Tensor volume exploration using attribute space representatives

Jochen Jankowai*

Robin Skånberg[†]

Daniel Jönsson[‡]

Anders Ynnerman[§]

Ingrid Hotz[¶]

Linköping University, Norrköping, Sweden

Figure 1: Attribute space filtered volume and corresponding classification (atlas) renderings of various tensor data sets. Renderings are supplemented with legends showing attribute representatives as domain specific glyphs. Color-coding across glyphs, atlas and volumes is consistent. (a) Brain DTI data set clustered in the barycentric attribute space spanned by linear, planar, and spherical anisotropy. The selected cluster (red) describes near-isotropic behaviour and is augmented on the left by tensor lines for additional directional information. (b) Rotating neutron star data set, segmented using the attribute space given by the eigenvalues of the tensors. The volume rendering on the left shows concentric shells of gravity distortion, complemented by directional textures. The right rendering shows the selected segment of the atlas.

ABSTRACT

While volume rendering for scalar fields has been advanced into a powerful visualisation method, similar volumetric representations for tensor fields are still rare. The complexity of the data challenges not only the rendering but also the design of the transfer function. In this paper we propose an interface using glyph widgets to design a transfer function for the rendering of tensor data sets. Thereby the transfer function (TF) controls a volume rendering which represents sought after tensor-features and a texture that conveys directional information. The basis of the design interface is a two-dimensional projection of the attribute space. Characteristic representatives in the form of glyphs support an intuitive navigation through the attribute space. We provide three different options to select the representatives: automatic selection based on attribute space clustering, uniform sampling of the attribute space, or manually selected representatives. In contrast to glyphs placed into the 3D volume, we use glyphs with complex geometry as widgets to control the shape and extent of the representatives. In the final rendering the glyphs with their assigned colors play a similar role as a legend in an atlas like representation. The method provides an overview of the tensor field in the 3D volume at the same time as it allows the user to explore the tensor field in an attribute space. We demonstrate the flexibility of our approach on tensor fields for selected data sets with very different characteristics.

Index Terms: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Clustering; I.3.5 [Computer Graph-

ics]: Computational Geometry and Object Modeling—Object hierarchies; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.3.8 [Computer Graphics]: Applications;

1 INTRODUCTION

Tensors are widely used in engineering, physics, mathematics, and many other scientific fields as descriptors for anisotropic physical entities. They can describe states such as stresses in a material, material properties like elasticity or diffusivity, or geometric properties as curvature and metrics. Often, they also appear as intermediate products in computations or during the analysis of scalars or vector fields as directional derivatives. A prominent example is the Jacobian which plays a central role in vector field analysis. Every application is accompanied by its own terminology and interpretation.

From a visualisation point of view we are confronted with all these different appearances of tensors which often seem to be entirely unrelated and completely different entities. Stress and diffusion tensors are one example. This diversity and the complexity of the tensor itself make it very difficult to design general visualisation concepts. A further challenge is that there is no long tradition of tensor field visualisation. While there is a strong need for visualisation, the willingness to adapt to novel representations is relatively low while the mistrust in visual representations is comparably high.

The concept behind the approach presented in this paper builds on the experiences and the success of volume rendering to visualise and explore scalar data sets. A crucial component in volume rendering is not only a high quality rendering but also intuitive interaction options for transfer function design. Often histograms or scatter plots are provided to support the transfer function interaction. Moving from scalar to more complex data, like tensors, these concepts are challenged in several ways. Firstly, there is no easy way to encode the multi-dimensional information via simple colour mapping. Secondly, the dimension of the interaction space for transfer function interaction increases with data dimensionality and rank and is not easily accessible.

The tensor volume rendering concept proposed in this paper

*e-mail: jochen.jankowai@liu.se

[†]e-mail: robin.skanberg@liu.se

[‡]e-mail: daniel.jonsson@liu.se

[§]e-mail: anders.ynnerman@liu.se

[¶]e-mail: ingrid.hotz@liu.se

Figure 2: Concept for tensor field volume rendering summarising the major steps of the pipeline. The first part in the pipeline configures the attribute space by selecting an attribute space of interest and defining representatives within this space either through clustering, manual selection or uniform sampling. The representatives, rendered as domain specific glyphs, then serve as widgets in an interaction panel. They are displayed in a linear fashion and their size indicate the maximum occurring eigenvalue within the cluster so the user gets an idea of the relation between clusters. For the manually picked representatives, either all feature points within a spherical neighbourhood of the selected point are assigned to it or each feature point is assigned to its closest representative point. The tensor transfer function (TTF) can be designed in various ways. (i) Voxel-wise classification of the data through cluster labels. (ii) Radial fall-off with an adjustable Gaussian shape within an adjustable filter shape. (iii) Feature level-sets in attribute space are defined as points with equal distance to the selected representative. We generate surfaces which highlight regions with a fixed distance to the representative by mapping the attribute space surface back to the spatial domain. The final rendering consists of a set of linked views of the selection panel including legend, filtering interface, volume rendering, and, if applicable, atlas.

tackles both of these challenges. To meet the above mentioned challenges, we propose a general concept for volume rendering of tensor data that combines an intuitive interaction interface for tensor transfer function (TTF) design with a textured volume rendering approach. The central idea is to identify representatives to parametrise a two or higher dimensional attribute space. This is done through automatic selection based on attribute space clustering, uniform sampling of the attribute space, or manually selected representatives. Domain specific glyphs are then assigned to the representatives to parametrise the attribute space. The glyphs play a two-fold role: at first they are used as interaction widgets for the transfer function design; second, they serve as legend for the rendering. The domain expert can navigate the attribute space in their familiar context; interacting with a few glyphs in a two-dimensional interaction panel to assign colour and opacity values. Based on these assignment the TTF is assembled either through classification, linear blending or feature level-set generation [13]. Our approach adheres to three design principles; (i) Use as much domain specific knowledge as possible for the interaction with the data while having one underlying concept. This means building on familiar attribute spaces and glyphs. (ii) Put the user in full control over their data without any hidden assumptions, data filtering or smoothing. (iii) Use a spatial rendering concept that can convey the complete tensor information using a glyph-based legend for the scalar tensor information and textures for the directional information.

We demonstrate the utility of the method in the context of data sets from three different applications: Stress tensor field from material simulations, diffusion tensor data MRI imaging and the deformation tensor from the simulation of a rotating neutron star.

The main contributions of the paper can be summarised as:

- A cross-application systematic approach to volumetric tensor visualisation while maintaining domain specific choices of glyphs and other design parameters,
- Automatic, or semi-automatic, assignment of glyph representatives for attribute space parametrisation and navigation for

transfer function settings.

The paper is structured as follows. After a short summary of related work in Sec. 2 we briefly give some background in Sec. 3. In Sec. 4 we explain the general visualisation concept and the pipeline. Some implementation details are given in Sec. 5. The results of the exploration of three different data sets is described in Sec. 6. The paper is concluded in Sec. 8.

2 RELATED WORK

The method presented in this work uses aspects from tensor visualisation, transfer function design and attribute space interaction. The previous work is presented with respect to these three topics.

TENSOR VISUALISATION – Tensors are complex entities and as such challenging to visualise, especially in three dimensional volumes. Commonly used approaches either show a subset of the tensor information in continuous visualisations or the entire tensor information in selected discrete positions. Typically used visualisations in applications render derived scalar measures on cutting planes or on surfaces neglecting the directional information. In contrast, fibre tracts for diffusion tensors [11], or tensor lines for stress tensor fields [25], focus on a single principal direction. The complete tensor information can be encoded in glyphs. When using glyphs in visualisations, the placements and the design of the glyphs are challenging tasks. Various approaches for the placement of glyphs have been proposed [10, 16, 20]. Also, some effort has been put in the development of perceptual optimised glyphs like superquadrics [14, 26]. However, glyph-based visualisations are limited to simple glyphs and a limited resolution, leading to cluttered images for 3-dimensional data. A concept for volume rendering of diffusion tensor data considering more than one scalar tensor invariant has been introduced by Kindlmann et. . [15]. They propose to use a two dimensional barycentric anisotropy space to specify the opacity. The extreme values in the barycentric spaces are displayed using respective tensor glyphs. This interaction space can be considered as a special case of the approach presented in this paper. There have been few other efforts to combine directional and

Figure 3: Filtering pipeline. (a) Attribute points are transformed into the respective cluster space by applying the calculated PCA basis. (b) Using the sliders in the filtering interface, the user then deforms an n -dimensional sphere, where n = number of feature dimensions, to filter along every axis. (c) The points are projected into a perfect n -dimensional sphere where they are assigned an alpha value based on a user-defined Gaussian-shaped fall-off.

scalar information using hybrid rendering. For interactive exploration of tensor data, Kratz et al. [21] propose to link attribute space plots with three-dimensional hybrid visualisations. Brushing and linking techniques are used to select features of interest for a hybrid visualisation. Thereby, volume rendering serves as context, glyphs and tensorlines are used to add more detail. Dick et al. [5] have developed a volumetric visualisation of stress tensors for implant design. They combine tensorlines for the major and minor eigenvector fields with volume rendering. Therefore, they introduce a colour mapping that distinguishes between compressive and tensile forces. Interaction with the data is not focus of the paper. A more detailed overview about tensor field visualisation can be found in [3, 19].

TRANSFER FUNCTION DESIGN FOR HIGHER DIMENSIONAL DATA – Most of the previous work dealing with transfer functions have been dedicated to scalar data, while comparably few address vector or tensor data. An overview on the current state of the art in transfer functions can be found in the survey by Ljung et al. [23]. Here, we focus on transfer functions using multi-dimensional attribute spaces as required for tensor fields. Note that many of the existing multi-dimensional transfer functions have been designed for scalar fields with derived attributes, i.e. gradient magnitude or other statistical properties [1, 2, 4, 24, 27, 30]. They are therefore not directly applicable to tensor fields and do not utilise the spatial information inherent to the tensor. Kniss et al. [17] introduced 2D widgets for interacting with projections of multi-dimensional data. They also presented 3D widgets for interacting with colour data sets. Falk et al. [9] presented a more intuitive way of interacting with colour data sets based on projections of the CIE $L^*u^*v^*$ colour space.

A popular technique for reducing transfer function complexity or dealing with multidimensional data has been to incorporate clustering algorithm in the transfer function method [2, 6, 27, 30]. Wang et al. [27] applied clustering to 2D density plots, which limits their technique to 2D data. Dobrev et al. [6] used density based clustering of high dimensional data in combination with a hierarchical tree widget and parallel coordinates for interaction. We also use clustering for information reduction but instead exploit tensor widgets for interaction.

ATTRIBUTE SPACE INTERACTION – Attribute space interaction is a common way to deal with complex data. Doleisch et al. [7] introduced an interactive visual analysis framework following the concepts of ‘multiple linked views’ and ‘focus and context’ visualisation. Thus, it is possible to integrate multiple derived attributes in the analysis process. Attribute space clustering has been applied by Dobrev et al. [6] for interactive visual analysis of volumetric data.

The user interacts with an abstract cluster tree to assign material properties to the clusters. This approach has some similarities to our concept as a discreet set of clusters is considered for interaction and could be considered as a special case of our concept. Jänicke et al. [12] propose a transfer function design for a high-dimensional attribute space. They generate a so-called attribute cloud by projecting the attributes to a 2D interaction space such that points with similar multivariate attributes are located close to each other. However, this attribute cloud is not very intuitive and it is hard to search for specific features. There have been more applications using attribute space interaction similar to these papers. In most cases however the interaction remains very abstract and it is hard to navigate the attribute space to find features of interest.

3 BACKGROUND

The work in this paper has been developed for three dimensional tensor fields of second order. Such tensors T can be defined as a linear operator from a n -dimensional vector space V into itself $T : V \rightarrow V$. A tensor field assigns a tensor to each point in the domain. In the following we reduce our considerations to three dimensional vector spaces and domains. Given a fixed frame of reference (e_1, e_2, e_3) of the vector space V , a tensor is represented by a three by three matrix. For the analysis of the tensor fields we decomposed them in a symmetric part $S = \frac{1}{2}(T + T^T)$ and an asymmetric part $A = \frac{1}{2}(T - T^T)$. A symmetric tensor S is uniquely defined by its n real eigenvalues (principal values) λ_i and n mutually orthogonal eigenvectors (principal directions) e_i . Commonly, the space spanned by the eigenvalues is referred to as *shape space* and points in this space as *tensor shape* [8]. The *tensor orientation* is given by its eigenvectors. We assume an ordering of the eigenvalues in a descending order, i.e., $\lambda_1 \geq \lambda_2 \geq \lambda_3$. This leads to the so called *ordered shape space* [8] which is a subset of the original shape space. In contrast to the tensor components, the eigenvalues and vectors

Figure 4: Common domain specific glyphs, (a) Reynolds glyph (b) HWY glyph (c) Mohr circle (d) Superquadric glyph

represent inherent properties of the tensor that do not depend on the chosen frame of reference. Such properties are called *tensor invariants*. All scalar tensor invariants can be expressed as functions of the eigenvalues.

4 CONCEPT FOR TENSOR VOLUME RENDERING

In the following, we give an overview of our concept for tensor field volume rendering and interaction. The concept is based on four essential parts: the attribute space setup, the attribute space interaction, the tensor transfer function generation, and the rendering. An overview of the pipeline is shown in Fig. 2. In the following, each part will be described in more detail.

4.1 Attribute space setup

The first part in the pipeline configures the attribute space by selecting features of interest and defining attribute space representatives.

4.1.1 Attribute space selection

Appropriate application specific attribute spaces constitute the centre of the tensor transfer function (TTF) design and ensure a familiar context. We use application specific attribute spaces and glyphs for tensor fields as introduced by Kratz et al. [18, 21]. It is assembled by domain specific tensor invariants and possibly other additional features of interest. To specify the tensor completely, in general three independent invariants are necessary. However, for many applications the set of invariants of interest are not complete. In the following we list some common application specific attribute spaces that we apply in our examples.

SET OF THREE EIGENVALUES – The simplest attribute space represents the three eigenvalues of the tensor. This is a default space that is basically applicable to all symmetric tensor fields. This space is often well suited for a first inspection of the data. A glyph conveying the information of the entire shape space also for non-positive definite tensors is the Reynolds glyph, Fig. 4(a).

LINEAR AND PLANAR ANISOTROPY – A typical attribute space used for diffusion tensor fields combines various anisotropy measures and the type of anisotropy. A barycentric attribute space [28] that is also the basis for the definition of the superquadric glyph, Fig. 4(d), is spanned by linear, planar, and spherical anisotropy, respectively defined as $c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}$, $c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}$, and $c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$. The extrema $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$ can immediately be used as attribute space representatives [15]. This space spans a two dimensional subspace of the complete shape space.

SHEAR-SHAPE-FACTOR SPACE – The shear-shape-factor space is an example for a two-dimensional space for stress tensors from mechanical engineering. Central entities are the maximum shear stress $\tau_{max} = \frac{\lambda_1 - \lambda_2}{2} \in [0, \infty)$ which is responsible to material failure in many failure models. The second invariant is the shape factor $R = \frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3} \in [0, 1]$. It gives insight into the distribution of the directions related to maximum shear. For $R = 0.5$, there are exactly two distinguished normal directions. For $R = 0.0$ and $R = 1.0$, the corresponding normal directions are spanning two cones. The shape factor is well represented by the normalised HWY glyph [22], Fig. 4(b). This reduced space could be completed by adding the mean stress $c = \frac{\lambda_1 + \lambda_2}{2} \in (-\infty, \infty)$ as additional invariant, the complete tensors are then represented by the Mohr diagram, Fig. 4(c).

SHEAR, ISOTROPIC SCALING, ROTATION – The Jacobian derived from flow data is a non-symmetrical tensor T . A typical decomposition is $T = A + \frac{1}{2}Tr(S)I + D$. Thereby, A is the asymmetric part which relates to rotation, S the symmetric part whose trace relates to isotropic scaling, I is the unit tensor, and D is the deviator expressing pure shear [29]. This decomposition leads to a three dimensional attribute space is spanned by $\|A\|$ and $Tr(S)$ and $\|D\|$. There is no established glyph for this decomposition. Mostly rotation and shear are expressed with separate glyphs.

4.1.2 Representative selection

To make the interaction with the abstract attribute space more intuitive, we select a small number of points in attribute space that serve as representatives. These representatives play the central role for the interaction with the data. We support three different options to assemble the set of representatives: automatic generation using attribute space clustering, uniform sampling of the attribute spaces, manual selection and refinement.

AUTOMATIC GENERATION – For some applications the distribution of points in attribute space exhibits distinguished clusters. In this case we define the representatives as maxima in the density field using mean shift clustering. A further cluster analysis using a principal component analysis (PCA) for the clusters provides additional information for a more detailed interaction, Sec. 4.2.

UNIFORM SAMPLING OF ATTRIBUTE SPACE – If there are no distinguished clusters in attribute space a uniform sampling can be applied to generate a predefined number of clusters. This however would also sample empty parts of the attribute space. A good compromise is provided by k-means clustering to generate the representatives.

MANUAL SELECTION – We also provide the option of manually specifying representatives. In some cases interesting attribute space configurations are known from the application context; this can be a critical tensor that would lead to material failure for stress tensor data. In context of diffusion tensor imaging the barycentric space for tensor type is very common and familiar to the domain scientist and the typical types ‘linear’, ‘planar’ and ‘isotropic’ can be used as representatives. Besides these presets, we also allow the user to pick glyphs in a glyph based rendering on 2D slices of the data.

4.2 Attribute space interaction for TTF design

To convey the semantics of the data while navigating in an abstract attribute space we introduce a widget-based interaction panel.

4.2.1 Interaction panels

The representatives defined in Sec. 4.1 serve as widgets in the panel. They are rendered as domain or application specific glyphs. Such glyphs are often complex and not well suited to be rendered in the volume for the entire data set but they are perfect to display the complete information of few representatives.

LEGEND – The automatically generated representative selection assembles the representatives in a linear fashion, similar to an atlas legend. The glyphs indicate the size of the respective cluster, see Fig. 7. For the manually picked representatives either the frequency of feature points within a spherical neighbourhood of the glyph are displayed or the closed cluster is used to define the frequency. By clicking, representatives can be selected, see Fig. 13a and Fig. 13b.

TREEMAP – A treemap shows the segmentation/clusters of the selected attribute space, arranged as nested rectangles where the size of each rectangle indicates the size of the cluster. This method provides a quick overview of the segmentation/cluster structure. Segments/clusters can be selected in the treemap which triggers a highlighting of all associated voxels in the volume rendering.

STAR PLOT – To get a better understanding of the distribution of the clusters and fine-tuning the TTF, the main variations of the clusters are displayed in a star plot. For this, we perform a PCA of the clusters and for each principal axis of the cluster we convey the variation in this direction by adding glyphs, see Fig. 5. The sliders can then be moved along the axes to adjust the TTF for the volume rendering, see Sec. 4.2.2.

We propose two different implementations of the star plot widget, see Fig. 5 and Fig. 10. Fig. 5 shows a conservative layout with filled areas indicating the extent of the cluster in the projected PCA space. Here, the glyphs only indicate general directions but do not show the actual configuration at the extremes of the axes. In addition, a representative of the cluster centre is lacking. The plot shown in

Fig. 10 tackles the aforementioned shortcomings by using glyphs that represent the configurations at the axes endpoints as well as at the centre.

Figure 5: Two point load data set where one pushing and one pulling force are applied to the top of the block. (a) and (b) show star plots for two selected clusters. The filled areas show the extents of the cluster along the axes. In this example, the attribute space axes are given by shear ($\rightleftarrows$, $\rightarrow$), rotation ($\curvearrowright$, $\curvearrowleft$), and expansion/compression ($\otimes$, $\odot$). (c) and (d) show the voxels that belong to the clusters, coloured respectively. (d) shows how filtering along one axis affects the volume rendering.

4.2.2 Tensor transfer function design

Based on the assignment of colour and opacity to the representatives we support various options to assemble the TTF for the entire data set.

CLASSIFICATION – This is the simplest option to map the representatives to a transfer function. Thereby, every representative is assigned one colour and is either turned off or on. Labelling of the data set either using nearest neighbour interpolation in attribute space or the attribute space clusters assigns values to every data point. We refer to this representation as the *data atlas*.

RADIAL FALL-OFF – A cluster interaction panel shows the variance of the cluster either for each attribute space dimension or for its principal components. One axis is added for each dimension showing the variation in the respective direction by adding two glyphs for the extremal points in this cluster, see figures 5 and 10. Filtering sliders enable the user to control the shape of the filter which is applied to the cluster and forms the transfer function for the volumetric rendering. Before applying the filter, the features within the clusters are mapped into a filter space defined by the values of the sliders. The fall-off of the filter in filter space is defined by a Gaussian which shape is configurable by the user. Applying logical operations between the resulting transfer functions facilitates a combination of different tensor characteristics in on image and provides a large variety of design options.

FEATURE LEVEL-SETS – We utilise the concept of feature level sets [13] for a closer inspection of critical regions in the data set.

The idea is to define level-sets in attribute space as points with equal distance to selected representatives. Mapping the attribute space surface back to the spatial domain generates surfaces which highlight regions with a fixed distance to the representative. This is for example interesting when analysing a material with respect to critical stress states that would lead to material failure. In contrast to the commonly used colour mapping of one feature dimension this allows to consider more complex failure criteria.

4.3 Rendering and application of the system

To convey the complete tensor information, scalar and directional aspects of the tensors have to be considered. The *tensor shape*, defined by the scalar tensor properties, is displayed using a volume rendering approach with glyphs as legend. Directional tensor properties are encoded by textures or tensor lines. The feature level sets are displayed as additional surfaces. Selected cluster representatives are rendered on a canvas alongside as legend and bar chart.

Some default configurations for the TTF are provided to obtain an automatic rendering. This is an atlas representation for all clusters, using a k-means clustering in eigenvalue space with 6 clusters or a volume rendering with uniform falloff radii for all clusters. The user can either click use panel for the default setting to start the exploration of the data by changing the transfer function or adding textures or tensor lines. Alternatively other attribute spaces settings can be selected from a pull down menu before starting the exploration. For two or three dimensional attribute spaces also the distribution of the data in attribute space is displayed as scatter plot.

5 IMPLEMENTATION

In the following we will give a brief technical description of the pipeline described in Fig. 2.

The application has been developed in *Inviwo*; an open source visualisation framework based upon modern C++ and OpenGL. The clustering and principal component analysis (PCA) is done through Python package *scikit-learn*, which is conveniently accessible through an interface in *Inviwo*. It offers many clustering methods such as affinity propagation, agglomerative clustering, Birch, DB-SCAN, feature agglomeration, KMeans, mini batch KMeans, mean shift, and spectral clustering.

The clustering techniques could be further extended with additional methods if needed, however the main focus of this paper is not on the clustering in attribute space. Our method uses the result of the clustering as a starting point for feature representatives that the user can then use to explore the attribute space and iteratively filter out the features which are of interest to the user. The application has been designed to make it easy for users to define new attribute spaces that they want to explore. All that needs to be done is to define a attribute space dimensionality and a function for every dimension to calculate the feature. For data sets that contain low-confidence data or noise, we offer an option to ignore that data during the clustering process. This data will be assigned to a designated cluster so it can still be examined. Additionally, feature data can be normalised to reshape the attribute space into the domain $[0, 1]$ on every axis. This can help create desired clusters/segmentation where the ranges of data values between axes vary significantly. After clustering, PCA is applied on each cluster to find a new basis which tighter encapsulates the features of the cluster. While this will likely be the desired configuration, we still provide the option to leave out the PCA transform.

We use consistent color-coding of clusters across views. For pleasant colour pallets and high distinguishability we use qualitative colour maps from *colorbrewer*. For the atlas we create a volume where each voxel stores the cluster ID of the corresponding tensor volume. This way standard raycasting can be used to render the atlas. In addition to this, a transfer function that assigns a colour from the

aforementioned palette to each cluster is automatically created to later render the volume.

As entrance into the filtering stage of the pipeline, we find the tensors in the original data set whose features are closest to the cluster centres and render them along with their shares of the attribute space (bar chart). If the user selects one of the representatives, filtering for that cluster may be performed. Simultaneously, the atlas rendering is updated accordingly. If no cluster is selected, all parts of the atlas will be shown. If the user clicks on a glyph, the corresponding region in the atlas will be rendered.

Then, for each dimension within the cluster we find the min and max feature values per axis which in turn forms the extents of the filter sliders for the representatives. By moving the sliders, the user implicitly controls the shape and cut-off of the filter which is later used as a cluster specific transfer function. For each positive and negative slider dimension, the user shapes a Gaussian kernel which has its maximum intensity of 1 at the mean point of the cluster and then falls off along the axes.

When the volume is rendered, each voxel and its corresponding feature are transformed into the filter space of its respective cluster and an alpha value can be computed by the euclidean distance to the centre. The alpha of the value is not only used for direct volume rendering, but also as a proportional seeding probability for the texture / directional lines. While the slider values define the cut-off extents of the filter, an additional parameter σ can be configured by the user to define the 'sharpness' of the Gaussian-filter in filter space such that with a given distance d alpha value $a = e^{-0.5d^2/\sigma^2}$. According to our preliminary tests, this seems to provide the user with enough control to shape the transfer function.

6 RESULTS

For the evaluation of our method we applied it to three different applications which have very different characteristics. The first application is the analysis of the human brain using diffusion tensor imaging (DTI). Thereby, we use two different data sets, one with a mask to exclude the noise outside of the head and one containing all noise, challenging the method. The second application is the simulation of a rotating neutron star which deforms the space in its vicinity. The first two applications provide positive definite tensor fields. In the following we will discuss these applications in more detail and describe how we derived our results.

6.1 Diffusion tensor of the human brain

One of the most prominent applications of tensor field visualisation is concerned with tensor fields from diffusion tensor imaging modalities. The goal of this exploration is to get an overview of the data. The first data set (Fig. 1 and 6), is a data set with a mask for the brain. This excludes the noise outside of the head from the visualisation. For the exploration we started with the usual attribute space used for such applications: the barycentric space spanned by the type of anisotropy values. Looking at the scatter plot in this space there are no dominant cluster and we start with our default setting: a k-means segmentation for 4 clusters and an atlas like representation rendering of the brain.

The second data set is a volume filled with tensors without a mask for the brain volume. For this data set we started with the same configuration as with the previous, however did not succeed to get any useful results with this attribute space. It turns out that the characteristics of the noise is uniformly spread over the entire space and it is not possible to filter it out.

As a second try we switched to a attribute spaced spanned by the three eigenvalues. The respective scatterplot with 4 clusters is shown in Fig. 8a. The cluster representatives using superquadrics are shown in the legend of Fig. 7. For this configuration, the noise can clearly be related to the largest cluster (green) and can thus be easily removed. Besides the representatives and the cluster size,

Fig. 7 shows two renderings, both using our default settings. The left image is a volume rendering of the cluster with a uniform fall-off radius for the transfer function for all clusters except for the noise cluster. In addition we added short tensorlines seeded in the red cluster. The right image is the atlas representation. For both volumes we clipped away the top part. In contrast to the barycentric attribute space, the linear behaviour could not be extracted easily.

The distribution of the largest cluster of this space is shown in Fig. 10. Using the superquadric glyphs to represent the variance of the cluster, it is clearly shown that the cluster contains linear, planar and isotropic tensors in an almost even distribution.

6.2 Simulation of a rotating neutron star

As for the diffusion data set, this is a positive definite tensor field. A first assumption might therefore be to use the barycentric attribute space. But looking at the distribution it becomes obvious this is not an optimal representation since the data set does not contain any planar features and all data points are aligned along one axis. Applying the default attribute space using the convectors, the 3D scatterplot shows that the tensors live in a two dimensional subspace, see Fig. 8b. The clustering results in a uniform partitioning of the space of clusters with increasing linearity. Using normalised superquadrics, the representatives can hardly be distinguished. One might think about other glyphs that are more appropriate for this data, see Fig. 9. The rendering of these clusters shows the structure of the data set nicely represented by shells of increasing distortion, Fig. 9 top row. By changing parameters and seeding within particular representative clusters, a number of different configurations for tensor lines and textures can easily be generated, Fig. 1 and 11.

6.3 Numerical simulation of a solid block with two applied forces

The last data set comes from a numerical material simulation of stresses inside of a block where two forces have been applied. For this data set, we started with the two dimensional shear-shape attribute space. These measures are related to the most commonly used failure criterion for isotropic material. The number of clusters has been set to 12 to increase to granularity of the sampling of the attribute space. We use the corresponding HWY glyph to convey the cluster representatives. In Fig. 12 we show one volume rendering of a combination of three clusters and the atlas representation for the two largest clusters.

In a second step we use glyph picking to select two representative of interest from one slice rendered using Reynolds glyphs. The selected glyphs are highlighted in the slice displayed in Fig. 13b. One of the selected glyphs relates to a state with a high shear stress close to the introduction of the load. The other glyph relates to an area with low shear stress but high pressure. We used these representatives to generate a feature level set of glyph number one. The corresponding surface is displayed in Fig. 13a. It encloses the most critical regions in the block. Since this attribute space does not distinguish between the mean of the eigenvalues, the surface extracts both critical regions in the area of compression and in the area of extension. The surface is embedded in a volume rendering using a uniform fall-off transfer function for both representatives. For glyph number two also a directional texture is displayed.

7 DISCUSSION

While the system provides a number of methods to explore and analyse, we have encountered several shortcomings and pitfalls during the design and implementation process. In the following, we will shortly touch on a few of them and discuss potential solutions.

There are common limitations of the plots used in this application. Star plots, for example, are rather limited in the number of dimensions they can accommodate. For higher order attribute spaces a rearrangement into parallel coordinate plots could prove beneficial.

Figure 6: Brain data set displaying four clusters in the barycentric attribute space.

Additionally, the star plot area suggests a linear behaviour between axes within the projected space which is most likely not true. One could conceptualise a method that will project the data points from attribute space into the projected space, thus showing the real shape of the cluster therein. Spring-mass systems, dimensionality reduction via PCA, or radial basis functions could be used to determine the projected positions. We could not agree on a method that would guarantee to convey a reasonable projection. We thus decided to use a filled region within the star plot as the implementation, comparison and evaluation of all aforementioned methods would exceed the scope of this paper.

Depending on the cluster distribution, tree maps may suffer from disproportionate cluster sizes, such that one or several clusters take up most of the visible plot space.

Finally, the semantic meaning of PCA axes poses an obstacle to interface design. As shown in Fig. 5, it could be sensible in some applications to label the axes with common domain specific variables such as shear, rotation and expansion/compression. While the axes in attribute space are clearly defined, the decomposition may yield axes that are very dissimilar to the original ones. In such situations, the automatic labelling of axes becomes virtually impossible.

Figure 7: Volume rendering and the segment atlas of the brain data set, sliced along the y axis. For texturing, tensor lines have been seeded in the red segment whose corresponding representative has also been highlighted.

Figure 8: (a) The attribute space used for the brain data set as rendered in Fig. 7. Here we have used eigenvalues as the attribute space as the noise, in this data set is characterised through negative and small eigenvalues, neatly fits into one of the representative clusters which can later be discarded when rendering the volume. (b) The attribute space of the star data set. In this case the three dimensional eigenvalues have been used.

Figure 9: Top row: Atlas segments of the star data set. Since the segments form concentric shells that occlude one another, we have placed them next to each other in order of appearance. Bottom row: legend of the glyph representative for the clusters together with their frequency.

8 CONCLUSION

In this work we have introduced a flexible and intuitive volume rendering concept for tensors fields. It facilitates domain specific attribute spaces and glyphs as interaction widgets. Applying the method to different data sets, we could confirm the usefulness of the tool and were able to generate a high number of expressive pictures. However, it also turned out that not for all applications there are domain specific glyphs available that can serve as a basis. For the example of the rotating star the glyph representatives that we used are not very expressive. While there were no distinct clusters present in the data-sets that we examined in this paper, clustering still provides the user with a sensible segmentation of the attribute space and

Figure 10: The filtering sliders for the first cluster representative with the corresponding colour in the brain data set, according to Fig 7.

Figure 11: Volume rendering of the star data set with different segments used for seeding the tensor lines which forms the texture.

thus supports the user in further exploration of the data. The major advantage of such an explorative method is to provide an overview of the data and an intuitive way to obtain flexible representations. This can however not replace application specific visualisation methods focusing on specific features as e.g. fibre tracking for DTI data. It might be interesting to explore further possibilities to combine the here proposed approach with such methods.

Figure 12: Two point load data set with two dominating features (purple and yellow) selected from the many clusters available. The cluster representatives are rendered as HWY glyphs. The size between the glyphs indicate the ratio between the maximum eigenvalues occurring in the respective clusters. The attribute space used for clustering was spanned by the eigenvalues.

(a) Two point load data set with two manually selected representatives (assigned colours are light green and purple, respectively) and a level-set rendered at a fixed distance (in attribute space) from the green cluster representative. Short tensor lines have been seeded throughout the volume to convey directional information of the stresses. The attribute space used was defined by shear stress, shape factor, and mean stress.

(b) The fifth slice (XY plane) of the two point load data set rendered using Reynolds glyphs. The two manually picked representatives are used to produce the rendering in Fig. 13a

Figure 13: Demonstration of how the linked views and manual selection of glyphs can be used to steer the TTF design and hence the volume rendering.

REFERENCES

- [1] Volume visualization based on statistical transfer-function spaces. In *IEEE Pacific Visualization Symposium (PacificVis 2010)*, pp. 17–24, 2010.
- [2] L. Cai, B. P. Nguyen, C.-K. Chui, and S.-H. Ong. A two-level clustering approach for multidimensional transfer function specification in volume visualization. *The Visual Computer*, 33:163–177, 2017.
- [3] L. Cammoun, C. Castaño-Moraga, E. Muñoz-Moreno, D. Sosa-Cabrera, B. Acar, M. Rodriguez-Flrido, A. Brun, H. Knutsson, and J. P. Thiran. A review of tensors and tensor signal processing. In *Tensors in Image Processing and Computer Vision*. Springer London, 2009.
- [4] C. Correa and K.-L. Ma. Size-based transfer functions: A new volume exploration technique. *IEEE Transactions on Visualization and Computer Graphics*, 14:1380–1387, 2008.
- [5] C. Dick, J. Georgii, R. Burkart, and R. Westermann. Stress tensor field visualization for implant planning in orthopedics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1399–1406, 2009.
- [6] P. Dobrev, T. V. Long, and L. Linsen. A cluster hierarchy-based volume rendering approach for interactive visual exploration of multi-variate volume data. In *VMV*, pp. 137–144, 2011.
- [7] H. Doleisch, H. Hauser, M. Gasser, and R. Kosara. Interactive focus+context analysis of large, time-dependent flow simulation data. *Simulation*, 82(12):851–865, 2006.
- [8] D. B. Ennis and G. Kindlmann. Orthogonal tensor invariants and the analysis of diffusion tensor magnetic resonance images. *Magnetic Resonance in Medicine*, 55(1):136–146, 2006.
- [9] M. Falk, I. Hotz, P. Ljung, D. Treanor, A. Ynnerman, and C. Lundström. Transfer function design toolbox for full-color volume datasets. In *Proc. IEEE PacificVis '17*, 2017.
- [10] L. Feng, I. Hotz, B. Hamann, and K. Joy. Anisotropic noise samples. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):342–354, 2008.
- [11] T. Isenberg. A survey of illustrative visualization techniques for diffusion-weighted mri tractography. In I. Hotz and T. Schultz, eds., *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*, Mathematics and Visualization, pp. 235–256. Springer, 2015.
- [12] H. Jänicke, M. Böttinger, and G. Scheuermann. Brushing of attribute clouds for the visualization of multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1459–1466, 2008.
- [13] J. Jankowai and I. Hotz. Feature Level-Sets: Generalizing Iso-surfaces to Multi-variate Data. *IEEE Transactions on Visualization and Computer Graphics*, 26(2):1308–1319, Feb 2020. doi: 10.1109/TVCG.2018.2867488
- [14] G. Kindlmann. Superquadric tensor glyphs. In *Proceedings of IEEE TVCG/EG Symposium on Visualization '04*, pp. 147–154, 2004.
- [15] G. Kindlmann, D. Weinstein, and D. Hart. Strategies for direct volume rendering of diffusion tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):124–138, 2000.
- [16] G. Kindlmann and C.-F. Westin. Diffusion tensor visualization with glyph packing. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1329–1336, 2006.
- [17] J. Kniss, G. Kindlmann, and C. Hansen. Multi-dimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.
- [18] A. Kratz, C. Auer, and I. Hotz. Tensor invariants and glyph design. In C.-F. Westin, A. Vilanova, and B. Burgeth, eds., *Visualization and Processing of Tensors and Higher Order Descriptors for Multi-Valued Data*, pp. 17–34. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
- [19] A. Kratz, C. Auer, M. Stommel, and I. Hotz. Visualization and Analysis of Second-Order Tensors: Moving Beyond the Symmetric Positive-Definite Case. *Computer Graphics Forum - State of the Art Reports*, 32(1):49–74, 2013.
- [20] A. Kratz, D. Baum, and I. Hotz. Anisotropic Sampling of Planar and Two-Manifold Domains for Texture Generation and Glyph Distribution. 19(11):1782–1794, 2013.
- [21] A. Kratz, B. Meyer, and I. Hotz. A Visual Approach to Analysis of Stress Tensor Fields. In H. Hagen, ed., *Scientific Visualization: Interactions, Features, Metaphors*, vol. 2 of *Dagstuhl Follow-Ups*, pp. 188–211. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2011.
- [22] R. D. Kriz, E. H. Glaessgen, and J. MacRae. Eigenvalue-eigenvector glyphs: Visualizing zeroth, second, fourth and higher order tensors in a continuum. *Workshop on Modeling the Development of Residual Stresses during Thermoset Composite Curing*, University of Illinois, 1995.
- [23] P. Ljung, J. Krüger, E. Gröller, M. Hadwiger, C. Hansen, and A. Ynnerman. State of the art in transfer functions for direct volume rendering. *Computer Graphics Forum*, 35(3):669–691, 2016.
- [24] J.-S. Prašni, T. Ropinski, and J. Mensmann. Shape-based transfer functions for volume visualization. In *Pacific Visualization Symposium (PacificVis)*, 2010 *IEEE*, 2010.
- [25] 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 Tensors and Higher Order Descriptors for Multi-Valued Data (Dagstuhl'14)*. Springer-Verlag, 2015.
- [26] T. Schultz and G. Kindlmann. Superquadric glyphs for symmetric second-order tensors. *IEEE Transactions on Visualization and Computer Graphics (to appear)*, 2010.
- [27] Y. Wang, J. Zhang, D. J. Lehmann, H. Theisel, and X. Chi. Automating transfer function design with valley cell-based clustering of 2d density plots. *Computer Graphics Forum*, 31(3), 2012.
- [28] C.-F. Westin, S. Peled, H. Gudbjartsson, R. Kikinis, and F. A. Jolesz. Geometrical diffusion measures for MRI from tensor basis analysis. In *Proceedings of the International Society for Magnetic Resonance Medicine (ISMRM)*, p. 1742, 1997.
- [29] E. Zhang, H. Yeh, Z. Lin, and R. S. Laramée. Asymmetric Tensor Analysis for Flow Visualizations. *IEEE Transaction on Visualization and Computer Graphics*, 15(1):106–122, 2009.
- [30] T. Zhang, Z. Yi, J. Zheng, D. C. Liu, W.-M. Pang, Q. Wang, and J. Qin. A clustering-based automatic transfer function design for volume visualization. *Mathematical Problems in Engineering*, p. 13, 2016.