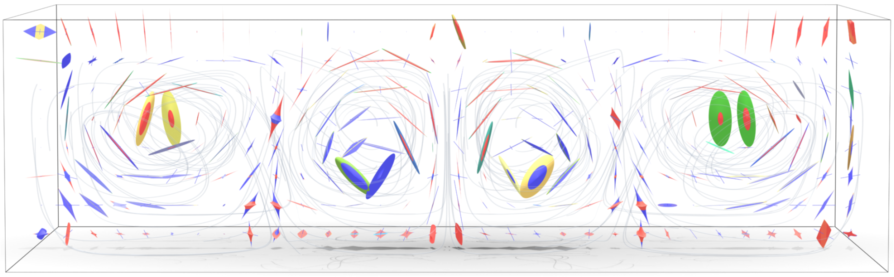


Figure 5.10: The glyphs depict the Jacobian matrices of a flow field from the simulation of a Rayleigh-Bénard convection. The illuminated streamlines give an impression of the flow.

5.5 HOW TO READ THE GLYPHS

The glyphs proposed in this chapter encode a significant amount of information. Moreover, the requirements (a)-(e) postulated in Section 5.1 constrain design choices and make the appearance of glyphs different from previous works. In this section, we show that despite this fact the new glyphs are easy to use by providing a few simple rules on how to read the proposed glyphs.

SHAPE. *Convex* shapes indicate that all real eigenvalues have the same sign or positive determinant, while *concave* shapes indicate different signs of eigenvalues.

An *ellipse* in 2D indicates that there are no unique real eigenvectors, for either case of complex or identical real eigenvalues.

An *ellipsoid*, i. e., a smooth shape without discontinuities, in 3D is only possible as a sphere, i. e., for three identical real eigenvalues (see Figure 5.1, left). Shape discontinuities at *sharp corners* in 2D and in addition *sharp edges* in 3D (see Figure 5.1, 2nd left) encode the direction of real eigenvectors for symmetric and asymmetric tensors without rotation.

Figure 5.11: The shape of the 2D tensor glyph indicates both, the relation of eigenvalue signs, as well as information, whether the eigenvalues are real-valued or complex.

COLOR. *Red* indicates positive real eigenvalues, i. e., an outflow, and *blue* indicates negative real eigenvalues, i. e., an inflow.

Yellow indicates counterclockwise swirling, and *green* indicates clockwise swirling.

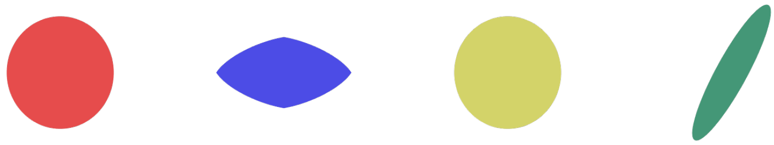


Figure 5.12: The color of the 2D tensor glyph indicates flow direction as well as swirling direction.

5.6 DISCUSSION

In this section, we verify that the proposed glyphs fulfill the requirements (a)-(e) postulated in Section 5.1, and we discuss our various design decisions.

5.6.1 *Fulfillment of Requirements*

(a) *Invariance under isometric domain transformations.* The property holds because all constructions are based on scaled eigenvectors $\mathbf{X}\,\Lambda$ of $\mathbf{T}$ (in the real case) or on scaled left singular vectors $\mathbf{U}\,\Sigma$ (in the complex case) and their linear combinations for determining control points. Both, $\mathbf{X}$ and $\mathbf{U}$ are invariant under isometric domain transformations $\mathbf{Q}$, as for $\mathbf{T} = \mathbf{X}\,\Lambda\,\mathbf{X}^{\mathrm{T}} = \mathbf{U}\,\Sigma\,\mathbf{V}^{\mathrm{T}}$, $\mathbf{Q}\,\mathbf{T}\,\mathbf{Q}^{\mathrm{T}}$ has eigenvectors $\mathbf{Q}\,\mathbf{X}$ and singular vectors $\mathbf{Q}\,\mathbf{U}$ and $\mathbf{Q}\,\mathbf{V}$ regardless of the dimension.

(b) *Scaling Invariance* follows directly from the construction.

(c) *Direct encoding of real eigenvalues and eigenvectors.* This follows also directly from the construction as all scaled eigenvectors appear visually as points or curves of C^0 continuity, i. e., as sharp bends due to discontinuities of the glyph's tangent field.

(d) *Uniqueness.* In 2D, it is sufficient to show that for every point in the (γ, r) phase space there is a different glyph, i. e., two such glyphs are not identical after rotation. For real eigenvalues, this follows directly from (c) because a tensor is uniquely defined by its eigenvalues and eigenvectors. In the complex case, the glyph is the characteristic ellipse. Since its aspect ratio uniquely encodes r and its color uniquely encodes γ, the glyph must be unique. Note that uniqueness also holds for 2D tensors of rank 1: they are encoded as sticks of a unique color that depends on γ.
In 3D, in the case of a well-defined base plane and non-zero third eigenvalue, the property follows from the 2D case in the base plane and the unique visibility of the third eigenvector. In the case of three all

positive (or all negative) real eigenvalues, all eigenvectors are visible, and the property follows from (c). With this, *weak* uniqueness in 3D is shown.
A special case in 3D is $\text{rank}(\mathbf{T}) = 2$, then the glyph is a 2D figure in the plane of the non-zero eigenvectors. In this case, the direction of the eigenvector corresponding to the 0 eigenvalue is not encoded in the glyph's shape and color. Hence, we do not have uniqueness. The solution is the optional addition of the eigensticks (Section 5.3.3) that ensures the uniqueness of this case.
Another special case in 3D is $\text{rank}(\mathbf{T}) = 1$, which gives a stick as glyph. To ensure uniqueness in this case, we additionally have to encode the eigenplane of the zero eigenvalues. Since the space of all possible eigenplanes is two-parametric, it cannot be encoded solely by color, hence, the glyph is not unique in this case. We decided not to introduce additional features to remedy this for the sake of avoiding further visual clutter only for a case of minor practical relevance.

(e) *Continuity.* To show continuity, we have to consider all cases of equal eigenvalues – and therefore undefined eigenvectors – as well as all transitions from real to complex eigenvalues. In 2D, equal real eigenvalues result in a circular glyph, whereas at the transition between real and complex eigenvalues, the glyph is always the characteristic ellipse.
In 3D, the following additional events have to be checked: The first case is the transition from a unique base plane to all positive (or all negative) real eigenvalues: the choice of the blend weights in Equation (5.6) ensures that in the transition event exactly one weight is non-zero, which gives continuity. The other case is the event of equal eigenvalues if all eigenvalues are real and positive/negative: here also only one weight is non-zero, giving continuity.

5.6.2 *A Critical Review of Requirements*

The set of requirements determines the glyph design and should be chosen carefully because every desired property constrains the space of admissible glyphs. In the extreme case, this space may even be empty for contradicting requirements. More generally, the imposed conditions may be "too strong" in the sense that the admissible glyphs do not feature an intuitive interpretation anymore. This, however, is crucial in any application of glyphs. In this case, one option is to remove or to relax certain conditions, e. g., demanding only partial fulfillment like "almost everywhere".
We strongly advocate for meeting *all* requirements (a)-(e) for the general design of tensor glyphs: these properties constitute a standard choice that was established by Schultz and Kindlmann [165]. Moreover, Kindlmann and Scheidegger [94] proposed three general visualization *design*

principles: representation invariance, unambiguous data representation, and visual-data correspondence. The conditions (a), (b), (d), (e) formally implement these principles and guarantee their fulfillment. Missing in this list is condition (c): the direct visual encoding of real eigenvalues and eigenvectors, if present, seems to be nontrivial. At the same time, this information is of such importance for the characterization of a tensor in essentially every application that condition (c) appears as an obvious choice for glyph design. As we have shown in Section 3.2.3, for Jacobian matrices of vector fields for example, the eigenvalues and eigenvectors give a direct classification of the flow around critical points. For sure, different applications may require emphasis on different properties of the tensor and therefore relax certain conditions in favor of a more intuitive interpretation. However, this holds for the particular application or task and comes at the cost of losing possibly important parts of the information. For a *generic* glyph design that is not a priory tailored toward a specific application, the theoretically sound and hence "safe" option is to implement *all* conditions (a)-(e). This is not necessarily complicated: in Section 5.5 we give a few simple rules on how to read the proposed glyphs. This gives evidence that it is feasible to learn to read the new glyphs such that they can be used in a variety of applications.
Finally, this contribution focuses on the generic construction of a design space for 2D and 3D glyphs that meet all of the mentioned requirements, which has not been done before. It does neither formally measure the intuitiveness of glyphs nor does it systematically explore this space with the goal of finding glyphs within the constraints that are in some sense "optimal" for a specific application. Possible directions are discussed below.

5.6.3 *Design Decisions*

The space of all possible solutions to the construction of glyphs fulfilling (a)-(e) is huge. In this chapter, we give only one sample (and to the best of our knowledge the first one). This raises the question of how to further explore the space.

Color schemes. We use a rather straightforward color scheme to encode one continuous value, in 2D this is γ. We do so to ensure comparability with similar approaches [57], [165] that proposed similar colors. Other and in particular more perception-oriented color maps are possible. We remark finally that [57], [165] only use few discrete colors whereas Proposition 5.2.1 states that we have to use a continuous color wheel.

Encoding of eigenvectors. Our approach is based on an encoding of the eigenvectors as discontinuities in the glyph's shape. Even though the human visual system reacts rather sensitive to discontinuities in a shaded scene, perception can be increased by ridge enhancement methods. This makes illustrative techniques interesting candidates for rendering our shapes. The main challenge here is to ensure a smooth cease and disappearance of the enhancements in the case of equal eigenvalues.

Perceptional considerations. Small changes in the tensors should lead to equally small changes in the perception of the glyphs. In order to proceed in this direction, we first need a metric in the space of all tensors, and it seems not at all obvious which one to choose. Only then a study on the perception of the glyphs is meaningful.

Similarity to existing special cases. Another design goal for glyphs could be the similarity to well-established glyphs in special cases, e. g., to Schultz' and Kindlmann's superquadrics [165] for symmetric tensors. Our current approach disregards this goal.

5.6.4 *Comparison with Existing Techniques*

Parallel to this work, Seltzer and Kindlmann developed and published an approach with the same goal: glyphs for general second-order tensors [169]. Their construction applies only to the 2D case, seemingly without a straightforward extension to 3D, and we give a comparison with our 2D construction.
Seltzer and Kindlmann use a parametrization of the space of 2D tensors that is similar to the (γ, r)-plane: they span the space in terms of three parameters D, S, R, similar to the decompositions described in Section 3.1.1, that generate isotropic, traceless symmetric, and antisymmetric parts of the tensor. They similarly factor domain rotations and constrain the Frobenius norm $||\mathbf{T}||_F = \sqrt{D^2 + S^2 + R^2} = 1$ (instead of the spectral norm as for (γ, r), which, after projection, leads to a planar parameter space defined by barycentric coordinates. Given a tensor $\mathbf{T} := \mathbf{T}/||\mathbf{T}||_F$ with $||\mathbf{T}||_F = 1$, the following relations hold for D, S, R in [169] and γ, r:

$$\tan\gamma = \frac{R}{D} \qquad \text{and} \qquad r = 1 - S^2 \ .$$

The first equation can be easily verified as follows. In 2D, the polar decomposition of a matrix $\mathbf{T} = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ can be expressed in closed form: find a rotation matrix, parametrized by angle γ, that makes $\mathbf{T}$ symmetric. This leads to $\tan\gamma = \frac{c-b}{a+d}$ and with Eq. (9) in [169] to the above equation. The second equation is already given as Eq. (37) in [169]: in $\det(\mathbf{T}) = 1/2 - S^2$ substitute $r = 1/2 - \det(\mathbf{T})$ from Section 3.1.1.
For the (D, S, R)-parametrization, Seltzer and Kindlmann give a tangible interpretation: $D = 0$ refers to traceless tensors, for $S = 0$ tensors are rotations (without shear or nonuniform scaling) and hence exhibit

rotational symmetry, and $R = 0$ refers to symmetric tensors without rotation. In the (γ, r)-space this refers to $\gamma \in \{\pi/2, 3\pi/2\}$ or $r = 0$ and any γ for traceless tensors, $r = 1$ for rotational symmetry, and $\gamma \in \{0, \pi\}$ for symmetric tensors. Finally, the loci of $\det(\mathbf{T}) = 0$ are $S^2 = 1/2$ and $r = 1/2$, respectively. Note that the projection of D, S, R to planar barycentric coordinates is nonlinear, and the projection of the algebraic curve $S^2 = 1/2$ is a conic section rather than a line.
Seltzer and Kindlmann also show – similar to this contribution – that shape alone is not sufficient to ensure a unique encoding. Different from this work, they propose a texture on the glyph. This gives an intuitive encoding of the rotation/swirling at the price that continuity and rotation invariance are not completely fulfilled anymore. Furthermore, [169] demand as an ultimate design goal that for the special case of symmetric tensors the well-established superquadric glyphs [93] appear as a solution. Our approach is different, and the glyphs differ significantly from superquadric glyphs: rather than treating symmetry as a special case, our construction always encodes the direction of real eigenvectors (and smoothly changes to left singular vectors in the complex case). [169] lose this property for asymmetric tensors with real eigenvectors.

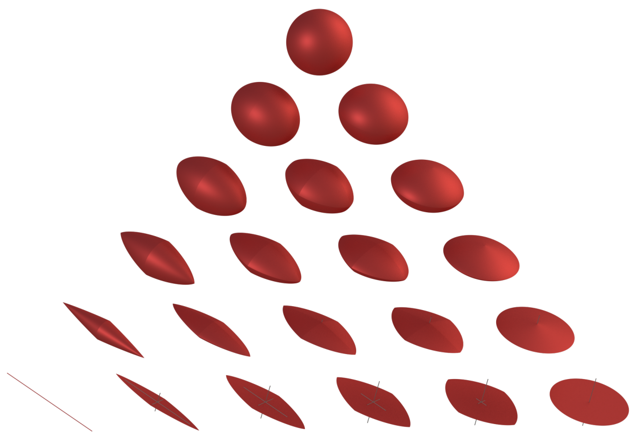


Figure 5.13: The proposed glyphs for symmetric positive definite 3D tensors in a similar arrangement as in [93]: in contrast to Kindlmann's superquadric glyphs, the shape discontinuities prevent view-dependent visual ambiguities.

We finally compare this work to the superquadric glyphs by Kindlmann [93]. This refers to 2D and 3D glyphs, but only for the special case of symmetric positive definite tensors. Figure 5.13 shows our glyphs in a similar arrangement as in Figure 7 in [93]. Note that our solution is different but shares an important design goal: due to the discontinuities in the shape, there is no visual ambiguity regardless of which perspective the glyph is rendered/viewed. This means that two of the proposed 3D glyphs can *always* be distinguished in the projection to 2D images. In contrast, Kindlmann points out that for his superquadric tensors the 2D projections are not always unique, e. g., in case the 3D glyph is an ellipsoid.

5.7 LIMITATIONS AND FUTURE WORK

The main theoretical limitation of our approach is the non-uniqueness of 3D tensors of rank 1. We decided not to fix this shortcoming because we consider this case as having only low relevance in practice. A direct road-map for future research comes from the discussion in Section 5.6.3. This includes in particular illustrative or stylized rendering to emphasize relevant information such that visual perception is taken into account *and* all postulated requirements are fulfilled.

6

GLYPHS FOR SPACE-TIME JACOBIANS OF TIME-DEPENDENT VECTOR FIELDS

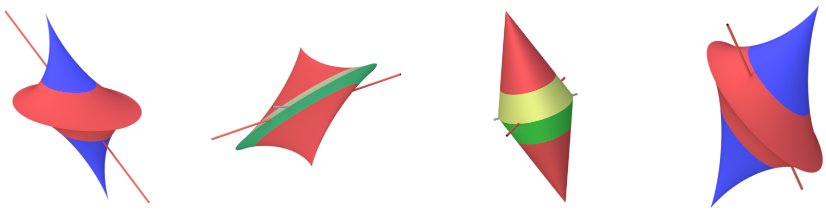


Figure 6.1: Glyphs representing both spatial and temporal derivatives encoded in different time-dependent 3D Jacobian matrices of vector fields.

This chapter is based on the publication:

T. Gerrits, C. Rössl, and H. Theisel
Glyphs for Space-Time Jacobians of Time-Dependent Vector Fields
Journal of WSCG, 2017

With the newly proposed glyph technique from the previous chapter, there now exists a glyph that is capable of representing, amongst other tensors, the 2D or 3D Jacobian matrix of velocity fields at a given location. The construction is, however, only based on the information of time-independent vector fields, where no temporal information is encoded within the Jacobian matrix.

In this chapter, we discuss the following problem:
given an n-dimensional ($n = 2, 3$) time-dependent vector field $\mathbf{v}(\mathbf{x}, t)$, can we construct an n-dimensional glyph that encodes the *space-time* Jacobian matrix of $\mathbf{v}$, i. e., all first-order derivatives, both spatial and temporal? This means that we have to find a glyph representation for a $(n+1) \times (n+1)$ Jacobian tensor. While this is straightforward for $n = 2$ (ending up in visualizing a 3×3 matrix), it is challenging for $n = 3$ because this requires the 3D visualization of a 4×4 space-time Jacobian tensor. This specific matrix, however, which is *not* a general 4D second-order tensor, has some properties that allow for a 3D glyph visualization that seamlessly extends existing 3D tensor glyphs.

6.1 VISUALIZATION OF TIME-DEPENDENT FLOW

We coarsely classify visualization techniques for flow data in different groups: topology-based techniques [146], dense flow visualization [105], geometric flow visualization [120], and glyph-based approaches. While several works exist for the first two groups, only a few known approaches belong to the latter which we covered in Section 4.1. Techniques such as the flow radar glyphs [72] or the pathline glyphs [73] focus on representing time-dependent flow by showing the temporal development of either a location within a vector field or a particle transported within a time-dependent vector field over time. Most of the information encoded by the vector field Jacobian matrix at the glyph locations is ignored.

There exists, to the best of our knowledge, no glyph technique that is capable of representing a time-dependent Jacobian tensor.

6.2 EXTENSION FOR TIME-DEPENDENT TENSOR GLYPHS

As the glyphs introduced in the previous Chapter 5 can visualize any given 2D or 3D Jacobian matrix as long as the feature is steady, we use them as a construction foundation to build upon. We further want to make use of the same construction principles defined in Section 5.1. This implies, that these glyphs need to be altered or extended in some way, such that they are able to represent the additional information encoded in time-dependent Jacobian matrices while still following the properties defined in the wish list. To find a suitable extension, we need to analyze the differences between the steady and unsteady case and discuss, how a suitable mapping of the additional data to the same dimension as the glyphs we build upon can be found. First, we do this for Jacobian matrices of 2D unsteady vector fields and present an addition to the glyph that keeps the requirements from Section 5.1 intact and later extend the idea to the 3D case.

6.2.1 *Time-Dependent 2D Tensor Glyphs*

To make the extension clearer, it makes sense to revisit some of the basic information on vector fields. The introduction on vector fields in Section 2.1.3 has already stated, that a steady 2D flow at a location $\mathbf{x} = (x, y)$ can be described by $\mathbf{v}(\mathbf{x})$ and the local behavior around it by its Jacobian matrix $\mathbf{T}$. As this is the spatial gradient of the vector field, we hence call it the *spatial Jacobian*. Using eigendecomposition, we obtain the eigenvalues λ_1, λ_2, and the corresponding eigenvectors

$\mathbf{e}_1, \mathbf{e}_2$. An unsteady flow, however, has time as an additional dimension. We define

$$\tilde{\mathbf{v}}(\mathbf{x}, t) \;=\; \begin{pmatrix} u(\mathbf{x}, t) \\ v(\mathbf{x}, t) \\ 1 \end{pmatrix}$$

to be a time-dependent 2D vector field and the corresponding *space-time Jacobian* as

$$\tilde{\mathbf{T}}(\mathbf{x}, t) \;=\; \begin{pmatrix} u_x & u_y & u_t \\ v_x & v_y & v_t \\ 0 & 0 & 0 \end{pmatrix}$$

with eigenvalues $\lambda_1,\ \lambda_2, 0$. The associated eigenvectors are

$$\begin{pmatrix} \mathbf{e}_1 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{e}_2 \\ 0 \end{pmatrix} \;, \quad \tilde{\mathbf{f}} \;\; \text{where} \;\; \tilde{\mathbf{f}} := \begin{pmatrix} a \\ b \\ c \end{pmatrix} \;.$$

This Jacobian matrix must not be mistaken with a general 3×3 matrix. Due to the fact, that the last row of $\tilde{\mathbf{T}}$ is entirely made up of zeros, two of the eigenvectors are simply the eigenvectors $\mathbf{e}_1$ and $\mathbf{e}_2$ of $\mathbf{T}$ with an additional zero as their component in the new dimension. The additional eigenvector $\tilde{\mathbf{f}}$ with its components $a, b, c \in \mathbb{R}$ is associated with the zero eigenvalue and fully encodes the temporal derivative, included in $\tilde{\mathbf{T}}$. We use the notation $\tilde{\mathbf{f}}$ in reference to the works of Theisel, Seidel and Weinkauf [179], [199] who show that this vector can be used to track features over time in a derived vector field known as the *feature flow field*. We can therefore use $\mathbf{e}_1$ and $\mathbf{e}_2$ to build the corresponding 2D glyph $G(\mathbf{T})$, which we call spatial glyph, using the glyph construction from Chapter 5, and use only $\tilde{\mathbf{f}}$ as the information that needs to be added to it. As we want our new glyph $G(\tilde{\mathbf{T}})$ to be of the same dimension as the spatial glyph, we require a projection of $\tilde{\mathbf{f}} \in \mathbb{R}^3$ to a vector $\mathbf{g} \in \mathbb{R}^2$ on the visualization plane. To define an appropriate and unique projection, we demand

1. Given two eigenvectors $\tilde{\mathbf{f}}_1$, $\tilde{\mathbf{f}}_2$ corresponding to the temporal derivative and the projected 2D vectors $\mathbf{g}_1$, $\mathbf{g}_2$, if $\tilde{\mathbf{f}}_1$ and $\tilde{\mathbf{f}}_2$ are parallel, $\mathbf{g}_1$ and $\mathbf{g}_2$ have to be identical.

$$\tilde{\mathbf{f}}_1 \parallel \tilde{\mathbf{f}}_2 \quad \Rightarrow \quad \mathbf{g}_1 \;=\; \mathbf{g}_2 \;.$$

2. If $\tilde{\mathbf{f}}_1$ and $\tilde{\mathbf{f}}_2$ are not parallel, $\mathbf{g}_1$ and $\mathbf{g}_1$ must never be identical

$$\tilde{\mathbf{f}}_1 \nparallel \tilde{\mathbf{f}}_2 \quad \Rightarrow \quad \mathbf{g}_1 \neq \mathbf{g}_2 \;.$$

3. When the field is stationary, $\mathbf{g}$ should not be visible. In this case, the resulting glyph is identical to the glyph based on the stationary Jacobian matrix $G(\mathbf{T}) = G(\tilde{\mathbf{T}})$. Therefore, the corresponding vector $\mathbf{g}$ should be the zero vector. Additionally, the transition from unstable to stable should result in a smooth transition to the zero vector.

$$\tilde{\mathbf{f}} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \mathbf{g} \rightarrow \begin{pmatrix} 0 \\ 0 \end{pmatrix} .$$

We propose the following projection that satisfies the above requirements:

$$\mathbf{g} \;=\; \frac{1}{||\tilde{\mathbf{f}}||} \begin{pmatrix} a \\ b \end{pmatrix} ,$$

where a and b are the first two components of $\tilde{\mathbf{f}}$.
This vector can then be visualized by adding two identical sticks to the glyph, one representing $\mathbf{g}$, the other $-\mathbf{g}$ and both given a length of $s||\mathbf{g}||$, where $s > 0 \in \mathbb{R}$ can be used as a constant scaling factor. Both orientations of $\mathbf{g}$ need to be rendered due to the fact, that $\tilde{\mathbf{f}}$ is an eigenvector of $\tilde{\mathbf{T}}$, and therefore must satisfy the same symmetry properties. To reduce visual clutter, we move these sticks along the lines, given by their directions to the locations where the line intersects the boundary of the underlying spatial glyph's shape. Figure 6.2a illustrates this construction.

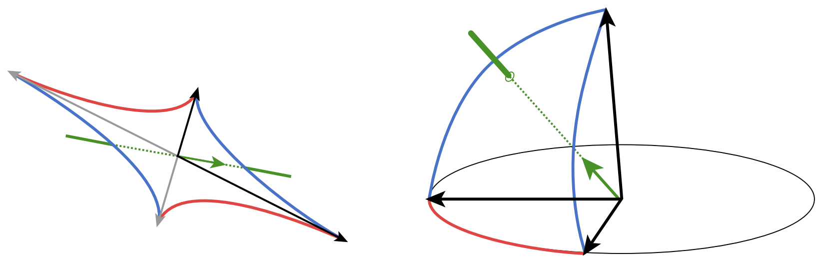
6.2.2 *Time-Dependent 3D Tensor Glyphs*

Finding new glyphs representing 3D time-dependent Jacobian matrices is analogous to the 2D case. The additional temporal information encoded by the Jacobian matrix $\tilde{\mathbf{T}} \in \mathbb{R}^{4\times 4}$ is given by the additional eigenvector $\tilde{\mathbf{f}} \in \mathbb{R}^4$, where $\tilde{\mathbf{f}} = (a\ b\ c\ d)^{\mathrm{T}}$.

We propose projecting $\tilde{\mathbf{f}}$ onto the 3D vector $\mathbf{g} \in \mathbb{R}^3$ by using

$$\mathbf{g} \;=\; \frac{1}{||\tilde{\mathbf{f}}||} \begin{pmatrix} a \\ b \\ c \end{pmatrix} ,$$

and visualizing it by adding tubes to the spatial glyph, created by using eigenvalues and eigenvectors of $\mathbf{T}$. These tubes are then moved along their vector directions as well until they reach the points, where their corresponding elongations would intersect the glyph patch. In that way, they are always visible and not rendered within the glyph, unless the



(a) (b)

Figure 6.2: Adding sticks to the base glyphs allows the representation of time-dependent Jacobians. The eigenvector $\tilde{\mathbf{f}}$ corresponding to the time derivation is projected onto the vector $\mathbf{g}$ ( ). A line cast from the center of the glyph in forward and backward direction of $\mathbf{g}$ intersects the boundary of the glyph exactly twice unless $\mathbf{g}$ is the zero vector. Sticks ( ) representing $\mathbf{g}$ and $\mathbf{-g}$ are then added at those locations. (a) Construction of the 2D time-dependent Jacobian tensor glyph. (b) Construction of one patch and one stick representing the eigenvalue $\tilde{\mathbf{f}}$ of a 3D time-dependent Jacobian tensor glyph.

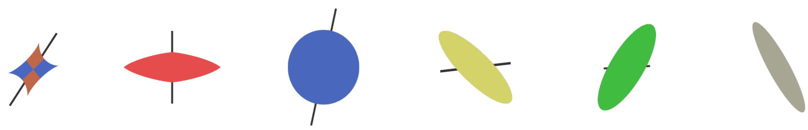


Figure 6.3: Glyphs representing different 2D Jacobian matrices. The underlying features are less temporally stable to the left and more stable to the right. The stick has vanished in the last glyph, which shows that this feature is completely stable.

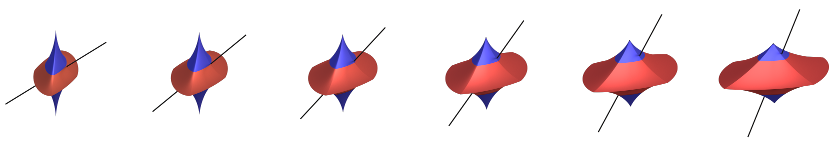


Figure 6.4: 3D Glyphs representing Jacobians sampled at the same location in an unsteady flow field over time. The glyph as well as the stick representing the temporal derivative change smoothly over time.

temporal derivative is zero, in which case the new vector becomes the zero vector as demanded.

Because both new constructions, 2D and 3D alike, follow the presented set of rules, they are suitable for creating unique tensor glyphs for any given 2D or 3D Jacobian matrix, unsteady or steady, and also follows all of the glyph design requirements that were discussed earlier.

6.3 RESULTS

First, we visualize a collection of 2D time-dependent Jacobian matrices. Figure 6.3 shows a selection of 2D glyphs for different classes of Jacobian tensors. The temporal derivative encoded within the Jacobian matrix decreases from left to right. The spatial derivatives within the Jacobians matrix result in glyphs based upon real-valued as well as complex-valued eigenvalues and eigenvectors. The additional sticks are always moved to the boundary of the spatial glyph, for any given shape.

In Figure 6.6 and Figure 6.7, our construction is applied to build glyphs representing the Jacobian matrices at sampled locations of one time slice of a 2D unsteady flow behind a cylinder. A variety of different flow behaviors is present within the flow as can be seen by the variety of different spatial glyphs. As time proceeds, alternating vortices, as illustrated by the glyphs using yellow and green colors, are created and transported to the right. Therefore, Jacobian matrices at several locations comprise strong temporal derivatives, indicated by the additional sticks being clearly visible. Locations where the derivative vanishes are analogously indicated by small or even no sticks. While Figure 6.6 shows the glyphs superimposed to an additional line integral convolution (LIC) texture of the underlying flow field, Figure 6.7 displays the same glyphs in front of a different LIC texture which represents the feature flow field at the selected time. The projected additional eigenvectors $\mathbf{f}$ are therefore tangent to this field at the given location. Two closeups for each field show zoomed-in areas of interest inside those fields.

To further highlight the sticks, the same domain is rendered without any supporting background LIC texture in Figure 6.8.

Figure 6.4 presents the new 3D glyphs, as it shows sampled time steps of the development of a 3D Jacobian matrix at the same location evolved over time. The underlying changing Jacobian matrix is computed by linear interpolation of two time slices of the vector field. The spatial glyph changes independent of the temporal derivative, whereas the added tubes change direction due to the projected vector and location due to the change of glyph shape, as seen in Figure 6.5.

In Figure 6.9, the glyphs are used to visualize regularly sampled locations in the 3D unsteady Jacobian matrix field of an analytical flow with one moving center in the middle of the field. The whole flow is steadily moved to the right over time. To illustrate the underlying flow field, a set of illuminated streamlines is added. Here, too, the new glyphs show a variety of different underlying Jacobian matrices, including constructions based upon tensors with complex and real-valued eigenvalues.

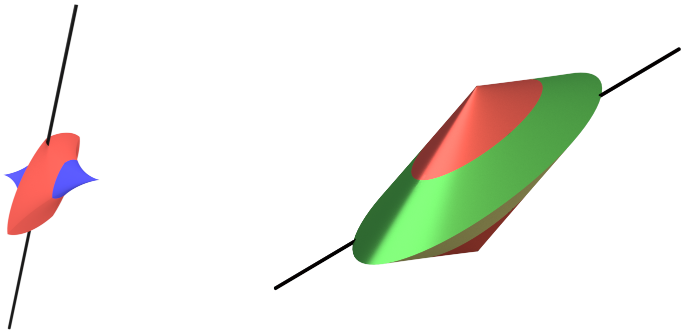


Figure 6.5: Sticks visualizing the temporal derivative are always moved along their directions to the boundary of the glyph, so they are always visible, no matter whether the spatial glyph is small (left) or large (right).

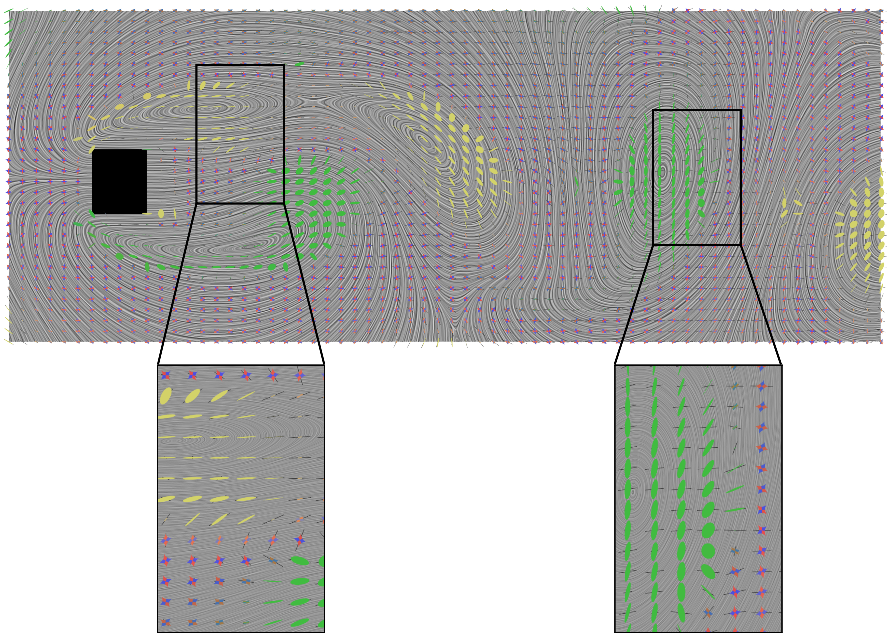


Figure 6.6: Glyphs visualizing the Jacobian matrix of the flow around a square cylinder with two closeups. The underlying LIC image visualizes the fluid flow.

6.4 DISCUSSION

In this section, we evaluate, whether the new glyph construction is capable of fulfilling all requirements of our wish list. We further discuss how the glyphs can be read and possible alterations to the design.

By basing our construction on the newly introduced glyphs from the previous chapter, we ensure that the spatial glyph will always follow all

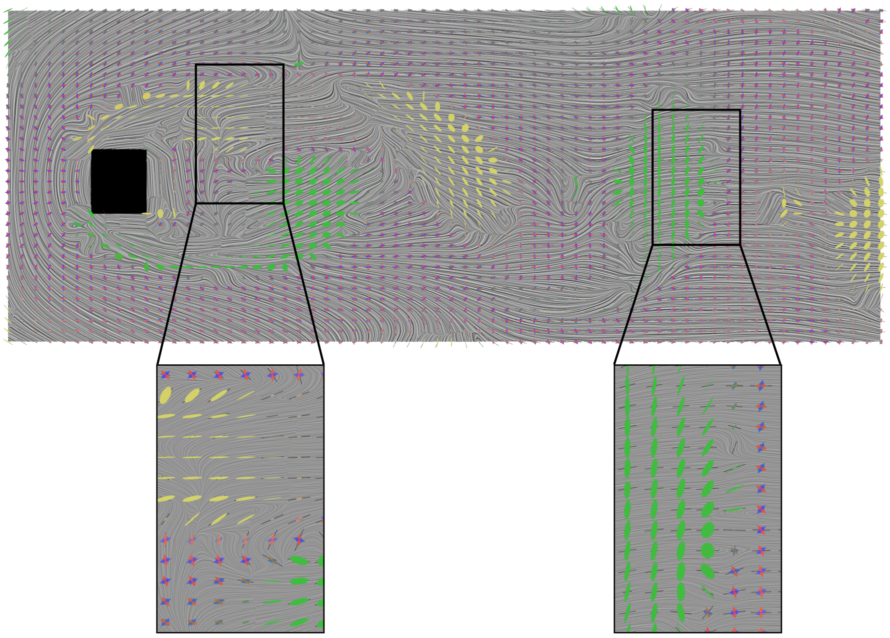


Figure 6.7: Glyphs visualizing the Jacobian matrix of the flow around a square cylinder with two closeups. The underlying LIC image visualizes the feature flow.

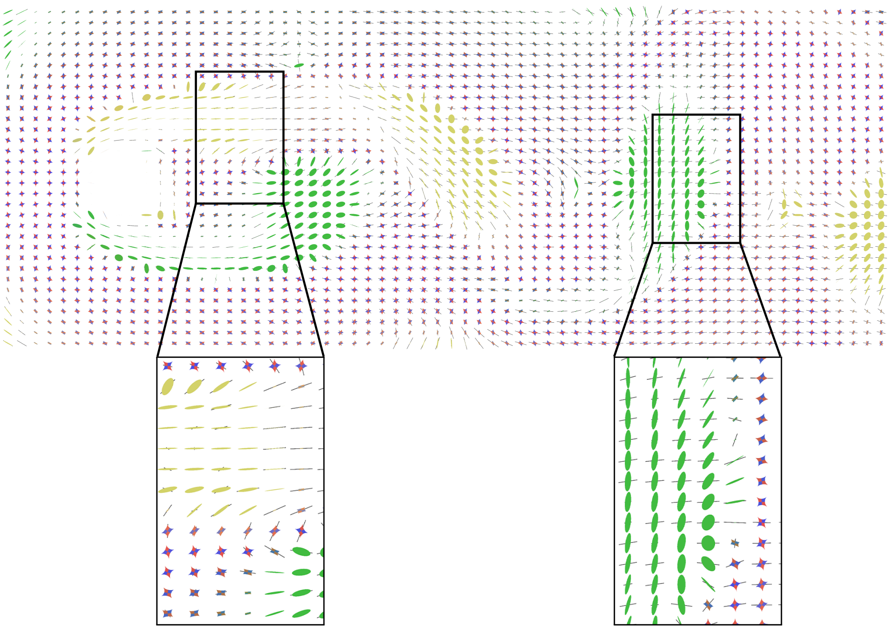


Figure 6.8: Glyphs visualizing the Jacobian matrix of the flow around a square cylinder without additional supporting LIC textures with two closeups.

postulated properties. Finding a mapping onto the same visualization plane and moving it on the shape boundaries, encoding the additional temporal information has not changed the spatial glyph such that

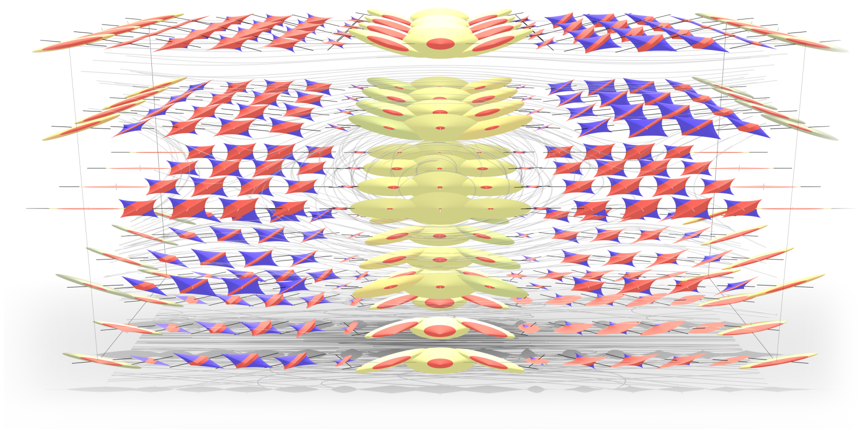


Figure 6.9: 3D time-dependent flow with a moving vortex in the center. All features are moving to the right over time. The newly constructed glyphs are rendered at sampled locations at one time slice. Illuminated streamlines illustrate the underlying flow.

the expressiveness is in no way impaired. Comparing Figure 6.6 and Figure 5.8 from the previous chapter, the new 2D glyphs allow to see the same structures of the underlying flow field as the steady or spatial 2D glyphs before. Therefore, rotational sections as well as laminar flows can still be easily determined in the given example. Now, however, they also convey the temporal information. The same statement holds for the 3D case as displayed in Figure 6.9. We have to show, that this is also true for our proposed extension.

By rendering the sticks or tubes added to the glyphs in both directions, they follow eigenvector symmetry and are always visible, even in the 3D case, as long as the underlying Jacobian matrix is unsteady. As the eigenvalue corresponding to $\tilde{\mathbf{f}}$ is always zero, the temporal information is completely encoded by $\tilde{\mathbf{f}}$. Due to our proposed mapping, the added sticks also follow invariance to domain scaling and rotation and the mapping is also unique. Further, a change of vector direction or vector length of the temporal information is also smooth, which is displayed in Figure 6.4, where a time series of the glyphs at the same location over time is shown.

As $\tilde{\mathbf{f}}$ also encodes the feature flow field, which allows tracking critical points over time (see, e. g., [179]), the glyphs offer an insight into the progression of flow. We can predict glyphs with longer sticks to be moving or changing over time, while shorter or no sticks indicate that a feature is quite stable. This is shown in Figure 6.7, where the flow around the cylinder has vortices going along one axis to the right, which is also indicated by the sticks of the glyphs in those areas pointing in this direction. We can also remove all supporting LIC textures as in Figure 6.8 and still understand the flow itself. Inquiring the analytic flow shown in Figure 6.9, all the glyphs indicate a similar temporal behavior as the added tubes are almost identical in terms of length and direction, as the whole underlying flow is moving horizontally along one axis over time.

An appropriate scaling factor or a change of thickness can be chosen to further emphasize this addition if necessary.

6.5 LIMITATIONS AND FUTURE WORK

Even though these extensions for the general second-order tensor glyphs can be applied to any temporal derivative of first-order tensor fields, this is not a construction method for general 4D second-order tensors. It is the special property of the time derivative, that allows us to utilize glyphs constructed in the remaining subspaces. This added dimension can then be projected onto the subspace and added to the glyph.
Similar to the previous glyphs, this chapter did not address deeper insights on visual perception of colors, controlled sampling of the underlying domain, or user studies, about the acceptance of the newly constructed glyphs. Dealing with cases of non-uniqueness when visualizing 3D tensors of rank 1 remains another inherited limitation of the glyph design introduced in Chapter 5.
Our decision to move the sticks to the boundary of the glyph is mainly due to reducing visual clutter as well as to ensure visibility in the 3D case. However, in the 2D case, these sticks may give the impression to be only overlapped by the geometry and therefore be much longer, when the glyph is larger. As the two sticks represent the symmetry property of an eigenvector, their directions are identical and only reflected. They cannot, however, provide any information about which choice of sign represents the actual change of position of the feature to the next time step.

7 TOWARDS GLYPHS FOR UNCERTAIN SYMMETRIC SECOND-ORDER TENSORS

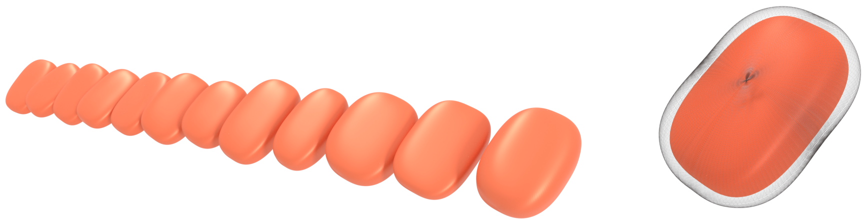


Figure 7.1: Left: superquadric tensor glyph representations of 12 members of a tensor field ensemble sampled in each field at the same location. Right: a superquadric tensor glyph representing the mean glyph augmented with a wireframe offset surface encoding uncertainty.

This chapter is based on the publication:

T. Gerrits, C. Rössl, and H. Theisel
Towards Glyphs for Uncertain Symmetric Second-Order Tensors
Computer Graphics Forum (Proc. EuroVis), 2019

Uncertainty visualization is one of the current challenges in scientific visualization. Modern visual data analysis does not only focus on properties, features, and correlations in the data but also on their uncertainties. As discussed in the introduction, this additional consideration comes with a significant increase in data to be processed and visualized: instead of scalar/vector/tensor samples at domain points, either ensembles of scalar/vector/tensor samples or distribution functions have to be processed. This challenge also applies to tensor data. In this chapter, we investigate *3D uncertain symmetric second-order tensor* fields under the assumption of a *normal distribution*. These fields are usually obtained from ensembles of tensor fields, which consist of multiple measurements of a tensor per grid point. Such an uncertain symmetric tensor is represented by a mean tensor (consisting of 6 coefficients) and – after embedding the tensors into a 6D vector space – by a 6×6 covariance matrix (consisting of 21 coefficients).

In this chapter, we aim to find a generic approach to designing glyphs that represent both simultaneously the 6 coefficients of the mean tensor and the 21 coefficients of the covariance matrix. Similar to the last chapter, it makes sense to start the new construction from some established glyph representation for the mean tensor by a closed glyph surface. We show that the information encoded in the covariance matrix can be captured by a scalar field that lives on the glyph surface. The scalar function encodes the local perturbation of the glyph surface under applying a normal random perturbation to the mean tensor, where the latter perturbation of the mean tensor is modeled by the covariance matrix. We demonstrate this by visualizing the scalar field as an offset surface to the surface that represents the mean glyph. This provides an understanding of the impact of perturbations on the geometry of the mean glyph or equivalently its shape variation under the given uncertainty distribution function. We show that this new glyph uniquely encodes the covariance matrix if the chosen mean glyph is "complicated enough", which is the case, e. g., for the standard representation by superquadrics. In addition, we can measure the "stability" of the mapping between any uncertain glyph and the associated covariance matrix as a single number. We apply the technique to three standard glyphs for the mean tensor: an ellipsoid representation for positive definite tensors, superquadric glyphs [93], [165], and the glyphs introduced in Chapter 5 for symmetric tensors. We show that the ellipsoid representation does not give full coverage of the information encoded in the covariance matrix whereas superquadric glyphs do. We provide examples and experiments and apply our technique to ensembles of DT-MRI data and mechanical stress tensors.

Throughout this chapter, we use $\mathbf{S}$ as a notation for the given tensor to emphasize, that it is a symmetric second-order tensor and further we make use of the fact, that such symmetric second-order tensors can be represented as vectors $\mathbf{v}(\mathbf{S})$ as shown in Section 3.1.3. This was done similarly in [1], [11] and not only leads to a more compact representation but further allows us to describe the uncertainty of second-order tensors in terms of standard matrix and vector operations instead of non-standard higher-order tensor operations. As a reminder, we use the standard nabla operator notation throughout this chapter for derivatives of scalar functions $g(\mathbf{s})$ and vector fields $\mathbf{g}(\mathbf{s})$ w.r.t. to a vector $\mathbf{s} = (s_1, \ldots, s_n)$:

$$\nabla_{\mathbf{s}} g = \frac{\partial g}{\partial \mathbf{s}} = \left(\frac{\partial g}{\partial s_1}, \ldots, \frac{\partial g}{\partial s_n}\right)^{\mathrm{T}} \text{ and } \nabla_{\mathbf{s}} \mathbf{g} = \frac{\partial \mathbf{g}}{\partial \mathbf{s}} = \left(\frac{\partial \mathbf{g}}{\partial s_1}, \ldots, \frac{\partial \mathbf{g}}{\partial s_n}\right)^{\mathrm{T}}.$$

7.1 THE VISUALIZATION PROBLEM

We assume an uncertain tensor under normal distribution that is described by the distribution function

$$p(\mathbf{S}) = \frac{1}{\sqrt{(2\pi)^n \ \det \mathbf{C}}} \ \exp\{-\tfrac{1}{2}\mathsf{v}(\mathbf{S}-\bar{\mathbf{S}})^{\mathrm{T}}\, \mathbf{C}^{-1}\, \mathsf{v}(\mathbf{S}-\bar{\mathbf{S}})\,\}\ . \quad (7.1)$$

This function has two parameters: the mean tensor $\bar{\mathbf{S}}$ and the covariance matrix $\mathbf{C}$, which describes not only the variance of the individual coefficients of $\mathsf{v}(\mathbf{S})$ but also their linear dependencies. So, we define an *uncertain tensor* as a pair $(\bar{\mathbf{S}}, \mathbf{C})$. Note that $\mathbf{C}$ is a symmetric positive definite matrix. In 2D, it is a 3×3 matrix with 6 distinct entries, while in 3D it is a 6×6 matrix with 21 entries.

Assuming a Gaussian distribution to describe uncertainty in tensor data is common and widely accepted in the literature [1], [11], [13]. Alternative models exist and are used in settings, where this assumption does not hold. For instance, [208], [209] use a modified mean tensor.

Given m tensor samples $\mathbf{S}_1, \ldots \mathbf{S}_m$, e. g., from an ensemble data set with m members, the best-fitting uncertain tensor is given by

$$\bar{\mathbf{S}} \;=\; \frac{1}{m}\sum_{i=1}^{m}\mathbf{S}_i \quad \text{and} \quad \mathbf{C} = \frac{1}{m}\sum_{i=1}^{m}\mathsf{v}(\mathbf{S}_i-\bar{\mathbf{S}})\,\mathsf{v}(\mathbf{S}_i-\bar{\mathbf{S}})^{\mathrm{T}}\ . \qquad (7.2)$$

As symmetric tensors are closed under linear combination [214], $\bar{\mathbf{S}}$ will also be symmetric.
We search for glyphs that encode both $\bar{\mathbf{S}}$ and $\mathbf{C}$ and satisfy a list of properties. In order to express these properties, we first need to specify the terms *scaled* and *rotated uncertain tensor*. Assume that the same rotation or scaling is applied to all tensor samples in Equation (7.2). Then scaling by a factor $\rho > 0$ gives the scaled uncertain tensor $(\rho\,\bar{\mathbf{S}}\,,\,\rho^2\,\mathbf{C})$, and rotation by $\mathbf{R}$ gives the rotated uncertain tensor $(\mathbf{R}\,\bar{\mathbf{S}}\,\mathbf{R}^{\mathrm{T}}\,,\,\hat{\mathbf{R}}\,\mathbf{C}\,\hat{\mathbf{R}}^{\mathrm{T}})$. The construction of $\hat{\mathbf{R}}$ from $\mathbf{R}$ was worked out by Theisel and Rössl and is given in Appendix D.

7.2 AN EXTENDED WISH LIST FOR UNCERTAIN GLYPHS

As we have discussed before, *the* one and only perfect glyph to represent data like this usually does not exist. So, again we want to set up some required properties to constrain the search to conforming glyphs. We, therefore, extend our “wish list” defined in Section 5.1 to apply to uncertain tensors. Let $(\bar{\mathbf{S}}, \mathbf{C})$ be an uncertain tensor and $G(\bar{\mathbf{S}}, \mathbf{C})$ its glyph representation.

1. *Rotation invariance:* $G(\mathbf{R}\,\bar{\mathbf{S}}\,\mathbf{R}^{\mathrm{T}},\,\hat{\mathbf{R}}\,\mathbf{C}\,\hat{\mathbf{R}}^{\mathrm{T}}) = \mathbf{R}\,G(\bar{\mathbf{S}},\mathbf{C})$ for any rotation matrices $\mathbf{R}$ and $\hat{\mathbf{R}}$.
2. *Scaling invariance:* $G(\rho\,\bar{\mathbf{S}},\,\rho^2\,\mathbf{C}) = \rho\,G(\bar{\mathbf{S}},\mathbf{C})$ for a positive scaling factor ρ.
3. *Continuity:* Small changes in tensor or covariance matrix should result in small changes in the glyph:
$(\bar{\mathbf{S}}_1,\mathbf{C}_1) \approx (\bar{\mathbf{S}}_2,\mathbf{C}_2) \Rightarrow G(\bar{\mathbf{S}}_1,\mathbf{C}_1) \approx G(\bar{\mathbf{S}}_2,\mathbf{C}_2)$.
4. *Uniqueness*: The glyph should contain information to uniquely reconstruct the uncertain tensor: no two different tensors should have the same glyph representation:
$(\bar{\mathbf{S}}_1,\mathbf{C}_1) \neq (\bar{\mathbf{S}}_2,\mathbf{C}_2) \Rightarrow G(\bar{\mathbf{S}}_1,\mathbf{C}_1) \neq G(\bar{\mathbf{S}}_2,\mathbf{C}_2)$.
5. *Direct encoding of real eigenvectors/eigenvalues of* $\bar{\mathbf{S}}$*:* Since the eigenvectors/eigenvalues of $\bar{\mathbf{S}}$ have a well-defined meaning in most applications, they should be directly encoded in $G(\bar{\mathbf{S}},\mathbf{C})$.
6. *Convergence for* $\mathbf{C} \to \mathbf{0}$*:* For vanishing uncertainty, $G(\bar{\mathbf{S}},\mathbf{C})$ should converge to a well-defined glyph encoding all information of $\bar{\mathbf{S}}$.
7. *Intuitiveness*: The glyph should be easily readable and should have an intuitive interpretation.

Properties (1.–5.) are direct generalizations of the glyph properties for the certain case, as formulated in Chapter 5. Property (6.) requires that the certain case should be a well-defined special case in all uncertain tensor glyphs. Property (7.) is the only one that cannot be shown by mathematical proof, due to the lack of a mathematical definition of the concept of intuitiveness. The main contribution in this chapter is to *prove* that our new glyphs fulfill the properties (1.–6.). We then search for intuitive glyphs in the subspace of all possible glyphs given by properties (1.–6.).

7.3 RELATED WORK

The aim of this work is to extend current glyph-based visualization techniques for second-order tensors such as those presented in Chapter 5 or techniques presented in the related work Section 4.1 by additionally encoding uncertainty. While uncertainty in tensor data is briefly covered in Section 3.3, where we list a selection of works, we want to review some of them in more detail to further analyze if they fulfill the newly introduced properties.

The *cones of uncertainty* introduced by Jones [90] fail to show distributions as unique glyphs. Similarly, the *HiFiVE* glyphs by Schultz et al. [166] visualize uncertainty in addition to the main direction of a given

probability distribution but this construction leads to sudden changes of the glyph when other directions are similarly strong as in isotropic locations.
The technique used to create the Shape Inclusion Probability (SIP) glyphs by Jiao et al. [89], where glyphs are created by superimposing a large number of samples from the given distribution, is related to our work because it is driven by geometry and even some of their glyphs may appear similar to ours. However, it is also very different: firstly, it lacks (provable) adherence to design principles summarized in the previous section, e.g., directions of eigenvectors or magnitude of eigenvalues of the mean tensor may not be directly visualized. Secondly, the glyph encodes the uncertainty in a 3D scalar field, while we want to encode the uncertainty in a 2D scalar field on the glyph surface. Finally, its glyph computation is based on generating a large number of random samples, which are “fused” to a glyph representation. In contrast, our approach constructs a well-defined shape as a parametric or implicit function.
The two visualizations proposed by Basser et al. [11], [13] also need to be analyzed separately: while the independent visualization of the covariance matrix provides a useful representation of the covariance itself, it raises a number of problems. Firstly, the same covariance matrix shows a different impact on different mean tensors. Secondly, only a subset of the full covariance matrix is visualized. Thirdly, this visualization is not invariant under rotation of the coordinate system or tensors, respectively. The rendering of the mean tensor and the variance as isosurfaces represented a first approach to visualizing the impact of the covariance matrix on the mean tensor, however, only the totally symmetric part of the covariance is used, resulting in a violation of property (4.). These shortcomings are also discussed in the work of Abbasloo et al. [1] who propose to render the six distinct glyphs that show the impact of the covariance on the mean tensor. While this gives a complete picture of the uncertain tensor, it does not satisfy the continuity property (3.): if two eigenvalues of the covariance tensor get close to each other, the corresponding eigenvectors (and therefore the 6 visualizations) may show discontinuities. Finally, the work of Zhang et al. [208] only encodes a three-dimensional subspace of the complete 21-dimensional space spanned by an uncertain tensor.
The following table compares existing work on uncertain symmetric tensor glyphs with respect to the design properties (1.–6.). The additional last column ($\pm$) indicates whether the glyph visualization distinguishes between indefinite and definite general symmetric second-order tensors. Properties that could not be decided are indicated as "?".

method / satisfies	(1.)	(2.)	(3.)	(4.)	(5.)	(6.)	±
Jones et al. [90]	✓	✓	✗	✗	?	?	✗
Basser et al. [13]	✓	✓	✓	✓	✗	✓	✗
Jiao et al. [89]	✓	✓	✓	✗	?	✓	✗
Schultz et al. [166]	✓	✓	✗	✗	?	?	✗
Abbasloo et al. [1]	✓	✓	✗	✓	✗	✓	✓
Zhang et al. [208]	✓	✓	✗	✗	✓	✓	✗
Our extended ellipsoid	✓	✓	✓	✗	✓	✓	✗
Our extended superquadric	✓	✓	✓	✓	✓	✓	✓
Our extended glyph from Chapter 5	✓	✓	✓	✓	✓	✓	✓

We conclude this review of related works with the statement that – to the best of our knowledge – no glyph for uncertain tensors exists that fulfills properties (1.–6.). Moreover, [1] even state that due to the high data complexity, "*it does not seem promising to try and visualize all aspects of tensor covariance simultaneously*". We disagree with this statement and propose a solution to this visualization problem in the following.

7.4 GLYPHS FOR UNCERTAIN SYMMETRIC TENSORS

We propose a generic approach that extends any glyph definition for a "certain" tensor $\mathbf{S}$ to the uncertain case to provide a glyph for $(\bar{\mathbf{S}}, \mathbf{C})$. A variety of glyph definitions exist for the certain case. These glyphs are often described as closed surfaces (or curves in 2D), sometimes with additional color information. A glyph surface is either given in *implicit* form

$$g(\mathbf{S}, \mathbf{x}) = 0 \tag{7.3}$$

or in *parametric* form

$$\mathbf{g}(\mathbf{S}, \theta, \phi) \ , \tag{7.4}$$

with surface parametrization (θ, ϕ). For the uncertain case, we represent the mean tensor $\bar{\mathbf{S}}$ by a standard glyph surface. In addition, we define a non-negative scalar field q on the glyph surface that encodes the impact of the covariance $\mathbf{C}$ on $\bar{\mathbf{S}}$. We write q short for $q(\mathbf{S}, \mathbf{x})$ (for $g(\mathbf{S}, \mathbf{x}) = 0$) and $q(\mathbf{S}, \theta, \phi)$, likewise, and define

$$q = \sqrt{\mathbf{q}^{\mathrm{T}} \, \mathbf{C} \, \mathbf{q}} \tag{7.5}$$

with

$$\mathbf{q} = \mathbf{q}(\mathbf{S}, \mathbf{x}) = \frac{\nabla_{\mathbf{S}} \, g}{||\nabla_{\mathbf{x}} \, g||} \tag{7.6}$$

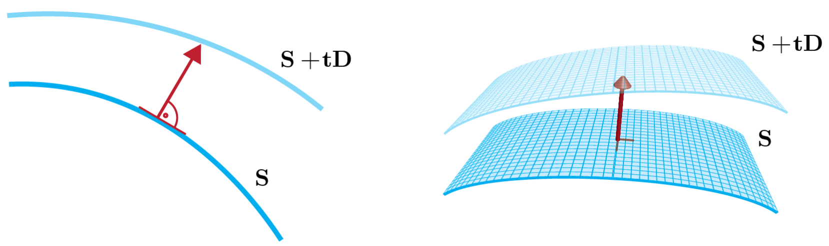


Figure 7.2: The normal velocity r of a time-dependent surface in 2D (left) and 3D (right).

for the implicit case and

$$\mathbf{q} = \mathbf{q}(\mathbf{S}, \theta, \phi) = (\nabla_{\mathbf{s}}\, \mathbf{g})\mathbf{n} \tag{7.7}$$

for the parametric case with $\mathbf{s} = \mathsf{v}(\mathbf{S})$, and $\mathbf{n} = \frac{1}{||\cdot||}\left(\frac{\partial \mathbf{g}}{\partial \theta} \times \frac{\partial \mathbf{g}}{\partial \phi}\right)$ is the surface normal. Note that for the implicit case, $\nabla_{\mathbf{s}}\, g$ is a 3-vector in 2D and a 6-vector in 3D. For the parametric case, $\nabla_{\mathbf{s}}\mathbf{g}$ is a 3×2 matrix in 2D and a 6×3 matrix in 3D.

Derivation and Explanation

In order to explain the idea of the field q, we consider the concept of *normal velocity of a time-dependent surface*. Assume a time-dependent surface either in an implicit representation $g(\mathbf{x}, t) = 0$ or in a parametric representation $\mathbf{g}(\theta, \phi, t)$. The surface changes shape and location under variation of the time parameter t. The normal velocity of the surface describes the change of the surface in the direction of the surface normal as

$$r = -\frac{g_t}{||\nabla_{\mathbf{x}}\, g||} \tag{7.8}$$

with $g_t = \frac{\partial g}{\partial t}$ for the implicit case and

$$r = \frac{\partial \mathbf{g}}{\partial t}^{\mathrm{T}} \mathbf{n} = \mathbf{g}_t{}^{\mathrm{T}} \mathbf{n} \tag{7.9}$$

with $\mathbf{g}_t = \frac{\partial \mathbf{g}}{\partial t}$ for the parametric case. Figure 7.2 gives an illustration. Returning to tensors, we observe how a glyph surface behaves under a perturbation

$$\mathbf{S} \rightarrow \mathbf{S} + t\, \mathbf{D} \tag{7.10}$$

to the tensor for a small t. We want to observe the directional derivative of $\mathsf{v}(\mathbf{S})$ in the direction $\mathsf{v}(\mathbf{D})$. We do so by considering the time-dependent surfaces

$$g(\mathsf{v}(\mathbf{S}) + t\, \mathsf{v}(\mathbf{D}), \mathbf{x}) = 0 \quad \text{and} \quad \mathbf{g}(\mathsf{v}(\mathbf{S}) + t\, \mathsf{v}(\mathbf{D}), \theta, \phi)\ .$$

Computing their normal velocity for $t = 0$ gives

$$r(\mathbf{S}, \mathbf{D}, \mathbf{x}) = -\mathsf{v}(\mathbf{D})^{\mathrm{T}}\mathbf{q} \tag{7.11}$$

with $\mathbf{q}$ as defined in Equation (7.6) for the implicit case and

$$r(\mathbf{S}, \mathbf{D}, \theta, \phi) = \mathsf{v}(\mathbf{D})^{\mathrm{T}}\mathbf{q} \tag{7.12}$$

with $\mathbf{q}$ as defined in Equation (7.7) for the parametric case.

Equation (7.11) and Equation (7.12) describe the normal velocity of the glyph surface under a *particular* perturbation $\mathbf{D}$. In order to consider the behavior of the glyph surface under *all* possible perturbations, we consider r^2 as

$$r^2 = \mathbf{q}^{\mathrm{T}}\mathsf{v}(\mathbf{D})\,\mathsf{v}(\mathbf{D})^{\mathrm{T}}\,\mathbf{q}, \tag{7.13}$$

and replace $\mathsf{v}(\mathbf{D})\,\mathsf{v}(\mathbf{D})^{\mathrm{T}}$ by the covariance matrix $\mathbf{C}$. This gives $q = \sqrt{r^2}$ as defined in Equation (7.5). So, the field q describes the mean absolute values of the normal velocities of an arbitrary glyph surface perturbation $\mathbf{D}$ with the distribution of $\mathbf{D}$ given by $\mathbf{C}$.

Computation and Visualization

Although the computation of q by Equations (7.5) to (7.7) is conceptually simple, the concrete implementation becomes involved for glyph representations that require a spectral decomposition of the tensor: in this case, the derivatives of the decomposition with respect to the glyph components need to be computed, which is difficult and generally unpractical in closed-from. However, the algorithmic computation of q is indeed simple if derivatives are approximated numerically by finite differences. This is what we use in our implementation. One potential pitfall remains: care has to be taken when different parametric representations exist for the same tensor (e. g., the superquadric glyphs). In this case, we have to ensure that the same parametrization is used for all samples required for estimating a derivative.

For visualization, we render two closed surfaces: the mean glyph surface G, and a surface Q defined as $Q = G + q\,\mathbf{n}$ where $\mathbf{n}$ is the surface normal on G. This means that Q is a scaled offset surface of G where q dictates the normal distance between G and Q. The joint visualization of the two nested surfaces G and Q is a standard problem for visualization. Here, we apply a straightforward rendering using semi-transparency.

A special case that should be considered consists of a mean glyph surface G that is C^0 continuous at certain locations, i. e., sharp edges or corners. These locations result in a locally discontinuous Q, i. e., a surface with boundaries at “jumps”. This can happen at single points as well as along a closed line. We close such boundary loops with ruled surfaces to maintain a closed surface. At singularities, where different values can be mapped to the same point, we chose to set the offset to zero.

7.5 ANALYSIS

In this section we show that the proposed glyphs fulfill properties (1.-7). Properties (1.,2.,3.,5.) are generic properties. If they are fulfilled by the underlying glyph for the mean tensor, Equations (7.5) to (7.7) ensure that they carry over directly to the glyph for the uncertain tensor. Property (6.) is simple: Equation (7.5) gives that for $\mathbf{C} \to \mathbf{0}$, we get $q \to 0$ and therefore convergence to the "certain" glyph.
Properties (4., uniqueness), and (7., intuitiveness) are harder to show. Uniqueness is not generic: different choices of glyphs for the mean tensor result in different statements about uniqueness.

7.5.1 *Uniqueness*

To prove or counter-prove the uniqueness of a glyph for an uncertain tensor $(\bar{\mathbf{S}}, \mathbf{C})$, we assume that the mean tensor $\bar{\mathbf{S}}$ is uniquely represented by the glyph surface itself. It remains to show that all 21 coefficients of $\mathbf{C}$ can be uniquely derived from the scalar field q on the glyph surface. In fact, to show uniqueness, we have to show that there exist 21 sample points $\mathbf{g}_1, \ldots, \mathbf{g}_{21}$ on the glyph surface such that the corresponding samples $q_1, \ldots, q_{21}$ of the field $q(\mathbf{g})$ enable a unique reconstruction of the covariance matrix $\mathbf{C}$. Let $\mathbf{q}_1, \ldots, \mathbf{q}_{21}$ denote the corresponding samples of the vector field $\mathbf{q}$ given in Equation (7.5) such that $q_i^2 = \mathbf{q}_i^{\mathrm{T}} \mathbf{C}\, \mathbf{q}_i$. Then mapping the symmetric tensor in a vector space and Equations (3.1) and (7.5) give

$$(q_1^2, \ldots, q_{21}^2)^{\mathrm{T}} = \mathbf{M}^{\mathrm{T}} \mathsf{v}(\mathbf{C}) \tag{7.14}$$

with

$$\mathbf{M} = \left(\mathsf{v}(\mathbf{q}_1 \mathbf{q}_1^{\mathrm{T}}), \ldots, \mathsf{v}(\mathbf{q}_{21} \mathbf{q}_{21}^{\mathrm{T}})\right) \in \mathbb{R}^{21 \times 21}. \tag{7.15}$$

In order to show that $\mathsf{v}(\mathbf{C})$ can be computed from $(q_1^2, \ldots, q_{21}^2)^{\mathrm{T}}$ (and vice versa), we have to show that $\mathbf{M}$ has full rank for the chosen sample points.
Equations (7.14) and (7.15) show that the uniqueness of a glyph depends on the behavior of $\mathbf{q}$ on the glyph surface. Let $\mathcal{Q}$ be the *set of all vectors* $\mathbf{q}$ *on the glyph surface.* A characterization of $\mathcal{Q}$ is the key to study uniqueness.

Lemma 7.5.1. *An uncertain glyph is* not *unique iff all* $\mathbf{q} \in \mathcal{Q}$ *live on a common quadric,i. e., there exists a non-zero matrix* $\mathbf{A}$ *such that* $\mathbf{q}^{\mathrm{T}} \mathbf{A}\, \mathbf{q} = 0$ *for all* $\mathbf{q} \in \mathcal{Q}$.

A proof for Lemma 7.5.1 has been worked out by Theisel and Rössl and is given in Appendix E and using it allows us to analyze particular uncertain tensor glyphs. Therefore, we can make the following Theorems:

Theorem 7.5.2. *Uncertain* ellipsoid glyphs *for positive definite tensors are* not *unique.*

We summarize the proof of Theorem 7.5.2 for uncertain tensors in 2D in Appendix E. It shows that we can find a non-zero matrix $\mathbf{A}$, such that $\mathbf{q}\,\mathbf{A}^{\mathrm{T}}\,\mathbf{q} = 0$ for all $\mathbf{q}$ on the surface of ellipsoid glyphs. The general construction is identical for the 3D case, however, the generated expressions are significantly more complex.

Theorem 7.5.3. *Uncertain* superquadric glyphs *for positive definite tensors are unique if all eigenvalues of the mean tensor are nonzero and distinct.*

A proof of this theorem for the 2D case is found in Appendix E and the basic idea is simple: find 6 samples of points on the glyph surface such that the matrix $\mathbf{M}$ in Equation (7.15) has full rank. The difficulty of the proof consists in the fact that the construction of superquadrics involves a change of coordinates using the spectral basis of the mean tensor $\bar{\mathbf{S}}$, i. e., it is parametrized by eigenvalues and eigenvectors, whereas partial derivatives must be computed w.r.t. the entries $\mathsf{v}(\bar{\mathbf{S}})$.

Quantifying uniqueness

As mentioned above, a formal proof that a new uncertain glyph is unique can be difficult. The reason is that many glyph definitions – like the superquadric glyphs – rely on the spectral decomposition of the tensor. This makes finding a formal proof seemingly the hardest task when establishing a new glyph for uncertain tensors – significantly harder than the definition and implementation.

To cope with this, we introduce a measure of the "uniqueness" of an uncertain glyph: we measure how stably the covariance matrix $\mathbf{C}$ can be reconstructed from m samples ($m \geq 21$) $q_1, \ldots, q_m$ of the function q at the sample points $\mathbf{g}_1, \ldots, \mathbf{g}_m$ on the glyph surface G. In the ideal case, a small perturbation in $\mathbf{C}$ results in small changes in $q_1, .., q_m$, and vice versa. A low uniqueness number is therefore intuitively given, if strong changes in $\mathbf{C}$ only lead to little or no changes in $q_1, .., q_m$ and conversely, if small changes in $\mathbf{C}$ lead to strong changes in $q_1, .., q_m$. The reconstruction of q from $\mathbf{C}$ is defined by the linear mapping in Equation (7.14) if $m = 21$. For $m > 21$ samples, the map is given by the corresponding least-squares solution to

$$\mathbf{M}\mathbf{M}^{\mathrm{T}}\,\mathsf{v}(\mathbf{C}) \;=\; \mathbf{M}\left(q_1^2, \ldots, q_m^2\right)^{\mathrm{T}} \;.$$

The condition number $\kappa = \kappa(\mathbf{M})$ measures the stability of this map (for any $m \geq 21$). The condition number is defined as the ratio of largest and smallest singular values of $\mathbf{M}$. This implies $\kappa \geq 1$ and $\kappa \to \infty$ if $\mathbf{M}$ does not have full rank (i. e., $\mathbf{M}\,\mathbf{M}^{\mathrm{T}}$ is not invertible). Numerical applications commonly prefer specification of the reciprocal condition number in order to have values in a finite interval. For the same reason, we define the uniqueness number as

$$u(G) \;=\; \frac{1}{\kappa(\mathbf{M})} \;\in\; [0,1]\ ,$$

which has the following properties:

- $u(G)$ depends only on the shape of the mean glyph. It is a measure of how well an arbitrary covariance matrix $\mathbf{C}$ can be reconstructed from sampling q on the mean glyph surface.
- $u(G)$ is invariant under rotation and scaling.
- $0 \leq u(G) \leq 1$. The larger $u(G)$, the better $\mathbf{C}$ can be reconstructed from sampling q.
- $u(G) = 0$ indicates that the glyph is not unique.

The uniqueness number depends on the number $m \geq 21$ of samples as well as on the sampling positions $\mathbf{g}_i$. Ideally, we would like to compute

$$\inf\{\, u(G) \mid \text{given any possible sampling of } G \,\}\ ,$$

which is infeasible. However, *any* computed $u(G) > \epsilon$ provides evidence of uniqueness for a suitable $\epsilon \to 0$, and *any maximum* of computed values (e. g., for different samplings) gives a conservative estimate or a *lower bound* on uniqueness.

We illustrate these properties and the behavior of $u(G)$ in a few numerical experiments. We start with 21 uniformly distributed random samples on the glyph surface and compute the uniqueness number. We observe that the particular sampling is generally not critical: the computed values of u typically do not vary much. As the uniqueness number depends on the selection of the sample points, one might be tempted to construct a "smart selection" or use "deterministic samples". We decided to use random samples because deterministic sampling would incorporate the orientation of the eigenvectors of $\bar{\mathbf{S}}$. This changes discontinuously in regions of equal eigenvalues, which leads to a violation of the continuity condition (3.) for glyph design. By incorporating additional samples, the number of rows in $\mathbf{M}$ typically increases, as this typically "adds" new information, and the condition number of $\mathbf{M}$ tends to decrease. This means the uniqueness number typically increases. The more samples, the less likely are additional samples to capture new information. Therefore, uniqueness changes at a slower and slower rate and is expected to converge to a limit. This is illustrated in Figure 7.3 for the same mean tensor and two different glyph constructions. With a minimum at 21 values, $u(G)$ rises rapidly until including about 150 additional samples and remains stable from thereon.

Since $u(G)$ is independent of rotation and scaling, we can systematically compute $u(G)$ for all glyphs of a certain glyph type. For this, we consider the three eigenvalues of the mean glyph $\lambda_{1,2,3}$ as $\lambda_1 = 1$, $\lambda_{2,3} \in [-1, 1]$ and compute $u(G)$ for each $(\lambda_2, \lambda_3) \in [-1, 1]^2$. The resulting plot for the glyphs from Chapter 5 is shown in Figure 7.4. It shows that $u(G) > 0$ if $\lambda_{1,2,3}$ are distinct and nonzero. Hence Figure 7.4 shows that the uncertain glyphs based on the 3D tensors glyphs introduced in Chapter 5 are unique. The height surface in Figure 7.4 was computed on a 151×151 sampling grid.

An uncertain glyph is considered to be unique for $u(G) > \epsilon$ such that it is numerically clearly distinguishable from 0. In practice, we expect a significantly lower uniqueness for a close to minimal sampling ($m = 21$) than for higher m (see Figure 7.3). However, even such cases correspond to condition numbers in the range of 10^5, which is perfectly tolerable for solving a linear system.

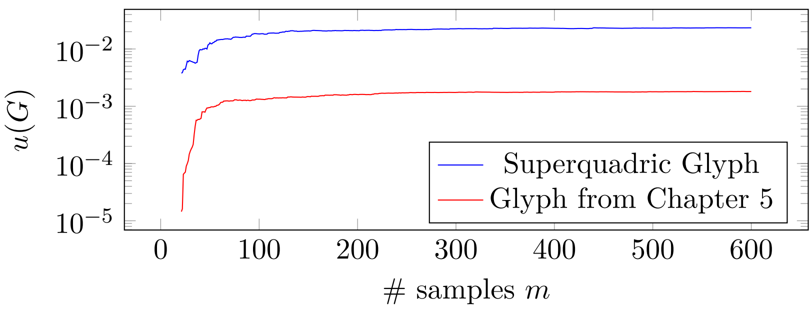


Figure 7.3: For an increasing number of samples on G, uniqueness u increases. The experiment shows random samples on the tensor $\mathbf{v}(\bar{\mathbf{S}}) = (1, 0.5, 0.4, 0, 0, 0)^{\mathrm{T}}$. For both choices of Q, we observe a converging behavior of u.

7.5.2 *Intuitiveness*

It remains to show that the field q on the mean glyph surface provides an intuitive encoding of uncertainty. This cannot be proven formally. Instead, we motivate and explain intuition with help of a few exemplary settings in 2D.

Figure 7.5 shows different visualizations of an ensemble of 2D tensors, which follow a given normal distribution. A 2D tensor can be considered as a point in the three-dimensional vector space of the tensor components $s_{11}, s_{22}, \sqrt{2}s_{12}$. Figure 7.5 (left) shows the tensors as 3D points. The red point denotes the mean tensor, and the overlaid ellipsoid denotes the covariance. Figure 7.5 (center) shows the same set of tensors by overlaying their corresponding transparent superquadric glyph

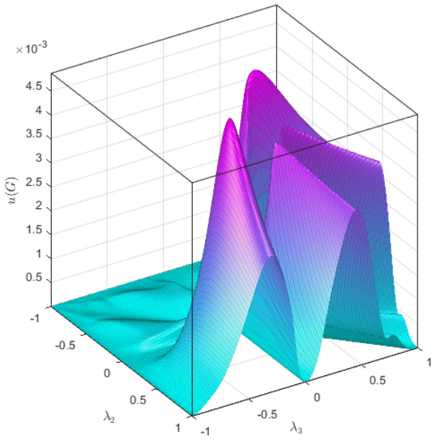


Figure 7.4: Uniqueness $u(G)$ for different mean tensors using the glyphs proposed in Chapter 5. $\lambda_1 = 1$ is fixed, and $\lambda_2, \lambda_3 \in [-1, 1]$ vary. The tensor is unique if $u(G) \neq 0$.

surface (here: superellipses in 2D). We see that there are regions where many curves coincide, whereas in other regions there is a larger spread. Figure 7.5 (right) shows the same tensor ensemble with our visualization: the orange curve is a superquadric representation of the mean tensor, the field q is shown as the region bounded by offset curves in positive and negative normal directions. The relation to Figure 7.5 (center) is visually noticeable: in regions of high spread among the sampled glyph curves, the offset q is rather large. Figure 7.5 also shows that our glyph shows similarities with curve boxplots [122], even though the definition and properties are different.

To further study the meaning of the field q, we conduct the following experiment: we generate 5 samples of 2D tensors by varying properties in the spectral domain. Then we compute the best-fitting uncertain tensor by applying Equation (7.2) and visualize its glyph. The top row of Figure 7.6 shows several collections of 5 tensors as overlaid superquadric glyphs. In the columns we vary (from left to right) 1. one eigenvalue (same signs), 2. one eigenvalue (opposite signs), 3. both eigenvalues (same sign) with inverse correlation, 4. both eigenvalues (opposite signs) with a positive correlation of the magnitudes. The eigenvectors remain constant. The bottom row in Figure 7.6 shows the corresponding uncertain glyphs. The relation between the overlaid superquadric glyphs and our uncertain superquadric glyphs is clearly noticeable.

In Figure 7.7 we conduct the same experiment with constant eigenvalues and varying directions of eigenvectors. The amount of variation decreases from left to the right. The top row shows overlaid superquadrics, and the bottom row shows the corresponding uncertain glyphs. As before, the relation between the overlaid samples and the fitted distributions shown as uncertain glyphs is clearly noticeable.

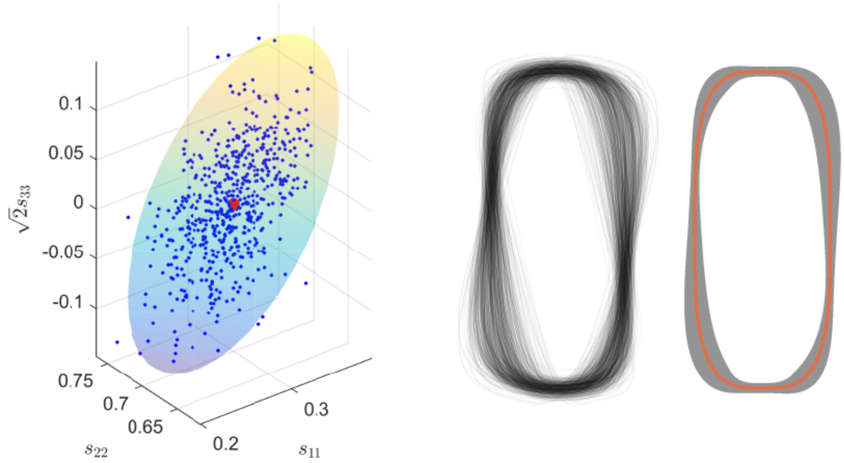


Figure 7.5: Three visualizations of the same 2D tensor ensemble: as points in the vector space of tensor components $\mathbf{s} = \mathbf{v}(\mathbf{S})$, the red point denotes the mean tensor $\bar{\mathbf{S}}$ (left); as overlaid superquadric glyph curves (center); as uncertain glyph: mean $G(\bar{\mathbf{S}})$ is depicted as orange curve, the filled region is bounded by the outward/inward offset curves defined by q (right).

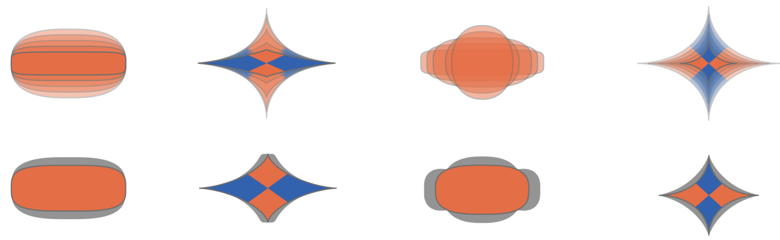


Figure 7.6: Four different sets of tensor samples. Each set consists of 5 tensors that are generated by varying the eigenvalues (with constant eigenvectors). The top row shows overlaid superquadric glyphs. The bottom row shows our corresponding uncertain glyphs.

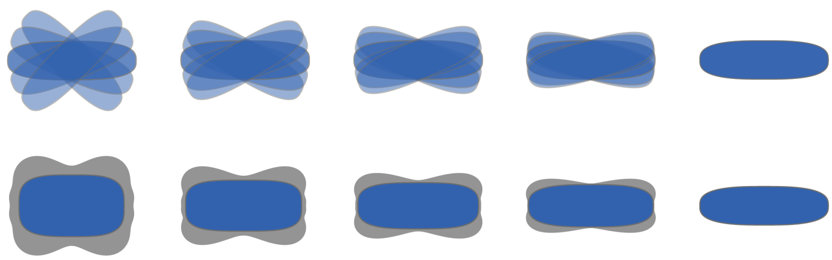


Figure 7.7: Five different sets of tensor samples. Each set consists of 5 tensors that are generated by varying the direction of eigenvectors (with constant eigenvalues). The top row shows overlaid superquadric glyphs. The bottom row shows our corresponding uncertain glyphs.

We conduct similar experiments for 3D glyphs. Here, we compare different classes of glyphs that are extended to visualize uncertain ten-

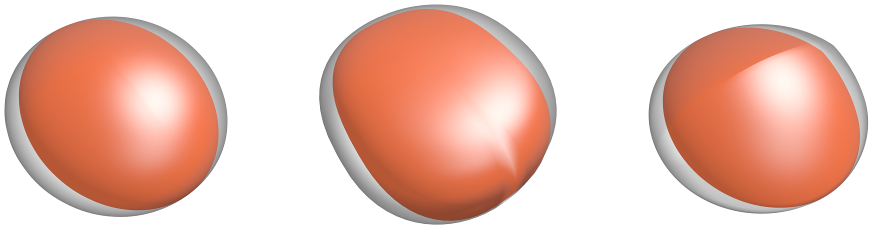


Figure 7.8: Glyphs for uncertain tensor with $\mathbf{v}(\bar{\mathbf{S}}) = (1, 0.8, 0.5, 0, 0, 0)^{\mathrm{T}}$, $\mathbf{C} = \operatorname{diag}(0, 0, 0.2, 0, 0, 0)$. The tensor varies in one principal direction. From left: ellipsoid glyph with $u(G) = 0$, superquadric glyph with $u(G) \approx 9.1 \cdot 10^{-4}$, glyph based on Chapter 5 with $u(G) \approx 1 \cdot 10^{-4}$.

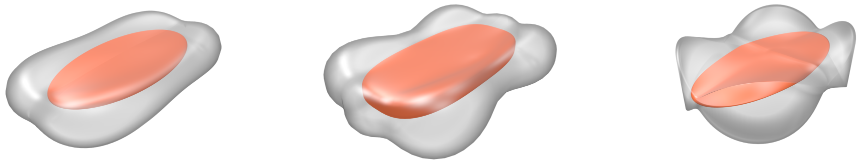


Figure 7.9: Glyphs for uncertain tensor with $\mathbf{v}(\bar{\mathbf{S}}) = (0.9, 0.7, 0.3, 0, 0, 0)^{\mathrm{T}}$ and covariance that corresponds to varying plane rotation of eigenvectors. From left: ellipsoid glyph with $u(G) = 0$, superquadric glyph with $u(G) \approx 3.4 \cdot 10^{-4}$, glyph based on Chapter 5 with $u(G) \approx 2 \cdot 10^{-6}$.

Figure 7.10: Glyphs for uncertain tensor with indefinite mean $\mathbf{v}(\bar{\mathbf{S}}) = (1, 0.6, -0.5, 0, 0, 0)^{\mathrm{T}}$ and $\mathbf{C} = \operatorname{diag}(0, 0.65, 0.03, 0, 0, 0)$. The tensor varies in one principal direction. Left: superquadric glyph with $u(G) \approx 1.5 \cdot 10^{-4}$. Right: glyph based on Chapter 5 with $u(G) \approx 3 \cdot 10^{-6}$. (There exists no ellipsoid glyph in the indefinite case.)

sors: simple ellipsoid glyphs (for positive-definite tensors), superquadric glyphs [165], and the glyphs presented in Chapter 5 (for the case of symmetric tensors). Figure 7.8 and Figure 7.10 show glyphs for ensembles with one varying eigenvalue, whereas Figure 7.9 shows glyphs for tensors that are varied by a plane rotation of eigenvectors. Figure 7.11 visualizes a randomly chosen uncertain tensor $(\bar{\mathbf{S}}, \mathbf{C})$. For each glyph, we provide a uniqueness number $u(G)$ that was computed from sampling at 21 random points on the glyph surface. We emphasize again, that this is no formal proof of intuitiveness but aims towards giving new insights into the visualization of tensor uncertainty.

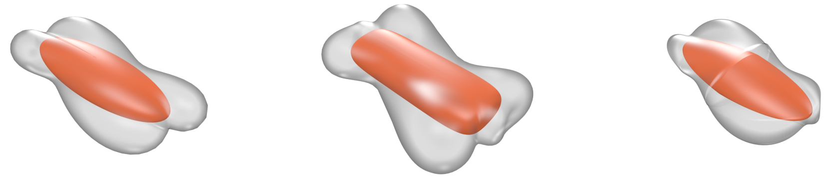
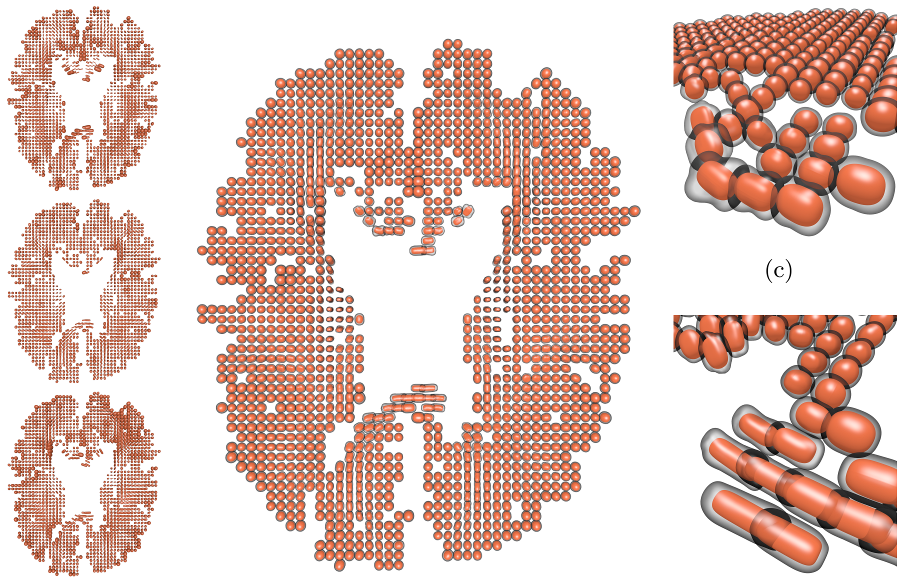


Figure 7.11: Glyphs for uncertain tensor with

$$\bar{\mathbf{S}} = \begin{pmatrix} 1 & 0.2 & 0.1 \\ 0.2 & 0.4 & 0.03 \\ 0.1 & 0.03 & 0.2 \end{pmatrix} \text{ and } \mathbf{C} = 10^{-3} \cdot \begin{pmatrix} 103 & 69 & 69 & 0 & 0 & 0 \\ 69 & 66 & 36 & 0 & 0 & 0 \\ 69 & 36 & 66 & 0 & 0 & 0 \\ 0 & 0 & 0 & 300 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 200 \end{pmatrix}.$$

From left: ellipsoid glyph with $u(G) = 0$, superquadric glyph with $u(G) \approx 3.6 \cdot 10^{-4}$, glyph based on Chapter 5 with $u(G) \approx 5.3 \cdot 10^{-6}$.



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

Figure 7.12: Uncertain superquadric glyphs for an ensemble of Diffusion Tensor Imaging data of the human brain.

7.6 RESULTS

We demonstrate how our new uncertainty glyph can be used as a tool for investigating uncertainty in tensor data by applying it to data from medical imaging as well as mechanical engineering. First, we apply our new tensor visualization to an ensemble of positive-definite symmetric diffusion tensor data given in the DTI multiple atlas set. The Human Brain Atlas was provided by the Johns Hopkins Medical Institute and the Laboratory of Brain Anatomical MRI. A horizontal slice is sampled for fourteen distinct members and a non-linear registration is applied on a rectangular grid as seen in Figure 7.12a. The measured tensors vary in magnitude and direction. We compute the mean tensor and covariance matrix for each sample location. Figures 7.12b to 7.12d show

the superquadric tensor glyph visualization and the offset surface that indicates uncertainty. The produced glyphs allow to see the mean tensor throughout the ensemble members for all locations as well as the local uncertainty. Large offset surfaces indicate stronger variations among the members and provide a geometric insight of this variation. Glyphs shown at the bottom of Figure 7.12c encode a high uncertainty and show, that tensors vary in rotation. Especially tensor data measured close to the lateral ventricles show uncertainty.
Figure 7.13 shows one selected tensor of the same dataset. To illustrate the effect of uncertainty and its correspondence to derived uncertainty measures, we construct a traceless matrix $\mathbf{C}' = \mathbf{C} - \mathrm{diag}(\mathrm{trace}(\mathbf{C}))$ and use it as covariance matrix. The top row shows a blending $\alpha\,\mathbf{C}'$ with the zero matrix for $\alpha \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. This increases uncertainty which is shown by the growing offsets. This offset is close to zero near the glyph axes that represent eigenvector directions. Note that visualizing only the trace of the covariance, which is often used as a derived uncertainty measure, would give the impression that all tensors are equal and certain. For the second row, we blend $\mathbf{C}'$ and the original covariance $\mathbf{C}$ stepwise from left to right as $(1-\alpha)\mathbf{C}' + \alpha\,\mathbf{C}$, which results only in a change of trace. This leads to an overall increase of the offset, also close to the axes. The overall volume of the surface allows for a quick understanding of the level of uncertainty, the glyphs, however, also include spatial information and indicates the type of uncertainty.

We further show the visualization of a stress tensor ensemble from static simulations of stresses applied to a steel cylinder. For the simulation, the bottom end geometry is fixed, and rotational momentum is applied to each axis of the top end. While a mean rotation is applied to the longitudinal axis, three different additional torques are applied and varied for each simulation following a Gaussian distribution to form an ensemble of 10 different tensor fields where tensors are indefinite symmetric stress tensors. Again, a slice orthogonal to the mean rotation axis is sampled on a uniform grid to compute the mean tensor and covariance matrix for each location and then visualized by applying our technique to the superquadric tensor glyph. The resulting image in Figure 7.14 clearly shows the rotational axis in the center of the slice, where the tensor vanishes. For most glyphs, the offset surface is close to the mean glyph surface, indicating a low uncertainty for the location. Only tensors at the left and right border show a stronger uncertainty, where eigenvalues vary while eigenvectors are stable.

PARAMETER DISCUSSION

While the scalar field q defined on the glyph surface is parameter-free, its visual representation is not. A global scaling parameter of the glyph itself has been used to have the sampled tensors densely cover the

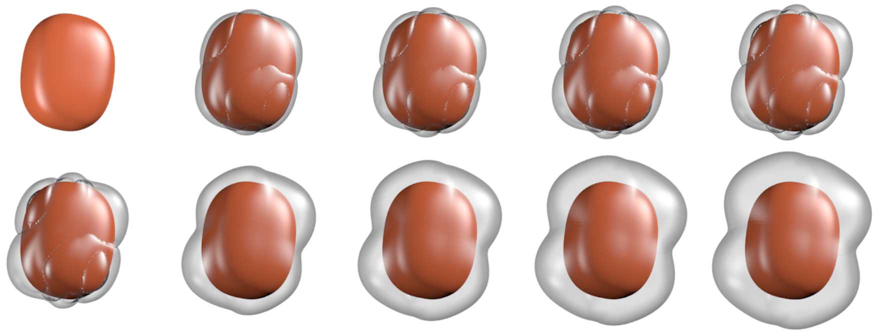


Figure 7.13: Top: linear blending of zero matrix and traceless matrix. Bottom: linear blending of traceless matrix and original covariance matrix.

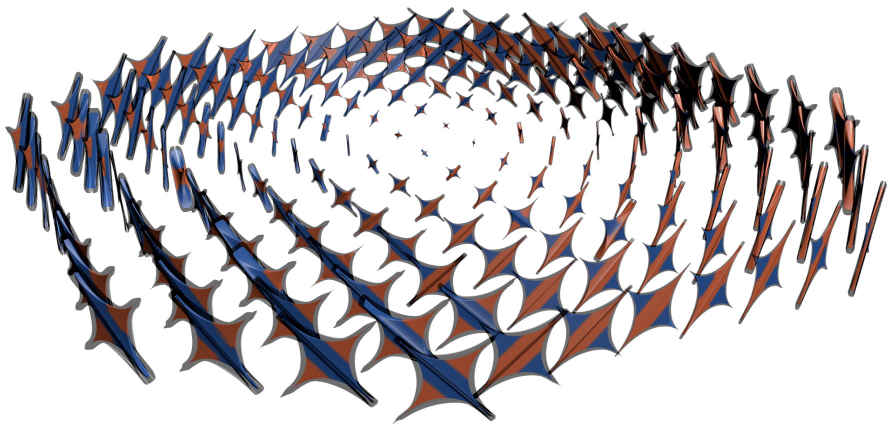


Figure 7.14: Uncertain superquadric glyphs for an ensemble of simulated stress tensors from changing torque applied to a steel cylinder. The colors indicate the signs of eigenvalues, the transparent offset surfaces indicate uncertainty.

area. Further, scaling the offset from the mean surfaces is possible to emphasize uncertainty but has not been used in this work. As the offset surface is encasing the mean tensor, we chose opacity to solve the problem of overlapping. A suitable rendering needs to be applied such that shape and color of both, offset as well as mean surface can be perceived well. Other techniques for visualizing scalar fields on a surface might be applicable, as long as they do not lead to a violation of our wish list described in Section 7.1. Further parameters are related to sampling: When using finite differences to approximate derivatives, the step size between discrete points affects the accuracy. As described in Section 7.5.1, determining a uniqueness measure relies on sampling the glyph surface. For determining the uniqueness of the uncertain glyphs shown in Figures 7.8 to 7.11 we chose the minimum sample number of 21. Values in Figure 7.4 are computed for a selection of 600 samples to ensure the value is close to the tensor's experimental limit.

COMPARISON TO EXISTING GLYPHS

To give a better understanding of how our contribution improves glyph visualization of uncertain tenors, we compare them with existing glyph techniques. As different mean tensors are affected differently by the same covariance, we compare our uncertain tensor glyphs to visualizations by Basser et al. [13] and Abbasloo et al. [1]. As mentioned in Section 7.3, Basser et al. [13] use a radial projection of mean and covariance tensors. Isosurfaces indicate mean tensor as well as standard deviation. Besides the visual complexity produced by three superimposed surfaces, the glyph construction cannot ensure unique representations. Due to the projection, mean tensors that only differ by the sign of eigenvalues will be mapped to the same scalar field. The same can be shown for the covariance, as only the totally symmetric part of the tensor is represented. Both mappings are not bijective. The bottom row of Figure 7.15 demonstrates this behavior: the surfaces for the mean tensor (shown in green) and standard deviation $\pm\sigma_{\text{std}}$ ($-\sigma_{\text{std}}$ red, $+\sigma_{\text{std}}$ blue) are superimposed and rendered translucent. Columns (a) and (b) show glyphs for the same mean tensor $\mathbf{v}(\bar{\mathbf{S}}) = (1, 2, 5, 0, 0, 0)^{\mathrm{T}}$ but different matrices as covariance tensors. While the first matrix can be written as $\mathbf{C}_1 = \operatorname{diag}(0, 0, 0, 1, 2, 3)$, the second places the same non-zero values on different off-diagonal locations. The two matrices used as covariance tensors are given as

$$\mathbf{C}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}, \mathbf{C}_2 = \begin{pmatrix} 0 & 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Columns (a) and (c) of Figure 7.15 show glyphs for the same covariance matrix but different mean tensors, as the sign of the minor eigenvalue is flipped such that $\mathbf{v}(\bar{\mathbf{S}}) = (-1, 2, 5, 0, 0, 0)^{\mathrm{T}}$. The three resulting glyph visualizations by [13] are identical, which is a violation of property (4.). In comparison, the top row shows our new glyphs for the same input tensors. They are clearly distinguishable and are capable to represent each combination uniquely.

Abbasloo et al. [1] visualize the impact on the mean tensor by decomposing the covariance into its eigentensors and rendering the effect of each eigenmode separately. They offer an animation to show how the mean tensor changes based on the different eigenmodes. Alternatively, these glyphs can be presented as overlays, to indicate confidence intervals of tensor distributions. The authors propose to add and subtract the eigentensor scaled by three times the corresponding eigenvalue to the

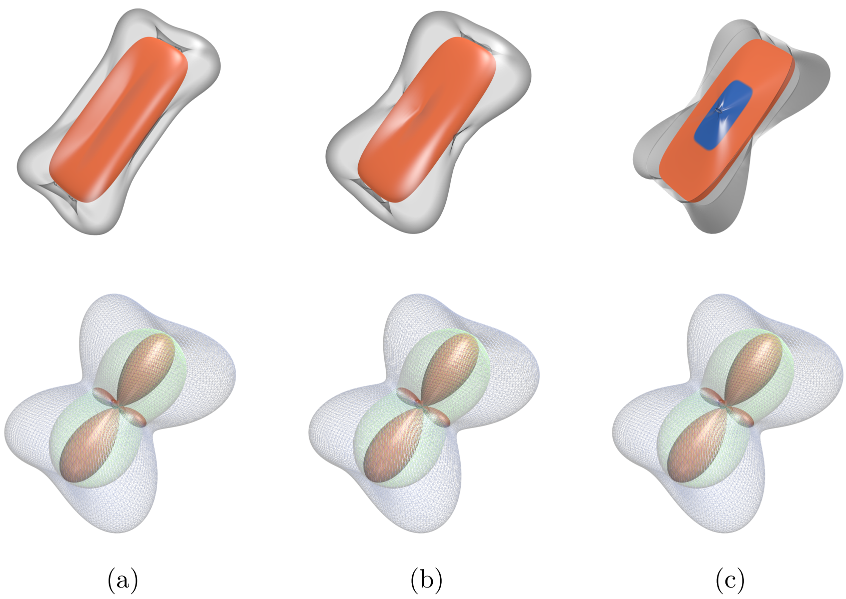


Figure 7.15: Three different uncertain tensors visualized by our method (top row) and [13] (bottom row). While our method clearly shows different glyphs, the glyphs by [13] are identical: [13] is not unique.

mean tensor and render two superimposed superquadric glyphs. As the original mean tensor and thus its eigenvector directions are no longer visualized, this poses as a violation of property (5.). Eigentensors do, however, change in a discontinuous way when the covariance tensor is nearly isotropic, which leads to a sudden change in visualization even though the covariance tensors are virtually identical. This sudden change can be observed in Figure 7.16. Both (b) and (d) show visualizations for the same mean tensor $\mathbf{v}(\bar{\mathbf{S}}) = (1, 2, 5, 0, 0, 0)^{\mathrm{T}}$. The six eigenvalues σ_i and eigentensors $\mathbf{E}^i$ extracted from the fourth-order covariance tensor are used to create six views. Each showing superquadric glyph representations for $\mathbf{D}_{\text{blue}} = \bar{\mathbf{S}} - 3\sigma_i \mathbf{E}^i$ and $\mathbf{D}_{\text{red}} = \bar{\mathbf{S}} + 3\sigma_i \mathbf{E}^i$ and labeled as eigenmode i. Note, that we used a simple translucent rendering of both glyphs, while [1] render both separately, adding a white core to areas where they overlap. For both covariance tensors used in (b) and (d), we use $\mathbf{C} = 0.3 \cdot \mathbf{I}$ and add random symmetric noise in the order of 10^{-8}. This slight noise leads to a sudden change in the eigentensors and therefore in the visualizations, which is a violation of the continuity property (3.). Our new glyph construction accounts for this problem. Uncertain tensor glyphs for the given tensors are shown in (a) and (c). The minimal change between both covariances results in a minimal change between both uncertain glyphs.

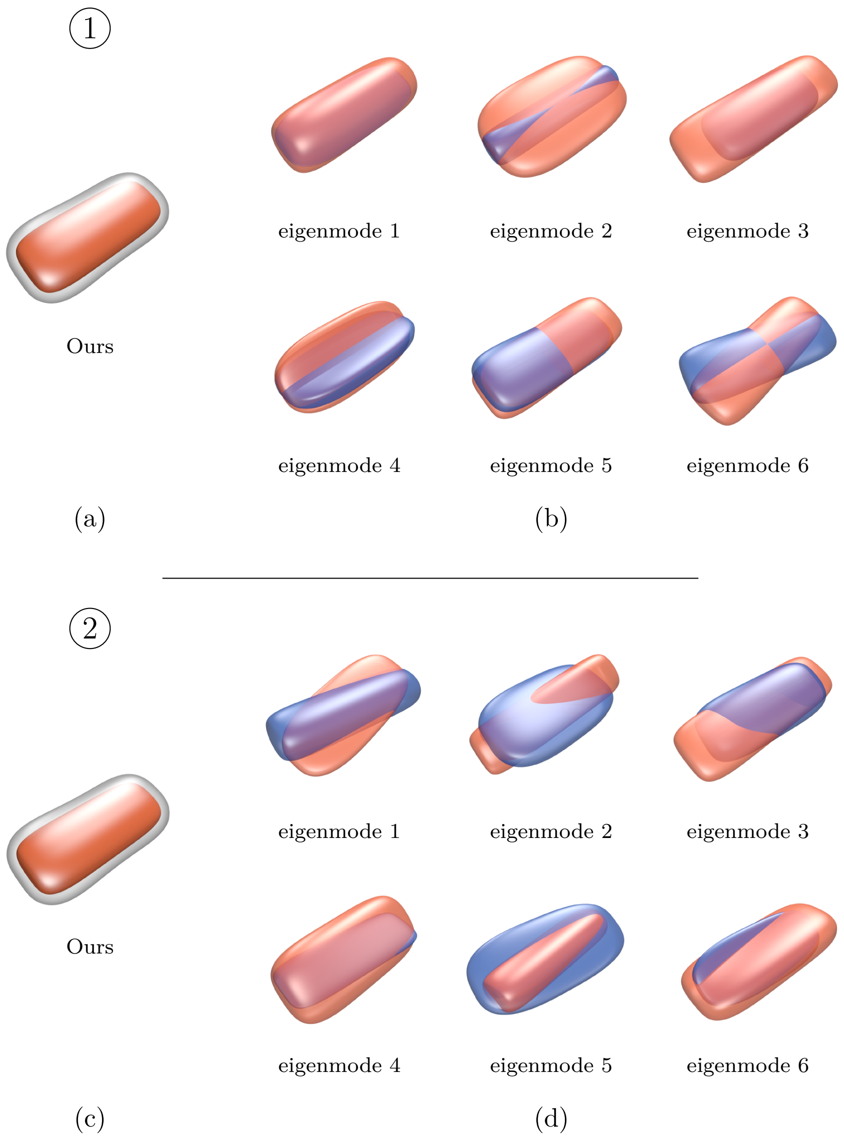


Figure 7.16: Two almost identical uncertain tensors (①,②) having almost identical glyphs by our method ((a) and (c)) but significantly different glyphs in the visualizations of [1] ((b) and (d)): [1] is not continuous.

7.7 LIMITATIONS AND FUTURE WORK

The technique proposed in this chapter is a first step towards uncertain glyphs. Therefore, there are limitations and areas that need further research: the glyph extension described here is built upon and therefore restricted to uncertain tensor data assuming a Gaussian distribution. There are data sets where this model is not appropriate and other distributions are more capable of capturing the underlying data. Naturally, all limitations arising from the chosen underlying glyph construction also apply to the augmented glyph. Our approach opens future research in several directions:

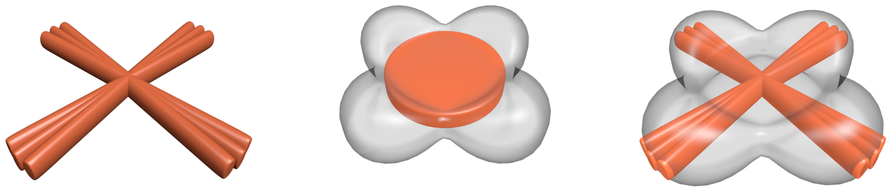


Figure 7.17: 6 tensors simulating a HARDI data set. Left: the tensors in overlaid superquadric representation. Center: uncertain glyph. Right: overlay of input glyphs and the offset surface Q.

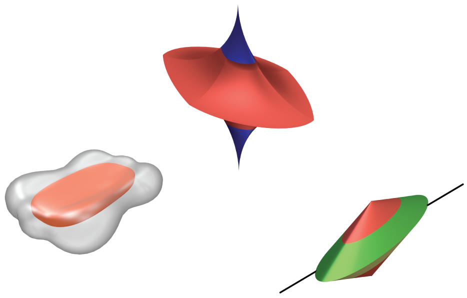
RENDERING. We distinguish surfaces G of the base glyph and the offset surface Q, which encodes uncertainty. Our current rendering style with a solid surface for the mean tensor and a transparent surface for uncertainty is straightforward. More advanced rendering techniques are possible, which may be optimized towards a simultaneous perception of the shapes of both G and Q. This includes illustrative approaches, opacity optimization for surfaces [53], or a piecewise rendering [209].

OPTIMAL UNIQUENESS SAMPLING. While the current implementation relies on a random point sampling for computing uniqueness numbers, better sampling strategies may result in even smaller, i. e., better, uniqueness numbers for the uncertain glyphs. This, however, does not affect the actual glyph design or visualization, it only provides better information about uniqueness.

APPLICATION TO HARDI DATA. In Section 3.2.1 we already discussed the limitations of diffusion tensors to represent crossing fiber structures [75], [167] which has led to the use of high angular resolution diffusion-weighted imaging (HARDI) data to better capture such structures. However, since multiple directional diffusion values can be interpreted as an ensemble of second-order tensors, the higher-order information could be represented by the covariance matrix. To test this, we created a collection of diffusion tensors that represent such a case. Figure 7.17 shows this “HARDI data simulation”: Figure 7.17 (left) shows 6 cigar-shaped tensors in superquadric glyphs with their orientation grouped into two clusters. This represents two crossing fiber directions. Figure 7.17 (middle) shows our uncertain glyph based on superquadric glyphs: while in the mean glyph the directional information is canceled out, it is clearly visible in the offset surface Q that encodes covariance. Figure 7.17 (right) is an overlay of the input glyphs and the Q surface. We note that while this is an indicator of the applicability of our method to HARDI data, a formal establishment and comparison with other HARDI visualization techniques is left to future research.

EXTENSION TO GENERAL TENSORS. Similar to Chapter 5, it seems desirable to extend the construction of tensor glyphs to general second-order tensors. Such a general (non-symmetric) 3D tensor is, however, represented by 9 coefficients for the mean tensor and 45 for the covariance matrix. It seems to be challenging but not hopeless to extend our approach to general tensors in future research.

8 CONCLUSIONS



In the Chapters 5 to 7 of this part, we covered the analysis and scientific visualization of second-order tensor data by the use of tensor glyphs. Even though such glyphs are a powerful and well-known tool, we showed that most techniques are focused on symmetric tensors only and exclude non-symmetric tensors where the eigenvectors can be non-orthogonal and eigenvalues complex. We therefore presented a new technique for visualizing general 2D and 3D tensors with glyphs in Chapter 5. To find such glyphs, we first proposed our "wish list" for tensor properties based on mathematical considerations of the underlying tensor data. Therefore, the construction follows a specific set of carefully chosen desired properties which consist of *invariance to isometries and scaling, direct encoding of all real eigenvalues and eigenvectors, one-to-one relation between the tensors and glyphs* and finally *glyph continuity under changing the tensor* and is based on piecewise rational Bézier curves and surfaces. These new glyphs therefore offer the first glyph visualization technique that can be applied to general 2D and 3D tensors. As vector field Jacobian matrices are one typical application where such non-symmetric tensors appear, we further introduced an extension to these new glyphs in Chapter 6. By finding a suitable mapping of the additional temporal information onto a vector that lives in the same dimension as the spatial glyph, these glyphs make it possible to represent time-dependent Jacobian matrices. The new method provides a visualization of the steadiness or unsteadiness of a vector field at a given instance of time. This approach, too, follows the list of desired tensor glyph properties and the extension is constructed such that the glyph's capability to encode the spatial derivatives is in no way impaired. As the high dimensionality and complex nature of tensor data makes it complicated to find suitable visualizations, including uncertainty presents an even bigger challenge. The work proposed in Chapter 7 presents – to the best of our knowledge – the first approach to direct visualization of uncertain tensors that incorporates *all* 21 parameters when assuming Gaussian distribution in a single glyph. The new uncertain glyph is based on some standard glyph for certain tensors, which represents the mean tensor, and enriched by a scalar field that represents tensor covariance. As variance of intrinsic tensor properties can be derived from the covariance matrix, the full uncertainty information is encoded. The construction of the uncertain glyph is again based on an extended version of the design principles introduced in Chapter 5 and provides a bijective map between the glyph and the uncertain tensor (i. e., mean tensor and covariance). We derive formal criteria for uniqueness that can be used in formal proofs as well as for measuring "uniqueness" empirically for glyph instances. The empirical study is helpful because although the approach to proving or disproving uniqueness is simple, the complexity of the formal expressions may "explode" if the basis glyph is defined w.r.t. a spectral basis. The visual comparison of the uncertain glyph for a best-fitting distribution with overlaid glyphs of the given ensemble members, indicates that the additional uncertainty can be encoded in a way that provides an idea of the given distribution. This is also emphasized by experiments where ensembles are generated by varying spectral parameters of the glyph. With this in mind, we believe that this chapter provides a valuable insight into encoding the effect of covariance on symmetric order tensors and the new glyphs provide a valuable tool for visual assessment of uncertain tensor data. All techniques have been applied to a number of data sets in 2D and 3D. We sincerely hope that these extend the palette of available tools for tensor visualization and therefore provide a valuable contribution for tensor analysis.

Part III

VISUALIZATION OF VECTOR FIELD ENSEMBLES

9 INTRODUCTION TO VECTOR FIELD ENSEMBLE DATA

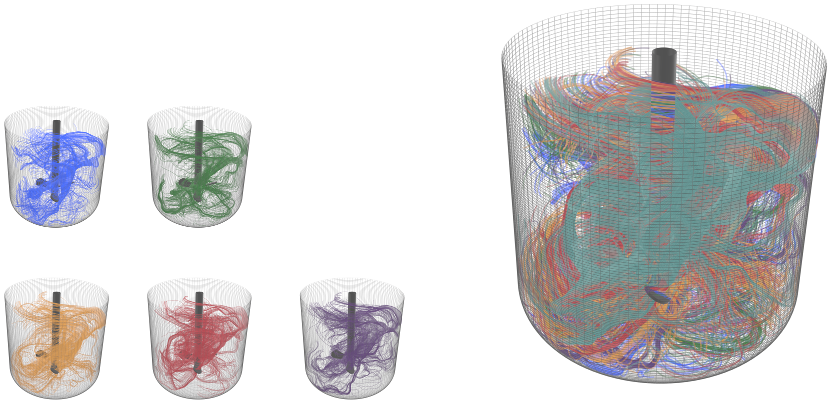


Figure 9.1: Vector field ensemble generated from sampling a simulated vector field describing flow inside a container with a rotating mixer at different time steps. Streamlines are used to visualize the flow in the distinct members (left). A spaghetti plot (right) of all streamlines combined can be used to visualize the whole ensemble but suffers greatly from visual cluttering and overlapping elements.

In Section 2.2 we defined an ensemble as a collection of "certain" fields which in total represent the uncertainty of the data [20]. For vector field ensembles we therefore deal with n vector fields $\mathbf{v}_1, ..., \mathbf{v}_n$ as members which all share the same domain and data dimension. Typically, such fields are produced by means of numerical simulations where either the simulation models, simulation parameters or initial configurations are changed for each simulation run [52]. Ensembles can however also arise from multiple measurements such as repeated experiments or medical observations [69]. Typical applications of vector field ensembles not only include meteorological and climatological research [19], [148], [159], but also data from oceanography [77], [78] as well as blood flow simulations [4]. For a general discussion and very detailed overview on ensemble data visualization, including other data types, we refer to Wang et al. [193].

Using ensembles allows to make stronger statements on the reliability of the data and therefore for better decisions and more accurate representations of complex phenomena. How to analyze the data is strongly dependent on the task and ranges from just gaining an overview of all the acquired information to comparing distinct ensemble members, also known as comparative visualization [50], [134]. Further tasks include testing and adapting simulation model parameters [194] or forecasts such as weather or hurricane track prediction [28]. The challenge lies within the drastically increased amount of data: not only do acquisition, storing, and processing need to be repeated for each member of the ensemble, several additional problems arise when trying to apply standard visualization techniques. Each vector field can be investigated individually by visualizing extracted features such as critical points or vortex core lines. To get an overview of the whole ensemble, however, these techniques can often not be directly applied to ensemble data. Especially when the number of members is high, going through each field individually and comparing them to each other is not only mentally challenging but also time consuming. Therefore, new visualization techniques need to be used and developed.

In this part of the thesis, we search for new approaches for the visualization of vector field ensembles. We first introduce background information that helps understanding feature extraction and visualization of single vector fields and put a special focus on line-type features. We then discuss existing works and approaches for dealing with the visualization of vector field ensembles in 2D and 3D. Finally, based on the observations made, we introduce a new operator that extracts features of vector field ensembles, where all vectors are approximately the same. We test this new operator on a number of vector field ensembles and discuss the results, challenges and future work.

9.1 BACKGROUND AND RELATED WORKS

Defining, extracting and visualizing features of multifield or ensemble data – either derived from one field or independent fields – is a challenging task with a variety of applications as presented by Verma and Pang [191]. Features can either be extracted from each single field or from information given by a combination of multiple or all fields as discussed by Obermaier and Peikert [126]. Depending on the data type of the members within the ensemble, different challenges arise. It makes sense to start the considerations with a single "certain" field of that type. When working with vector fields, there exist a variety of visualization techniques, which – following Post et al. [147] – can be grouped into four different groups. *Direct flow visualization*, *Texture-based flow visualization*, *geometric flow visualization* and *feature-based flow visualization*. As an in-depth discussion on each of the categories is beyond the scope of this work, we refer to related literature [106], [146], [147], [158] that offer an extensive overview.

Section 2.1.3 already lists several well-known and widely used visualization methods for certain vector fields including simple arrow glyphs and streamlines. We want to put a stronger focus on techniques that are not only used in the visualization of single vector fields but can also be found in vector field ensemble visualizations. Whereas isocontours can often be found when working with scalar fields, line-type features, that are frequently used for vector field data analysis, are streamlines and pathlines as well as feature lines such as vortex core lines.

Integral Lines

Integral lines such as pathlines and streamlines have become a well-established visualization method, as they make it possible to visualize and analyze the trajectory of massless particles within the flow. By only following these paths, domain experts can classify regions within the flow that indicate turbulent or laminar behavior, rotational flow or separation structures. In general, integral lines can be defined as

$$\frac{d}{d\tau}\mathbf{x}(\tau) = \mathbf{v}(\mathbf{x}(\tau)) \text{ with } \mathbf{x}(0) = \mathbf{x}_0$$

where τ denotes the arc-length coordinate along the curve and can also be interpreted as the integration time and $\mathbf{x}_0$ is the starting or seeding position. This means, that the tangent of every location on the curve shares exactly the same direction as the underlying vector field. When dealing with steady vector fields, pathlines and streamlines are identical. In time-dependent vector fields, pathline integration uses the vector information based on the changing time parameter, while streamline integration uses only the vector information of one specific time step. In single vector fields, there exists only one unique curve for each case of integral line at each location and point in time. For ensemble vector fields, this number is evidently increased, such that each ensemble member adds another possible curve per case. These lines can take different paths and intersect one another several times as is indicated in Figure 9.1. Besides information about the flow, integral lines can further encode additional scalar values using color. Often, they are colored based on the magnitude of the underlying vector field at sampled locations or based on the ensemble member they belong to.

Vector Field Features and Topology

Instead of direct visualization of each vector value, which introduces a lot of information at each location, important features can be extracted that represent the structure of a vector field. Several approaches therefore focus on the definition, extraction and visualization of such features to aid analysis and deeper understanding of such fields. Depending on the given data, features can be understood as an abstraction of the underlying field, as they often represent a certain aspect of interest, such as physically meaningful structures or patterns. In flows, we might be interested in locations where particles advected by the flow exhibit specific properties. Features can be extracted directly from the given field itself, or they take into account additional derived information like the acceleration field. Especially, when dealing with flow fields in a 3D domain, the areas where flow follows a swirling or rotating motion are often of particular interest. In applications ranging from engineering to medicine, extracting and visualizing such locations can help to understand the flow and its impact on phenomena. These include mixing of gases in combustions, testing aerodynamic properties of objects or blood flow through aneurysms as indicated in Figure 9.2. Finding a suitable definition for rotating flow, as well as finding and extracting these features has produced numerous works and we refer to Günther and Theisel for an extensive overview [54] of such techniques. While definitions such as the Q-criterion [85] or the λ_2-criterion [86] result in regions that are considered to belong to a vortex, we want to focus on the extraction of vortex core lines.

9.1.1 *Vortex Core Lines*

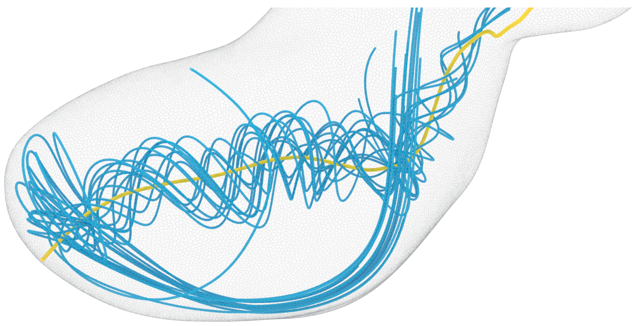


Figure 9.2: The path taken by the streamlines (blue) indicate regions of rotational flow behavior in a blood flow simulation through an aneurysm. The center of such rotational flow can be represented by a vortex core line (yellow).

A well-known feature to represent locations of rotating motion in flow fields are vortex core lines. These lines represent the centers of areas of swirling behavior. They do, however, not provide any additional information such as the border of rotational regions or the strength of the rotation. Especially the definition and extraction method of Sujudi and Haimes [175] has been used as a reliable technique for finding vortex core lines. They propose looking for structures where the velocity vector is parallel to the acceleration vector $\mathbf{v} \parallel (\nabla \mathbf{v})\, \mathbf{v}$ and the Jacobian matrix $(\nabla \mathbf{v})$ must have complex eigenvalues. The eigenvectors corresponding to the complex eigenvalues span a plane in which the swirling motion occurs. In other words, to fulfill the above criterion, the projected flow vector onto the plane needs to be the zero vector, which means that particles advected on the core line only move in the direction of the line itself and exhibit no swirling. This however favors straight lines, while core lines can form any arbitrary closed curve. Roth and Peikert [155] use higher-order derivatives to improve results by finding locations where velocity vectors are parallel to the second-order derivatives of particle motion $\mathbf{v} \parallel (\nabla \mathbf{c})\,\mathbf{v}$ where $\mathbf{c} = (\nabla \mathbf{v})\,\mathbf{v}$.

9.1.2 *The Parallel Vectors Operator*

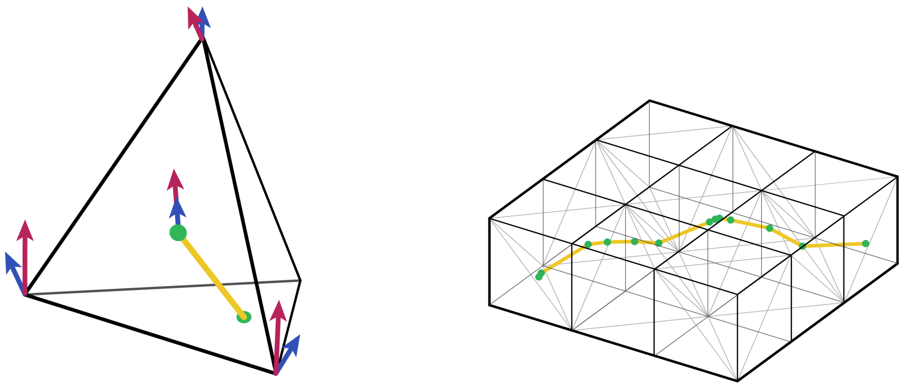


Figure 9.3: When two vector fields (red and blue) are defined on the same domain, the parallel vector operator finds all locations where both are parallel. When the domain is discretized on a mesh of tetrahedra, solutions (green) can be searched on the triangle surfaces first (left). Solutions belonging to the same tetrahedron are connected by lines (yellow) and solutions that are shared by neighboring tetrahedra are further connected (right).

The well-known *parallel vectors operator* was introduced by Peikert and Roth [142] and is used to find line-type features in a pair of vector or scalar fields. Formally, it is defined as follows: given two vector fields $\mathbf{v}_1(\mathbf{x}), \mathbf{v}_2(\mathbf{x})$ the parallel vectors ($\mathbf{PV}$) operator yields all locations where these fields are parallel, i. e.,

$$\mathbf{PV}(\mathbf{v}_1, \mathbf{v}_2) \;=\; \{\mathbf{x} \mid \mathbf{v}_1(\mathbf{x}) \parallel \mathbf{v}_2(\mathbf{x})\} \;.$$

The set $\mathbf{PV}(\mathbf{v}_1, \mathbf{v}_2)$ generally represents line structures also called *parallel vectors lines*.

For $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$, $\mathbf{PV}$ can be implemented as finding the roots of $\mathbf{v}_1(\mathbf{x}) \times \mathbf{v}_2(\mathbf{x})$. For $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^2$, the cross-product is replaced by the determinant of the matrix $(\mathbf{v}_1 \,|\, \mathbf{v}_2)$.

There are different algorithmic approaches to numerical root finding, the Newton-Raphson method is one of the most well-known. Their success often depends on an initial guess and the behavior of the function like multiplicity of roots or crossing zero versus touching without change of sign. Root finding is a non-trivial numerical task in general, even if the vector fields are given as polynomials, i. e., from interpolated data.

The setting is simpler for an appropriate discretization of the domain: Peikert and Roth [142] give an analytic solution for piecewise linear vector fields. The domain is partitioned such that parallel vector locations are searched on triangles. For instance, a bounded domain is partitioned into tetrahedral pieces, and the search space is restricted to their triangular faces. Each triangle supports a linear piece of the vector fields. The restriction to a locally two-dimensional search domain yields parallel vectors locations – if any – as isolated points, which are connected to line features in a post-process as displayed in Figure 9.3. Within each triangle Δ, solutions $\mathbf{PV}(\mathbf{v}_1(\mathbf{x}), \mathbf{v}_2(\mathbf{x}))|_{\mathbf{x}\in\Delta}$ are found by solving a generalized eigenvalue problem. This concept has been applied to a wide variety of applications. Roth and Peikert list several problems, including finding ridge and valley lines [63] or separation lines [91]. A lot of effort has been put into finding the centers of vortices which are the aforementioned vortex core lines. Whereas many methods differ in the extraction of solution points and their connection, most of them can be reformulated to be the result of the parallel vectors operator applied to the velocity field and a second, derived field.

A number of works extend and improve the operator. In general, these approaches can be divided into two stages: first, finding solution points on a grid, and second, tracing feature lines from these locations. Banks and Singer [8] use a predictor-corrector scheme that uses pressure information for line tracing, which was later combined with the λ_2 method [86] by Stegmaier [171]. Theisel et al. [180] show that tracing solution lines from extracted points or from seed points can be reformulated as a streamline integration in the feature flow field [179], [199]. This is extended for higher-order data by Pagot et al. [135]. Sukharev et al. [176] define an analytical tangent instead for tracing solution lines. A generalization of both of these is used by *PVSolve*, a method introduced by Van Gelder and Pang [189].

9.2 VISUALIZATION OF VECTOR FIELD ENSEMBLES

The previous section has introduced approaches and their realization for the visualization of single vector fields. When one is given ensemble data, e. g., as produced by different Computational Fluid Dynamics simulations, it is not only interesting to get an understanding of each individual field but rather to gain additional insight in similarities or dissimilarities of all ensemble members. Most techniques for single vector fields, however, cannot be directly used or adapted to offer such insight. To be able to make statements about the uncertainty encoded in the ensemble, the variability within the ensemble members needs to be analyzed. The more similar the members are, the lower the uncertainty and vice versa. To help with the analysis of such data, there exist two major approaches: either aggregate information of all members to single

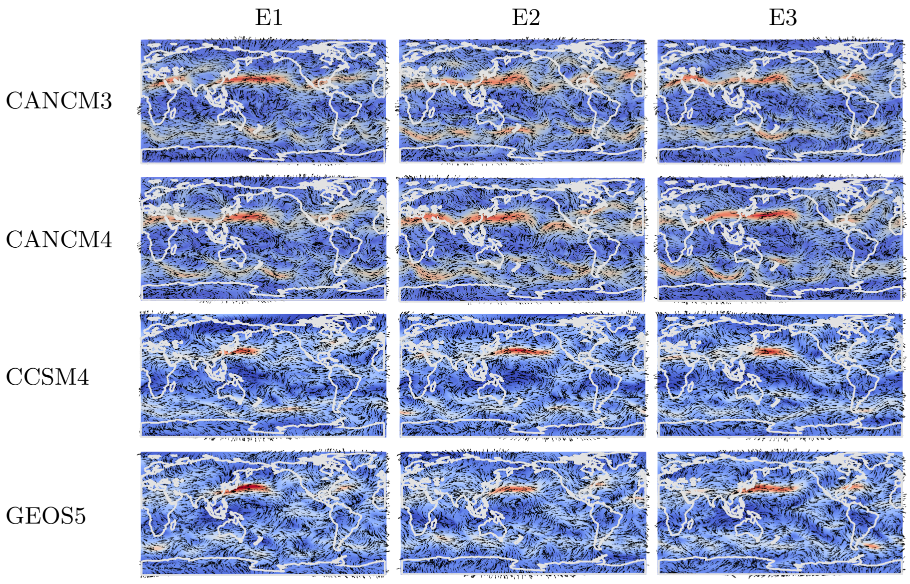


Figure 9.4: Comparative visualization via juxtaposition of selected members of the North American Multi-Model Ensemble (NMME): multiple side-by-side visualizations of different ensemble members with varying models and initial conditions for meteorological simulation. Vector direction indicated by arrow glyphs and vector magnitude mapped to color.

data values and visualize the resulting field or use visual composition of extracted features. Sometimes, a combination of both can also be applied. A straightforward implementation of the latter is to deal with each and every member separately and juxtapose the resulting visualization. This leads to what is known as a stamp map as displayed in Figure 9.4. For a very small numbers of flow fields, comparative visualization techniques such as the UFLOW system [116] which compares pairs of streamlines seeded at the same location, can be used and Verma and Pang [191] give an extended overview of such techniques for flow data. These techniques, however, struggle when a high number of ensemble members are present. This is not only due to the limitation of space, but especially due to massive amount of mental work that has to be used to compare each and every member with another and spot trends, differences, or similarities. When dealing with large ensembles of curves, like pathlines or other feature lines, that are extracted from each ensemble member, collecting and visualizing them collectively is often a fast and simple technique. This is known as a spaghetti plot [28], [36], which typically results in visual clutter as can be observed in Figure 9.1. Their limitations are discussed in several recent publications [42], [44], [122], [202]. The challenge of dealing with too many objects also applies to further compositing approaches such as superimposing or nesting [84].

Aggregating information of vector fields usually makes use of statistical tools to describe the variability at a given location or the whole vector field [37]. These range from simply calculating mean values of vectors to fitting distribution functions to the data. How similar ensemble members at given location are, can be expressed by comparing derived vector quantities such as magnitude, pairwise angles or even combinations. Sanyal et al. [159] propose to visualize the differences of the values of each ensemble to the mean with the help of their graduated uncertainty glyphs, which stack circles on top of each other. Also considering the orientation of the vectors, Jarema et al. [83] propose lobular glyphs that indicate the distribution of vector directions at each sample point.

As line-type features are a powerful tool to convey the movement and paths of particles, that are advected by vector fields, curve-oriented aggregation approaches have sparked a variety of visualization research. There exist several approaches that try to extract only meaningful selections from the different curve ensembles to give further insight. This selection can be based on geometric properties [177] or the point of view [55], but often takes into consideration their relevance to the whole ensemble. For instance, Guo et al. [56] create a variation field that is filtered for pathlines that best characterize the differences between different fields. Lui et al. [115] create a similar variation field, based on what they call the Longest Common Subsequences of pathlines. An alternative approach is using features of a single field, which are then modulated by the additional data, as seen in modulated streamlines or streamtubes [187].
A standard approach that reduces the number of objects visible into trends is clustering, only rendering representatives of each cluster or bundling of similar curves. Clustering similar curves into groups has been successfully applied to stream- and pathlines [127], [128], [154]. When pathlines pass through a shared location, several approaches make use of visualization techniques known from statistics. Ferstl et al. [42], [44] cluster such pathlines in major trends and provide a median as well as a region of confidence, which they call variability plot. While a similar technique was introduced for isocontours [202], Mirzargar et al. [122] summarize trends of stream- and pathlines by introducing curve boxplots, additionally showing outliers and selected pathline members. Kern and Westermann [92] propose a clustering technique for ensembles of core lines of jet-streams, that are often disconnected line segments and thus highly fragmented line sets.
A very different approach, that forms a research field on its own, is to deal with an uncertain vector field instead of a vector field ensemble. In such a field, each location has an uncertain vector value assigned to it, which can be represented by a probability density function. This allows for new definitions and extraction of known vector field features for uncertain vector fields such as integral lines or topological features as proposed by Otto et al. [131], [132] and Petz et al. [143]. Further, this led to new definitions for vortices in uncertain vector fields as

proposed by Otto and Theisel [133], the introduction of texture-based visualizations [3], [22] and glyphs [205]. Obermaier and Joy [125] clarify however, that uncertain data neither contains information on how different ensemble outcomes relate, nor does it allow for a discrete analysis of the input parameters, which is an important task for domain experts, when improving their models.

Another field of research is concerned with giving domain experts the option to explore more than one quantity at the same time, thus combining multiple linked visualizations [148]. This might include one or more of the techniques mentioned above as well as further tools not restricted to vector fields relevant to the application. Especially weather forecast ensembles have led to a variety of tools for the visual analysis in combined visualizations [43], [148], [150]. Whereas many of the visualization approaches are automatic or require the user to define some parameters, Liu et al. [113], [114] propose a framework, that allows the user to define features themselves such as user defined pathlines and find these features within the ensemble.

9.3 CHALLENGES IN VECTOR FIELD ENSEMBLE VISUALIZATION

It is only due to the recent advancements in computing hardware that repeated calculations of complex simulations as well as storing and processing massive amounts of data has become a feasible task. Therefore, the visualization of ensemble data, especially vector field ensembles, is still a young and advancing field. This means, that not only known techniques need to be adapted to this new kind of data, but also new approaches and tools must be developed. As the goal of visualization is to enable the effective and efficient analysis and exploration of data, such techniques and tools must be designed with several things in mind: not only the efficient handling of massive amounts of data in terms of storing, processing and rendering must be dealt with algorithmically. Also, the support of analysis tasks of the users, which leads to reduced complexity and faster and more accurate understanding of the data should be a major intention in the development process.

The previous section has already listed several of these adaptations, especially when dealing with line-type features. We may treat each input field individually, compute such feature lines and then chose a suitable visual representation. Whereas techniques such as clustering could be considered, they all require the notion of distance, and many different distance measures are available. Even though, clustering has been successfully applied to isocontours, stream- and pathlines, it is, however, not applicable to several feature lines such as vortex core lines, which are an important tool for vector field analysis. Such line features that result from the ensemble members may differ significantly in shape and topology: they may consist of several unconnected parts, for example, due to the filtering, or some of the ensemble members may give no such lines at all. There is no straightforward answer to the question how to incorporate these cases into a stable clustering algorithm. Calculating statistics to show representatives, as done with the aforementioned curve boxplot or variability plots may be applied to the set of core lines. These approaches rely either on implicit curve representations or on a common parametrization of the curves. Such parametrization does not exist for **PV** lines as input curves. Moreover, due to the unconnectedness and even non-existence of core lines in individual fields, we are not aware of straightforward approaches to construct a common parametrization. And similarly, we are not aware of straightforward extensions of variability plots to sets of core lines.

In the following chapter, we seek to build upon the extraction of core lines in vector fields and extend it for vector field ensembles. We focus on extracting a set of line-type features that take into account the information given by all ensemble members at the same time.

10 AN APPROXIMATE PARALLEL VECTORS OPERATOR FOR MULTIPLE VECTOR FIELDS

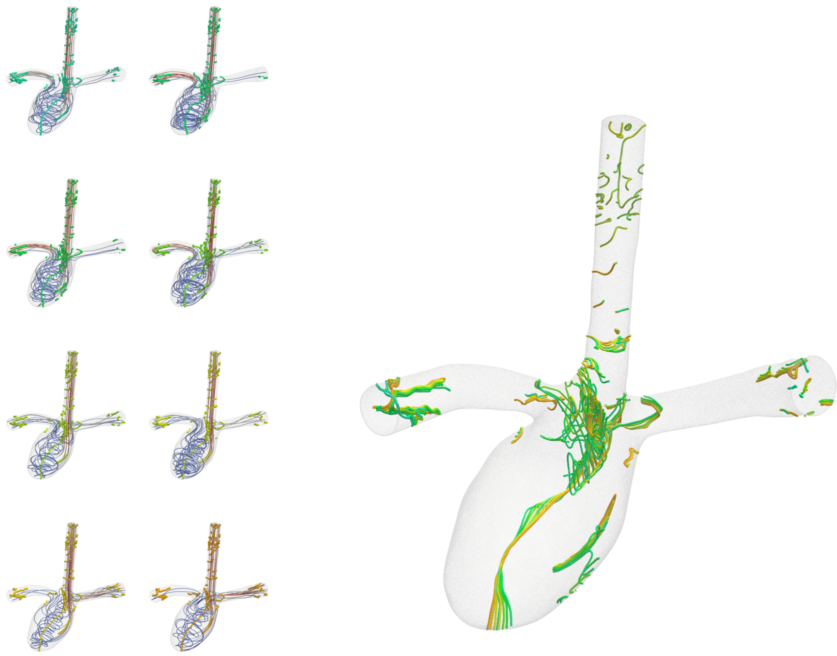


Figure 10.1: Left: eight different members of an ensemble dataset that represent different outcomes of CFD simulations of blood flow through an aneurysm with varying pressure parameters. Each member is visualized with several streamlines seeded at the flow inlet as well as parallel vectors feature lines calculated from the derived acceleration field. Every member has a different color defined for parallel vectors lines. Right: a spaghetti plot visualization of all parallel vectors lines.

This chapter is based on the publication:

T. Gerrits, C. Rössl, and H. Theisel
An Approximate Parallel Vectors Operator for Multiple Vector Fields
Computer Graphics Forum (Proc. EuroVis), 2018

The parallel vectors operator, illustrated in Section 9.1.2, is a concept that enjoys a high popularity in visualization and other communities because it is conceptually simple, generic, fast and easily computable,

and applicable to a variety of problems. As we have listed, it can be used for the extraction of vortex core lines, finding ridge structures, or finding bifurcation lines in flow fields. The **PV** operator yields all locations where two vector fields are parallel. These are structurally stable line structures. Different types of input vector fields open a variety of applications for the **PV** operator.
As we are dealing with ensemble flow data sets, i. e., a number of vector fields in a common spatial domain, all describing the same flow phenomenon with slightly varying parameters, in this chapter, we want to search locations of vortex core lines that simultaneously describe the vortical behavior of all fields best. Analyzing the literature in Section 9.2, this problem can be solved using one of two general strategies: either extract vortex core lines for each of the velocity fields along with a visual representation of the resulting multiple line sets – including line bundling, line clustering, or finding best representatives –, or directly extract line structures that represent the vortices of all fields best in an approximate sense. In this chapter, we present our approach using the second strategy. We introduce a new generic concept called the *Approximate Parallel Vectors (**APV**) Operator* that is applied to an arbitrary number of vector fields.

Given m 3D vector fields $\mathbf{v}_1, ..., \mathbf{v}_m$, it is generally not useful to search for locations where all vectors are parallel. For $m > 2$, such structures – if there exist any at all – are structurally unstable: adding noise to the fields will destroy line structures with all parallel vectors. Instead, we develop a new operator that gives stable line structures at locations where all m fields are maximally – but generally not perfectly – parallel. The **APV** operator is rather simple in terms of computation: from the given $\mathbf{v}_1, ..., \mathbf{v}_m$ we compute two derived fields $\mathbf{a}, \mathbf{b}$ and apply the **PV** operator to these. Despite its computational simplicity, the **APV** operator requires a rigorous mathematical analysis of its properties to make it applicable.

10.1 THE APPROXIMATE PARALLEL VECTORS OPERATOR

Given are m vector fields $\mathbf{v}_i(\mathbf{x})$ with $\mathbf{v}_i : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ and $i = 1, \ldots, m$. We assume simultaneous evaluation at the same location $\mathbf{x}$ and write $\mathbf{v}_i$ for short.
The *parallel vectors operator* is defined for $m = 2$: $\mathbf{PV}(\mathbf{v}_1, \mathbf{v}_2)$ gives all locations where $\mathbf{v}_1(\mathbf{x})$ and $\mathbf{v}_2(\mathbf{x})$ are parallel. These are typically *line* structures. For $m > 2$ distinct fields, e. g., multiple fields of an ensemble, we would generally expect no such locations or just isolated points if we require that $\mathbf{v}_i \parallel \mathbf{v}_j$ for all $i \neq j$. The higher m the “more restrictive” this condition is. Our goal is the construction of a new operator that

- relaxes the condition and measures if m vector fields are *approximately* parallel, and

- does so in a parameter-free way by measuring if *two derived fields* are parallel using the *parallel vectors* operator $\mathbf{PV}$.

We stack all vectors $\mathbf{v}_i$ as columns in the matrix $\mathbf{V} = (\mathbf{v}_1 \,|\, \dots \,|\, \mathbf{v}_m)$ and compute the average

$$\mathbf{a} \;=\; \frac{1}{m}\sum_{i=1}^{m}\mathbf{v}_i \;=\; \frac{1}{m}\mathbf{V}\,\mathbf{1}\ ,$$

where $\mathbf{1} \in \mathbb{R}^m$ is a column vector with all entries equal to 1. If we subtract the mean vector field from all fields, we obtain columns $\mathbf{v}_i - \mathbf{a}$ in the matrix

$$\mathbf{D} \;=\; \mathbf{V} - (\mathbf{a} \,|\, \dots \,|\, \mathbf{a}) \;=\; \mathbf{V} - \mathbf{a}\,\mathbf{1}^{\mathrm{T}}\ .$$

Then the symmetric operator

$$\mathbf{D}\mathbf{D}^{\mathrm{T}} \;=\; \sum_{i=1}^{m}(\mathbf{v}_i - \mathbf{a})(\mathbf{v}_i - \mathbf{a})^{\mathrm{T}} \;\in \mathbb{R}^{3\times 3}$$

measures the covariance of the vector fields and thus how much and in which directions the fields “spread away” from the average field. The quadratic form $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ is positive definite if $\mathbf{v}_i$ are linearly independent. Its spectral decomposition gives the directions of minimum and maximum variance as eigenvectors. This is also known as the Principle Component Analysis (PCA).
We take the mean vector field $\mathbf{a}$ as a representative for the whole ensemble $\{\mathbf{v}_i\}$. We define $\{\mathbf{v}_i\}$ being *approximately parallel* if their variance obtains a maximum in direction of the mean $\mathbf{a}$. A necessary condition is that $\mathbf{a}$ must be an *eigenvector* of $\mathbf{D}\mathbf{D}^{\mathrm{T}}$, i. e.,

$$\mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a} \;=\; \lambda\,\mathbf{a}\ . \tag{10.1}$$

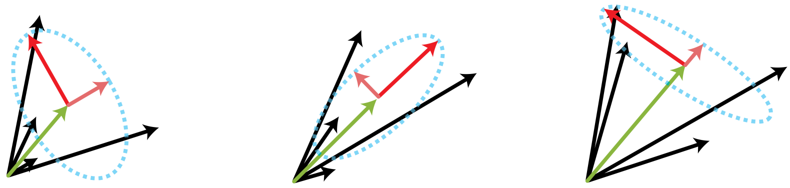
A further condition requires that the corresponding eigenvalue is maximal, i.e.,

$$\lambda = \lambda_{\max}(\mathbf{D}\mathbf{D}^{\mathrm{T}})\ . \tag{10.2}$$

APPROXIMATE PARALLEL VECTORS. We define the *Approximate Parallel Vectors* operator ($\mathbf{APV}$) as follows:
Let $\mathbf{b} = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a}$ then

$$\mathbf{APV}(\mathbf{v}_1, \dots, \mathbf{v}_m) \;=\; \mathbf{PV}(\mathbf{a}, \mathbf{b})\ .$$

In this definition the necessary condition 10.1 is expressed by the *parallel vectors operator* as $\mathbf{PV}(\mathbf{a}, \mathbf{b}) \Leftrightarrow \mathbf{a} \parallel \mathbf{b} \Leftrightarrow \mathbf{a} = \lambda\,\mathbf{b}$.
We reduced the definition of $\mathbf{APV}$ to the standard $\mathbf{PV}$ operator. This reduces the problem of finding $\mathbf{APV}$ lines to the application of $\mathbf{PV}$ and makes the implementation straightforward.



$\mathbf{x} \notin \mathbf{APV}$ $\mathbf{x} \notin \mathbf{fAPV}$ | $\mathbf{x} \in \mathbf{APV}$ $\mathbf{x} \in \mathbf{fAPV}$ | $\mathbf{x} \in \mathbf{APV}$ $\mathbf{x} \notin \mathbf{fAPV}$

Figure 10.2: A location $\mathbf{x}$ is part of an **APV** line if the mean vector $\mathbf{a}$ is an eigenvector of the covariance matrix $\mathbf{DD}^{\mathrm{T}}$. The filtered **fAPV** operator requires additionally that this is the eigenvector corresponding to the largest eigenvalue.

FILTERED APV. The **APV** operator computes eigenvectors and uses only the necessary condition (10.1). Among all **APV** lines, we are only interested in those where the mean vector $\mathbf{a}$ is the major eigenvector that corresponds to the *largest eigenvalue*. We implement the missing condition (10.2) as a "filter" and define the *Filtered Approximate Parallel Vectors* operator (**fAPV**) as

$$\begin{aligned}\mathbf{fAPV}(\mathbf{v}_1,\ldots,\mathbf{v}_m) &= \{\mathbf{x}\,|\,\mathbf{DD}^{\mathrm{T}}\mathbf{a} \;=\; \lambda_{\max}(\mathbf{DD}^{\mathrm{T}})\,\mathbf{a}\}\\ &= \{\mathbf{x}\in\mathbf{PV}(\mathbf{a},\mathbf{b})\,|\,\mathbf{b}=\lambda_{\max}\mathbf{a}\}\\ &\subset\; \mathbf{APV}(\mathbf{v}_1,\ldots,\mathbf{v}_m)\;.\end{aligned}$$

Figure 10.2 shows – from left to right – examples of a non-feature point (no alignment), a point that is in **fAPV** and in **APV** (alignment with the major eigenvector), and a point that is in **APV** but not in **fAPV** (alignment with a minor eigenvector).

10.2 PROPERTIES OF APV

In this section, we summarize a number of properties of the **APV** operator. The proofs have been established by Theisel and Rössl and are provided in the Appendix F. In the remainder of this section, the matrix of stacked vector fields $\mathbf{V}$, the mean vector field $\mathbf{a}$ and the derived field $\mathbf{b}$ are defined as in Section 10.1.

INDEPENDENCE OF ORDER. For any permutation π of $(1,2,\ldots,m)$:

$$\mathbf{APV}(\mathbf{v}_{\pi_i},\ldots,\mathbf{v}_{\pi_m}) \;=\; \mathbf{APV}(\mathbf{v}_i,\ldots,\mathbf{v}_m)\;. \qquad (\mathbf{P1})$$

RELATION TO PV. For $m=2$, **fAPV** and **PV** coincide:

$$\mathbf{fAPV}(\mathbf{v}_1,\mathbf{v}_2) \;=\; \mathbf{PV}(\mathbf{v}_1,\mathbf{v}_2)\;. \qquad (\mathbf{P2})$$

DEPENDENCE ON SCALING. The **PV** operator is invariant to scaling, i. e., $\mathbf{PV}(\mathbf{v}_1,\mathbf{v}_2) = \mathbf{PV}(s_1\mathbf{v}_1,s_2\mathbf{v}_2)$ for any nonzero scalars

s_1, s_2. By construction, the **APV** operator depends on the scaling of the vector fields as different scales weight their contributions to the covariance matrix $\mathbf{DD}^\mathrm{T}$. We study $\mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_m, s\mathbf{v}_{m+1})$ for the edge cases $s = 0$ and $s \to \infty$:

$$\begin{aligned} \mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_m, 0\,\mathbf{v}_{m+1}) &= \mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_m) && (\mathbf{P3}) \\ \lim_{s\to\infty} \mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_m, s\,\mathbf{v}_{m+1}) &= \mathbf{PV}(\mathbf{a}, \mathbf{v}_{m+1}) && (\mathbf{P4}) \end{aligned}$$

The second Property (**P4**) is remarkable: if one single field is scaled extremely such that it "dominates" all other fields, and both the average $\mathbf{a}$ as well as $\mathbf{b}$ converge to $\mathbf{v}_{m+1}$, the **APV** operator gives a well-defined line.

ADDING NEW VECTOR FIELDS. The **APV** operator is invariant to adding a scaled mean vector field or zero fields. For any scalar s:

$$\begin{aligned} \mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_m, s\sum_i \mathbf{v}_i) &= \mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_m) && (\mathbf{P5}) \\ \mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_m, \mathbf{0}, \ldots, \mathbf{0}) &= \mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_m) && (\mathbf{P6}) \end{aligned}$$

If we add the same field $\mathbf{w}$ extremely often, **APV** still yields a well-defined result:

$$\lim_{k\to\infty} \mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_m, \underbrace{\mathbf{w}, \ldots, \mathbf{w}}_{k \text{ times}}) = \mathbf{PV}(\mathbf{VV}^\mathrm{T}\mathbf{w} - m\,||\mathbf{w}||^2\,\mathbf{a}, \mathbf{w})\ . \qquad (\mathbf{P7})$$

10.3 DISCRETIZATION AND VISUALIZATION

All datasets are given as sets of piecewise linear vector fields that are defined w.r.t. a tetrahedral partition of the domain. We apply the **APV** operator on all triangular faces of the tetrahedra. This way, we find point locations on faces that are connected by line segments within tetrahedra, which gives discrete **APV** lines. This is the same modus operandi as for parallel vectors **PV**as illustrated in Figure 9.3 of Section 9.1.2.
At each feature point location $\mathbf{x}$, we can quantify the "spread" of vectors. For eigenvalues $\lambda_1 \le \lambda_2 \le \lambda_3$ of $\mathbf{DD}^\mathrm{T}$, we measure the ratio

$$\varepsilon \;=\; \frac{\lambda_3 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \cdot \chi \quad \text{with} \quad \chi = \begin{cases} +1 & \text{if } \mathbf{x} \in \mathbf{fAPV} \\ -1 & \text{else} \end{cases}$$

We use the additional sign χ to distinguish between locations that are part of the filtered **fAPV** and locations the are non-filtered **APV** features but not part of **fAPV**. For the latter, the mean vector is aligned to one of the minor eigenvectors corresponding to λ_1 or λ_2. We color code ε (see Figure 10.6) and we place spheres in regions where

$\varepsilon \approx 0$, i. e., $\lambda_3 \approx \lambda_2$ and thus the major eigenvector is undefined. As **PV** feature lines always form closed lines, so do **APV** lines. Figure 10.7 compares the different filtering options:

- Figure 10.7 (a) shows *all* closed feature lines *without* filtering.
- In Figure 10.7 (b), connected components are discarded if $\varepsilon < 0$ for *all* locations, i. e., the remaining lines contain at least one **fAPV** location ($\varepsilon > 0$). This is a non-local filter criterion on structures that maintains closed lines.
- In Figure 10.7 (c), all line segments which are spanned by a location with $\varepsilon < 0$ are discarded. This is essentially the "pointwise" **fAPV** filter, which generally yields a set of open feature lines.

10.4 APPLICATIONS AND RESULTS

In the following, we demonstrate the approximate parallel vector fields operator. We start with an analytic ensemble and then examine three different ensemble datasets from numerical simulations. For all shown applications, the data consists of a number of 3D velocity fields $\mathbf{v}_1, ..., \mathbf{v}_m$. We use the **APV** operator to analyze them in two ways:

- To analyze the alignment of the velocity fields, we compute $\mathbf{APV}(\mathbf{v}_1, ..., \mathbf{v}_m)$. This gives corelines of best alignment of the velocity fields. Along these lines the ensemble members have a locally maximal parallelity to each other.
- To analyze the alignment of the vortex core lines of all fields, we additionally consider the accelerations fields $\mathbf{c}_1 = (\nabla \mathbf{v}_i)\ \mathbf{v}_i$ of all ensemble members. Instead of computing the vortex core lines $\mathbf{v}_i \parallel \mathbf{c}_i$ of each ensemble member, our approach gives the best approximated vortex core lines of all fields by considering $\mathbf{APV}(\mathbf{v}_1, ..., \mathbf{v}_m, \mathbf{c}_1, ..., \mathbf{c}_m)$.

The acceleration fields are estimated on the same tetrahedral partition as the velocity fields and represented as linear pieces in the tetrahedral cells.

10.4.1 *Linear Vector Field Ensemble*

A family of linear vector fields is given by

$$v(x, y, z) = \begin{pmatrix} 0 & a & 0 \\ -a & 0 & 0 \\ 0 & 0 & b \end{pmatrix} \begin{pmatrix} x + x_0 \\ y + y_0 \\ z + z_0 \end{pmatrix}.$$

We created an ensemble of 250 velocity fields with randomly chosen parameters $a, b, x_0, y_0, z_0 \in [-1, 1]$. Then each member describes a rota-

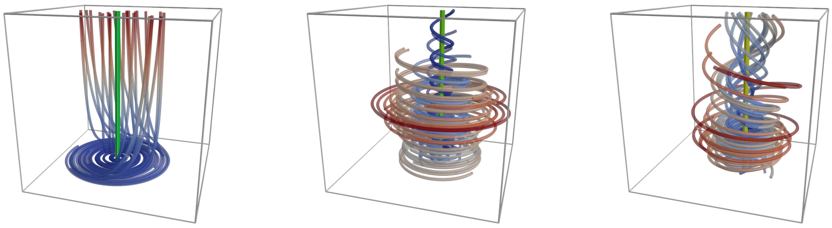


Figure 10.3: Streamlines and corresponding **PV** lines for three different members of an ensemble of random linear flows rotating around a vertical axis with varying locations, rotational direction and speed.

tional flow around a vertical core line with random location, rotational speed and direction. Figure 10.3 displays some streamlines. For each member and its acceleration, their **PV** lines are vertical lines that intersect the x-y-plane close to the origin. Figure 10.4 shows seven of these **PV** lines, colored in different shades of green and yellow.

A naïve approach to finding feature lines of the ensemble, consists in applying the standard **PV** operator to the mean vector field and its acceleration field. In the example, this gives the blue line in Figure 10.4, which appears at a significant offset from the individual members' **PV** lines. In contrast, the **APV** line – displayed in red – intersects the x-y-plane as expected close to the origin, i. e., which is obviously the better representative or "mean feature line". The reason for the misalignment of the blue line lies in the fact that averaging the ensemble members results in a field with an ill-conditioned Jacobian matrix, i. e., the matrix' determinant is close to zero. Finding extremal lines in linear vector fields with such Jacobians gives unstable results.

In order to compare and demonstrate alignment, we display glyphs for velocity and acceleration vectors of all members sampled at three different locations: on a **PV** solution line of a single member, on the **PV** line of the mean velocity field, and on the **APV** line. The **APV** solution indeed shows the smaller spread and hence the better alignment of vectors. This is an indicator for the plausibility of our approach.

Figure 10.5 shows two scalar fields that are derived from the velocity ensemble: the accumulated angles (Section 10.5.2) and the accumulated norm of the cross products (Section 10.5.2) and provide an alternative measure for "how parallel" vectors are at a domain point. The lowest values in both fields are found in the center of the domain around a vertical line which coincides with the location of the **APV** feature line. This fact again indicates plausibility of **APV** features. We will discuss in Section 10.5.2, why we prefer the definition of **APV** over alternative concepts of approximately parallel line features.

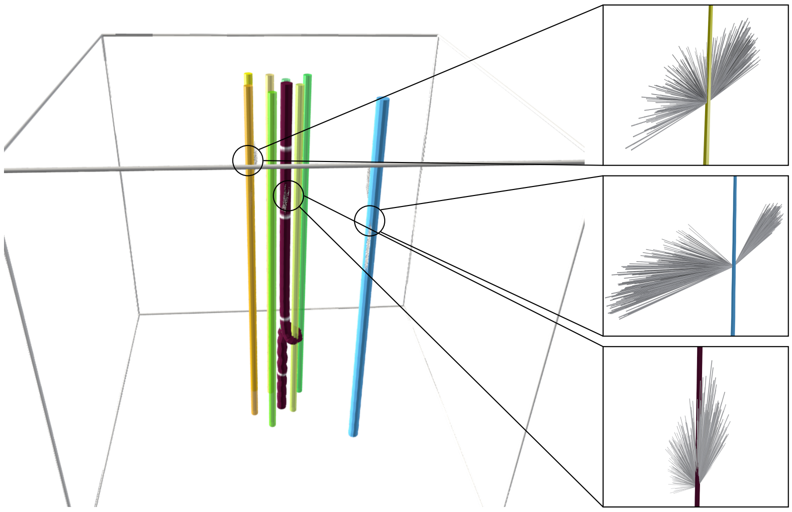


Figure 10.4: Several **PV** feature lines of the ensemble are shown in different shades of green and yellow. The blue **PV** line of the mean velocity and its acceleration appears at a significant offset, while the red **APV** line is centered within the members' **PV** lines and runs closely through the origin. The close-ups right display velocity and acceleration vectors of all members displayed at three distinct points on a member, on **PV** and on **APV**: The alignment of vectors seems best for the **APV** sample.

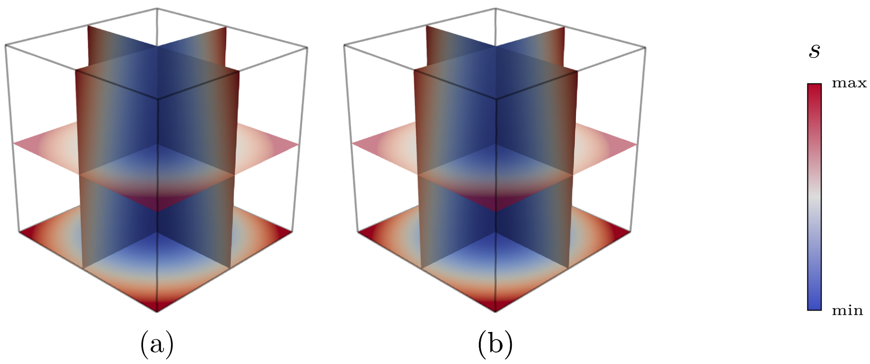


Figure 10.5: Scalar fields s that measure local alignment of ensemble members. (a) Accumulated norm of the cross product of all vectors at a given location (see Section 10.5.2). (b) Accumulated angle of all vectors at a given location (see Section 10.5.2). Both fields indicate that locations of high alignment of all vectors lie vertically in the center of the domain.

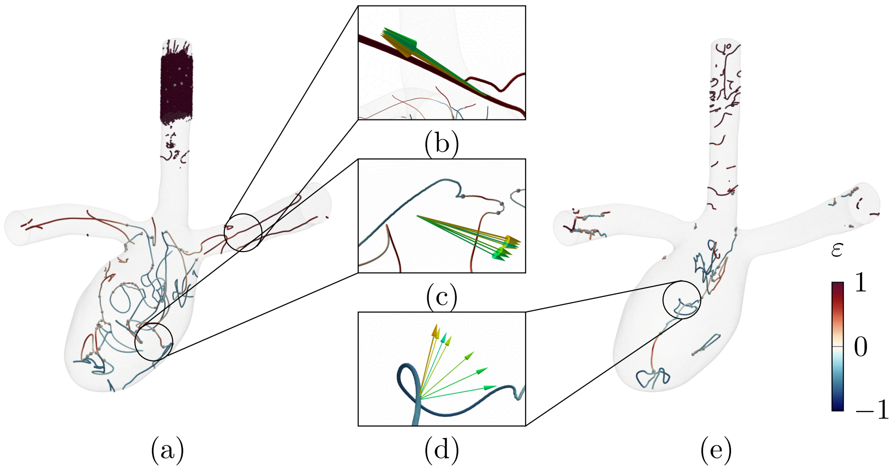


Figure 10.6: **APV** lines for aneurysm ensemble: (a) using all eight velocity fields. (e) using all eight velocity fields and their acceleration fields. The closeups give examples for the color coding (right): (b) Locations with input vectors aligned closely with the major eigenvector are depicted in red color. (c) Is an example of a non-feature location. (d) Blue lines refer to locations with vectors aligned closely to one of the minor eigenvalues. They would be removed in filtered **fAPV** in Figure 10.7.

10.4.2 *Aneurysm Ensemble*

The Computational Fluid Dynamics Rapture Challenge 2013 [17], [81] presented a velocity field ensemble data that was created by different hemodynamics simulations inside an aneurysm geometry. Eight different blood flow fields were simulated by varying the outlet boundary conditions. This included a zero-pressure condition for both outlets as well as seven simulations, where the pressure was split between the outlets and changed in steps of 10% from 20% to 80% and vice versa. Figure 10.1 shows streamlines of blood flow of each of the eight members of the ensemble as well as the feature lines extracted with the standard **PV** operator. Each member has a specific color assigned ranging from green to yellow so the corresponding feature line can be located in the combined spaghetti plot visualization on the right.
Figure 10.6 (a) shows **APV** feature lines derived from eight ensemble members. Near the inlet (top) there is no significant difference between members, and therefore many insignificant features are found. As the flow progresses, the members start to divert. The closeups (b)-(d) show single point locations $\mathbf{x}$ of different regions with vectors at $\mathbf{x}$ drawn as arrows: (b) $\mathbf{x} \in$ **fAPV**, (c) $\mathbf{x} \notin$ **APV**, and (d) $\mathbf{x} \in$ **APV** but $\mathbf{x} \notin$ **fAPV**. The latter is a mean vector aligned with a minor eigenvector and would be removed by filtering. The resulting feature lines are shown in Figure 10.6 (e).

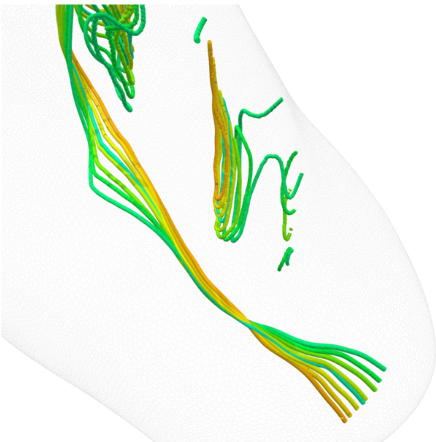
(a) (b) (c)

Figure 10.7: The **APV** operator finds structures of closed lines, which can be filtered: (a) **APV** lines w/o filtering. (b) Only lines with at least one **fAPV** segment. (c) Structures of **fAPV** line segments may be non-closed.

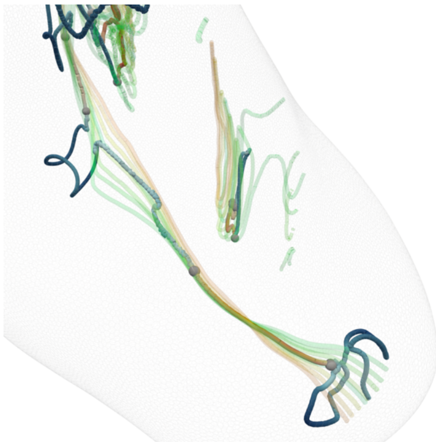
The resulting **APV** lines give the locations, where all **PV** lines of the ensemble are best aligned. This behavior can be observed in Figure 10.8: we observe high ε near the locations where all **PV** lines are close to each other. However, in regions where the **PV** lines start to diverge from each other, ε decreases and eventually turns negative. In these locations, the mean vector is aligned with a minor eigenvector as seen in Figure 10.6 (d). Finally, we remark that computing **PV** lines for each ensemble member and its derived acceleration field consumes significantly more time than one single application of the **APV** operator for the same data.
To reduce visual clutter, we apply filtering as seen in Figure 10.7.

10.4.3 *Helicopter in Ground Proximity*

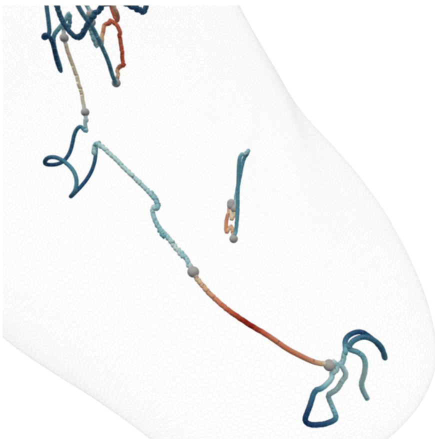
This dataset by Kutz et al. [103] simulates wind flow near a helicopter that is hovering over ground. Figure 10.9 shows streamlines for a single time step. We sampled the flow field uniformly in time such that the rotor revolution increases by 10° for each time step and collect six time steps in total in an ensemble. The difference between the ensemble members is relatively small. This can be seen by computing **PV** features for each member and its derived acceleration field as depicted in Figure 10.11 (a). Figure 10.10 (a) shows all **PV** feature lines in a combined visualization. We compute **APV** features on all velocity fields of the ensemble. This is shown in Figure 10.10 (b) and (c). Figures (d) and (f) include in addition the derived acceleration fields.
Note that this dataset is special, due to the lack of variance between **PV** features. In this case the **APV** features resemble the "mean locations" of **PV** features (although it is unclear how to average core lines). Indeed, the comparison of **PV** and **APV** features in Figure 10.11 suggests that the latter express the essence of the "ensemble" of **PV** core lines. The



(a)



(b)



(c)

Figure 10.8: (a) Spaghetti plot of all **PV** feature lines. (b) The **APV** operator finds locations where the **PV** feature lines are close to each other. (c) **APV** lines of all velocity and acceleration fields.

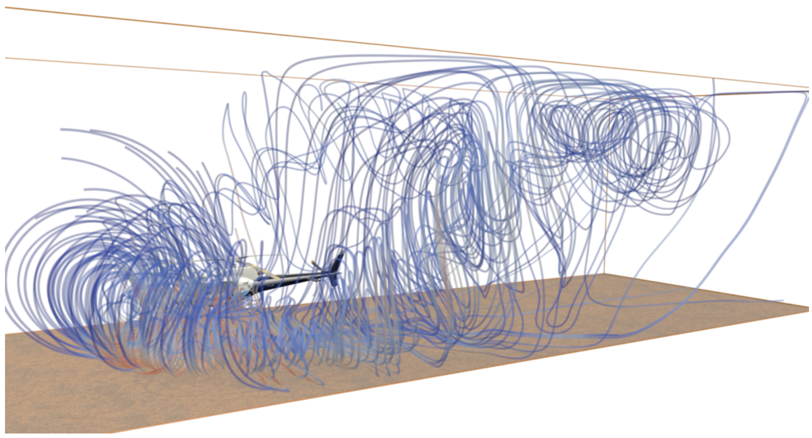


Figure 10.9: Streamlines of the wind flow around a hovering helicopter for a fixed time step. Swirling behavior can be seen behind the helicopter.

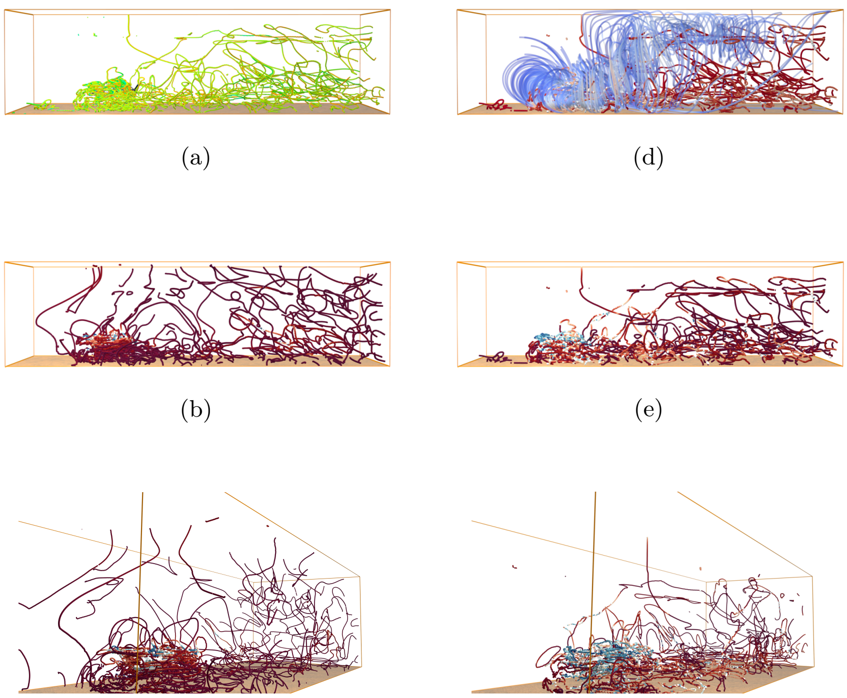
additional streamlines for one single ensemble member in Figure 10.10 (d) suggests that **APV** lines can yield locations similar to vortex core lines.

10.4.4 *Rotating Mixer*

The last ensemble was created by sampling a CFD simulation of flow inside a container with a rotating mixer. Six time steps were chosen in a way, that the three blades inside the container rotate by 120° each time such that the blade geometry overlaps exactly for each ensemble member. This is a turbulent flow and the members vary greatly. The **PV** feature lines for one single member's velocity and acceleration shows already a complex behavior, which makes it difficult to identify interesting structures. This is shown in Figure 10.13 (a). Figure 10.13 (b) shows **PV** feature lines for all members. We compute **APV** features for velocity and acceleration fields. The result is shown in Figure 10.13 (c). Filtering reduces the feature regions significantly: Figure 10.13 (c) shows **fAPV** features. The colors in (d) show regions with low "spread" of ensemble members near the blades of the mixer. Figures (e) and (f) show **APV** and **fAPV** features for only the given velocity members without additional acceleration fields.

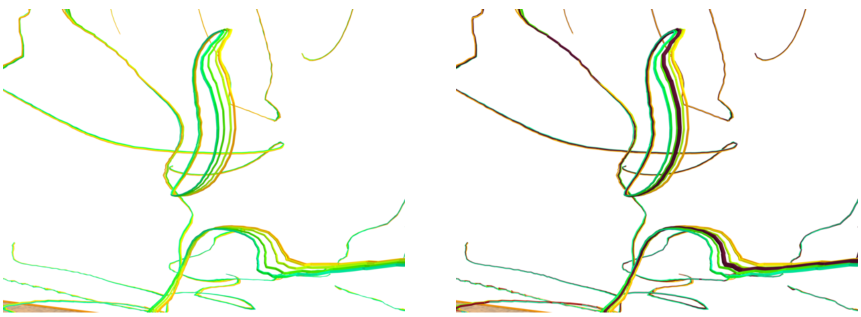
10.4.5 *Performance*

Table 10.1 summarizes the sizes of the ensemble data and timings for feature extraction. All times were measured on an Intel Core i7-6700K CPU at 4GHz with 32GB RAM available, this was always enough memory to store the data. All algorithms were implemented in C++. The time that is required for computing the derived fields $\mathbf{a}$ and $\mathbf{b} = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a}$ depends linearly on the number of input fields. For



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

Figure 10.10: Helicopter dataset. (a) All **PV** core lines from each ensemble member displayed in different shades of green. They lie very close to each other. (b)-(c) **APV** features for all velocity fields and derived acceleration fields. (d)-(f): **APV** features for only the velocity members. (d) additionally shows streamlines for one of the members.



(a) (b)

Figure 10.11: (a) A combined visualization of all core lines extracted with the **PV** operator from each ensemble member and its derived acceleration field. (b) Same with an additional overlay of **APV** feature lines for the same input data.

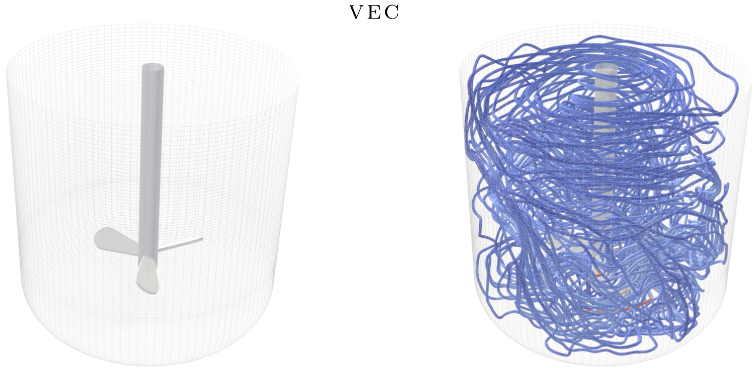


Figure 10.12: Streamlines for one time step of rotating mixer dataset.

Dataset	#Fields	#Vertices	#Vectors	**APV**	**PV**
Aneurysm Velocity	8	$3,501,487$	$28,011,896$	974ms	$261,571$ms
Aneurysm Vel. + Acc.	16	$3,501,487$	$56,023,792$	$1,538$ms	$261,668$ms
Helicopter Velocity	6	$4,810,000$	$28,860,000$	$1,231$ms	$310,260$ms
Helicopter Vel. + Acc.	12	$4,810,000$	$57,720,000$	$1,884$ms	$288,672$ms
Mixer Velocity	6	$1,258,759$	$7,552,554$	312ms	$99,083$ms
Mixer Vel. + Acc.	12	$1,258,759$	$15,105,108$	465ms	$97,529$ms

Table 10.1: Timings for the given ensemble data. The column **APV** refers to the computation of the derived fields, the mean $\mathbf{a}$ and $\mathbf{b} = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a}$. The **PV** column refers to the extraction of $\mathbf{PV}(\mathbf{a}, \mathbf{b})$.

a moderate number of ensemble members this cost is not significant compared to the next step: the time for the subsequent computation of $\mathbf{PV}(\mathbf{a}, \mathbf{b})$ for all triangular faces is constant for the given tetrahedral partition. Filtering (**fAPV** versus **APV**) and color coding require a spectral decomposition of $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ at feature locations. The cost for their computation is negligible as the number of feature locations is small compared to the total number of triangular faces.

10.5 DISCUSSION AND COMPARISON

The **APV** operator computes core lines for ensembles of velocity fields. In this section, we compare the new approach with other algorithms for computing core lines for flow ensemble data.

10.5.1 *Multiple Line Sets*

When treating each input field individually and extracting core lines using the standard **PV** operator, we end up with an ensemble of curves. For these, a number of visual representations could be considered, that have already been discussed in Section 9.2. While techniques such as a simple *spaghetti plot* visualization comes with several limitations, we already state in Section 9.3 that approaches that use *clustering* of line features cannot be directly applied due to the nature of vortex core lines as a feature, that might not appear in every ensemble member.

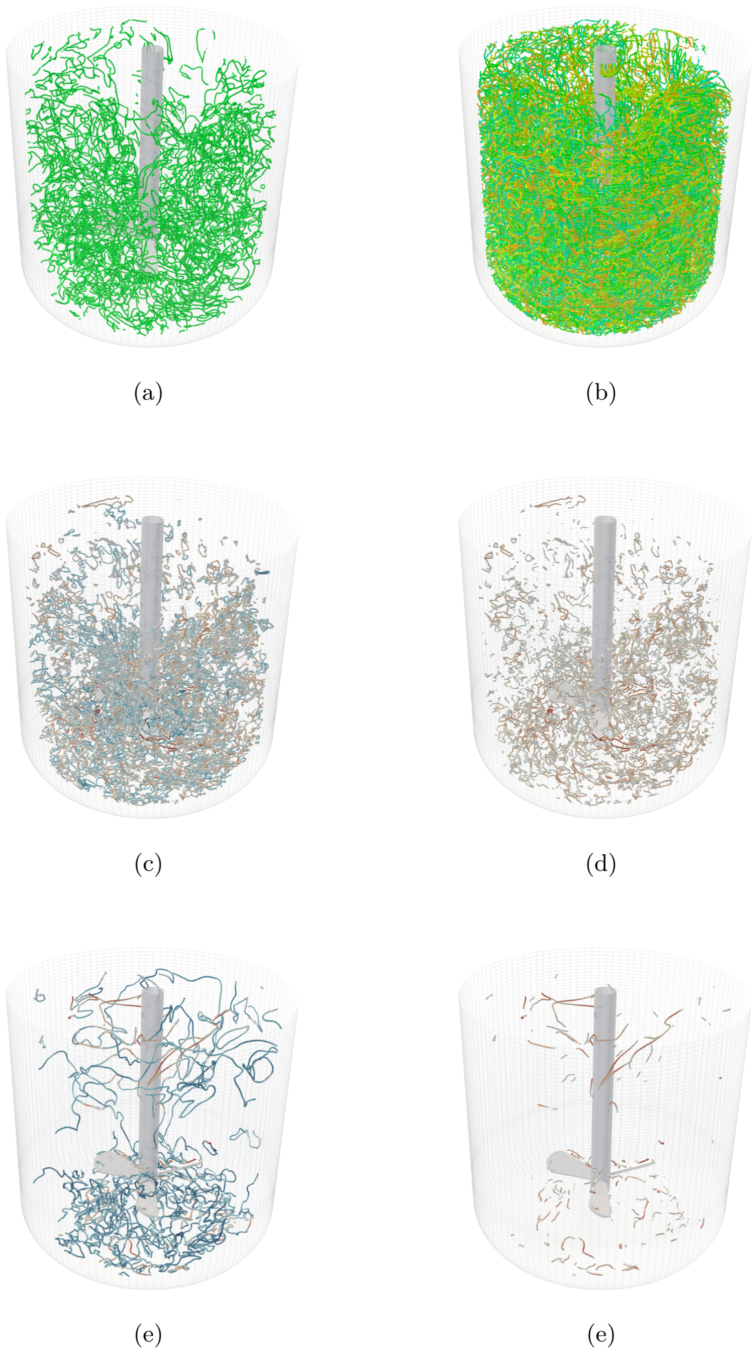


Figure 10.13: Flow inside a rotating mixer. Top: (a) **PV** core lines extracted from one member and the derived acceleration field. (b) Combined visualization of all **PV** core lines extracted from each ensemble member and derived acceleration fields. Middle: **APV** features for the same data, unfiltered (c) and filtered (d). Bottom: **APV** features for velocity members only, unfiltered (e) and filtered (f).

Further, as such curves are neither implicitly defined nor do they share a common parametrization, techniques similar to the *curve boxplot* or *Variability plots* are not directly applicable.

10.5.2 *Lines on Derived Fields*

As introduced in Section 10.4.1, in order to find core lines of multiple fields, one may also consider *extremal curves* in a scalar field s derived from $\mathbf{v}_1, ..., \mathbf{v}_m$ such that summed angles or cross products of each pair of vectors are considered as a measure of how "parallel" the m fields are. The scalar field could be chosen as

$$s = \sum_{i=1}^{m} \sum_{j=1}^{m} |\angle(\mathbf{v}_i, \mathbf{v}_j)|, \quad \text{or} \tag{10.3}$$

$$s = \sum_{i=1}^{m} \sum_{j=1}^{m} \|\mathbf{v}_i \times \mathbf{v}_j\|. \tag{10.4}$$

Ridge lines of s may also be interpreted as lines of "maximal parallelity" of $\mathbf{v}_1, ..., \mathbf{v}_m$. However, we see the following potential problems with this approach:

- *Scaling:* (as in Equation (10.3)) gives unstable results in areas of small vectors $\mathbf{v}_i$.
- *Numerical Stability:* The numerical ridge extraction requires the gradient and Hessian of s, e. g., as input to the **PV** operator. The use of, possibly estimated, first- and second-order derivatives makes the feature extraction significantly more sensitive to noise in the data. Note that alternative ridge extraction methods are, although well-understood, generally less stable than applying the **PV** operator, because ridge extraction is based on searching for local extrema. In contrast, **PV** is based on searching for zero crossings of functions. In regions of strongly varying fields, zero crossing is less prone to missing results than searching for extrema.

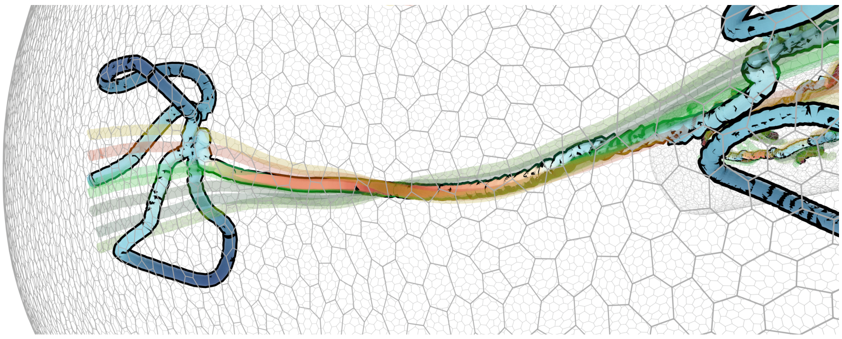
We are not aware of any existing approaches of this category that find core lines of multiple velocity fields.

10.6 LIMITATIONS AND FUTURE WORK

APV feature lines provide an intuitive interpretation for locations where the mean vector is aligned with the major eigenvector. In the visualization, this was shown in red ($\varepsilon > 0$, and ideally $\varepsilon \to 1$). For locations where we observe the alignment with one of the minor eigenvectors – these are missing in **fAPV** – there is no such interpretation (shown in blue as $\varepsilon < 0$). This is the reason why we offer a filter for them. It may be worthwhile to try to derive some information that is meaningful and helps understanding the data also from these locations.

In the future, the presented approach could be extended to an out-of-core method for ensembles with very large numbers of members such that members can be "streamed" into memory. This requires only an "online" update of the mean $\mathbf{a}$ and the matrix $\mathbf{DD}^{\mathrm{T}}$ such that only mean vectors and covariance matrices are held in main memory. This type of update is certainly possible, however, the straightforward formulas suffer significantly from numerical round-off errors. An out-of-core method would require finding a balance between numerical efficiency and sufficient accuracy.
Finally, the domain of applications can be extended by applying **APV** to fields other than velocity and acceleration such as second derivatives of particle trajectories, or pressure gradients.

11 CONCLUSIONS



In the second part of the thesis, we studied the visualization of vector field ensembles. While the visualization of single vector fields has been extensively studied over the years, the latest advancements in computing hardware has enabled the use of ensemble vector fields to analyze complex physical phenomena. The additional ensemble member dimension not only leads to an increased amount of data that needs to be processed, it also poses a challenge for visualization, as most of the known approaches cannot be directly applied or adapted. Especially the extraction of line-type features such as vortex core lines has mainly been handled by using combined visualizations such as spaghetti plots, which are limited in communicating trends of ensembles.

In this part, we developed a new tool in the visualization and analysis of vector field ensembles. After carefully studying the current literature on such data in Chapter 9 as well as discussing recent techniques in the definition and extraction of vortex core lines, we introduced a new operator for vector field ensembles in Chapter 10, called *The Approximate Parallel Vectors (APV) Operator*. This new generic feature extraction method for multiple 3D vector fields extracts lines where all fields are approximately parallel. The definition of the APV operator is based on the application of the Parallel Vectors (PV) operator for two vector fields that are derived from the given set of fields. The APV operator enables the direct visualization of features of vector field ensembles without processing fields individually and without causing visual clutter. We give a theoretical analysis of the APV operator and demonstrate its utility for a number of ensemble data.

Part IV

CONCLUSION AND FUTURE RESEARCH

12 CONCLUSION AND FUTURE RESEARCH

12.1 CONCLUSION

The advancements in computational power as well as the improvements in measuring techniques and algorithms has not only increased the amount of data to process, but also the complexity of the data. Visualization is a powerful tool to grasp what is hidden in the data and must therefore adapt to these new challenges. In this thesis we developed new approaches for the visualization of tensor and vector field ensemble data to aid the analysis and exploration.

In the first part we discussed general ideas within the field of scientific visualizations as well as basic visualization approaches and concepts that are used throughout the thesis. The contributions within this work can be divided into two parts:

Part ii was dedicated to the visualization of second-order tensor data. First, we reviewed important properties of second-order tensors, as well as applications and visualization approaches and focused on the use of tensor glyphs. Most of these geometric objects are based on vectors and values of a tensor decomposition such as the eigenvalues and eigenvectors. After reviewing the literature, we showed that most glyph visualization approaches are limited to symmetric cases of tensors only. These tensors are easier to handle as their eigenvalues are always real-valued and their eigenvector orthogonal which allows for a straightforward mapping to geometry. The decomposition of general tensors however, which often appear in the context of fluid flow data, might result in complex eigenvalues and non-orthogonal eigenvectors and therefore pose a challenge to visualization. We developed the first tensor glyph construction technique for general 2D and 3D second-order tensors that is able to visualize any given tensor of that type, regardless of symmetry. As the design space for possible glyphs is very large, we started the new construction by finding useful limitations. This was done by developing a list of carefully chosen properties or wishes that ensure that the newly developed glyphs behave in desired ways. This included invariance under domain rotation and scaling, the unique representation of each class of tensor as well as continuity. Additionally, we chose the direct rendering of real eigenvector and evaluated current tensor glyph techniques in terms of how they do or do not fulfill these wishes. We proposed a new glyph construction for 2D tensors based on

a characteristic ellipse, that would transition smoothly between cases where eigenvalues are real-valued and complex-valued and further use color to indicate eigenvalue sign and direction of rotation. We then used this construction as a base for developing glyphs for 3D tensors as well. The new techniques were presented on a variety of 2D and 3D datasets. Building upon this work, we further analyzed vector field Jacobian matrices as they represent a special case of such a general second-order tensor. Whereas Jacobian matrices of steady flow fields can be represented by general second-order tensors and can therefore be visualized by the new glyph technique, Jacobian matrices of unsteady flow fields additionally describe the temporal derivative at a given location. We showed that such a Jacobian tensor can be decomposed into a spatial and a temporal component, whereas the spatial component resembles the information that would be given by a Jacobian matrix of a steady flow. We thus proposed using the newly developed glyphs as a base to represent this information. For the additional temporal information, we proposed suitable mappings into the visualization domain and developed the first glyphs capable of rendering the full information encoded by a time-dependent vector field Jacobian matrix. This was, again, presented by applying the visualization on a selection of 2D and 3D datasets.
We then analyzed uncertain symmetric second-order tensor data. When assuming Gaussian distribution, such data can be described by a mean tensor and a covariance tensor that encodes the uncertainty. We extended our list of desirable properties for uncertain tensors, requiring each combination of mean and covariance to follow the same rules such as uniqueness, while additionally requiring a certain tensor with no uncertainty to be a well-defined special case. To solve this, we proposed visualizing the mean tensor by any suitable tensor glyph technique such as superquadric tensor glyphs or our newly proposed glyphs. We showed that the covariance encodes the change of the mean glyph surface which we encoded as a scalar field on the surface of the mean glyph. We used the fact that symmetric second-order tensors have repeated components, which allows the embedding into a vector space and thus easier handling of the covariance. We proposed visualizing the resulting scalar field as an offset surface and further evaluated if different certain glyphs could be extended by our technique while fulfilling all requirements.

To summarize, we not only offered an overview on the visualization of second-order tensors with the help of glyphs, but further introduced two new glyph construction techniques as well as an augmentation for existing glyphs to also represent uncertain tensor data.

In Part iii, we focused on vector field ensemble data. After explaining the usefulness of such data as a tool to represent uncertainty, we stated that it also poses a challenge to domain expert, due to its size and complexity. We first reviewed well-known visualization techniques for single vector fields and emphasized vortex core lines as the result of the Parallel Vectors (**PV**) Operator as these lines are able to capture important structures within flow data. We showed that adapting or extending visualization techniques especially based on extracting line-type features to be used on vector field ensembles is not a trivial task. After reviewing current strategies in state-of-the-art approaches, we concluded, that techniques such as clustering cannot be applied to core lines and using approaches such as spaghetti plots comes with significant limitations. We therefore presented a new ensemble vector field operator called the Approximate Parallel Vectors (**APV**) operator as a generalization of the **PV** operator for an arbitrary number of vector fields. The operator extracts all locations, where vectors of the different ensemble members are approximately parallel to each other. We defined vectors to be approximately parallel, if the direction of the largest spread from the mean value is parallel to the mean vector. As this reduced the problem of finding approximately parallel vector to finding parallel vector in two newly derived vector fields, we showed that a standard **PV** operator can be applied which results in closed line structures. We further analyzed the behavior of the operator for different inputs and provided a solution on how to do the calculations efficiently. The operator was applied to several vector field ensembles.

We introduced several new tools for domain experts that can be used to not only get an overview of the fields provided, but to further gain a deeper understanding and find new structures within the high quantity of data. Source code for the glyph construction in Chapter 5 has been made publicly available and some of the techniques presented in this thesis already have been or are currently being added to general purpose scientific visualization software such that they can be used by the public.

12.2 FUTURE RESEARCH

There are numerous directions for future research on the visualization of scientific data. Some individual improvements and ideas are already raised within the chapters such as finding glyphs that are capable of representing general uncertain second-order tensors which is challenging due to the high number of parameters encoded within the tensor. For all proposed visualization techniques for second-order tensors with the help of glyphs in this thesis, a rigorous evaluation of design choices might further improve the glyphs. Similar studies have been applied to vector glyphs [172], [173] and existing tensor glyphs [121]. This includes color palettes as well as suitable rendering of geometry like the eigensticks or the offset surface for uncertain tensors. As the glyphs are meant to be a valuable tool for domain experts, user studies might be used to examine and improve readability, efficiency and effectiveness of the glyphs. Further, searching general glyphs for tensors of higher-order such as the stiffness tensor or elasticity and compliance tensors which are tensors of fourth-order is subject of ongoing research [74], [163], [164] with the aim to find more efficient ways to convey the information within the data.

The effective visualization of vector field ensembles also offers several directions to pursue. This includes incorporating temporal information, thus finding new features to analyze ensembles of unsteady vector fields. Extracting stable features such as other representative lines or surfaces that better capture the gist of a vector field ensemble is still a new and open challenge. New challenges also arise from dealing with ensembles of other data types such as tensor field ensembles. Seeing if similar concepts can be applied to such data and what challenges arise from it is definitely an exciting direction to go. Finally, all approaches might be improved in terms of algorithmic efficiency, which is important when dealing with large amounts of data. This includes looking for possible parallelization opportunities as well as efficient data management.

Part V

APPENDIX

A

ADDITIONAL MAPPING OF TENSOR PROPERTIES IN ADDITION TO SHAPE IS NECESSARY

This proof has been developed by Theisel and Rössl in [47].

In Section 5.2.1 we state that shape alone cannot fully encode a general 2D or 3D tensor. To prove this, consider tensors $\mathbf{T}$ that are invariant under domain rotation, i. e.,

$$\mathbf{Q}\,\mathbf{T}\,\mathbf{Q}^{\mathrm{T}} \;=\; \mathbf{T} \tag{A.1}$$

for any rotation matrix $\mathbf{Q}(\gamma)$. Choose two such tensors of equal scale (norm) $\mathbf{T}_1 = \left[\begin{smallmatrix}1 & 0\\ 0 & 1\end{smallmatrix}\right]$, $\mathbf{T}_2 = \left[\begin{smallmatrix}0 & -1\\ 1 & 0\end{smallmatrix}\right]$. Then Equation (5.1) and Equation (A.1) require that $\mathbf{Q}\,G(\mathbf{T}_1) = G(\mathbf{T}_1)$ and $\mathbf{Q}\,G(\mathbf{T}_2) = G(\mathbf{T}_2)$, i. e., an arbitrary rotation of the glyph gives the same glyph. The only shape that fulfills this requirement is the circle. Hence, both $\mathbf{T}_1$ and $\mathbf{T}_2$ must be encoded as the same circle. If glyphs are only determined by shape, this violates the requirement for uniqueness (d) as listed in Section 5.1. In general, every tensor of the form $\mathbf{T} = \mathbf{Q}(\gamma)$ is constant under domain rotation and must therefore be mapped to the circle. Thus, we need at least one additional continuous channel to encode the angle γ in the glyph. For our work, we use color.

B

PROPERTIES OF THE CHARACTERISTIC ELLIPSE

The insights on the following properties have been developed by Theisel and Rössl in [47].

The glyph construction presented in Chapter 5 is based on a quadric that interpolates both the scaled eigenvector endpoints as well as the left singular vectors of a singular value decomposition. We prove this property and give a possible representation in Bernstein-Bézier form.

CHARACTERISTIC ELLIPSE

Given is a second-order tensor $\mathbf{T} \in \mathbb{R}^{2\times 2}$ and *real* eigenvalues $\lambda_{1,2}$ and eigenvectors as columns of $\mathbf{X}$. Its characteristic ellipse is determined by Equation (5.3).

Proposition B.0.1. *The implicit curve defined by Equation* (5.3) *is an ellipse that interpolates* $\pm\lambda_i \mathbf{X}_{\cdot i}$ *for* $i = 1, 2$*, i. e., the eigenvectors scaled by the eigenvalues.*

Proof. The equation $\mathbf{x}^{\mathrm{T}}\left(\mathbf{T}\mathbf{T}^{\mathrm{T}}\right)^{-1}\mathbf{x} = 1$ defines a quadric, i. e., the implicit curve is a conic section [41]. It is indeed an ellipse as $\mathbf{T}\mathbf{T}^{\mathrm{T}}$ is symmetric and positive definite and so is its inverse as $\left(\mathbf{T}\mathbf{T}^{\mathrm{T}}\right)^{-1} = \left(\mathbf{T}^{-1}\right)^{\mathrm{T}}\mathbf{T}^{-1}$.
The tensor has spectral decomposition $\mathbf{T} = \mathbf{X}\Lambda\mathbf{X}^{-1}$ with unit length eigenvectors $||\mathbf{X}_{\cdot i}|| = 1$ and singular value decomposition $\mathbf{T} = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$. We verify that Equation (5.3) holds for $\mathbf{x}_i = \lambda_i \mathbf{X}_{\cdot i}$ for $i = 1, 2$. In matrix notation with $\mathbf{x}_i$ as the two columns of $\mathbf{X}\Lambda$ we write

$$\Lambda\mathbf{X}^{\mathrm{T}}\left(\mathbf{T}\mathbf{T}^{\mathrm{T}}\right)^{-1}\mathbf{X}\Lambda \;=\; \mathbf{X}^{\mathrm{T}}\mathbf{V}\Sigma\mathbf{U}^{\mathrm{T}}\left(\mathbf{U}\Sigma^{-2}\mathbf{U}^{\mathrm{T}}\right)\mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}\mathbf{X} \;=\; \mathbf{X}^{\mathrm{T}}\mathbf{X}$$

using the identities $\mathbf{T}\mathbf{T}^{\mathrm{T}} = \mathbf{U}\Sigma^2\mathbf{U}^{\mathrm{T}}$ and $\mathbf{X}\Lambda = \mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}\mathbf{X}$, obtained from the factorizations, and then exploiting orthogonality of $\mathbf{U}$ and $\mathbf{V}$. The diagonal entries $\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)_{ii}$ evaluate to 1 and are by construction equal to $\mathbf{x}_i\left(\mathbf{T}\mathbf{T}^{\mathrm{T}}\right)^{-1}\mathbf{x}_i$, which shows that the equation holds. Obviously, the same holds for $-\mathbf{X}\Lambda$ and hence $-\mathbf{x}_i$. $\square$

Equation (5.3) holds equally for *non-real* $\pm\mathbf{x}_i \in \mathbb{C}$, $i = 1, 2$. However, the interpretation of the ellipse in the real plane (for $\mathbf{x} \in \mathbb{R}$) changes: the complex eigenvectors are "replaced" by the left singular vectors $\mathbf{U}$, which span the orthogonal principal axes, and it is easy to verify that the ellipse interpolates the column vectors of $\pm\mathbf{U\Sigma}$. For $\text{rank}(\mathbf{T}) = 1$, the characteristic ellipse degenerates to a line segment.

RATIONAL PARAMETRIZATION OF THE CHARACTERISTIC ELLIPSE

The quadric described by Equation (5.3) is an implicit representation for the characteristic ellipse, which can be parametrized as a rational quadratic curve [41]. A particularly simple construction is as follows:

Express the rational curve in Bernstein-Bézier form as

$$\mathbf{f}(t) = \frac{w_0 \mathbf{b}_0\,(1-t)^2 \,+\, w_1 \mathbf{b}_1\,2(1-t)t \,+\, w_2 \mathbf{b}_2\,t^2}{w_0\,(1-t)^2 \,+\, w_1\,2(1-t)t \,+\, w_2\,t^2}\;,\quad t \in [0,1]\;,$$

with control points $\mathbf{b}_i \in \mathbb{R}$. The weights $w_i \in \mathbb{R}$ can always be chosen such that at the end points $w_0 = w_2 = 1$. Figure 5.3a shows an example. Assume $\mathbf{T}$ has eigenvalues $\lambda_1 = \lambda_2 = 1$, i. e., $\mathbf{T} = \mathbf{XIX}^{-1}$, and construct a parametrization of one arc of the unit circle that is enclosed by the eigenvectors $\mathbf{X}_{\cdot i}$. From end point interpolation of rational Bézier curves the left and right control points $\mathbf{b}_0$ and $\mathbf{b}_2$ are determined as the unit length eigenvectors $\mathbf{X}_{\cdot i}$. The tangents of the curve at the end points are given as tangents to the unit circle or a 90 degree rotation of $\mathbf{X}_{\cdot i}$. End points and tangent directions define two lines, and the center control point $\mathbf{b}_1$ is determined as their intersection. Let α denote the angle enclosed by the eigenvectors. To verify that ω in Equation (5.4) is to be set as $\mathbf{b}_1 = \omega\,(\mathbf{b}_0 + \mathbf{b}_2)$, consider the following: w.l.o.g. use eigenvectors $(\cos(\alpha/2)\,, \sin(\alpha/2))$ to exploit symmetry, and obtain $\mathbf{b}_1 = (\frac{1}{\cos(\alpha/2)}, 0)$, then compare to $\mathbf{b}_0 + \mathbf{b}_2 = (2\cos(\alpha/2)\,, 0)$ to determine the factor $\omega = (1 + \cos\alpha)^{-1}$. The weights associated with the three control points are $w_0 = w_2 = 1$ and for the center $w_1 = \cos(\alpha/2)$. For the general construction with unconstrained eigenvalues, i. e., arcs of an ellipse, the control points are transformed linearly as $\mathbf{T}\,\mathbf{b}_i$ for $i = 0, 1, 2$. The weights remain unchanged. Due to the affine invariance property of rational Bézier curves, mapping the circular arc results in the same curve as performing the construction directly for a general ellipse. The proposed parametrization can be applied similarly or symmetrically to all four arcs of the characteristic ellipse. Finally, the construction is straightforward for the degenerated case when $\text{rank}(\mathbf{T}) = 1$ (or $\cos\alpha = -1$): in this case $\mathbf{b}_1$ is undefined, however, as $w_1 = \sin\alpha = 0$ the curve is just the line segment spanned by $\mathbf{b}_0$ and $\mathbf{b}_2$.

C

CREATION OF SURFACE PATCHES FOR GENERAL 3D TENSOR GLYPHS

The following construction has been developed by Theisel and Rössl in [47].

For the construction of tensor glyphs for general 3D second-order tensors as introduced in Chapter 5, we construct surface patches from three quadratic rational boundary curves in Bernstein-Bézier form. Although, three such curves determine a triangular rational quadratic surface patch, such patches are parts of ellipsoids, and in particular spheres, only in special configurations [41]. In general, a rational surface patch of total degree four is required. We follow the construction in [144] that uses a rational bi-quadratic patch that is "degenerated" by the map to the triangle domain. Then the 3×3 control points and weights $(\mathbf{b}_{ij}, w_{ij})$ of the patch are determined as

$$\begin{array}{ccc} (\mathbf{x}_3, 1) & (\mathbf{x}_3, w_{12}) & (\mathbf{x}_3, 1) \\ (\nu_{12}(\mathbf{x}_3 + \mathbf{x}_1), \sqrt{2}/2) & (\mathbf{b}_{11}, \sqrt{2}/2\, w_{12}) & (\nu_{12}(\mathbf{x}_2 + \mathbf{x}_3), \sqrt{2}/2) \\ (\mathbf{x}_1, 1) & (\mu_{12}(\mathbf{x}_1 + \mathbf{x}_2), w_{12}) & (\mathbf{x}_2, 1) \end{array}$$

with the center control point $\mathbf{b}_{11} = \nu_{12}\left(\mu_{12}\left(\mathbf{x}_1 + \mathbf{x}_2\right) + \mathbf{x}_3\right)$, and $\pm\mathbf{x}_i = \lambda_i \mathbf{X}_{\cdot i}$ are scaled eigenvectors. Assume that the boundary curve from $\mathbf{x}_1$ and $\mathbf{x}_2$ is located in the reference plane. This determines the surface patches for the 3D glyph in the case of a well-defined base plane (see Section 5.3) as reference plane, which gives the weight w_{12} and the factor μ_{12} as in the 2D case.

In the case that no base plane can be established, we construct a continuous surface patch by blending the results of the evaluation of the three possible patches. While Section 5.3 discusses and justifies the choice of blend weights, we need to clarify the surface evaluation, which defines a different, non-standard class of surface patch.

Let $\mathbf{f}_i : [0,1] \times [0,1] \to \mathbb{R}^3$, $i = 0,1,2$, denote the three rational patches with base curves from $\mathbf{x}_i$ to $\mathbf{x}_{i+1 \bmod 3}$. The core idea is a reparametrization of $\mathbf{f}_i(u,v)$ as $\mathbf{f}_i(\beta_1, \beta_2, \beta_3)$ with barycentric coordinates $\beta_1 + \beta_2 + \beta_3 = 1$ such that

$$(\beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \beta_3 \mathbf{x}_3) \parallel \mathbf{f}_i(u,v) \;, \tag{C.1}$$

i. e., the barycentric combination of scaled eigenvectors that span the patch yields a vector that is parallel to the position vector $\mathbf{f}(u,v)$. In this parametrization, we can evaluate the blend patch as

$$\mathbf{f}(\beta_1,\beta_2,\beta_3) \;=\; \frac{1}{\sum_0^2 W_i} \sum_0^2 W_i\, \mathbf{f}_i(\beta_1,\beta_2,\beta_3)\;,$$

using the blend weights W_i defined in Equation (5.6).
In order to construct the parametrization (C.1) – w.l.o.g. for $\mathbf{f}_1$ – consider the roots of the norm of the cross product. Due to symmetries, the solution can be expressed as follows. Find $u(\beta_1,\beta_2,\beta_3) \in [0,1]$ as solution of the quadratic equation

$$(1-u)^2 b_0 \;+\; 2\,(1-u)u\,b_1 \;+\; u^2 b_2 \;=\; 0\;,$$

for $b_0 = \beta_2$, $b_1 = w_{12}\mu_{12}\,(\beta_2-\beta_1)$, $b_2 = -\beta_1$. The expression in the quadratic Bernstein polynomials reveals immediately that there exists one solution $u_0 \in [0,1]$. Then find $v(\beta_1,\beta_2,\beta_3) \in [0,1]$ similarly as the root

$$(1-v)^2 c_0 \;+\; 2\,(1-v)v\,c_1 \;+\; v^2 c_2 \;=\; 0\;,$$

with $u := u(\beta_1,\beta_2,\beta_3) = u_0$, $\tilde{u} := 1-u$, $\tilde{\beta} := \beta_3-\beta_1-\beta_2$, and

$$\begin{aligned}
&c_0 = \sqrt{2}\beta_3\,(\tilde{u}^2 + 4w_{12}\mu_{12}\,u\,\tilde{u} + u^2)\;,\\
&c_1 = \nu_{12}(\tilde{\beta}\tilde{u}^2 + 2w_{12}\,(2\mu_{12}\,\beta_3-\beta_1-\beta_2)\,u\,\tilde{u} + \tilde{\beta}u^2)\;,\ \text{and}\\
&c2 = -\sqrt{2}(\beta_1+\beta_2)(\tilde{u}^2 + 2w_{12}\,u\,\tilde{u} + u^2)\;.
\end{aligned}$$

D

ROTATIONS IN TENSOR SPACE

These proofs have been developed by Theisel and Rössl in [49].

Given is a symmetric tensor $\mathbf{S} \in \mathbb{R}^{3\times 3}$. For any rotation matrix $\mathbf{R}$ in Equation (3.2) as defined in Section 3.1.3 that acts on $\mathbf{S}$, the corresponding rotation $\hat{\mathbf{R}}$ that acts on $\mathsf{v}(\mathbf{S}) \in \mathbb{R}^6$ can be derived as

$$\hat{\mathbf{R}} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{pmatrix} \quad \text{with} \tag{D.1}$$

$$\begin{aligned} \mathbf{R}_{11} &= \mathbf{R} \circ \mathbf{R} \\ \mathbf{R}_{21} &= \sqrt{2}\,\mathbf{R}_{(132)(123)} \circ \mathbf{R}_{(213)(123)} \\ \mathbf{R}_{12} &= \sqrt{2}\,\mathbf{R}_{(123)(132)} \circ \mathbf{R}_{(123)(213)} \\ \mathbf{R}_{22} &= 2\,\mathbf{R}_{(132)(132)} \circ \mathbf{R}_{(213)(213)} - \mathbf{R}_{(321)(321)} \ , \end{aligned}$$

where $\circ$ denotes the entrywise Hadamard product of matrices and the subindices (ijk) denote permutations of matrix rows and columns, respectively. Note that [66] provides a different construction based on plane rotations.

E

UNIQUENESS OF UNCERTAIN TENSOR GLYPHS

These proofs have been developed by Theisel and Rössl in [49].

In order to proof if the scalar field q on the surface of a glyph from Chapter 7 uniquely represents an uncertain tensor $(\tilde{\mathbf{S}}, \mathbf{C})$, we need to show that the covariance matrix $\mathbf{C}$ can be reconstructed from sampling the scalar field.

PROOF OF LEMMA 7.5.1

Lemma 7.5.1 states that an uncertain glyph is not unique if all $\mathbf{q} \in \mathcal{Q}$ live on a common quadric, which in turn means that there exists a non-zero matrix $\mathbf{A}$ such that $\mathbf{q}^{\mathrm{T}} \mathbf{A}\, \mathbf{q} = 0$ for all $\mathbf{q} \in \mathcal{Q}$. To prove this, we first show that a glyph is *not* unique if such a matrix exists. Assume that there exists a non-zero matrix $\mathbf{A}$ that fulfills $\mathbf{q}^{\mathrm{T}} \mathbf{A}\, \mathbf{q} = 0$ for all $\mathbf{q} \in \mathcal{Q}$. Then for any selection of sample points $\mathbf{g}_1, \ldots, \mathbf{g}_{21}$ we have $\mathbf{q}_i^{\mathrm{T}} \mathbf{A}\, \mathbf{q}_i = 0$ for $i = 1, \ldots, 21$. Writing this in matrix form using Equation (3.1) gives

$$\mathbf{M}^{\mathrm{T}}\ \mathsf{v}(\mathbf{A}) = \mathbf{0}_{21} \tag{E.1}$$

with $\mathbf{M}$ as in Equation (7.15). Since $\mathsf{v}(\mathbf{A})$ is non-zero, Equation (E.1) can only hold if $\mathbf{M}^{\mathrm{T}}$ has a zero eigenvalue with the corresponding eigenvector $\mathsf{v}(\mathbf{A})$. Therefore, $\mathbf{M}$ has a rank deficit for any selection of sample points, there is no unique solution to Equation (7.14), and therefore the uncertain glyph is not unique.
For the reverse direction, we assume that an uncertain glyph is *not* unique, i. e., we have chosen sample points $\mathbf{g}_1, \ldots, \mathbf{g}_{21}$ such that the matrix $\mathbf{M}$ is singular. Then there exists a (non-zero) eigenvector $\mathsf{v}(\mathbf{E})$ that corresponds to a zero eigenvalue, i. e., $\mathbf{M}^{\mathrm{T}}\mathsf{v}(\mathbf{E}) = \mathbf{0}_{21}$. Due to Equation (3.1) and the definition of $\mathbf{M}$ in Equation (7.15), the latter condition is equivalent to

$$\mathsf{v}(\mathbf{q}_i{\mathbf{q}_i}^{\mathrm{T}})^{\mathrm{T}} \mathsf{v}(\mathbf{E}) = {\mathbf{q}_i}^{\mathrm{T}} \mathbf{E}\, \mathbf{q}_i = 0 \quad \text{for} \quad i = 1, \ldots, 21\ ,$$

which implies that the samples live on the common quadric defined by $\mathbf{E}$. $\square$

SKETCH OF PROOF OF THEOREM 7.5.2

Given is an uncertain tensor $(\bar{\mathbf{S}}, \mathbf{C})$ with $\mathbf{v}(\bar{\mathbf{S}}) = \mathbf{s}$. The surface of the ellipsoid glyph of the mean tensor is given as implicit surface $g(\bar{\mathbf{S}}, \mathbf{x}) = 0$ with $g(\bar{\mathbf{S}}, \mathbf{x}) = \mathbf{x}^{\mathrm{T}}\bar{\mathbf{S}}^{-2}\mathbf{x} - 1$ or equivalently as the image $\{\, \bar{\mathbf{S}}\mathbf{x} \,|\, \mathbf{x}^{\mathrm{T}}\mathbf{x} = 1 \,\}$. We want to show that for this choice of g there exists an $\mathbf{A} \neq \mathbf{0}$ such that $\mathbf{q}^{\mathrm{T}}\mathbf{A}\mathbf{q} = 0$, which is equivalent to $(\nabla_{\mathbf{s}} g)^{\mathrm{T}}\mathbf{A}\nabla_{\mathbf{s}} g = 0$ after dropping the normalization. Let $\mathbf{x} = (x_1, x_2)$ with $||x||^2 = x_1^2 + x_2^2 = 1$. Then the evaluation of the gradient at a surface point gives

$$\nabla_{\mathbf{s}} g(\bar{\mathbf{S}}, \bar{\mathbf{S}}\mathbf{x}) \;=\; \frac{2}{2s_{11}s_{22} - s_{12}^2} \begin{pmatrix} (2s_{22}x_1 - \sqrt{2}s_{12}x_2)\, x_1 \\ (\sqrt{2}s_{12}x_1 - 2s_{11}x_2)\, x_2 \\ s_{12} - \sqrt{2}(s_{11} + s_{22})\, x_1x_2 \end{pmatrix},$$

and for the further consideration we can drop the factor that is constant in $\mathbf{x}$. The remaining expression

$$\tilde{\mathbf{q}} := (s_{11}s_{22} - \tfrac{1}{2}s_{12}^2) \cdot \nabla_{\mathbf{s}} g(\bar{\mathbf{S}}, \bar{\mathbf{S}}\mathbf{x})$$

is linear in $\mathbf{s}$, and we can write

$$\tilde{\mathbf{q}} \;=\; \mathbf{B}\,\mathbf{r} \;:=\; \begin{pmatrix} 2s_{22} & 0 & -\sqrt{2}s_{12} \\ 0 & -2s_{11} & \sqrt{2}s_{12} \\ s_{12} & s_{12} & -\sqrt{2}(s_{11} + s_{22}) \end{pmatrix} \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1x_2 \end{pmatrix}.$$

We will now show that the glyph is not unique by Lemma 7.5.1 and use the fact that the above matrix $\mathbf{B}$ has full rank and is invertible.
We have $q^2 = \varphi\,\tilde{\mathbf{q}}^{\mathrm{T}}\tilde{\mathbf{q}}$ for some factor $\varphi \neq 0$ that is constant in $\mathbf{x}$, i. e., independent of the chosen surface point. Consider the vector $\mathbf{r}$ and find a matrix $\mathbf{S} \neq \mathbf{0}$ such that $\mathbf{r}^{\mathrm{T}}\mathbf{S}\,\mathbf{r} = 0$. In this case there are three distinct choices up to scaling, e.g.,

$$\mathbf{S} \;=\; \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

We use that $\mathbf{B}$ is invertible in order to construct $\mathbf{A} = \mathbf{B}^{-\mathrm{T}}\mathbf{S}\,\mathbf{B}^{-1}$, which is similar to $\mathbf{S}$, and for which $\varphi^{-1}\mathbf{q}^{\mathrm{T}}\mathbf{A}\mathbf{q} = \tilde{\mathbf{q}}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\tilde{\mathbf{q}} = \mathbf{r}^{\mathrm{T}}\mathbf{B}^{-\mathrm{T}}\mathbf{S}\,\mathbf{B}^{-1}\mathbf{r} = \mathbf{r}^{\mathrm{T}}\mathbf{S}\,\mathbf{r} = 0$ for all $\mathbf{r}$ and therefore also for all $||\mathbf{x}|| = 1$ and all surface points $\mathbf{S}\mathbf{x}$, respectively. With Lemma 7.5.1 this proves the theorem. □

SKETCH OF PROOF OF THEOREM 7.5.3

For the proof of Theorem 7.5.3, we restrict ourselves to the 2D case. The basic idea is simple: provide 6 sample points on the glyph curve and show that they give a full rank matrix $\mathbf{M}$. The technical difficulty consists is the fact that the superquadric glyphs are parametrized in the spectral basis, whereas partial derivatives must be computed w.r.t. to the tensor components. We only give the (intermediate) results for derivations. Given is a symmetric positive definite tensor $\mathbf{S}$ and its spectral decomposition $\mathbf{S} = \mathbf{R}\,\Lambda\,\mathbf{R}^T$ where $\mathbf{R}$ is a rotation matrix with eigenvectors as columns, and $\Lambda = \left(\begin{smallmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{smallmatrix}\right)$ has the eigenvalues $0 < \lambda_2 \leq \lambda_1$ on its diagonal. Then the parametric representation of the glyph is

$$\mathbf{g}(\mathbf{S},\theta) = \mathbf{R}\,\Lambda \begin{pmatrix} \cos^{\alpha}\theta \\ \sin^{\alpha}\theta \end{pmatrix} \tag{E.2}$$

with $\alpha = \left(\frac{2\lambda_2}{\lambda_1+\lambda_2}\right)^{\gamma}$, $x^{\alpha} = \operatorname{sgn}(x)\ |x|^{\alpha}$, and $\gamma \geq 0$ serving as a shape parameter. W.l.o.g. we assume that $\mathbf{S}$ is diagonal, i. e., $\mathbf{R} = \mathbf{I}$. Note that this can always be achieved by a change of the coordinate system. However, the main diagonal $\mathbf{S} = \Lambda$ generally has non-vanishing partials w.r.t. the off-diagonal entries s_{12}, because any (infinitesimal) change of the rotation results in a change of s_{12}.
We compute the gradient for the factors of $\mathbf{g}$ and summarize the results as

$$\begin{aligned}
\nabla_{\mathbf{s}}\mathbf{R} &= \left[\left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right),\ \left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right),\ \frac{\sqrt{2}}{2(\lambda_1-\lambda_2)}\left(\begin{smallmatrix}0&-1\\1&0\end{smallmatrix}\right)\right],\\
\nabla_{\mathbf{s}}\Lambda &= \left[\left(\begin{smallmatrix}1&0\\0&0\end{smallmatrix}\right),\ \left(\begin{smallmatrix}0&0\\0&1\end{smallmatrix}\right),\ \left(\begin{smallmatrix}0&0\\0&0\end{smallmatrix}\right)\right],\\
\nabla_{\mathbf{s}}\alpha &= \tfrac{\alpha\,\gamma}{\lambda_1+\lambda_2}\left(-1\ ,\ \tfrac{\lambda_1}{\lambda_2}\ ,\ 0\right)^{\mathrm{T}}.
\end{aligned}$$

Note that these expressions are well defined only for $\lambda_1 \neq \lambda_2$ and $\lambda_1, \lambda_2 \neq 0$ as required in Theorem 7.5.3. This gives

$$\nabla_{\mathbf{s}}\mathbf{g} = (\nabla_{\mathbf{s}}\mathbf{R}\ \Lambda + \mathbf{R}\ \nabla_{\mathbf{s}}\Lambda)\begin{pmatrix}\cos^{\alpha}\theta\\ \sin^{\alpha}\theta\end{pmatrix} + \mathbf{R}\ \Lambda\ \nabla_{\mathbf{s}}\alpha \begin{pmatrix}\cos^{\alpha}\theta\ \ln\cos\theta\\ \sin^{\alpha}\theta\ \ln\sin\theta\end{pmatrix}.$$

Now we select 6 sample points as

$$\mathbf{g}_i = \mathbf{g}(\mathbf{S}, i\,\frac{\pi}{6}) \quad \text{for} \quad i = 0, \ldots, 5\ .$$

Then computing the gradients $\nabla_{\mathbf{s}}\mathbf{g}(\mathbf{S}, i\,\frac{\pi}{6})$ at the sample points, where common factors that are constant in $\mathbf{s}$ – they do not affect the rank of $\mathbf{M}$ – are dropped and some symmetries are exploited, gives sample vectors $\mathbf{q}_i$ of the form

$$(\mathbf{q}_1, \ldots, \mathbf{q}_6) = \begin{pmatrix} 1 & a & c & 0 & -c & -a\\ 0 & b & d & 1 & -d & -b\\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} \tag{E.3}$$

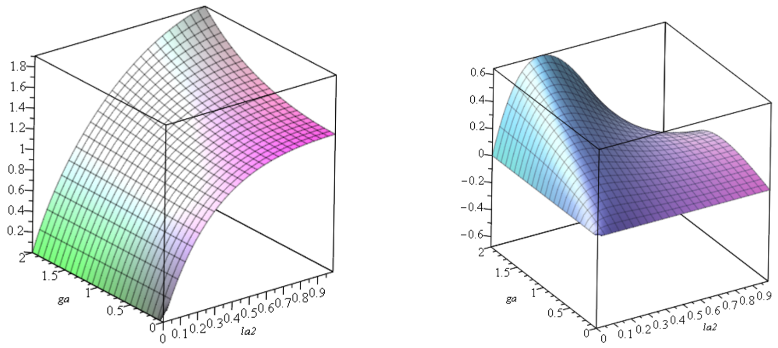


Figure E.1: The proof of Theorem 7.5.3 involves two functions that must not vanish iff $\mathbf{M}$ has full rank. The plots visualize f_1 (left) and f_1 (right) for $\lambda_1 = 1$ in the range $\lambda_2 \in [0, 1]$ and $\gamma \in [0, 2]$

for certain terms a, b, c, d, which depend on $\lambda_1, \lambda_2, \gamma$. Applying Equation (7.15) (for the 2D case) to Equation (E.3) gives

$$\mathbf{M} = \begin{pmatrix} 1 & a^2 & c^2 & 0 & c^2 & a^2 \\ 0 & b^2 & d^2 & 1 & d^2 & b^2 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & \sqrt{2}ab & \sqrt{2}cd & 0 & \sqrt{2}cd & \sqrt{2}ab \\ 0 & \sqrt{2}a & \sqrt{2}c & 0 & -\sqrt{2}c & -\sqrt{2}a \\ 0 & \sqrt{2}b & \sqrt{2}d & 0 & -\sqrt{2}d & -\sqrt{2}b \end{pmatrix} .$$

Since $\det \mathbf{M} = 8\sqrt{2}\, f_1\, f_2$ with

$$f_1 = f_1(\lambda_1, \lambda_2, \gamma) = (ad - bc)$$
$$f_2 = f_2(\lambda_1, \lambda_2, \gamma) = (ab - cd) \ ,$$

$\mathbf{M}$ has full rank if neither f_1 nor f_2 vanish, which is the case. Instead of summarizing formal expressions for f_1, f_2 and a formal proof, we provide "visual evidence" that $f_1 f_2 \neq 0$: Figure E.1 shows the functions f_1 and f_2 plotted as height fields for $\lambda_1 = 1$ in the range $\lambda_2 \in [0, 1]$ and $\gamma \in [0, 2]$. It shows that f_1, f_2 do not vanish for $\gamma > 0$ and $0 < \lambda_2 < \lambda_1$. Showing Theorem 7.5.3 for the 3D case requires a sampling of 21 points on the glyph surface. The basic idea is same as for 2D, however, the involved expressions become significantly more complex.

F

THE APPROXIMATELY PARALLEL VECTORS OPERATOR

The following proofs have been developed by Theisel and Rössl in [48].

In the following, we give proofs for the Properties (**P1**)-(**P7**) summarized in Section 10.2.

Property (**P1**) holds by construction as neither the mean vector $\mathbf{a}$ nor the symmetric operator $\mathbf{D}\mathbf{D}^{\mathrm{T}} = \sum_{i=1}^{n}(\mathbf{v}_i - \mathbf{a})(\mathbf{v}_i - \mathbf{a})^{\mathrm{T}}$ (and hence $\mathbf{b}$) change on permutation of the columns $\mathbf{v}_i$ of $\mathbf{V}$. □

Property (**P2**). Assume the *parallel vectors condition* holds for $\mathbf{v}_1$ and $\mathbf{v}_2$, i. e., $\mathbf{v}_1 = \lambda\mathbf{v}_2$. Then with $\mathbf{V} = (\mathbf{v}_1|\lambda\mathbf{v}_1)$, $\mathbf{a} = \frac{1+\lambda}{2}\mathbf{v}_1$ and

$$\mathbf{D}\mathbf{D}^{\mathrm{T}} \;=\; (1+\lambda^2)\mathbf{v}_1{\mathbf{v}_1}^{\mathrm{T}} - \tfrac{1}{2}(1-\lambda)^2\mathbf{v}_1{\mathbf{v}_1}^{\mathrm{T}} \;=:\; \tilde{\lambda}\mathbf{v}_1{\mathbf{v}_1}^{\mathrm{T}}\,.$$

This implies $\mathbf{b} = \mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a} = \frac{1+\lambda}{2}\tilde{\lambda}\mathbf{a}$, or equally the **APV** condition. The symmetric matrix $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ has rank 1 and one single nonzero eigenvalue $\tilde{\lambda}{\mathbf{v}_1}^{\mathrm{T}}\mathbf{v}_1$ therefore also the **filtered APV** condition holds. The reverse direction (**fAPV** $\Rightarrow$ **PV**) is straightforward to show using the same argument. □

The proofs of the remaining, more general properties are slightly more complex. The essential argument, however, is similar.
Let $\mathbf{1}_n \in \mathbb{R}^n$ denote the constant vector with all entries equal to 1 and let $\mathbb{1}_n = \mathbf{1}_n\mathbf{1}_n^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ denote the constant matrix with all entries 1. For the sake of a concise notation, we may omit the dimension n whenever it is irrelevant or clear from the context.
Let $\mathbf{V} = (\mathbf{v}_1,\ldots,\mathbf{v}_n)$ with $\mathbf{v}_i \in \mathbb{R}^m$. Then $\mathbf{a} = \frac{1}{n}\mathbf{V}\mathbf{1}_n$ the *mean* of column vectors for a matrix $\mathbf{V} \in \mathbb{R}^{m\times n}$, and $\frac{1}{n}\mathbf{V}\mathbb{1}_n$ gives the matrix $(\mathbf{a},\ldots,\mathbf{a}) \in \mathbb{R}^{m\times n}$. So far, we considered (w.l.o.g.) the 3D case $m = 3$. The following arguments apply for any dimension $m \geq 2$.
The proposed algorithm requires the matrix $\mathbf{D} = \mathbf{V} - \frac{1}{n}\mathbf{V}\mathbb{1}_n = \mathbf{V}\mathbf{P}$ with $\mathbf{P} = \mathbf{I} - \frac{1}{n}\mathbb{1}_n$. ($\mathbf{I}$ is the identity.)
In a slightly more general setting, we first summarize properties of symmetric matrices $\mathbf{P}_n(\alpha) := \mathbf{I} - \alpha\mathbb{1}_n$.

Lemma F.0.1 (Spectral decomposition)**.** *For $\alpha > 0$, matrices $\mathbf{P}_n(\alpha)$ have* eigenvalues *$\lambda_i = 1 - \alpha\mu_i$ for $\mu_i = 0,\ldots,0,n$, and they share* eigenvectors *up to order and orientation.*

Proof. The following holds for eigenvalues λ and eigenvectors $\mathbf{x}$.

$$\begin{aligned}(\mathbf{I}-\alpha\mathbb{1})\mathbf{x}=\lambda\mathbf{x} \quad\Leftrightarrow\quad \mathbf{x}-\alpha\mathbb{1}\mathbf{x} \;=\; \lambda\mathbf{x} \quad\Leftrightarrow\quad \alpha\mathbb{1}\mathbf{x} \;=\; (1-\lambda)\mathbf{x}\\ \mathbb{1}\mathbf{x}=\mu\mathbf{x} \quad \text{with eigenvalues} \quad \mu=\tfrac{1}{\alpha}(1-\lambda)\end{aligned}$$

The matrix $\mathbb{1}=\mathbb{1}_n$ has rank 1 and eigenvalue 0 with multiplicity $n-1$, and the remaining eigenvalue must be equal to n. It is straightforward to confirm that the first $n-1$ *eigenvectors* of $\mathbb{1}_n$ provide an orthonormal basis of the kernel $\{\mathbf{x}\,|\,\mathbb{1}_n\mathbf{x}=\mathbf{0}\}$, and the remaining last (unit length) eigenvector is the constant vector $\mathbf{q}:=\frac{1}{\sqrt{n}}\mathbf{1}_n$. The eigenvalues are shifted for $\mathbf{P}(\alpha)$, and the eigenvectors are the same as for $\mathbb{1}$ up to order and orientation. $\square$

Note that the constant eigenvector $\mathbf{q}$ of $\mathbf{P}(\alpha)$ appears for the smallest eigenvalue due to the shift of eigenvalues. In the following, we write the eigenvectors as columns of the orthogonal matrix $\mathbf{Q}=(\mathbf{q}|\hat{\mathbf{Q}})$, where the columns of $\hat{\mathbf{Q}}$ span the kernel of $\mathbb{1}$ (or equally the image of $\mathbf{P}(\alpha)$). For the *special case* $\mathbf{P}=\mathbf{I}-\frac{1}{n}\mathbb{1}_n$, this gives eigenvalues $\lambda_i=0,1,\ldots,1$. The symmetric matrix $\mathbf{P}$ has rank $n-1$ and is *idempotent* (i.e., $\mathbf{PP}=\mathbf{P}$). It can be written in terms of its spectral decomposition as $\mathbf{P}=\mathbf{Q}\,\mathrm{diag}(0,1,\ldots,1)\,\mathbf{Q}^{\mathrm{T}}=\hat{\mathbf{Q}}\hat{\mathbf{Q}}^{\mathrm{T}}$.
In the following, we consider the cases of n and $n+1$ vectors:
Let $\mathbf{V}=(\mathbf{v}_1|\ldots|\mathbf{v}_n)\in\mathbb{R}^{3\times n}$ and $\bar{\mathbf{V}}=(\mathbf{v}_1|\ldots|\mathbf{v}_{n+1})\in\mathbb{R}^{3\times(n+1)}$. Likewise, we denote

$$\begin{aligned} n\,\mathbf{a} &=\mathbf{V}\mathbf{1}_n \quad\text{and}\quad \mathbf{b}=\mathbf{D}\mathbf{D}^{\mathrm{T}}\mathbf{a}=\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbb{1}_n, \quad\text{and}\\ (n+1)\,\bar{\mathbf{a}} &=\bar{\mathbf{V}}\mathbf{1}_{n+1} \quad\text{and}\quad \bar{\mathbf{b}}=\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}}\bar{\mathbf{a}}=\bar{\mathbf{V}}\bar{\mathbf{P}}\bar{\mathbf{V}}^{\mathrm{T}}\bar{\mathbf{V}}\mathbb{1}_{n+1}\;.\end{aligned}$$

For the sake of a concise notation, the bar over a quantity denotes dimension $n+1$, e.g., $\bar{\mathbf{D}}\in\mathbb{R}^{3\times(n+1)}$ and $\bar{\mathbf{P}}\in\mathbb{R}^{(n+1)\times(n+1)}$, and we omit explicit subscripts.
We show how the *parallel vectors condition* $\bar{\mathbf{a}}=\lambda\bar{\mathbf{b}}$ can be expressed in terms of $\mathbf{a}$ and $\mathbf{b}$ using block decompositions of $\bar{\mathbf{V}}=(\mathbf{V}|\mathbf{v}_{n+1})$ and

$$\bar{\mathbf{P}} \;=\; \left(\begin{array}{c|c} \hat{\mathbf{P}} & -\frac{1}{n+1}\mathbf{1} \\ \hline -\frac{1}{n+1}\mathbf{1}^{\mathrm{T}} & \frac{n}{n+1}\end{array}\right) \quad\text{with}\quad \hat{\mathbf{P}}=\mathbf{P}_n(\tfrac{1}{n+1})=\mathbf{I}-\tfrac{1}{n+1}\mathbb{1}_n\;.$$

Then

$$\begin{aligned}\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}}=\bar{\mathbf{V}}\bar{\mathbf{P}}\bar{\mathbf{V}}^{\mathrm{T}} &= \left(\mathbf{V}\hat{\mathbf{P}}-\tfrac{1}{n+1}\mathbf{v}_{n+1}\mathbf{1}^{\mathrm{T}}\;\middle|\;\tfrac{n}{n+1}\mathbf{v}_{n+1}-\tfrac{1}{n+1}\mathbf{V}\mathbf{1}\right)(\mathbf{V}|\mathbf{v}_{n+1})^{\mathrm{T}}\\ &=\mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}}-\tfrac{1}{n+1}\mathbf{v}_{n+1}\mathbf{1}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}}+\tfrac{1}{n+1}\left(n\mathbf{v}_{n+1}-\mathbf{V}\mathbf{1}\right)\mathbf{v}_{n+1}^{\mathrm{T}}\;.\end{aligned} \tag{F.1}$$

Using Lemma F.0.1, we can express $\hat{\mathbf{P}}$ as a rank-1 update of $\mathbf{P}$ — the zero eigenvalue becomes $\frac{1}{n+1}$ — and obtain

$$\hat{\mathbf{P}} \;=\; \mathbf{P}+\tfrac{1}{n+1}\mathbf{q}\mathbf{q}^{\mathrm{T}} \;=\; \mathbf{P}+\tfrac{1}{(n+1)n}\mathbb{1}\;.$$

We use Equation (F.1) to show Properties (**P3**), (**P4**) and (**P5**), and we use a similar block decomposition to show (**P7**).

Property (**P3**). Let $\mathbf{v}_{n+1} = \mathbf{0}$. Then Equation (F.1) reduces to $\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}} = \mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}}$. With $(n+1)\,\bar{\mathbf{a}} = n\,\mathbf{a}$ the *parallel vectors condition* $\bar{\mathbf{b}} = \lambda\bar{\mathbf{a}}$ becomes

$$\begin{aligned}\mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{1} =& \lambda\mathbf{V}\mathbf{1} \quad \Leftrightarrow\\ \mathbf{V}\left(\mathbf{P} + \tfrac{1}{(n+1)n}\mathbf{1}\mathbf{1}\right)\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{1} =& \lambda\mathbf{V}\mathbf{1} \quad \Leftrightarrow\\ \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{1} \;+\; \tfrac{1}{(n+1)n}\mathbf{V}\mathbf{1}\mathbf{1}\mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{1} =& \lambda\mathbf{V}\mathbf{1} \quad \Leftrightarrow\\ \mathbf{b} \;+\; \tfrac{1}{(n+1)n}\mathbf{V}\mathbf{1}\mathbf{1}\mathbf{V}^{\mathrm{T}}\mathbf{a} =& \lambda\mathbf{a}\;.\end{aligned}$$

Assume that $\mathbf{b} = \mu\mathbf{a}$ holds. Then there exists an eigenvalue

$$\lambda = \mu + \frac{n}{n+1}\mathbf{a}^{\mathrm{T}}\mathbf{a}$$

for which the above equation becomes true because

$$\mathbf{b} + \tfrac{1}{(n+1)n}\mathbf{V}\mathbf{1}\mathbf{1}\mathbf{V}^{\mathrm{T}}\mathbf{a} = \lambda\mathbf{a} \quad \Leftrightarrow \quad \mu\mathbf{a} + \tfrac{1}{(n+1)n}\,n^2\mathbf{a}(\mathbf{a}^{\mathrm{T}}\mathbf{a}) = \lambda\mathbf{a}\;.$$

It is straightforward to show the reverse by fixing λ rather than μ. □

Property (**P4**). Assume $\bar{\mathbf{b}} = \lambda\bar{\mathbf{a}}$ holds, i. e., an eigenvalue λ exists. We consider the left-hand-side of the equation $\bar{\mathbf{b}} - \lambda\bar{\mathbf{a}} = \mathbf{0}$ and substitute $\mathbf{v}_{n+1} = s\,\mathbf{v}$. With $\mathbf{V}\mathbf{1} = n\,\mathbf{a}$, this gives

$$\begin{aligned}\bar{\mathbf{b}} - \lambda\bar{\mathbf{a}} =& \tfrac{1}{n+1}\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}}(n\mathbf{a} + s\mathbf{v}) - \lambda\tfrac{1}{n+1}(n\mathbf{a} + s\mathbf{v})\\ =& \;\tfrac{n}{n+1}\mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}}\mathbf{a} - \tfrac{sn^2}{(n+1)^2}\mathbf{v}\mathbf{a}^{\mathrm{T}}\mathbf{a} - \tfrac{sn^2}{(n+1)^2}\mathbf{a}\mathbf{v}^{\mathrm{T}}\mathbf{a} + \tfrac{s^2n^2}{(n+1)^2}\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{a}\\ &+ \tfrac{s}{n+1}\mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}}\mathbf{v} - \tfrac{s^2n}{(n+1)^2}\mathbf{v}\mathbf{a}^{\mathrm{T}}\mathbf{v} - \tfrac{s^2n}{(n+1)^2}\mathbf{a}\mathbf{v}^{\mathrm{T}}\mathbf{v} + \tfrac{s^3n}{(n+1)^2}\mathbf{v}\mathbf{v}^{\mathrm{T}}\mathbf{v}\\ &- \lambda\tfrac{n}{n+1}\mathbf{a} - \lambda\tfrac{s}{n+1}\mathbf{v}\;.\end{aligned}$$

We choose and substitute

$$\lambda = \tfrac{n}{n+1}\,(s^2 - 2\mu s)\,\mathbf{v}^{\mathrm{T}}\mathbf{v}\;,$$

factor powers of s and obtain

$$\begin{aligned}\bar{\mathbf{b}} - \lambda\bar{\mathbf{a}} =& s^2\,\Big(-\tfrac{n}{(n+1)^2}\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{a} - \tfrac{n^2}{(n+1)^2}\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{a} + \tfrac{n^2}{(n+1)^2}\mathbf{a}^{\mathrm{T}}\mathbf{v}\mathbf{v} - \tfrac{n}{(n+1)^2}\mathbf{a}^{\mathrm{T}}\mathbf{v}\mathbf{v}\\ &+ \tfrac{2\mu n}{(n+1)^2}\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{v}\Big) + O(s)\\ =& s^2\,\tfrac{1}{(1+n)^2}\,\Big((n^2 - n)\mathbf{a}^{\mathrm{T}}\mathbf{v}\mathbf{v} - (n^2 + n)\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{a} + 2\mu n\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{v}\Big) + O(s)\;.\end{aligned}$$

The computation of the above term is somewhat lengthy but elementary. We give a few remarks: First, the choice of eigenvector λ requires a term that is *linear* in s. Second, s^3 is the highest power that appears in the derivation. However, the cubic terms sum to zero and disappear in

the result. (This is independent of the particular choice of λ. There the additional linear term generates a term in s^2 that is required to ensure no solution is lost.)
We can now evaluate the limit $\lim_{s\to\infty} \frac{1}{s^2}(\bar{\mathbf{b}} - \lambda\bar{\mathbf{a}})$ and obtain the equation

$$(n^2 - n)\mathbf{a}^{\mathrm{T}}\mathbf{v}\mathbf{v} - (n^2 + n)\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{a} + 2\mu n\mathbf{v}^{\mathrm{T}}\mathbf{v}\mathbf{v} \;=\; \mathbf{0} \;,$$

which holds iff the *parallel eigenvectors condition* $\mathbf{a} = \mu\mathbf{v}$ is satisfied. □

Property (**P5**). Let $\mathbf{v}_{n+1} = s\sum_{i=1}^{n}\mathbf{v}_i = s\,\mathbf{V}\mathbf{1}$. Then

$$\begin{aligned}
\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}} &= \mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}} - \tfrac{s}{n+1}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}} + \tfrac{s^2\,n}{n+1}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}} - \tfrac{s}{n+1}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}} \\
&= \mathbf{V}\hat{\mathbf{P}}\mathbf{V}^{\mathrm{T}} + \frac{n^2(s^2\,n - 2s)}{n+1}\mathbf{a}\mathbf{a}^{\mathrm{T}} \\
&= \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \tfrac{1}{(n+1)n}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}} + \frac{n^2(s^2\,n - 2s)}{n+1}\mathbf{a}\mathbf{a}^{\mathrm{T}} \\
&= \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \frac{n\,(1 - s\,n)^2}{n+1}\mathbf{a}\mathbf{a}^{\mathrm{T}} \;.
\end{aligned}$$

With

$$\bar{\mathbf{a}} \;=\; \tfrac{1}{n+1}(n\mathbf{a} + s\,\mathbf{V}\mathbf{1}) \;=\; \tfrac{1}{n+1}(n\mathbf{a} + s\,n\mathbf{a}) \;=\; \tfrac{1+s}{n+1}\mathbf{a}$$

we obtain

$$\begin{aligned}
\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}}\bar{\mathbf{a}} = \lambda\bar{\mathbf{a}} \quad &\Leftrightarrow \\
\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{a} + \frac{n\,(1 - s\,n)^2}{n+1}\mathbf{a}(\mathbf{a}^{\mathrm{T}}\mathbf{a}) = \lambda\mathbf{a} \;,&
\end{aligned}$$

which holds if $\mathbf{b} = \mu\mathbf{a}$. The argument is similar as for (**P3**) and uses $\mathbf{b} = \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{a} = \mu\mathbf{a}$. □

Property (**P6**) follows immediately from (**P3**) as

$$\begin{aligned}
\mathbf{APV}(\mathbf{v}_1, \ldots, \mathbf{v}_n, \underbrace{\mathbf{0}, \ldots, \mathbf{0}, \mathbf{0}}_{k+1}) \;&=\; \mathbf{APV}(\mathbf{v}_1\;\ldots, \mathbf{v}_n, \underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{k}) \;=\; \ldots \\
&=\; \mathbf{APV}(\mathbf{v}_1\;\ldots, \mathbf{v}_n) \;.
\end{aligned}$$

□

Property (**P7**). Let $\mathbf{W} = \mathbf{w}\mathbf{1}_k = (\mathbf{w}, \ldots, \mathbf{w}) \in \mathbb{R}^{m\times k}$. We consider $\bar{\mathbf{V}} = (\mathbf{V}|\mathbf{W}) \in \mathbb{R}^{m\times(n+k)}$. Similarly, as before, we define

$$\bar{\mathbf{P}} \;=\; \left(\begin{array}{c|c} \mathbf{P}_n & -\alpha\mathbb{1}_{n\times k} \\ \hline -\alpha\mathbb{1}_{k\times n} & \mathbf{P}_k \end{array}\right) \;\in \mathbb{R}^{(n+k)\times(n+k)}$$

where $\alpha = \frac{1}{k+n}$ and $\mathbf{P}_n = \mathbf{P}_n(\alpha)$ and $\mathbf{P}_k = \mathbf{P}_k(\alpha)$. Then

$$\begin{aligned}\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}} =& \bar{\mathbf{V}}\bar{\mathbf{P}}\bar{\mathbf{V}}^{\mathrm{T}} = \mathbf{V}\mathbf{P}_n\mathbf{V}^{\mathrm{T}} - \alpha\mathbf{W}\mathbb{1}_{k\times n}\mathbf{V}^{\mathrm{T}} - \alpha\mathbf{V}\mathbb{1}_{n\times k}\mathbf{W}^{\mathrm{T}} + \mathbf{W}\mathbf{P}_k\mathbf{W}^{\mathrm{T}} \\ =& \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \alpha\, k\, n\, \mathbf{A} \quad \in \mathbb{R}^{m\times m}\end{aligned}$$

with $\mathbf{A} = \mathbf{a}\mathbf{a}^{\mathrm{T}} - \mathbf{a}\mathbf{w}^{\mathrm{T}} - \mathbf{w}\mathbf{a}^{\mathrm{T}} + \mathbf{w}\mathbf{w}^{\mathrm{T}}$ and $\mathbf{P} = \mathbf{P}_n(\frac{1}{n})$. To confirm this equivalence, we study each term of the quadratic form:
From $\mathbf{P}_n = \mathbf{P} + \frac{k}{k+n}\mathbf{q}\mathbf{q}^{\mathrm{T}} = \mathbf{P} + \frac{k}{n(k+n)}\mathbb{1}$ (rank update) we obtain

$$\begin{aligned}\mathbf{V}\mathbf{P}_n\mathbf{V}^{\mathrm{T}} =& \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \frac{k}{n(k+n)}\mathbf{V}\mathbb{1}\mathbf{V}^{\mathrm{T}} = \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \frac{k\,n}{k+n}\mathbf{a}\mathbf{a}^{\mathrm{T}} \\ =& \mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \alpha k\, n\, \mathbf{a}\mathbf{a}^{\mathrm{T}}\ .\end{aligned}$$

We have

$$-\alpha\mathbf{W}\mathbb{1}\mathbf{V}^{\mathrm{T}} = -\alpha\, k\mathbf{w}\mathbf{1}_n^{\mathrm{T}}\mathbf{V}^{\mathrm{T}} = -\alpha\, k\mathbf{w}(\mathbf{V}\mathbf{1}_n)^{\mathrm{T}} = -\alpha\, k\, n\, \mathbf{w}\mathbf{a}^{\mathrm{T}}$$

and $-\alpha\mathbf{V}\mathbb{1}\mathbf{W}^{\mathrm{T}} = -\alpha\, k\, n\, \mathbf{a}\mathbf{w}^{\mathrm{T}}$.
And finally, $\mathbf{W}\mathbf{P}_k\mathbf{W}^{\mathrm{T}} = \mathbf{w}\mathbf{1}^{\mathrm{T}}\mathbf{P}_k\mathbf{1}\mathbf{w} = \frac{k\,n}{k+n}\mathbf{w}\mathbf{w}^{\mathrm{T}} = \alpha\, k\, n\, \mathbf{w}\mathbf{w}^{\mathrm{T}}$ using $\mathbf{1}^{\mathrm{T}}\mathbf{P}_k\mathbf{1} = (1-\alpha\, k)\, k = \alpha\, k\, n$.
With the mean $\bar{\mathbf{a}} = \alpha(n\,\mathbf{a} + k\mathbf{w})$ we write the condition

$$\bar{\mathbf{D}}\bar{\mathbf{D}}^{\mathrm{T}}\,\bar{\mathbf{a}} \;=\; \lambda\,\bar{\mathbf{a}} \quad\Leftrightarrow\quad (\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} + \alpha\, k\, n\, \mathbf{A})\,\mathbf{a} \;=\; \lambda\,\mathbf{a}\ ,$$

which gives

$$\begin{aligned}&\frac{n}{k+n}\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{a} + \frac{k}{k+n}\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{w} \\ &\qquad + \frac{k\,n^2}{(k+n)^2}\mathbf{A}\mathbf{a} + \frac{k^2\,n}{(k+n)^2}\mathbf{A}\mathbf{w} - \frac{\lambda\,n}{k+n}\mathbf{a} - \frac{\lambda\,k}{k+n}\mathbf{w} \;=\; \mathbf{0}.\end{aligned}$$

In the limit $k\to\infty$ this reduces to the condition

$$\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}}\mathbf{w} + n\left(\mathbf{a}\mathbf{a}^{\mathrm{T}} - \mathbf{a}\mathbf{w}^{\mathrm{T}} - \mathbf{w}\mathbf{a}^{\mathrm{T}} + \mathbf{w}\mathbf{w}^{\mathrm{T}}\right)\mathbf{w} - \lambda\mathbf{w} \;=\; \mathbf{0}\ ,$$

and with $\mathbf{V}\mathbf{P}\mathbf{V}^{\mathrm{T}} = \mathbf{D}\mathbf{D}^{\mathrm{T}} = \mathbf{V}\mathbf{V}^{\mathrm{T}} - n\,\mathbf{a}\mathbf{a}^{\mathrm{T}}$ we obtain

$$\begin{aligned}&\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{w} + n\left(\mathbf{w}\mathbf{w}^{\mathrm{T}} - \mathbf{a}\mathbf{w}^{\mathrm{T}} - \mathbf{w}\mathbf{a}^{\mathrm{T}}\right)\mathbf{w} - \lambda\mathbf{w} \;=\; \mathbf{0} \quad\Leftrightarrow \\ &\mathbf{V}\mathbf{V}^{\mathrm{T}}\mathbf{w} - n\,||\mathbf{w}||^2\,\mathbf{a} \;=\; \mu\mathbf{w} \quad \text{for} \quad \mu = \lambda + n(\mathbf{w}^{\mathrm{T}}\mathbf{a} - ||\mathbf{w}||^2)\ ,\end{aligned}$$

which is the postulated *parallel vectors condition.* □

BIBLIOGRAPHY

[1] A. Abbasloo, V. Wiens, M. Hermann, and T. Schultz, "Visualizing tensor normal distributions at multiple levels of detail," *IEEE TVCG*, vol. 22, pp. 975–984, 2016 (cit. on pp. 45, 80, 81, 83, 84, 97–99).

[2] A. Aldroubi and P. Basser, "Reconstruction of vector and tensor fields from sampled discrete data," *Contemporary Mathematics*, vol. 247, pp. 1–16, 1999 (cit. on p. 45).

[3] R. S. Allendes Osorio and K. W. Brodlie, "Uncertain flow visualization using LIC," in *Theory and Practice of Computer Graphics, Eurographics UK Chapter Proceedings*, Eurographics Association, 2009, pp. 215–222 (cit. on p. 115).

[4] P. Angelelli, S. R. Snare, S. A. Nyrnes, S. Bruckner, H. Hauser, and L. Løvstakken, "Live ultrasound-based particle visualization of blood flow in the heart," in *Proceedings of the 30th Spring Conference on Computer Graphics*, 2014, pp. 13–20 (cit. on p. 107).

[5] C. Auer and I. Hotz, "Complete tensor field topology on 2d triangulated manifolds embedded in 3d," in *Computer Graphics Forum*, Wiley Online Library, vol. 30, 2011, pp. 831–840 (cit. on p. 37).

[6] C. Auer, J. Kasten, A. Kratz, E. Zhang, and I. Hotz, "Automatic, tensor-guided illustrative vector field visualization," in *2013 IEEE Pacific Visualization Symposium (PacificVis)*, IEEE, 2013, pp. 265–272 (cit. on p. 44).

[7] M. M. Bahn, "Invariant and orthonormal scalar measures derived from magnetic resonance diffusion tensor imaging," *Journal of magnetic resonance*, vol. 141, no. 1, pp. 68–77, 1999 (cit. on p. 36).

[8] D. C. Banks and B. A. Singer, "A predictor-corrector technique for visualizing unsteady flow," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 151–163, 1995 (cit. on p. 112).

[9] A. H. Barr, "Superquadrics and angle-preserving transformations," *IEEE Computer graphics and Applications*, vol. 1, no. 1, pp. 11–23, 1981 (cit. on p. 42).

[10] P. J. Basser, "Inferring microstructural features and the physiological state of tissues from diffusion-weighted images," *NMR in Biomedicine*, vol. 8, no. 7, pp. 333–344, 1995 (cit. on p. 28).

[11] P. J. Basser and S. Pajevic, "A normal distribution for tensor-valued random variables: applications to diffusion tensor MRI," *IEEE transactions on medical imaging*, vol. 22, pp. 785–794, 2003 (cit. on pp. 39, 45, 80, 81, 83).

[12] P. J. Basser and C. Pierpaoli, "Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI," *Journal of magnetic resonance*, vol. 213, no. 2, pp. 560–570, 2011 (cit. on p. 27).

[13] P. J. Basser and S. Pajevic, "Spectral decomposition of a 4th-order covariance tensor: Applications to diffusion tensor MRI," *Signal Processing*, vol. 87, no. 2, pp. 220–236, 2007 (cit. on pp. 45, 81, 83, 84, 97, 98).

[14] T. E. Behrens, M. W. Woolrich, M. Jenkinson, H. Johansen-Berg, R. G. Nunes, S. Clare, P. M. Matthews, J. M. Brady, and S. M. Smith, "Characterization and propagation of uncertainty in diffusion-weighted MR imaging," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 50, no. 5, pp. 1077–1088, 2003 (cit. on p. 39).

[15] W. Benger and H.-C. Hege, "Tensor splats," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5295, Jun. 2004 (cit. on p. 44).

[16] W. Benger and H.-C. Hege, "Strategies for direct visualization of second-rank tensor fields," in *Visualization and Processing of Tensor Fields*, Springer, 2006, pp. 191–214 (cit. on p. 38).

[17] P. Berg, C. Roloff, O. Beuing, S. Voss, S.-I. Sugiyama, N. Aristokleous, A. S. Anayiotos, N. Ashton, A. Revell, N. W. Bressloff, *et al.*, "The computational fluid dynamics rupture challenge 2013–phase II: variability of hemodynamic simulations in two intracranial aneurysms," *Journal of biomechanical engineering*, vol. 137, no. 12, 2015 (cit. on p. 125).

[18] C. Bi, L. Yang, Y. Duan, and Y. Shi, "A survey on visualization of tensor field," *Journal of Visualization*, vol. 22, no. 3, pp. 641–660, 2019 (cit. on p. 38).

[19] A. Biswas, G. Lin, X. Liu, and H.-W. Shen, "Visualization of time-varying weather ensembles across multiple resolutions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 841–850, 2016 (cit. on p. 107).

[20] G.-P. Bonneau, H.-C. Hege, C. R. Johnson, M. M. Oliveira, K. Potter, P. Rheingans, and T. Schultz, "Overview and state-of-the-art of uncertainty visualization," in *Scientific Visualization*, Springer, 2014, pp. 3–27 (cit. on pp. 39, 107).

[21] R. Borgo, J. Kehrer, D. H. Chung, E. Maguire, R. S. Laramee, H. Hauser, M. Ward, and M. Chen, "Glyph-based visualization: Foundations, design guidelines, techniques and applications," *Eurographics State of the Art Reports*, pp. 39–63, 2013 (cit. on p. 41).

[22] R. P. Botchen, D. Weiskopf, and T. Ertl, "Texture-based visualization of uncertainty in flow fields," in *VIS 05. IEEE Visualization, 2005.*, IEEE, 2005, pp. 647–654 (cit. on p. 115).

[23] K. Brodlie, R. A. Osorio, and A. Lopes, "A review of uncertainty in data visualization," in *Expanding the frontiers of visual analytics and visualization*, Springer, 2012, pp. 81–109 (cit. on p. 39).

[24] K. W. Brodlie, L. A. Carpenter, R. A. Earnshaw, J. R. Gallop, R. J. Hubbold, A. M. Mumford, C. D. Osland, and P. Quarendon, *Scientific visualization: techniques and applications.* Springer Science & Business Media, 2012 (cit. on p. 9).

[25] B. Cabral and L. C. Leedom, "Imaging vector fields using line integral convolution," Lawrence Livermore National Lab., CA (United States), Tech. Rep., 1993 (cit. on p. 37).

[26] G. Chen, D. Palke, Z. Lin, H. Yeh, P. Vincent, R. S. Laramee, and E. Zhang, "Asymmetric tensor field visualization for surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1979–1988, 2011 (cit. on p. 44).

[27] 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*, vol. 15, no. 6, pp. 1433–1440, 2009 (cit. on p. 38).

[28] J. Cox, D. House, and M. Lindell, "VISUALIZING UNCERTAINTY IN PREDICTED HURRICANE TRACKS," *International Journal for Uncertainty Quantification*, vol. 3, no. 2, pp. 143–156, 2013, ISSN: 2152-5080 (cit. on pp. 108, 113).

[29] 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*, vol. 48, no. 12, pp. 2445–2465, 2000 (cit. on p. 36).

[30] P. Crossno, D. H. Rogers, R. M. Brannon, and D. Coblentz, "Visualization of Salt-Induced Stress Perturbations," in *Proc. IEEE Visualization*, 2004, pp. 369–376, ISBN: 0-7803-8788-0 (cit. on pp. 43, 50).

[31] W. C. de Leeuw and J. J. van Wijk, "A Probe for Local Flow Field Visualization," in *Proc. IEEE Visualization*, San Jose, California, 1993, pp. 39–45, ISBN: 0-8186-3940-7 (cit. on pp. 41, 43, 50).

[32] T. Delmarcelle and L. Hesselink, "Visualization of second order tensor fields and matrix data," in *Proceedings of the 3rd conference on Visualization'92*, IEEE Computer Society Press, 1992, pp. 316–323 (cit. on pp. 32, 37).

[33] T. Delmarcelle and L. Hesselink, "Visualizing second-order tensor fields with hyperstreamlines," *IEEE Computer Graphics and Applications*, vol. 13, no. 4, pp. 25–33, 1993 (cit. on p. 37).

[34] T. Delmarcelle and L. Hesselink, "The topology of symmetric, second-order tensor fields," in *Proceedings of the conference on Visualization'94*, IEEE Computer Society Press, 1994, pp. 140–147 (cit. on p. 38).

[35] 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 (Proceedings of IEEE Visualization 2009)*, vol. 15, no. 6, pp. 1399–1406, 2009 (cit. on pp. 36, 50).

[36] P. Diggle, P. J. Diggle, P. Heagerty, K.-Y. Liang, P. J. Heagerty, S. Zeger, *et al.*, *Analysis of longitudinal data.* Oxford University Press, 2002 (cit. on p. 113).

[37] H. Q. Dinh and L. Xu, "Measuring the similarity of vector fields using global distributions," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, Springer, 2008, pp. 187–196 (cit. on p. 114).

[38] R. Dodd, "A new approach to the visualization of tensor fields," *Graphical Models and Image Processing*, vol. 60, no. 4, pp. 286–303, 1998 (cit. on p. 38).

[39] S. Eichelbaum, M. Hlawitschka, B. Hamann, and G. Scheuermann, "Fabric-like visualization of tensor field data on arbitrary surfaces in image space," in *New Developments in the Visualization and Processing of Tensor Fields*, Springer, 2012, pp. 71–92 (cit. on p. 37).

[40] D. B. Ennis and G. Kindlmann, "Orthogonal tensor invariants and the analysis of diffusion tensor magnetic resonance images," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 55, no. 1, pp. 136–146, 2006 (cit. on pp. 27, 36).

[41] G. Farin, *Curves and Surfaces for CAGD*, 5th. Morgan Kaufmann, 2002 (cit. on pp. 52, 147–149).

[42] F. Ferstl, K. Bürger, and R. Westermann, "Streamline variability plots for characterizing the uncertainty in vector field ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 767–776, 2016 (cit. on pp. 113, 114).

[43] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann, "Time-hierarchical clustering and visualization of weather forecast ensembles," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 831–840, 2016 (cit. on p. 115).

[44] F. Ferstl, M. Kanzler, M. Rautenhaus, and R. Westermann, "Visual Analysis of Spatial Variability and Global Correlations in Ensembles of Iso-Contours," in *Computer Graphics Forum*, vol. 35, 2016, pp. 221–230 (cit. on pp. 113, 114).

[45] P. T. Fletcher and S. Joshi, "Principal geodesic analysis on symmetric spaces: Statistics of diffusion tensors," in *Computer Vision and Mathematical Methods in Medical and Biomedical Image Analysis*, Springer, 2004, pp. 87–98 (cit. on p. 45).

[46] M. Gardner, "Piet Hein's superellipse," *Mathematical Carnival: A New Round-Up of Tentalizers and Puzzles*, pp. 240–254, 1977 (cit. on p. 42).

[47] T. Gerrits, C. Rössl, and H. Theisel, "Glyphs for general second-order 2d and 3d tensors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 980–989, 2016 (cit. on pp. 33, 145, 147, 149).

[48] T. Gerrits, C. Rössl, and H. Theisel, "An approximate parallel vectors operator for multiple vector fields," in *Computer Graphics Forum*, vol. 37, 2018, pp. 315–326 (cit. on p. 157).

[49] T. Gerrits, C. Rössl, and H. Theisel, "Towards Glyphs for Uncertain Symmetric Second-Order Tensors," in *Computer Graphics Forum*, vol. 38, 2019, pp. 325–336 (cit. on pp. 151, 153).

[50] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, "Visual comparison for information visualization," *Information Visualization*, vol. 10, no. 4, pp. 289–309, 2011 (cit. on p. 108).

[51] A. Globus, C. Levit, and T. Lasinski, "A Tool for Visualizing the Topology of Three-dimensional Vector Fields," in *Proc. IEEE Visualization*, San Diego, California, 1991, pp. 33–40, ISBN: 0-8186-2245-8 (cit. on pp. 43, 49, 50).

[52] T. Gneiting and A. E. Raftery, "Weather forecasting with ensemble methods," *Science*, vol. 310, no. 5746, pp. 248–249, 2005 (cit. on p. 107).

[53] T. Günther, M. Schulze, J. Martinez Esturo, C. Rössl, and H. Theisel, "Opacity Optimization for Surfaces," *Computer Graphics Forum*, vol. 33, no. 3, pp. 11–20, 2014 (cit. on p. 100).

[54] T. Günther and H. Theisel, "The State of the Art in Vortex Extraction," *Computer Graphics Forum*, vol. 37, no. 6, pp. 149–173, 2018 (cit. on pp. 38, 110).

[55] T. Günther, C. Rössl, and H. Theisel, "Opacity Optimization for 3D Line Fields," *ACM Transactions on Graphics*, vol. 32, 120:1–120:8, Jul. 2013 (cit. on p. 114).

[56] H. Guo, X. Yuan, J. Huang, and X. Zhu, "Coupled ensemble flow line advection and analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2733–2742, 2013 (cit. on p. 114).

[57] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel, "Saddle Connectors – An Approach to Visualizing the Topological Skeleton of Complex 3D Vector Fields," in *Proc. IEEE Visualization*, 2003, pp. 225–232 (cit. on pp. 43, 50, 64).

[58] R. B. Haber, "Visualization techniques for engineering mechanics," *Computing Systems in Engineering*, vol. 1, no. 1, pp. 37–50, 1990 (cit. on pp. 43, 50).

[59] R. B. Haber and D. A. McNabb, "Visualization idioms: A conceptual model for scientific visualization systems," *Visualization in scientific computing*, vol. 74, p. 93, 1990 (cit. on p. 9).

[60] H. Hagen and S. Hahmann, "Generalized focal surfaces: A new method for surface interrogation," in *Proceedings Visualization'92*, IEEE, 1992, pp. 70–76 (cit. on p. 38).

[61] H. Hagen, S. Hahmann, and H. Weimer, "Visualization of Deformation Tensor Fields.," in *Scientific Visualization*, 1994, pp. 357–371 (cit. on p. 38).

[62] C. D. Hansen, M. Chen, C. R. Johnson, A. E. Kaufman, and H. Hagen, *Scientific visualization: uncertainty, multifield, biomedical, and scalable visualization.* Springer, 2014 (cit. on p. 9).

[63] R. M. Haralick, "Ridges and valleys on digital images," *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 1, pp. 28–38, 1983 (cit. on p. 112).

[64] Y. M. 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*, vol. 27, no. 7, pp. 603–626, 2003 (cit. on p. 43).

[65] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth, "A survey of topology-based methods in visualization," in *Computer Graphics Forum*, Wiley Online Library, vol. 35, 2016, pp. 643–667 (cit. on p. 38).

[66] K. Helbig, “Foundations of Anisotropy for Exploration Seismics,” in *Handbook of Geophysical Exploration: Seismic Exploration*, vol. 22, Elsevier, 1994, ch. Chapter 11 – Eigentensors of the elastic tensor and their relationship with material symmetry, pp. 393–470 (cit. on pp. 27, 151).

[67] J. Helman and L. Hesselink, “Representation and display of vector field topology in fluid flow data sets,” *Computer*, no. 8, pp. 27–36, 1989 (cit. on p. 31).

[68] J. L. Helman and L. Hesselink, “Visualizing vector field topology in fluid flows,” *IEEE Computer Graphics and Applications*, no. 3, pp. 36–46, 1991 (cit. on p. 31).

[69] M. Hermann, A. C. Schunke, T. Schultz, and R. Klein, “Accurate Interactive Visualization of Large Deformations and Variability in Biomedical Image Ensembles,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 708–717, 2016 (cit. on p. 107).

[70] L. Hesselink, Y. Levy, and Y. Lavin, “The topology of symmetric, second-order 3D tensor fields,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 1, pp. 1–11, 1997 (cit. on p. 37).

[71] L. Hesselink, F. H. Post, and J. J. van Wijk, “Research issues in vector and tensor field visualization,” *IEEE Computer Graphics and Applications*, vol. 14, no. 2, pp. 76–79, 1994 (cit. on p. 38).

[72] M. Hlawatsch, P. Leube, W. Nowak, and D. Weiskopf, “Flow Radar Glyphs& Static Visualization of Unsteady Flow with Uncertainty,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1949–1958, 2011 (cit. on pp. 44, 70).

[73] M. Hlawatsch, F. Sadlo, H. Jang, and D. Weiskopf, “Pathline glyphs,” in *Computer Graphics Forum*, Wiley Online Library, vol. 33, 2014, pp. 497–506 (cit. on pp. 44, 70).

[74] M. Hlawitschka, Y. Hijazi, A. Knoll, and B. Hamann, “Towards a High-quality Visualization of Higher-order Reynold’s Glyphs for Diffusion Tensor Imaging,” in *Visualization in Medicine and Life Sciences II*, Springer, 2012, pp. 209–225 (cit. on p. 142).

[75] M. Hlawitschka and G. Scheuermann, “HOT-Lines: Tracking Lines in Higher Order Tensor Fields,” in *16th IEEE Visualization Conference*, 2005, pp. 27–34 (cit. on p. 100).

[76] M. Hlawitschka, G. Scheuermann, and B. Hamann, “Interactive glyph placement for tensor fields,” in *International Symposium on Visual Computing*, Springer, 2007, pp. 331–340 (cit. on p. 45).

[77] T. Hollt, M. Hadwiger, O. Knio, and I. Hoteit, "Probability maps for the visualization of assimilation ensemble flow data," in *Workshop on Visualisation in Environmental Sciences (EnvirVis)*, The Eurographics Association, 2015 (cit. on p. 107).

[78] T. Höllt, A. Magdy, P. Zhan, G. Chen, G. Gopalakrishnan, I. Hoteit, C. D. Hansen, and M. Hadwiger, "Ovis: A framework for visual analysisof ocean forecast ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 8, pp. 1114–1126, 2014 (cit. on p. 107).

[79] 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'04*, IEEE Computer Society, 2004, pp. 123–130 (cit. on p. 36).

[80] I. Hotz, L. Feng, B. Hamann, and K. Joy, "Tensor-fields visualization using a fabric-like texture applied to arbitrary two-dimensional surfaces," in *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Springer, 2009, pp. 139–155 (cit. on p. 37).

[81] G. Janiga, P. Berg, S. Sugiyama, K. Kono, and D. Steinman, "The Computational Fluid Dynamics Rupture Challenge 2013—Phase I: Prediction of Rupture Status in Intracranial Aneurysms," *American Journal of Neuroradiology*, vol. 36, no. 3, pp. 530–536, 2015, ISSN: 0195-6108 (cit. on p. 125).

[82] T. Jankun-Kelly and K. Mehta, "Superellipsoid-based, real symmetric traceless tensor glyphs motivated by nematic liquid crystal alignment visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1197–1204, 2006 (cit. on p. 42).

[83] M. Jarema, I. Demir, J. Kehrer, and R. Westermann, "Comparative visual analysis of vector field ensembles," in *IEEE Visual Analytics Science and Technology (VAST)*, 2015, pp. 81–88 (cit. on p. 114).

[84] W. Javed and N. Elmqvist, "Exploring the design space of composite visualization," in *2012 ieee pacific visualization symposium*, IEEE, 2012, pp. 1–8 (cit. on p. 113).

[85] H. JCR, A. Wray, and P. Moin, "Eddies, stream, and convergence zones in turbulent flows," *Center for turbulence research report CTR-S88*, pp. 193–208, 1988 (cit. on p. 110).

[86] J. Jeong and F. Hussain, "On the identification of a vortex," *Journal of fluid mechanics*, vol. 285, pp. 69–94, 1995 (cit. on pp. 31, 110, 112).

[87] B. Jeremić, G. Scheuermann, J. Frey, Z. Yang, B. Hamann, K. I. Joy, and H. Hagen, "Tensor visualizations in computational geomechanics," *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 26, no. 10, pp. 925–944, 2002 (cit. on p. 37).

[88] R. Jianu, C. Demiralp, and D. Laidlaw, "Exploring 3D DTI fiber tracts with linked 2D representations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1449–1456, 2009 (cit. on p. 38).

[89] F. Jiao, J. M. Phillips, Y. Gur, and C. R. Johnson, "Uncertainty visualization in HARDI based on ensembles of ODFs," in *Visualization Symposium (PacificVis)*, 2012, pp. 193–200 (cit. on pp. 44, 83, 84).

[90] D. K. Jones, "Determining and visualizing uncertainty in estimates of fiber orientation from diffusion tensor MRI," *Magnetic Resonance in Medicine*, vol. 49, pp. 7–12, 2003 (cit. on pp. 44, 82, 84).

[91] D. N. Kenwright, "Automatic detection of open and closed separation and attachment lines," in *IEEE Visualization*, 1998, pp. 151–158 (cit. on pp. 31, 112).

[92] M. Kern and R. Westermann, "Clustering Ensembles of 3D Jet-Stream Core Lines," in *Vision, Modeling and Visualization*, H.-J. Schulz, M. Teschner, and M. Wimmer, Eds., The Eurographics Association, 2019 (cit. on p. 114).

[93] G. Kindlmann, "Superquadric tensor glyphs," in *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization*, 2004, pp. 147–154 (cit. on pp. 41, 42, 50, 66, 80).

[94] G. Kindlmann and C. Scheidegger, "An Algebraic Process for Visualization Design," *IEEE TVCG*, vol. 20, no. 12, pp. 2181–2190, Nov. 2014 (cit. on p. 63).

[95] G. Kindlmann and C.-F. Westin, "Diffusion tensor visualization with glyph packing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1329–1336, 2006 (cit. on p. 45).

[96] R. M. Kirby, H. Marmanis, and D. H. Laidlaw, "Visualizing multivalued data from 2D incompressible flows using concepts from painting," in *Proceedings of the conference on Visualization'99: celebrating ten years*, IEEE Computer Society Press, 1999, pp. 333–340 (cit. on p. 43).

[97] A. Kratz, "Three-Dimensional Second-Order Tensor Fields: Exploratory Visualization and Anisotropic Sampling," Ph.D. dissertation, Free University of Berlin, 2013 (cit. on p. 43).

[98] A. Kratz, C. Auer, and I. Hotz, "Tensor Invariants and Glyph Design," in *Visualization and Processing of Tensors and Higher Order Descriptors for Multi-Valued Data*, C.-F. Westin, A. Vilanova, and B. Burgeth, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 17–34 (cit. on p. 43).

[99] A. Kratz, C. Auer, M. Stommel, and I. Hotz, "Visualization and Analysis of Second-Order Tensors: Moving Beyond the Symmetric Positive-Definite Case," in *Computer Graphics Forum*, Wiley Online Library, vol. 32, 2013, pp. 49–74 (cit. on pp. 38, 44, 50).

[100] A. Kratz, B. Meyer, and I. Hotz, "A Visual Approach to Analysis of Stress Tensor Fields," in *Scientific Visualization: Interactions, Features, Metaphors*, H. Hagen, Ed., vol. 2, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2011, pp. 188–211 (cit. on pp. 36, 38, 43, 50).

[101] R. Kriz, M. Yaman, M. Harting, and A. Ray, "Visualization of zeroth, second, fourth, higher order tensors, and invariance of tensor equations," *Computers and Graphics*, vol. 21, no. 6, pp. 1–13, 2005 (cit. on p. 43).

[102] M. Kubicki, R. McCarley, C.-F. Westin, H.-J. Park, S. Maier, R. Kikinis, F. A. Jolesz, and M. E. Shenton, "A review of diffusion tensor imaging studies in schizophrenia," *Journal of psychiatric research*, vol. 41, no. 1-2, pp. 15–30, 2007 (cit. on p. 38).

[103] B. M. Kutz, U. Kowarsch, M. Keßler, and E. Krämer, "Numerical investigation of helicopter rotors in ground effect," in *30th AIAA Applied Aerodynamics Conference*, vol. 7, 2012 (cit. on p. 126).

[104] 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 Visualization'98*, IEEE, 1998, pp. 127–134 (cit. on p. 42).

[105] R. S. Laramee, G. Erlebacher, C. Garth, T. Schafhitzel, H. Theisel, X. Tricoche, T. Weinkauf, and D. Weiskopf, "Applications of texture-based flow visualization," *Engineering Applications of Computational Fluid Mechanics*, vol. 2, no. 3, pp. 264–274, 2008 (cit. on p. 70).

[106] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf, "The state of the art in flow visualization: Dense and texture-based techniques," in *Computer graphics forum*, Wiley Online Library, vol. 23, 2004, pp. 203–221 (cit. on p. 108).

[107] R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post, "Topology-based flow visualization, the state of the art," in *Topology-based methods in visualization*, Springer, 2007, pp. 1–19 (cit. on p. 38).

[108] Y. Lavin, R. Batra, and L. Hesselink, "Feature comparisons of vector fields using Earth mover's distance," *Proceedings Visualization'98*, pp. 103–109, 1998 (cit. on p. 33).

[109] Y. Lavin, Y. Levy, and L. Hesselink, "Singularities in nonuniform tensor fields," in *Proceedings. Visualization'97 (Cat. No. 97CB36155)*, IEEE, 1997, pp. 59–66 (cit. on p. 37).

[110] X. Liang, B. Li, and S. Liu, "The deformed cube: a visualization technique for 3d velocity vector field," in *International Computer Science Conference*, Springer, 1995, pp. 51–58 (cit. on p. 43).

[111] A. E. Lie, J. Kehrer, and H. Hauser, "Critical design and realization aspects of glyph-based 3D data visualization," in *Proceedings of the 25th Spring Conference on Computer Graphics*, ACM, 2009, pp. 19–26 (cit. on p. 41).

[112] Z. Lin, H. Yeh, R. S. Laramee, and E. Zhang, "2D Asymmetric Tensor Field Topology," in *Topological Methods in Data Analysis and Visualization II: Theory, Algorithms, and Applications*, R. Peikert, H. Hauser, H. Carr, and R. Fuchs, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 191–204 (cit. on pp. 32, 38).

[113] R. Liu, H. Guo, and X. Yuan, "A bottom-up scheme for user-defined feature comparison in ensemble data," in *SIGGRAPH Asia: Visualization in High Performance Computing*, 2015, p. 10 (cit. on p. 115).

[114] R. Liu, H. Guo, and X. Yuan, "User-defined feature comparison for vector field ensembles," *Journal of Visualization*, vol. 20, no. 2, pp. 217–229, 2017 (cit. on p. 115).

[115] R. Liu, H. Guo, J. Zhang, and X. Yuan, "Comparative visualization of vector field ensembles based on longest common subsequence," in *IEEE Pacific Visualization Symposium (PacificVis)*, 2016, pp. 96–103 (cit. on p. 114).

[116] S. K. Lodha, A. Pang, R. E. Sheehan, and C. M. Wittenbrink, "UFLOW: Visualizing uncertainty in fluid flow," in *Proceedings of Seventh Annual IEEE Visualization'96*, IEEE, 1996, pp. 249–254 (cit. on p. 113).

[117] H. Löffelmann, H. Doleisch, and E. Gröller, "Visualizing dynamical systems near critical points," in *In Spring Conference on Computer Graphics and its Applications*, 1998 (cit. on p. 43).

[118] M. Löffler and J. M. Phillips, "Shape Fitting on Point Sets with Probability Distributions," in *Algorithms – ESA*, A. Fiat and P. Sanders, Eds., Springer, 2009, pp. 313–324 (cit. on p. 45).

[119] B. H. McCormick, "Visualization in scientific computing," *Computer graphics*, vol. 21, no. 6, 1987 (cit. on p. 9).

[120] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen, "Over Two Decades of Integration-Based, Geometric Flow Visualization," in *Computer Graphics Forum*, Wiley Online Library, vol. 29, 2010, pp. 1807–1829 (cit. on p. 70).

[121] M. Meuschke, S. Voß, O. Beuing, B. Preim, and K. Lawonn, “Glyph-based comparative stress tensor visualization in cerebral aneurysms,” in *Computer Graphics Forum*, Wiley Online Library, vol. 36, 2017, pp. 99–108 (cit. on p. 141).

[122] M. Mirzargar, R. T. Whitaker, and R. M. Kirby, “Curve boxplot: Generalization of boxplot for ensembles of curves,” *IEEE TVCG*, vol. 20, no. 12, pp. 2654–2663, 2014 (cit. on pp. 91, 113, 114).

[123] J. G. Moore, S. A. Schorn, and J. Moore, “Education Committee Best Paper of 1995 Award: Methods of Classical Mechanics Applied to Turbulence Stresses in a Tip Leakage Vortex,” *Journal of turbomachinery*, vol. 118, no. 4, pp. 622–629, 1996 (cit. on p. 43).

[124] S. Mori and J.-D. Tournier, *Introduction to diffusion tensor imaging: And higher order models*. Academic Press, 2013 (cit. on p. 29).

[125] H. Obermaier and K. I. Joy, “Future challenges for ensemble visualization,” *IEEE Computer Graphics and Applications*, vol. 34, no. 3, pp. 8–11, 2014 (cit. on p. 115).

[126] H. Obermaier and R. Peikert, “Feature-Based Visualization of Multifields,” in *Scientific Visualization*, Springer, 2014, pp. 189–196 (cit. on p. 108).

[127] S. Oeltze, D. J. Lehmann, A. Kuhn, G. Janiga, H. Theisel, and B. Preim, “Blood Flow Clustering and Applications in Virtual Stenting of Intracranial Aneurysms,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 20(5), pp. 686–701, 2014 (cit. on p. 114).

[128] S. Oeltze-Jafra, J. R. Cebral, G. Janiga, and B. Preim, “Cluster Analysis of Vortical Flow in Simulations of Cerebral Aneurysm Hemodynamics,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 22 (1), no. 1, pp. 757–766, 2016 (cit. on p. 114).

[129] T. Oster, C. Rössl, and H. Theisel, “The Parallel Eigenvectors Operator,” in *Proc. of Vision, Modeling, and Visualization (VMV 2018)*, 2018, to appear (cit. on p. 38).

[130] T. Oster, C. Rössl, and H. Theisel, “Core Lines in 3D Second-Order Tensor Fields,” in *Computer Graphics Forum*, Wiley Online Library, vol. 37, 2018, pp. 327–337 (cit. on p. 38).

[131] M. Otto, T. Germer, H.-C. Hege, and H. Theisel, “Uncertain 2D vector field topology,” in *Computer Graphics Forum*, Wiley Online Library, vol. 29, 2010, pp. 347–356 (cit. on p. 114).

[132] M. Otto, T. Germer, and H. Theisel, “Uncertain topology of 3d vector fields,” in *2011 IEEE Pacific Visualization Symposium*, IEEE, 2011, pp. 67–74 (cit. on p. 114).

[133] M. Otto and H. Theisel, "Vortex analysis in uncertain vector fields," in *Computer Graphics Forum*, Wiley Online Library, vol. 31, 2012, pp. 1035–1044 (cit. on p. 115).

[134] H.-G. Pagendarm and F. H. Post, *Comparative visualization: approaches and examples.* Delft University of Technology, Faculty of Technical Mathematics and . . ., 1995 (cit. on p. 108).

[135] C. Pagot, D. Osmari, F. Sadlo, D. Weiskopf, T. Ertl, and J. Comba, "Efficient Parallel Vectors Feature Extraction from Higher-Order Data," in *Computer Graphics Forum*, vol. 30, 2011, pp. 751–760 (cit. on p. 112).

[136] S. Pajevic, A. Aldroubi, and P. J. Basser, "A continuous tensor field approximation of discrete DT-MRI data for extracting microstructural and architectural features of tissue," *Journal of magnetic resonance*, vol. 154, no. 1, pp. 85–100, 2002 (cit. on p. 45).

[137] S. Pajevic, A. Aldroubi, and P. J. Basser, "Continuous tensor field approximation of diffusion tensor MRI data," in *Visualization and Processing of Tensor Fields*, Springer, 2006, pp. 299–314 (cit. on p. 45).

[138] S. Pajevic and C. Pierpaoli, "Color schemes to represent the orientation of anisotropic tissues from diffusion tensor data: application to white matter fiber tract mapping in the human brain," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 42, no. 3, pp. 526–540, 1999 (cit. on p. 37).

[139] J. Palacios, H. Yeh, W. Wang, Y. Zhang, R. S. Laramee, R. Sharma, T. Schultz, and E. Zhang, "Feature Surfaces in Symmetric Tensor Fields Based on Eigenvalue Manifold," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 3, pp. 1248–1260, 2016 (cit. on p. 38).

[140] D. Palke, G. Chen, Z. Lin, H. Yeh, R. Laramee, and E. Zhang, "Asymmetric tensor visualization with glyph and hyperstreamline placement on 2D manifolds," 2009 (cit. on pp. 44, 50).

[141] A. Parker, C. Cristou, B. Cumming, E. Johnson, M. Hawken, and A. Zisserman, "The analysis of 3D shape: psychological principles and neural mechanisms," *Understanding Vision*, p. 2, 1992 (cit. on p. 42).

[142] R. Peikert and M. Roth, "The "Parallel Vectors" Operator - A Vector Field Visualization Primitive," *IEEE Visualization*, pp. 263–270, 532, 1999 (cit. on pp. 111, 112).

[143] C. Petz, K. Pöthkow, and H.-C. Hege, "Probabilistic local features in uncertain vector fields with spatial correlation," in *Computer Graphics Forum*, Wiley Online Library, vol. 31, 2012, pp. 1045–1054 (cit. on p. 114).

[144] L. Piegl and W. Tiller, "Curve and Surface Constructions Using Rational B-splines," *Compututer Aided Design*, vol. 19, no. 9, pp. 485–498, Nov. 1987, ISSN: 0010-4485 (cit. on pp. 56, 149).

[145] C. Pierpaoli and P. J. Basser, "Toward a quantitative assessment of diffusion anisotropy," *Magnetic resonance in Medicine*, vol. 36, no. 6, pp. 893–906, 1996 (cit. on pp. 27, 36, 42).

[146] A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matković, and H. Hauser, "The State of the Art in Topology-Based Visualization of Unsteady Flow," in *Computer Graphics Forum*, Wiley Online Library, vol. 30, 2011, pp. 1789–1811 (cit. on pp. 70, 108).

[147] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, "The state of the art in flow visualisation: Feature extraction and tracking," in *Computer Graphics Forum*, Wiley Online Library, vol. 22, 2003, pp. 775–792 (cit. on p. 108).

[148] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson, "Ensemble-vis: A framework for the statistical visualization of ensemble data," in *2009 IEEE International Conference on Data Mining Workshops*, IEEE, 2009, pp. 233–240 (cit. on pp. 107, 115).

[149] F. Raith, C. Blecha, T. Nagel, F. Parisio, O. Kolditz, F. Günther, M. Stommel, and G. Scheuermann, "Tensor field visualization using fiber surfaces of invariant space," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1122–1131, 2018 (cit. on p. 36).

[150] M. Rautenhaus, M. Böttinger, S. Siemen, R. Hoffman, R. M. Kirby, M. Mirzargar, N. Röber, and R. Westermann, "Visualization in meteorology—a survey of techniques and tools for data analysis tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 12, pp. 3268–3296, 2017 (cit. on p. 115).

[151] T. Ropinski, S. Oeltze, and B. Preim, "Survey of glyph-based visualization techniques for spatial multivariate medical data," *Computers & Graphics*, vol. 35, no. 2, pp. 392–401, 2011 (cit. on p. 43).

[152] T. Ropinski and B. Preim, "Taxonomy and usage guidelines for glyph-based medical visualization.," in *SimVis*, 2008, pp. 121–138 (cit. on p. 43).

[153] T. Ropinski, M. Specht, J. Meyer-Spradow, K. H. Hinrichs, and B. Preim, "Surface glyphs for visualizing multimodal volume data.," in *VMV*, 2007, pp. 3–12 (cit. on p. 45).

[154] C. Rössl and H. Theisel, "Streamline Embedding for 3D Vector Field Exploration," *IEEE Transactions on Visualiztion and Computer Graphics*, vol. 18-3, pp. 407–420, 2012 (cit. on p. 114).

[155] M. Roth and R. Peikert, "A higher-order method for finding vortex core lines," in *IEEE Visualization*, 1998, pp. 143–150 (cit. on p. 111).

[156] L. Roy, P. Kumar, Y. Zhang, and E. Zhang, "Robust and Fast Extraction of 3D Symmetric Tensor Field Topology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1102–1111, 2018 (cit. on p. 37).

[157] I. A. Sadarjoen and F. H. Post, "Detection, quantification, and tracking of vortices using streamline geometry," *Computers & Graphics*, vol. 24, no. 3, pp. 333–341, 2000 (cit. on p. 31).

[158] T. Salzbrunn, H. Jänicke, T. Wischgoll, and G. Scheuermann, "The State of the Art in Flow Visualization: Partition-based Techniques," in *Simulation and Visualization 2008*, H. Hauser, S. Strassburger, and H. Theisel, Eds., SCS Publishing House, SCS Publishing House, 2008, pp. 75–92 (cit. on p. 108).

[159] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead, "Noodles: A tool for visualization of numerical weather model ensemble uncertainty," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1421–1430, 2010 (cit. on pp. 107, 114).

[160] M. Schöneich, A. Kratz, V. Zobel, G. Scheuermann, M. Stommel, and I. Hotz, "Tensor Lines in Engineering: Success, Failure, and Open Questions," in *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*, Springer, 2015, pp. 339–351 (cit. on p. 37).

[161] W. J. Schroeder, B. Lorensen, and K. Martin, *The visualization toolkit: an object-oriented approach to 3D graphics*. Kitware, 2004 (cit. on p. 45).

[162] W. J. Schroeder, C. R. Volpe, and W. E. Lorensen, "The stream polygon-a technique for 3d vector field visualization," in *Proceeding Visualization'91*, IEEE, 1991, pp. 126–132 (cit. on p. 43).

[163] T. Schultz, "Topological features in 2D symmetric higher-order tensor fields," in *Computer graphics forum*, Wiley Online Library, vol. 30, 2011, pp. 841–850 (cit. on p. 142).

[164] T. Schultz and G. Kindlmann, "A Maximum Enhancing Higher-Order Tensor Glyph," in *Computer Graphics Forum*, Wiley Online Library, vol. 29, 2010, pp. 1143–1152 (cit. on p. 142).

[165] T. Schultz and G. Kindlmann, "Superquadric glyphs for symmetric second-order tensors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1595–1604, 2010 (cit. on pp. 41–43, 48, 50, 63–65, 80, 93).

[166] T. Schultz, L. Schlaffke, B. Schölkopf, and T. Schmidt-Wilcke, "HiFiVE: A Hilbert Space Embedding of Fiber Variability Estimates for Uncertainty Modeling and Visualization," *Computer Graphics Forum*, vol. 32, no. 3, pp. 121–130, 2013 (cit. on pp. 44, 82, 84).

[167] T. Schultz and H. Seidel, "Estimating Crossing Fibers: A Tensor Decomposition Approach," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1635–1642, 2008 (cit. on p. 100).

[168] T. Schultz, H. Theisel, and H.-P. Seidel, "Topological visualization of brain diffusion MRI data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1496–1503, 2007 (cit. on p. 39).

[169] N. Seltzer and G. Kindlmann, "Glyphs for Asymmetric Second-Order 2D Tensors," *Computer Graphics Forum*, vol. 35, no. 3, pp. 141–150, 2016 (cit. on pp. 44, 50, 51, 65, 66).

[170] 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 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM internation conference on Information and knowledge management*, ACM, 1999, pp. 17–20 (cit. on p. 42).

[171] S. Stegmaier, U. Rist, and T. Ertl, "Opening the can of worms: An exploration tool for vortical flows," in *IEEE Visualization*, 2005, pp. 463–470 (cit. on p. 112).

[172] A. H. Stevens, T. Butkiewicz, and C. Ware, "Hairy Slices: Evaluating the Perceptual Effectiveness of Cutting Plane Glyphs for 3D Vector Fields," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 990–999, 2016 (cit. on p. 141).

[173] A. H. Stevens, C. Ware, T. Butkiewicz, D. Rogers, and G. Abram, "Hairy Slices II: Depth Cues for Visualizing 3D Streamlines Through Cutting Planes," *Computer Graphics Forum*, vol. 39, no. 3, I. Viola, M. Gleicher, and T. Landesberger von Antburg, Eds., pp. 25–35, 2020 (cit. on p. 141).

[174] G. Strang, *Introduction to Linear Algebra*, 4th. Wellesley-Cambridge Press, 2009 (cit. on p. 23).

[175] D. Sujudi and R. Haimes, "Identification of swirling flow in 3-D vector fields," in *12th Computational Fluid Dynamics Conference*, 1995, pp. 792–799 (cit. on pp. 31, 110).

[176] J. Sukharev, X. Zheng, and A. Pang, "Tracing parallel vectors," in *Electronic Imaging*, International Society for Optics and Photonics, 2006, pp. 606 011–606 011 (cit. on p. 112).

[177] T. Weinkauf, H. Theisel, and O. Sorkine, "Cusps of Characteristic Curves and Intersection-Aware Visualization of Path and Streak Lines," in *Proc. TopoInVis*, 2011 (cit. on p. 114).

[178] A. C. Telea, *Data Visualization: Principles and Practice, Second Edition*, 2nd. Natick, MA, USA: A. K. Peters, Ltd., 2014, ISBN: 1466585269, 9781466585263 (cit. on p. 9).

[179] H. Theisel and H. Seidel, "Feature Flow Field," in *Proceedings of the symposium on Data visualisation*, vol. 2003, 2003 (cit. on pp. 71, 77, 112).

[180] H. Theisel, J. Sahner, T. Weinkauf, H.-C. Hege, and H.-P. Seidel, "Extraction of parallel vector surfaces in 3D time-dependent fields and application to vortex core line tracking," in *IEEE Visualization*, IEEE, 2005, pp. 631–638 (cit. on p. 112).

[181] H. Theisel and T. Weinkauf, "Vector field metrics based on distance measures of first order critical points," *Journal of WSCG*, vol. 10, no. 3, pp. 121–128, 2002 (cit. on pp. 33, 34, 52).

[182] C. Tominski, P. Schulze-Wollgast, and H. Schumann, "3d information visualization for time dependent data on maps," in *Ninth International Conference on Information Visualisation (IV'05)*, IEEE, 2005, pp. 175–181 (cit. on p. 44).

[183] X. Tricoche, G. Kindlmann, and C.-F. Westin, "Invariant crease lines for topological and structural analysis of tensor fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1627–1634, 2008 (cit. on p. 36).

[184] X. Tricoche and G. Scheuermann, "Topology simplification of symmetric, second-order 2D tensor fields," in *Geometric Modeling for Scientific Visualization*, Springer, 2004, pp. 275–291 (cit. on p. 37).

[185] X. Tricoche, G. Scheuermann, and H. Hagen, "Tensor Topology Tracking: A Visualization Method for Time-Dependent 2D Symmetric Tensor Fields," in *Computer Graphics Forum*, Wiley Online Library, vol. 20, 2001, pp. 461–470 (cit. on p. 39).

[186] X. Tricoche, X. Zheng, and A. Pang, "Visualizing the topology of symmetric, second-order, time-varying two-dimensional tensor fields," in *Visualization and Processing of Tensor Fields*, Springer, 2006, pp. 225–240 (cit. on p. 39).

[187] S.-K. Ueng, C. Sikorski, and K.-L. Ma, "Efficient streamline, streamribbon, and streamtube constructions on unstructured grids," *IEEE Transactions on Visualization and Computer Graphics*, vol. 2, no. 2, pp. 100–110, 1996 (cit. on p. 114).

[188] M. Uffinger, F. Sadlo, and T. Ertl, "A time-dependent vector field topology based on streak surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 379–392, 2013 (cit. on p. 38).

[189] A. Van Gelder and A. Pang, "Using PVsolve to analyze and locate positions of parallel vectors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 4, pp. 682–695, 2009 (cit. on p. 112).

[190] J. J. Van Wijk, "Image based flow visualization," in *ACM Transactions on Graphics (ToG)*, ACM, vol. 21, 2002, pp. 745–754 (cit. on p. 37).

[191] V. Verma and A. Pang, "Comparative flow visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 6, pp. 609–624, 2004 (cit. on pp. 108, 113).

[192] B. Wang and I. Hotz, "Robustness for 2D symmetric tensor field topology," in *Modeling, Analysis, and Visualization of Anisotropy*, Springer, 2017, pp. 3–27 (cit. on p. 37).

[193] J. Wang, S. Hazarika, C. Li, and H.-W. Shen, "Visualization and visual analysis of ensemble data: A survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 9, pp. 2853–2872, 2018 (cit. on p. 107).

[194] J. Wang, X. Liu, H.-W. Shen, and G. Lin, "Multi-resolution climate ensemble parameter analysis with nested parallel coordinates plots," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 81–90, 2016 (cit. on p. 108).

[195] M. O. Ward, "A taxonomy of glyph placement strategies for multidimensional data visualization," *Information Visualization*, vol. 1, no. 3-4, pp. 194–210, 2002 (cit. on p. 45).

[196] M. O. Ward, "Multivariate data glyphs: Principles and practice," in *Handbook of data visualization*, Springer, 2008, pp. 179–198 (cit. on p. 41).

[197] V. J. Wedeen, T. G. Reese, V. J. Napadow, and R. J. Gilbert, "Demonstration of primary and secondary muscle fiber architecture of the bovine tongue by diffusion tensor magnetic resonance imaging," *Biophysical Journal*, vol. 80, no. 2, pp. 1024–1028, 2001 (cit. on p. 42).

[198] J. Weickert and M. Welk, "Tensor field interpolation with PDEs," in *Visualization and processing of tensor fields*, Springer, 2006, pp. 315–325 (cit. on p. 45).

[199] T. Weinkauf, H. Theisel, A. Van Gelder, and A. Pang, "Stable feature flow fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 6, pp. 770–780, 2011 (cit. on pp. 71, 112).

[200] D. Weinstein, G. Kindlmann, and E. Lundberg, "Tensorlines: Advection-diffusion based propagation through diffusion tensor fields," in *Proceedings of the conference on Visualization'99: celebrating ten years*, IEEE Computer Society Press, 1999, pp. 249–253 (cit. on p. 37).

[201] C.-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis, “Processing and visualization for diffusion tensor MRI,” *Medical image analysis*, vol. 6, no. 2, pp. 93–108, 2002 (cit. on pp. 27, 36).

[202] R. T. Whitaker, M. Mirzargar, and R. M. Kirby, “Contour Boxplots: A Method for Characterizing Uncertainty in Feature Sets from Simulation Ensembles.,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2713–2722, 2013 (cit. on pp. 113, 114).

[203] A. Wiebel, S. Koch, and G. Scheuermann, “Glyphs for nonlinear vector field singularities,” in *Topological Methods in Data Analysis and Visualization II*, Springer, 2012, pp. 177–190 (cit. on pp. 43, 44).

[204] M. R. Wiegell, H. B. Larsson, and V. J. Wedeen, “Fiber crossing in human brain depicted with diffusion tensor MR imaging,” *Radiology*, vol. 217, no. 3, pp. 897–903, 2000 (cit. on p. 42).

[205] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha, “Glyphs for visualizing uncertainty in vector fields,” *IEEE transactions on Visualization and Computer Graphics*, vol. 2, no. 3, pp. 266–279, 1996 (cit. on pp. 17, 44, 115).

[206] H. Wright, *Introduction to Scientific Visualization.* Springer London, 2006, ISBN: 9781846284946 (cit. on p. 9).

[207] S. Zellmann, M. Aumüller, N. Marshak, and I. Wald, “High-Quality Rendering of Glyphs Using Hardware-Accelerated Ray Tracing,” in *Eurographics Symposium on Parallel Graphics and Visualization*, S. Frey, J. Huang, and F. Sadlo, Eds., The Eurographics Association, 2020 (cit. on p. 45).

[208] C. Zhang, M. Caan, T. Höllt, E. Eisemann, and A. Vilanova, “Overview + detail visualization for ensembles of diffusion tensors,” *Computer Graphics Forum*, vol. 36, no. 3, pp. 121–132, 2017 (cit. on pp. 45, 81, 83, 84).

[209] C. Zhang, T. Schultz, K. Lawonn, E. Eisemann, and A. Vilanova, “Glyph-based Comparative Visualization for Diffusion Tensor Fields,” *IEEE TVCG*, vol. 22, no. 1, pp. 797–806, 2016 (cit. on pp. 43, 81, 100).

[210] E. Zhang, H. Yeh, Z. Lin, and R. S. Laramee, “Asymmetric tensor analysis for flow visualization,” *IEEE TVCG*, vol. 15, no. 1, pp. 106–122, 2009 (cit. on pp. 32, 38, 44, 50).

[211] X. Zheng and A. Pang, “HyperLIC,” in *IEEE Visualization, 2003. VIS 2003.*, IEEE, 2003, pp. 249–256 (cit. on p. 37).

[212] X. Zheng and A. Pang, “2d asymmetric tensor analysis,” in *Visualization, 2005. VIS 05. IEEE*, IEEE, 2005, pp. 3–10 (cit. on pp. 32, 38, 44).

[213] X. Zheng, B. Parlett, and A. Pang, "Topological structures of 3D tensor fields," in *VIS 05. IEEE Visualization, 2005.*, IEEE, 2005, pp. 551–558 (cit. on p. 37).

[214] L. Zhukov and A. H. Barr, "Oriented tensor reconstruction: Tracing neural pathways from diffusion tensor MRI," in *Proceedings of the conference on Visualization'02*, IEEE Computer Society, 2002, pp. 387–394 (cit. on p. 81).

[215] L. Zhukov and A. H. Barr, "Heart-muscle fiber reconstruction from diffusion tensor MRI," in *IEEE Visualization, 2003. VIS 2003.*, IEEE, 2003, pp. 597–602 (cit. on p. 42).

[216] V. Zobel and G. Scheuermann, "Extremal curves and surfaces in symmetric tensor fields," *The Visual Computer*, vol. 34, no. 10, pp. 1427–1442, 2018 (cit. on p. 38).