Feature Level-Sets: Generalizing Iso-surfaces to Multi-variate Data

Jochen Jankowai and Ingrid Hotz

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-151231>

N.B.: When citing this work, cite the original publication.

Jankowai, J., Hotz, I., (2018), Feature Level-Sets: Generalizing Iso-surfaces to Multi-variate Data, IEEE Transactions on Visualization and Computer Graphics. <https://doi.org/10.1109/TVCG.2018.2867488>

Original publication available at:

<https://doi.org/10.1109/TVCG.2018.2867488>

Copyright:

Institute of Electrical and Electronics Engineers (IEEE)

<http://www.ieee.org/index.html>

©2018 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Feature Level-Sets: Generalizing Iso-Surfaces to Multi-Variate Data

Jochen Jankowai and Ingrid Hotz

Abstract—Iso-surfaces or level-sets provide an effective and frequently used means for feature visualization. However, they are restricted to simple features for uni-variate data. The approach does not scale when moving to multi-variate data or when considering more complex feature definitions. In this paper, we introduce the concept of *traits* and *feature level-sets*, which can be understood as a generalization of level-sets as it includes iso-surfaces, and fiber surfaces as special cases. The concept is applicable to a large class of traits defined as subsets in attribute space, which can be arbitrary combinations of points, lines, surfaces and volumes. It is implemented into a system that provides an interface to define traits in an interactive way and multiple rendering options. We demonstrate the effectiveness of the approach using multi-variate data sets of different nature, including vector and tensor data, from different application domains.

1 INTRODUCTION

Iso-surfaces or level-sets provide a powerful and intuitive way to visualize features in scalar fields. They are however restricted to very simple features, given through points in the one-dimensional attribute space. Carr et al. [1] recently introduced fiber surfaces to extend the concept of iso-surfaces to bi-variate fields, creating surfaces from polygons in the two-dimensional attribute space. When increasing the dimension of the attribute space further or when considering more complex features, there is no established theory or analogue to iso-surfaces. This lack of a multi-dimensional equivalent to iso-surfaces is a severe limitation, considering the fact that real-world data sets are rarely restricted to one or two scalar attributes. Among the applications involving multi-variate features, our work has largely been motivated by applications involving tensor fields and therein especially by two frequently encountered demands: There is at first the wish for ‘tensor level-sets’ defined as surfaces in a similar way as iso-surfaces. Secondly, there is the need for generic tools supporting failure analysis of mechanical components described by yield surfaces in tensor-invariant space. In both cases, the features are defined via an area in attribute space, in the first case as a point, and in the second case as a surface or volume.

In this paper, we introduce the concepts of *feature level-sets* and *traits*. Traits are defined as arbitrary geometries in attribute space while their equivalent counterparts in spatial domain are identified as features. Such traits can be of any dimension and structure which includes points, intervals, lines and volumes. The establishment of traits as new terminology serves as utility to formulate a framework wherein features in spatial domain and features in attribute are clearly defined and distinguished. This facilitates unambiguous formulation of feature-based algorithms as well as a defined mapping of features between spaces. Feature level-sets are then surfaces in spatial domain initialized by the distance field generated for the trait in attribute space,

providing a surface-based visualization for complex features in multi-variate data. Within this concept, the feature itself is represented as zero feature level-set, which in many cases might be empty. The classical iso-surface would be the zero feature level-set with respect to the trait defined by an iso-value, a point in one-dimensional attribute space.

Along with the technique, we present an expert tool providing an interface that allows for semantic querying of the data by defining traits in attribute space. This interactive visualization framework enables users to leverage their domain knowledge in the feature finding process and locate areas where a combination of critical parameters occur. For point traits we provide picking in spatial domain or attribute space. In the case of tensor fields, glyphs are used to give immediate feedback of the selected trait. Similarly, trait intervals or polygons can be defined. For higher dimensional attribute spaces, we support trait selection using parallel coordinates. Furthermore, explicit surface definitions or geometry can be imported. A first visual feedback is given using volumetric iso-surface rendering. Therefore, a default transfer function highlighting equidistant level sets is automatically adapted to the settings. The result can then be used to select levels of interest for explicit surface extraction.

We demonstrate the technique with three example data sets, each of which is of different nature. The first is a tensor data set containing a material simulation under stress. Here, we focus on the flexibility of interactive feature definition. The second data set is the flow around a cylinder (Kármán vortex street) which is a vector field, here considered as a multi-variate data set. Finally, the last data set is a multi-variate simulation of hurricane Isabel.

Our contributions include:

- Generalization of the concept of level-set from *uni-variate* to *multi-variate* data introducing the concept of feature level-sets.
- Interactive extraction of feature level-sets that can be seamlessly integrated into any existing visualization pipeline.

- Formal framework establishing the definition of traits, semantically separating them from features as they are often used interchangeably but do not reside in the same space.
- Domain- and data-related interface suggestions for interactive trait definition.

The first part of the paper discusses work related to the presented technique and puts it in relation to the presented method. Subsequently, feature level-sets and traits for multi-variate data are formally introduced in Sec. 3. Sec. 4 gives an overview over the method and the pipeline, followed by interface suggestions for some applications where intuitive interfaces can be defined and exposed straightforwardly in Sec. 5. The applicability and validity of the method is demonstrated by the means of three use cases in Sec. 6. Various aspects of the proposed method and possible future work are discussed in Sec. 7. Finally, conclusions are drawn in Sec. 8.

2 RELATED WORK

The method presented in this work falls into the domain of multi-variate visualization, and through the chosen use cases it is also strongly related to tensor visualization. Furthermore, there is previous work in the field of transfer function (TF) design and attribute space interaction which are relevant for our work.

MULTI-VARIATE VISUALIZATION – Multi-variate data has been listed among the major challenges in visualization during the Visualization seminar that took place in Dagstuhl, Germany in June 2011. The seminar led to a series of book chapters [2] summarizing challenges and the state of the art at that time. Two chapters also emphasize the need for feature-based visualization of multi-variate data [3], [4]. For an overview of early work on visualization of multi-variate data we refer to Fuchs et al. [5]. Since then, much has happened in advancing the visualization of multi-variate data, mostly in the context of attribute space exploration and filtering. This also includes various forms of dimension reduction techniques to reduce the dimensionality of the attribute space. Parallel coordinates and scatter plots are used to filter the data linked for explorative visualization [6], [7]. Buskin et al. [8] present an approach highlighting areas of similar characteristics respecting all attributes. The method is example-based, meaning that a user selects a point in the field to define a feature of interest. As such, it has some similarities with the trait definition that is proposed in this paper. Lu et al. [9] present a work-flow for interactive multi-variate transfer function design. The key idea is to allow the user to navigate through different subspaces and select appropriate subspaces for exploration. While exploration and data filtering are valuable means to get an overview over the data, in general no explicit feature definition is provided.

A method that focuses on an explicit feature extraction and which requires special attention in relation to our work is the extension of level-sets to bi-variate fields by Carr et al. [1]. They present an approach for the extraction of surfaces in bi-variate fields from sets of fibers, which are the bi-variate analogues of iso-lines. They allow the user to generate families of fiber-surfaces with respect to arbitrary

polygons in attribute space. Within our framework, these polygons would define the trait and the fiber-surface its zero level-set. To address the performance and accuracy limitations of the approach presented by Carr et al., Klačanský et al. [10] propose a parallelized implementation that leverages optimized data structures and tetrahedral meshes to generate exact fiber surfaces with a relative speedup of two orders of magnitude. Wu et al. [11] offer a method for direct volume rendering of fiber surfaces which provides per-pixel accurate rendering of fiber surface geometries while supporting the use of higher-order interpolation schemes. They demonstrate the performance advantage over geometry extraction which can be attributed to the small amount of pre-computational overhead required.

TENSOR VISUALIZATION – Even though tensors are more than multi-variate data, they share many properties and visualization challenges. A concise overview about tensor field visualization can be found in the work of [12], [13]. Here, we restrict the discussion to the multi-variate aspect. Most commonly used visualization approaches show only a subset of the tensor information. In many applications, there is a strong focus on the visualization of derived scalar measures on cutting planes or on surfaces, neglecting the directional information. A concept for volume rendering of diffusion tensor data considering more than one scalar tensor invariant has been introduced by Kindlmann et al. [14]. There are also some examples for visualization methods displaying the directional tensor properties. These are e.g. fiber tracts for diffusion tensors [15], and tensor lines for stress tensor fields [16]. Mostly related to our work are methods based on the exploration of the space spanned by tensor invariants [17], [18]. Kratz et al. [19] have used the invariant space as basis for interactive exploration of tensor data by facilitating concepts of brushing and linking. Here, volume rendering serves as context, glyphs and tensor-lines are used to add more detail.

TF DESIGN FOR HIGH-DIMENSIONAL DATA – Rendering of level-sets can also be interpreted as transfer function design. In this way, our feature level-sets are an example for multi-dimensional transfer functions. An overview on the current state of the art in transfer functions can be found in the survey by Ljung et al. [20]. Most of the existing multi-dimensional transfer functions have been designed for scalar fields with derived attributes, i.e., gradient magnitude or other statistical properties [21], [22], [23], [24], [25], [26]. They are therefore not directly applicable to general multi-variate data.

ATTRIBUTE SPACE INTERACTION – For complex data, interaction typically facilitates appropriate attribute spaces equipped with scatter plots using brushing and linking options for interaction. Feature space interaction is a common way to deal with complex data. Doleisch et al. [27] introduced an interactive visual analysis framework following the concepts of ‘multiple linked views’ and ‘focus and context’ visualization. Thus, it is possible to integrate multiple derived attributes in the analysis process. All views are linked in the sense that the features in focus are consistently emphasized in the visualization, whereas the rest of the data is shown as context.

3 THE CONCEPT OF FEATURE LEVEL-SETS

MULTI-VARIATE DATA AND ITS ATTRIBUTE SPACE – We consider general multi-variate data as a set of fields [28]

$$\{F_1, F_2, \dots, F_r\}, r \in \mathbb{N}, F_i : D_i \rightarrow R_i \quad (1)$$

where $D_i = D \subset \mathbb{R}^3$ for all i and R_i may be the sets of scalars, vectors or tensors. We assume that the fields are all continuous. Based on these fields, the *attribute space* can be assembled as a combination of the field values. Thereby, every scalar field contributes with one dimension, vectors with three dimensions and 2nd order tensors with six or nine dimensions, respectively. However, in many cases some of these attributes are not relevant for a feature definition or are missing. Therefore, we consider a more general *attribute space* $\mathcal{A}$ spanned by the field values of interest extended by derived quantities, such as the gradient of a scalar field. For the tensor case, the attributes of interest are often three scalar invariants, e.g., its eigenvalues, neglecting its directions. Using this attribute space we summarize multi-variate data as a mapping

$$f : D \rightarrow \mathcal{A} \subset \mathbb{R}^n \quad (2)$$

where n is the number of selected attributes plus the derived entities.

TRAITS AS GEOMETRIES IN ATTRIBUTE SPACE – In many applications, features of a field are defined via properties in attribute space. A common example are iso-surfaces, where the feature is defined via a point in the one-dimensional attribute space consisting of the scalar attribute. For vector field visualization, typical feature definitions rely on the Jacobian of the field and can be expressed as conditions in the attribute space containing the invariants of the Jacobian. For stress tensor fields, features are frequently defined via so-called yield-surfaces in the tensor invariant space which serves as attribute space in this context. Motivated through these observations, we define a *trait* T as a geometry or set of points in attribute space $T \subset \mathcal{A}$. As such, a trait can be a combination of points, lines, surfaces or higher dimensional entities in $\mathcal{A}$. We do not require that the parts of a trait are of the same dimension or connected to each other. We further define a *feature* as the pre-image of the trait in spatial domain $f^{-1}(T) = \{x \in D | f(x) \in T\}$. In the general case for an arbitrary trait in an attribute space of arbitrary dimension, the corresponding features will often be empty.

Using this terminology, iso-surfaces and fiber surfaces can be formulated as follows:

- Iso-surface: Let $f : D \rightarrow \mathbb{R}$ be a scalar function with the one-dimensional attribute space $\mathcal{A} = \mathbb{R}$. Defining the trait as an iso-value $c \in \mathcal{A}$ the respective feature will be the iso-surface $f^{-1}(c) = \{x \in D | f(x) = c\}$.
- Fibers and fiber surfaces: Let $f : D \rightarrow \mathbb{R}^2$ be a bi-variate field with the two-dimensional attribute space $\mathcal{A} = \mathbb{R}^2$. Defining the trait as a point $(c, d) \in \mathcal{A}$ the respective feature is the fiber $f^{-1}(c, d) = \{x \in D | f(x) = (c, d)\}$. In a similar way, a fiber surface of a polygon P can be described as feature of the one-dimensional trait $P \subset \mathcal{A}$.

FEATURE LEVEL-SETS AND THE TRAIT DISTANCE FIELD – To cope with the problem of empty features, we consider

not only the traits themselves but the distance to the traits in attribute space $\mathcal{A}$. In the case of features describing a critical or an optimal setting in attribute space, this distance can be interpreted as criticality receptively optimality. Therefore, we define the *trait distance field* which assigns a scalar distance or level to each point $a \in \mathcal{A}$

$$d_T : \mathcal{A} \rightarrow \mathbb{R}, d_T(a) = \min\{\|a - t\| \mid t \in T\} \quad (3)$$

where $\|\cdot\|$ represents the Euclidean norm in attribute space $\mathcal{A}$. The *feature distance field* $d_F = d_T \circ f$ is then defined as the pull-back of the trait distance field into the spatial domain, compare Fig. 2. We finally define *feature level-sets* f_c as level-sets of level c of the feature distance field

$$d_F^{-1}(c) = \{x \in D \mid d_F(x) = c\}. \quad (4)$$

The zero feature level-set would correspond to the feature in spatial domain matching the trait exactly. The smallest non-empty feature level-set highlights the points in the domain that are closest to the feature in attribute space.

4 METHOD

In the following subsections, we give an overview of our method for feature level-sets. The concept is based on three sequential steps: the trait specification, the feature level-set computation, and the rendering by various means in linked views (Fig. 1).

4.1 Trait specification

The first step is the selection of the attribute space by specifying a set of relevant scalar attributes. The selection of the attributes spanning this space can have a strong impact on the result since it also defines the metric that is later used for the distance field computation. As mentioned in the introduction, the proposed method is primarily targeted towards expert users seeking to extract the locations of certain critical parameter settings. Domain and data set specific knowledge is hence used to steer the trait design. Nonetheless, the method is not restricted to expert usage. In some cases, basic knowledge can be sufficient, such as the nature of the data. For example, if the user knows that the data contains weather information, they may intuitively query what they are interested in. In other cases, the user will be familiar with the domain of the data and therefore have an idea of what might be contained in the data and what features may be of interest. Possibilities for trait definition guidance systems are discussed in Sec. 7.

The first step in the trait specification process is to define sought-after behavior expressed as a subset in attribute space. This subset can be a single point, a line, a surface or any other body as long as its dimensionality is smaller or equal to the dimensionality of the attribute space. Often, there are several possibilities to model a trait depending on the chosen attribute space. A one-dimensional trait in three dimensional attribute space can be modeled as point in a two dimensional subspace. Our system provides an interface that supports several options to specify a trait, see Sec. 5.

Fig. 1. Pipeline for feature level-set extraction. Any multi-variate data serves as input. The user then defines the relevant attribute space, e.g. spanned by tensor invariants. Next, the feature or rather the feature-combination of interest is defined as a single or multiple points in attribute space, a line, a surface or any other body. Then, the distance field to the defined attribute is computed and subsequently probed at the corresponding locations of the data points. This results in a scalar volume where at each location the distance towards the selected feature is given. Finally, either the surface is extracted or the data rendered using direct volume rendering.

Fig. 2. The feature distance field is defined as the pull-back of the trait distance field to the domain D .

4.2 Feature level-set generation

Generating the feature level-sets for a chosen trait primarily consists of three steps:

First, the distance field is calculated according to the defined traits. For simple traits, e.g., a set of points, the distance field is computed directly as the minimum distance to the points. For the computation of the distance field in general dimensions we use an algorithm proposed by Saito et al. [29]. As distance metric we always use the Euclidean distance within attribute space. If the trait is a point picked from the data set, the zero-level-set cannot be empty, consisting of at least one point.

Second, the distance field is pulled back to the spatial domain. The pull-back is done by evaluating the distance field at the corresponding location of the data point in attribute space.

Third, either the surface is extracted using marching cubes (MC) or the data is rendered using direct volume rendering (DVR) as described in Sec. 4.3. This pipeline is an essential difference compared to the paper by Carr et al. [1] who incorporate an adjustment to MC to automatically extract a surface based on the selected 2D attribute space polygon. Our generalized approach using distance fields does not generate the necessity to alter MC, leaving

it directly applicable. This approach however can result in interpolation artifacts as discussed in Sec. 7.

4.3 Rendering

After the distance field has been computed, the resulting level-sets may be rendered by essentially two different means, namely as geometries with A-buffers, and DVR.

DIRECT VOLUME RENDERING – For a first visual inspection of data, DVR offers a high flexibility through interactive transfer function (TF) specification. This flexibility can be an advantage over explicit geometry rendering if the user wishes to interactively explore and filter the data. In our setting, DVR can be used to directly render iso-surfaces using an adapted raycasting technique and transfer function editor which has been used for all volume renderings of feature level-set figures throughout this paper.

GEOMETRY RENDERING – Leveraging the scalar nature of the distance field, explicit iso-surfaces can be extracted. Explicit surfaces segment the domain with respect to a specific trait and have the advantage that they can be further used for e.g., topological analysis. To get a sound rendering that shows multiple level-sets that properly nest within each other, the use of A-buffers is advised. With this technique, nested and intersecting geometries that are not fully opaque are blended correctly through per-pixel depth-based fragment sorting. We refer interested readers to the work of Maule et al. [30].

5 VISUALIZATION SYSTEM AND TRAIT DESIGN

The novel concept of feature level-sets has been integrated in a visualization system which allows an easy application in a multitude of settings. The major components of the system are described in the following subsections.

5.1 Trait design interfaces

A central part of the visualization system is the interface for the composition of traits. In our implementation it consists of partially application-specific, data-specific, or generally applicable parts from which to select. The final trait can be a combination of the individual parts from all interfaces.

Picking point-traits

IN SPATIAL DOMAIN – The first option to specify point-traits is picking one or more data points directly from the rendered data using any visualization at hand. To make the picking more intuitive, the system allows for interactive slicing through the data. In an add-in specifically designed for tensor data, we support picking of domain-specific glyphs rendered in the spatial domain. The chosen glyphs might not represent the complete tensor information, e.g., in Fig. 3a the glyphs are normalized to reduce clutter. When picking the glyph however, the complete tensor information at this position will be evaluated.

IN ATTRIBUTE SPACE – Alternatively, if the attribute space allows for a visual representation, usually meaning it has a dimensionality up to three, the user can pick features in attribute space. Therefore, we provide a scatter plot representation of the data set, see Fig. 3c.

Modeling polyhedral traits

We have implemented three different approaches to model geometric traits in attribute space: Connecting selected points in attribute space, using parallel coordinates for general multi-variate data and star plots for tensor fields.

PARALLEL COORDINATES (PC) FOR CUBOIDAL TRAITS – Every dimension of the attribute space is assigned to one parallel axis to initialize the PC. Either ranges or single points on the axis can be selected to construct a cuboidal trait, see Fig. 7b. The interface resembles PC-based filtering, the effect however is different. Even if the defined cuboid doesn't contain any data points, it is still a valid trait generating feature level-sets. Attribute axes that are irrelevant for the trait specification, i.e., axes where no selection is applied, are neglected for the distance field computation in the next step. Formally, this omission of axes in the trait specification can be considered as a projection to a lower dimensional attribute space yielding the same results.

STAR PLOTS FOR TENSOR-TRAITS – In the context of tensor fields, we chose to utilize a *star plot* which has the principal stresses on its axes, see Fig. 3b. Simply by dragging the points along the axes, the user can define the sought-after trait. The volume and/or geometry renderings are updated accordingly. The orientation of the axes in the tensor star plot is set up to easily distinguish between definite and indefinite tensors. If the triangle does not cover the center point, the eigenvalues have different signs and it is thus indefinite.

Implicit and explicit geometries

Geometries can either be defined as implicit function, such as many yield surfaces, or explicitly as geometries Fig. 3c.

5.2 Linked views

To support the exploration of the data set and get an understanding of the traits, the system integrates linked views of attribute and spatial domain. For attribute space dimensions up to three, it is possible to display the data set in attribute space, e.g., as point or glyph cloud, for higher dimensional attribute spaces the data is rendered into parallel coordinates. In our application, the user can pick a data item in this scatter plot and will get visual feedback about its location in data space and can thereby correlate between the two spaces, see Fig. 5. This interaction mode is especially helpful in combination with the exploration of surface traits, e.g., yield surfaces. The user can identify points that lie outside the geometry and locate these in spatial domain.

5.3 Trait normalization

When calculating distance fields in attribute spaces where the data values in the different dimensions have significantly different ranges, results get warped due to relative insignificance of small values. In order to compute meaningful distance fields we normalize the data in each dimension to the same range.

6 APPLICATIONS

To demonstrate the versatility and general purpose of the presented technique, we apply it to three data sets, each of which is of different nature. At first, a tensor data set containing a material simulation under stress is analyzed. Secondly, we utilize the method on a data set that describes the flow of a fluid around a cylinder (Kármán vortex street) which is a vector field. Third and last data set is the hurricane Isabel data set which by nature is multi-variate. In combination with each of these data sets we used different trait design interfaces. For those data sets that are directly setup as multi-variate data, we applied trait definition by parallel coordinate plot filtering. With a focus on tensor field visualization and exploration, we utilize several different approaches for trait design for the tensor field data set.

6.1 Numerical simulation of forces applied to a solid

The first data set is a result from a numerical material simulation of stresses inside a block where two forces have been applied. We will refer to the data set as 'two-point-load' in the following subsection. The stress tensor field is symmetric and as such has six independent degrees of freedom. The data is well-known and thus useful to achieve verifiable results. We use this data set to showcase the use of the concept of feature level-sets in all its variants in context with tensor field visualization. To design traits, we use different attribute spaces and different interfaces highlighting various aspects of the data. We subsequently use the following terminology. Eigenvalues and eigenvectors are referred to as principal stresses λ_i and principal stress direction e_i . They are ordered such that $\lambda_1 \geq \lambda_2 \geq \lambda_3$, named major, intermediate, and minor principal stress. One frequently used invariant is the maximum shear stress that is defined as $\lambda_1 - \lambda_3$.

Fig. 3. Different interfaces can be used to assemble a trait. (a) Here an interface for picking of glyphs rendered on a slice in spatial domain is shown. The glyphs are Reynolds glyphs colored by the sign of their principal stresses. (b) The selected tensor in two different representations, as glyph or as star plot which also can be used to modify the tensor. (c) The second option for picking is to select points from a scatterplot in attribute space, here the space spanned by the three principal stresses. As context also the yield surfaces using Tresca yield criterion for different material strength are displayed which can also be used as predefined traits.

Tensor Level-Sets – The first example is based on one of our main motivation for this work of creating level-sets for tensor fields using two different methods to specify the tensor of interest. First, we apply the picking interface in spatial domain. The data set is therefore rendered using domain specific glyphs. Interactive slicing through the data set makes a targeted selection possible, see Fig. 3a. Here we show Reynold glyphs which are frequently used for stress tensor fields [31]. Secondly, we design a tensor using the star plot where we can manipulate the principal stresses directly. In contrast to the tensor obtained from picking, the principal directions of the tensor are not specified. To define the trait based on the tensor of interest, one can chose between different attribute spaces. The full attribute space is six-dimensional and can be expressed using the six tensor components or any three scalar invariants, e.g., the principal stresses, and three directional attributes. In many examples however it is sufficient to consider lower dimensional attribute spaces.

The results for different configurations are shown in Fig. 8. The images in the first two columns display tensor level-sets for one trait, the first one defined through a picked glyph, the second one designed using the star plot. While both images 8d and 8g rely on the same tensor, different attribute spaces have been used. For image 8d, the attribute space is spanned by the three principal stresses. The image 8g below uses a reduced attribute space spanned by maximum shear stress and tensor trace. In this case, both locations of applied forces appear in a symmetric form, tension and expansion cannot be distinguished in this setting. For the third column of the figure the trait is composed of two glyphs. Both images 8f and 8i use maximum shear stress and trace as attributes, in image 8i the z -component of the direction of the major eigenvector is added as third dimension. The two chosen tensors represent stress states which result from each other by exchanging tension and compression. Therefore, the resulting feature level-sets are symmetric.

Manually Assembled Trait Using PC – For the second example we focus on directional aspects for the def-

inition of the trait. The attribute space is spanned by the major stress and the z -component of the major direction. Directional aspects of the stress tensor become especially interesting when drilling in the material is considered, analyzing the stability of the drilling boundary. The result shown in Fig. 4 shows that the critical region can be found in the part of the volume where the compressive force is applied.

Traits Defined by Yield Surfaces – In typical applications, stress tensor fields provide the basis for failure analysis of material parts. Depending on the material, the failure behavior is described by different failure models, which are mostly represented by yield surface in attribute space. In general a yield surface is defined in a space spanned by the six degrees of freedom of stress tensors.

Fig. 4. For these feature level-sets, the trait has been based on the directional aspects of the tensors. The attribute space is spanned by the major principal stress and the z -component of the major principal direction. This feature is expressed strongly in the part of the volume where the compressive force is applied.

Fig. 5. Two-point-load data set. (a) shows feature level-sets using Tresca yield surface as trait as shown in (b). Selecting a data point in attribute space highlights the corresponding region in the spatial domain as shown in (a).

The assumption is that the material with stresses lying inside the yield surface behaves elastically and becomes plastic outside of the surface. Thus, the region outside of the yield surface can be considered critical for the material. In this example we use a common criterion for isotropic material, the Tresca yield criterion. The Tresca yield surface is an irregular hexagon in the space spanned by the three principal stresses. The surfaces for different yield points (material property) are displayed in Fig. 5a. Using the yield-surface as trait results in the feature level-sets shown in Fig. 5b. They show that for the chosen yield point the load in the material is uncritical in most locations. This condition can also be verified when looking at the scatter plot embedded in attribute space (Fig. 5b), where only a few points lay outside the yield surface shown in red. For further exploration the system provides the possibility to select the points in the scatter plot which immediately highlight the respective region in spatial domain. As such the system allows inspection of the data set integrating exploratory and feature-based aspects.

6.2 Kármán vortex street

The data set results from a simulation of a three-dimensional flow around an obstacle, in this case a square cylinder, describing a repeating pattern of vortices, which are however hidden behind the dominating mean flow. The left boundary plane serves as inflow area and the right as the outflow area. The stream-wise direction is given by x -component, the vertical direction as z -component of the flow. Typically, the hidden vortices are extracted using various scalar vortex identifiers like λ_2 , acceleration or vorticity [32]. We do not try to reproduce these results but rather want to demonstrate the flexibility of the use of feature level-sets.

The interface to generate the feature level-sets is a parallel coordinate plot containing the flow components and some derived quantities like λ_2 and velocity, see Fig. 6. The interface always shows all attributes, while for the distance field computation only the actively used axes are considered.

In the first setting, shown in Fig. 6a, we have rendered iso-surfaces for λ_2 which serves mainly as reference to

the later examples. The iso-surfaces can be interpreted as features according to a trait defined by the point zero on λ_2 -axis. In the next two settings we have chosen the flow-components to define the traits. First, we define a trait defined by two intervals on the v_z and v_y axis containing high values. The result is shown in Fig. 6b. This trait generates features in similar locations to the λ_2 features highlighting the major vortices. In a second step we reduce the trait by setting additionally the v_x value to zero. This setting favors lower stream-wise velocities which reveals the secondary structures in the flow Fig. 6c.

Playing with different settings for the traits shows that the interesting structures of the Kármán vortex street can also be obtained by traits just based on the flow components which demonstrates that offering a system that supports playing with different traits can lead to discovery of known structures from a different perspective and the chance to find novel ones.

6.3 Hurricane Isabel

As a third data set we use a simulation of a hurricane from the National Center for Atmospheric Research provided for the IEEE visualization contest 2004. In September 2003, hurricane Isabel formed in the Atlantic ocean and moved towards land where it caused major damage from North Carolina up to Pennsylvania. The simulation data was later generated by the Weather Research and Forecast (WRF) model, courtesy of NCAR, and the U.S. National Science Foundation (NSF). The data consists of several time-varying scalar and vector variables and is as such a classical multi-variate data set. In our analysis we do not focus on the eye of the hurricane but on more specific critical weather conditions defined by several attributes. Our method enables the user to define one or more sets of critical weather conditions. For example, a combination of high wind velocity, low temperature, and high precipitation. In our example the corresponding trait has been designed using the parallel coordinate interfaces shown in Fig. 7b. The attributes used to define the trait are high wind speed and high precipitation. Considering the entire attribute space, the trait can be considered as a hyper prism with a ground shape of a

Fig. 6. Kármán vortex street. (a) Typical visualization using level set for λ_2 . (b) Feature level sets for a trait defined by two intervals at high values for the v_y and v_z component of the flow also highlight the vortex regions. (c) Adding to the trait in (b) v_x as additional dimension emphasizing low stream-wise velocity by setting $v_x = 0$ the secondary structures of the flow become visible. Red, green, and blue arrows correspond to x , y , and z direction respectively.

cube defined by the two selected attribute ranges extending into all other directions of the attribute space. The resulting feature level-sets can be seen in Fig. 7a. The surfaces show how close a region comes to the specified trait. As such, the level-sets do not only highlight the regions that fulfill these criteria but also summarize the severity of the weather conditions with respect to storm-like behavior defined by the trait. In this example, the highest values are reached in a certain distance from the center of the hurricane eye highlighted by the red surface. Though the highlighting of the hurricane eye was not the goal of the trait selection the symmetry of the feature around it also highlights the eye itself. Note that this result cannot be achieved with simple filtering on the parallel axis.

7 DISCUSSION AND FUTURE WORK

The presented concept of feature level-sets is directly applicable to a large class of data. The design of traits in attribute space offers an interface for the creation a large variety of features for multi-variate data in an intuitive way with a multitude of application areas. Thus, feature level-sets provide a way to overcome the limitations of iso-surfaces or the more general fiber surfaces that arise when increasing the dimensionality of the attribute space. While the major focus of the paper is the introduction of the novel concept, also an implementation of the concept in a workflow and a visualization system is suggested. The proposed implementation closely adheres to the presented concept without special care for optimization. This leaves

Fig. 7. (a) Feature level-sets: The color of the surfaces represents the severity of the weather conditions with respect to the trait defined in (b). The red surface highlights the area that comes closed to this trait. Observe that the weather conditions are less severe in the eye of the hurricane shown by a yellow feature level-set. Note that the image has been turned upside down to highlight the conditions close to the ground. (b) Trait design: The trait, represented by the yellow polygon, is defined using parallel coordinates. It is defined by high wind speed and high precipitation. We want to point out that the result shown in (a) cannot be achieved by filtering based on parallel coordinates. Note that the lines in the plot only represent 5% of the data and do not well represent the data in the hurricane eye.

space for extensions and possible improvements which will be discussed in the following. The main areas where the authors see potential and necessity for discussion are the choice of the distance metric in attribute space, the discernibility of equal-value data points in the final rendering, performance, interpolation artifacts, and advanced guidance for trait design. Each of these aspects will be discussed below with an outlook on possible future work.

DISTANCE METRIC – The choice of the distance metric in attribute space has a substantial impact on the result. Principally, the proposed technique allows to interchange the distance metric to whichever is necessary. The Euclidean distance metric, treating all attributes in the same way, has been used throughout this application to generate the distance field from which the level-sets are computed. This metric has the advantage of being intuitive as it corresponds to our natural understanding of distance, is trivially transferred into higher-dimensional spaces, and can be computed efficiently. However, the metric has the disadvantage that it

treats all attributes in the same way regardless the different relevance and units of the attributes. A weighted Euclidean distance metric where certain dimensions are considered more important and are therefore assigned a higher significance would solve many of these issues with the cost of an additional parameter that has to be set by the user. The normalization of the attribute values might give similar results in some cases. Depending on the application, the need for a more specific distance measure, possibly even asymmetric, may arise. An alternative to changing the metric achieving similar results would be a reparameterization of the attribute space.

DISCERNIBILITY – Inherent to the technique, points with equal distance to the trait cannot be distinguished in the final level-set rendering. In cases where the user has performed his initial query about the data set, e.g., with respect to some critical behavior, he or she may eventually need to further investigate what attribute exactly is responsible for the criticality. In such a case, the feature level-sets may

be computed as proposed but locally coloured or textured according to individual attributes. In the current implementation, color is used to encode the level of the feature level-sets which is the distance to the trait. When using color for other purposes this information has to be encoded separately somewhere else which is especially important when rendering multiple nested level-sets. Other options may be to augment the level-sets with local feature information in the field, such as glyphs.

PERFORMANCE – The core technical contribution of the proposed technique is the leveraging of distance field evaluation for dimensionality reduction and is therefore considered the main aspect when evaluating the performance of the system. With a non-optimized implementation of the technique employing only basic parallelization techniques, the distance field computation for the Isabel data set (500x500x100 voxels) and 4 attribute space dimensions takes about 1.5s on parameter change. However, the computational cost increases with the dimensionality of the attribute space and the complexity of the traits. Especially for higher dimensions, the computation of the entire distance field in attribute space might not be the optimal solution. The authors are confident that further investigations with regard to cache coherency, task parallelization, and acceleration structures would yield a system that improves interactivity.

INTERPOLATION – Two main aspects need to be considered regarding interpolation differences and artifacts. Firstly, as discussed by Wu et al. [11], the per-pixel evaluation of raycasting will yield results that differ from the surfaces extracted by marching cubes. We refer to [11] for an in-depth discussion. Secondly, special care is required regarding tensor interpolation. While in many cases a linear interpolation of attributes between data points is reasonable, e.g., for scalar fields, this is frequently not the case for tensor fields. Tensor invariants, often used as attributes, generally do not depend on the tensor components linearly. As a consequence, the behavior of the attributes may not even be convex between two sample points [33]. For example, in discrete tensor fields so-called degenerate points, where the tensor exhibits a maximum isotropy, usually do not occur on sample points but within cells. And while the trait distance field is only sampled at positions directly derived from sample points to generate the feature distance field, linear interpolation is performed on the GPU when using DVR as rendering method. This linear interpolation can lead to artifacts. We leave the issues related to interpolation and sampling artifacts and the improvement of the accuracy for the surface extraction to future work.

TRAIT DEFINITION – As briefly mentioned in Sec. 5, we have implemented feature level-sets in a framework targeted towards expert users. For a more exploratory application of the proposed method, guidance methods may be employed to support the user to find features. Additionally, further application-driven trait definition interfaces can improve user experience.

8 CONCLUSION

In this paper we have introduced traits and feature level-sets for multi-variate data as an extension to classical iso-values and iso-surfaces for univariate data. With the concept

of traits as geometries in attribute space a large class of typical features can be modeled including iso-surfaces as simple example but also more complex features that are characterized by multiple parameters. The concept is easy to integrate in common visualization systems employing typical rendering pipelines. In our case, the method has been integrated into a system that supports brushing and linking and different rendering styles. We link feature-based visualization and exploratory data analysis to be used in a complimentary way including an interface for trait and feature design which can be updated on demand.

A specific field that will benefit from the notion of feature level-sets is the area of tensor field visualization where the method opens a multitude of novel visualization and analysis options. Especially different failure models for failure analysis for different materials are directly integrated as traits into the system. As such it is easy to compare critical regions applying different models or materials.

ACKNOWLEDGMENTS

The authors would like to thank the ELLIIT program for financing the research. The application was realized in the open-source software Inviwo, financed by the Swedish e-Science Research Centre (SeRC). The authors would also like to thank Bill Kuo, Wei Wang, Cindy Bruyere, Tim Scheitlin, and Don Middleton of the U.S. National Center for Atmospheric Research (NCAR), and the U.S. National Science Foundation (NSF) for providing the Weather Research and Forecasting (WRF) Model simulation data of Hurricane Isabel. The tensor data was provided by tensorvis.org.

Finally, the authors would like to express their gratitude to Wito Engelke, Sathish Kottraval, and Robin Skånberg for inspiring discussions.

AUTHORS

specifically on tensor field exploration, and feature space exploration and analysis.

Jochen Jankowai received his Masters degree from Linköping University in 2016 and is now a PhD student in the Scientific Visualization group at Linköping University. His research interests are application-driven exploration of high-dimensional, high-order data. Therefore, his research focuses

Ingrid Hotz is a Professor in Scientific Visualization at the Linköping University in Sweden and has an affiliation with the Center for Medical Image Science and Visualization (CMIV) in Linköping. She received her PhD degree from the Computer Science Department at the University of Kaiserslautern, Germany. Her research interests lie in data analysis and scientific visualization, ranging from basic research questions to effective solutions to visualization problems in applications. This includes developing and applying concepts originating from different areas of computer sciences and mathematics, such as computer graphics, computer vision, dynamical systems, computational geometry, and combinatorial topology.

Fig. 8. Stress tensor level-sets for a material block with two applied forces. Column-wise: Left: One glyph picked in spatial domain. Middle: One tensor designed using the eigenvalue starplot. Right: Two glyphs picked from the spatial domain. The color of the glyph always refers to the sign of the eigenvalues: green for positive and red for negative values. Row-wise: Top row shows the feature selection/design interface with domain-relevant glyphs to convey feature characteristics. Second row shows uses a point-trait in the principal-stress space spanned by the three eigenvalues. Last row, (g) and (h) using point traits defined in anisotropy-trace space and (i) the direction of the largest eigenvalue is added as additional dimension in attribute space.

REFERENCES

- [1] H. Carr, Z. Geng, J. Tierny, A. Chattopadhyay, and A. Knoll, "Fiber Surfaces: Generalizing Isosurfaces to Bivariate Data," *Computer Graphics Forum*, vol. 34, no. 3, pp. 241–250, 2015.
- [2] C. D. Hansen, M. Chen, C. R. Johnson, A. E. Kaufman, and H. Hagen, *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*, ser. Mathematics and Visualization. Springer, 01 2014.
- [3] H. Obermaier and R. Peikert, "Feature-Based Visualization of Multifields," in *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*, ser. Mathematics and Visualization. Springer, 2014.
- [4] H. Carr, "Feature Analysis in Multifields," in *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*, ser. Mathematics and Visualization. Springer, 2014.
- [5] R. Fuchs and H. Hauser, "Visualization of Multi-Variate Scientific Data," *Computer Graphics Forum*, vol. 28, no. 6, pp. 1670–1690, 2009.
- [6] J. Blaas, C. P. Botha, and F. H. Post, "Interactive visualization of multi-field medical data using linked physical and feature-space views," in *Proceedings of Eurographics - IEEE VGTC Symposium on Visualization '07*, 2007, pp. 123–130.
- [7] J. Blaas, C. Botha, and F. Post, "Extensions of Parallel Coordinates for Interactive Exploration of Large Multi-Timepoint Data Sets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1436–1451, 2008.
- [8] S. Busking, C. Botha, and F. H. Post, "Example-based interactive illustration of multi-field datasets," *Computers & Graphics*, vol. 34, pp. 719–728, 2010.
- [9] K. Lu and H.-W. Shen, "Multivariate Volumetric Data Analysis and Visualization through Bottom-Up Subspace Exploration," in *IEEE Pacific Visualization (PacificVis)*, 2017.
- [10] P. Klacansky, J. Tierny, H. Carr, and Z. Geng, "Fast and exact fiber surfaces for tetrahedral meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1782–1795, July 2017.
- [11] K. Wu, A. Knoll, B. J. Isaac, H. Carr, and V. Pascucci, "Direct multifield volume ray casting of fiber surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 941–949, Jan 2017.
- [12] 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*, vol. 32, no. 1, pp. 49–74, 2013.
- [13] L. Cammoun, C. Castaño-Moraga, E. Muñoz-Moreno, D. Sosa-Cabrera, B. Acar, M. Rodriguez-Florido, A. Brun, H. Knutsson, and J. P. Thiran, "A Review of Tensors and Tensor Signal Processing," in *Tensors in Image Processing and Computer Vision*. Springer, 2009.
- [14] G. Kindlmann, D. Weinstein, and D. Hart, "Strategies for Direct Volume Rendering of Diffusion Tensor Fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 2, pp. 124–138, 2000.
- [15] T. Isenberg, "A Survey of Illustrative Visualization Techniques for Diffusion-Weighted MRI Tractography," in *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*, ser. Mathematics and Visualization. Springer, 2015, pp. 235–256.
- [16] 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*. Springer, 2015.
- [17] 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, pp. 2445–2465, 2000.
- [18] D. B. Ennis and G. Kindlmann, "Orthogonal Tensor Invariants and the Analysis of Diffusion Tensor Magnetic Resonance Images," *Magnetic Resonance in Medicine*, vol. 55, no. 1, pp. 136–146, 2006.
- [19] A. Kratz, B. Meyer, and I. Hotz, "A Visual Approach to Analysis of Stress Tensor Fields," in *Scientific Visualization: Interactions, Features, Metaphors*, ser. Dagstuhl Follow-Ups, Dagstuhl, Germany, 2011, vol. 2, pp. 188–211.
- [20] P. Ljung, J. Krüger, E. Gröller, M. Hadwiger, C. D. Hansen, and A. Ynnerman, "State of the Art in Transfer Functions for Direct Volume Rendering," in *Computer Graphics Forum - STAR*, vol. 35, no. 3, 2016, p. 23.
- [21] 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.
- [22] 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," *CComputer Graphics Forum*, vol. 31, no. 3, 2012.
- [23] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M. E. Gröller, "Volume visualization based on statistical transfer-function spaces," in *IEEE Pacific Visualization (PacificVis)*, 2010, pp. 17–24.
- [24] 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*, vol. 33, pp. 163–177, 2017.
- [25] C. Correa and K.-L. Ma, "Size-based Transfer Functions: A New Volume Exploration Technique," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, pp. 1380–1387, 2008.
- [26] J.-S. Pražni, T. Ropinski, and J. Mensmann, "Shape-based transfer functions for volume visualization," in *IEEE Pacific Visualization (PacificVis)*, 2010.
- [27] H. Doleisch, H. Hauser, M. Gasser, and R. Kosara, "Interactive Focus+Context Analysis of Large, Time-Dependent Flow Simulation Data," *Simulation*, vol. 82, no. 12, pp. 851–865, 2006.
- [28] I. Hotz and R. Peikert, "Definition of a Multifield," in *Scientific Visualization - Uncertainty, Multifield, Biomedical, and Scalable Visualization*. Springer, 2014.
- [29] T. Saito and J.-I. Toriwaki, "New Algorithms for Euclidean Distance Transformation of an n-dimensional Digitized Picture with Applications," *Pattern Recognition*, vol. 27, no. 11, pp. 1551–1565, 1994.
- [30] M. Maule, J. L. D. Comba, R. Torchelsen, and R. Bastos, "Memory-Efficient Order-Independent Transparency with Dynamic Fragment Buffer," in *SIBGRAPI Conference on Graphics, Patterns and Images*, 2012, pp. 134–141.
- [31] 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*, ser. Mathematics and Visualization. Springer, 2014, pp. 17–33.
- [32] J. Kasten, I. Hotz, B. Noack, and H.-C. Hege, "On the Extraction of Long-living Features in Unsteady Fluid Flows," in *Topological Methods in Data Analysis and Visualization. Theory, Algorithms, and Applications.*, ser. Mathematics and Visualization. Springer, 2011.
- [33] I. Hotz, J. Sreevalsan-Nair, H. Hagen, and B. Hamann, "Tensor Field Reconstruction Based on Eigenvector and Eigenvalue Interpolation," in *Scientific Visualization: Advanced Concepts*, ser. Dagstuhl Follow-Ups, vol. 1, 2010, pp. 110–123.