

Surface-Based Structure Analysis and Visualization for Multifield Time-Varying Datasets

Samer S. Barakat, *Student Member, IEEE*, Markus Rütten, and Xavier Tricoche, *Member, IEEE*

Abstract—This paper introduces a new feature analysis and visualization method for multifield datasets. Our approach applies a surface-centric model to characterize salient features and form an effective, schematic representation of the data. We propose a simple, geometrically motivated, multifield feature definition. This definition relies on an iterative algorithm that applies existing theory of skeleton derivation to fuse the structures from the constitutive fields into a coherent data description, while addressing noise and spurious details. This paper also presents a new method for non-rigid surface registration between the surfaces of consecutive time steps. This matching is used in conjunction with clustering to discover the interaction patterns between the different fields and their evolution over time. We document the unified visual analysis achieved by our method in the context of several multifield problems from large-scale time-varying simulations.

Index Terms—Multifield, time-varying, surface structures.

1 INTRODUCTION

Scientific simulations and experiments are producing increasingly large datasets comprised of multiple scalar, vector, and tensor fields tracked over time. To better understand the output of these simulations and advance scientific knowledge, domain experts need effective means to visually analyze these so-called *multifield* datasets. However, multifield visualization is made fundamentally challenging by the high dimensionality of the data: displaying multiple fields simultaneously either leads to visual clutter or produces a cognitive overload. Defining features of interest in this context on the other hand remains an open problem and no universally applicable model has been proposed to date. Yet, a transformation is required to reduce the complexity of the data and facilitate the visual assessment of its most significant properties. Furthermore, the relationship between the constitutive fields should remain visible to help elucidate the interplay of relevant physical attributes. Time-varying datasets require in addition to characterize the dynamics of the considered phenomenon. Studying the quantitative and qualitative evolution of the data over time requires to compute and track features efficiently.

In contrast to recent methods based on statistical correlation analysis or information theoretic concepts applied to voxels, we propose a geometrically motivated approach for the extraction, analysis, and visualization of 3D multifield datasets that focuses on the characterization of *distinguished surfaces* in the spatial domain of the problem. Specifically, the contribution of this paper is threefold. First, we propose an iterative simplification algorithm suitable for multifield problems that leverages the existing theory on skeleton derivation to extract and merge structures from individual fields. Our approach can be applied to any feature definition that results in a geometrical surface representation. It fuses structures from the different fields to eliminate redundancy and allows for a simplified representation of the multifield domain. The extracted features carry information about the individual fields in a compact form that subsequently permits efficient visualizations and analysis. Second, we devise a new method for non-rigid surface registration specifically tailored to the here relevant case of limited and spatially smoothly varying transformations of the original mesh. This matching is applied to capture structure evolution and associated variable changes over time. Finally, we present a practical

analysis technique that exploits the proposed multifield structure extraction and can be used to shed light on the interactions of multiple variables. In addition, our visualization solution for multifield datasets provides tools through which the user can test a hypothesis and view its association to structures in the volume.

2 PREVIOUS WORK

A comprehensive review of the multifield visualization literature is clearly beyond the scope of this paper. Instead we concentrate in this section on prior work that is most directly relevant to the present work. Specifically, we focus on methods that aim to address the visual complexity of multifield datasets, characterize their most interesting features, and give insight into the correlation between individual physical quantities.

To tackle the issue of visual complexity, brushing in the data space for interactive exploration of multifield data was introduced [22]. Doleisch and Hauser [5] combine brushing with a transfer function to show interesting flow features. Jänicke *et al.* [14] propose a transformation from the high-dimensional data space to a 2D space that preserves proximity. Brushing on the resulting point cloud is used to highlight interesting structures. In general, brushing allows one to selectively visualize different portions of the data space. However, it cannot show an overview of all the structures in the domain simultaneously and suffers from clutter and occlusion. In addition, brushing implicitly assumes that structures form connected regions in the data space rather than connected regions in the 3D space.

Correlation analysis was used to understand and display the relations between fields. Kniss *et al.* [18] present a method to combine multiple fields through compositing. Sauber *et al.* [26] propose a gradient similarity measure and a local correlation coefficient to visualize relationships in 3D multifields. Gosink *et al.* [9] define a correlation field as the normalized dot product between two gradient fields from two variables. The derived field was used to study variable interactions with a third variable. Qu *et al.* [25] introduce the standard correlation coefficient for calculating the correlation strengths between different data attributes in weather data analysis. They present a weighted complete graph where nodes corresponds to data attributes and weights encode the strength of correlation. While these methods give insight into the *correlation* between different variables, the visualization of the variables themselves remains an open question.

A standard approach to reduce the complexity of large datasets is to focus on their most remarkable features. While a significant literature has been dedicated to the topic in the context of individual fields, only few techniques exist that are suitable for multifield datasets. Jänicke *et al.* [15, 11] introduce methods for multifield data reduction based on statistical complexity. A visualization method based on block-wise

- S. Barakat and X. Tricoche are with Computer Science Department, Purdue University, E-mail: sbarakat@purdue.edu, xmt@purdue.edu.
- M. Rütten is with DLR Göttingen, E-mail: markus.ruetten@dlr.de.

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

importance analysis of the data in the joint feature-temporal space was discussed by Chaoli *et al.* [34]. Analyzing causal relations through information transfer is also discussed by Chaoli *et al.* [33]. However, the visualization of a large number of 3D fields remains a challenge in terms of both visual perception and computational cost.

Finally, query-driven visualization techniques were first discussed by Stockinger *et al.* [31] to reduce the computational complexity of visualization. A following work uses correlation fields to explore variable interactions within the domain space of query-regions [10]. However, structures of interest may require more complex definition than range queries.

3 FEATURES

In this paper, we propose a method to visualize 3D multifield volumes. Our proposed method aims at reducing the data complexity by focusing on finding the most relevant structures. In this paper we define features as geometrical surfaces that captures interesting boundaries, edges, or skeletons in the data. The basic idea behind our approach is to extract such surfaces from individual fields before simplifying and combining these structures into global multifield surface features.

3.1 The Challenge of Defining Features in Multifield

Visualizing features discovered for different variables in separate screen windows is very challenging and might require significant cognitive effort from the viewer to match these different structures across the fields displayed. In fact, the problem becomes intractable when the number of fields exceeds two or three. Additionally, the features associated with different fields will usually not be perfectly aligned and their spatial correspondence can be very ambiguous, as shown in figure 1. On the other hand, viewing all features in a single view might be very ineffective due to occlusion and clutter. For a large number of fields, we apply data reduction in order to provide an informative visualization. In order to identify features, our strategy consists in characterizing the skeleton of each multifield structure and use this information for the elimination of redundancies and the pruning of spurious structure. A spurious structure in turn is a surface patch that is inconsistent with the overall skeleton of the surface (*e.g.* branch). Therefore, we introduce an algorithm that will iteratively filter out small distortions, merge the individual features based on the spatial correlation of their geometrical signature, and connect pieces with close matching surfaces. The resulting surfaces form what we call *multifield features*.

The methods described in this paper can accommodate any surface-based feature definition applicable to the individual or combined fields. However, we have chosen to use crease surfaces in our implementation because of their compelling results in various applications in recent years. Creases are discussed in previous work [7, 19], and proved to be useful for the analysis and visual representation of scalar, vector, and tensor fields in medical and engineering applications [17, 32, 24, 28, 16]. Creases generalize the notion of point-wise extrema (where the gradient vanishes) of smooth scalar fields to objects of higher dimensions (*e.g.*, curves and surfaces). The *height ridge* definition proposed by Eberly [6] defines the local coordinate frame in terms of the eigenvectors of the Hessian matrix $\mathbf{H} = \nabla^2 f$ (second-order derivative of f) associated with the p smallest eigenvalues $\lambda_1 < \dots < \lambda_p$ with the additional requirement that these eigenvalues be negative. *Valleys* of f are similarly defined as ridges of $-f$. Spurious structures can be filtered out based on the value f and the *crease strength* $|\lambda_p|$ defined by the user. Several methods exist for the extraction of these surfaces [28, 2].

3.2 Proposed Multifield Feature Definition and Extraction

For two different feature surfaces S_A and S_B extracted from two fields f_A and f_B , the two surface patches \mathcal{M}_A and \mathcal{M}_B , where $\mathcal{M}_A \subset S_A$ and $\mathcal{M}_B \subset S_B$, are considered to match if $d_H(\mathcal{M}_A, \mathcal{M}_B) < \alpha$ where d_H is the Hausdorff distance. The distance limit α is a user-defined parameter to indicate spatial dependence or correlation between the two structures based on proximity. In other words, if two pieces from different fields surfaces are sufficiently close, we consider both to characterize the same feature. This parameter is also used to fill holes on feature

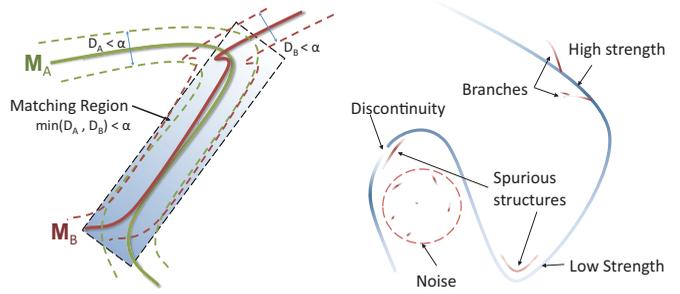


Fig. 1. Multifield feature definition. *Left*: Structures discovered in three different fields. *Right*: Example of a true structure shown in blue surrounded by insignificant structures shown in red.

surfaces and to prune small branches as we will discuss later in this section.

If $D_A(x)$ and $D_B(x)$ are the Euclidean distance transforms [4] for the patches \mathcal{M}_A and \mathcal{M}_B respectively, then we expect that:

$$D_A(x) < \alpha, \forall x \in \mathcal{M}_B$$

$$D_B(x) < \alpha, \forall x \in \mathcal{M}_A$$

Hence, the matching region is characterized by $\max(D_A(x), D_B(x)) < \alpha$. This latter observation has been used in the context of surface matching [21]. In our algorithm, we take it into account by using the distance field computation in order to test matching portions of structures. The distance field transform of all feature surfaces is used to create a scalar field \mathcal{F} . Our goal is to construct this scalar field \mathcal{F} such that it embeds all the features' shapes without noise or repetition. Hence, it is constructed as the shortest distance to any surface in the space $\mathcal{F}(x) = \min_{I \in M} (D_I(x))$ where M is the set of all surfaces from the individual fields. The skeleton of the regions in \mathcal{F} where the value is less than α represent our merged multifield surfaces. This skeleton connects pieces of the surfaces that are less than α apart and has a smoothing effect on their geometry. The topological skeleton of a volumetric shape can be defined as the ridges of its exterior distance field [13]. Therefore, the skeleton of the region where $\mathcal{F}(x) < \alpha$ is found by extracting the ridges of the distance field of the region $\mathcal{F}(x) > \alpha$. Notice that both cracks and noise in the range α around the original structures will be smoothed out and will not appear in the final skeleton. These steps are demonstrated in figure 2. This computation of the merged features is much more efficient computationally than to carry a combinatorial binary match search between all fields, especially for large datasets.

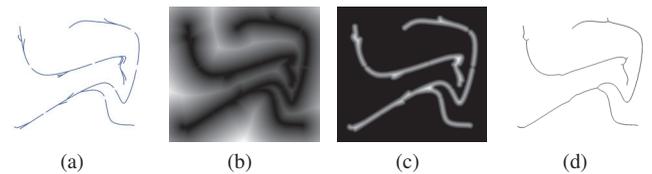


Fig. 2. (a) Original structures, (b) Distance field of the structures $\mathcal{F}(x)$, (c) Distance field for $\mathcal{F}(x) > \alpha$, (d) Ridges extracted from c.

For many applications, we have noticed that the field \mathcal{F} might become too dense as a result of adding surfaces from a number of individual fields. This high density leads to a skeleton that conveys information about a bounding region of size α around the forming structures rather than the individual feature shapes as shown in the top portion of figure 4. In order to overcome this problem and achieve a more informative visualization, the field \mathcal{F} is updated using the distance field of one connected component at a time. Every connected component is found such that it does not contain portions in the range where $\mathcal{F} < \alpha$. This constraint guarantees that the spatial density of

structures is limited and prevents the repetition of matching structures. It also implies that the order of the components considered must reflect the importance of these components, such that those with higher importance are considered first. We consider connected structures with higher ridge strength to be more relevant to the visualization, whereby the ridge strength of a component is defined as the average strength on that component. In order to use the ridge strength for ordering, there is a need to normalize the values across all structures discovered in the different fields. This is achieved by applying histogram equalization for each field followed by scaling the values to have an identical range. As components of higher strength values are inserted in the field \mathcal{F} , other components in their proximity are broken into smaller pieces since matched regions where $\mathcal{F} < \alpha$ are discarded. The user can filter small and low strength components using thresholding. The steps of this hierarchical filtering approach are summarized in Algorithm 1 and illustrated for a simple example in figure 3. In the bottom part of figure 4, we show the result obtained for an example dataset consisting of ten scalar fields. The user can also experiment with the individual fields and specify different ridge strength limits for each during the extraction of the surfaces or during the fusion of these surfaces.

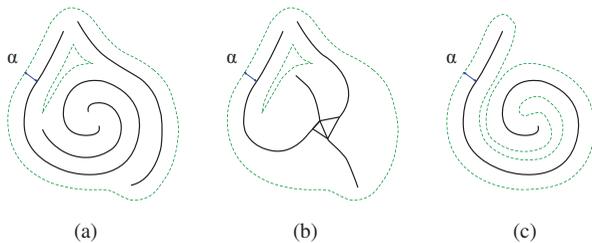


Fig. 3. (a) Original structures containing three components, (b) The skeleton of the region $\mathcal{F}(x) < \alpha$ when all components are added to $\mathcal{F}(x)$ at once, and (c) The skeleton of the region $\mathcal{F}(x) < \alpha$ when incremental addition of components is applied. Two of the components were excluded in the simplification because they are within an α distance from the first added component.

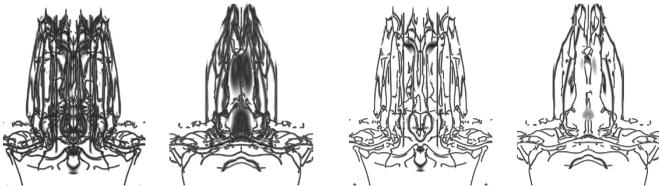


Fig. 4. Comparison between the global minimum field (left) and the field formed using hierarchical filtering (right) for two time steps.

One complication of this hierarchical filtering technique arises when large components are connected to small spurious structures, or when two large components of different shapes are connected through a small number of vertices. A corresponding example is shown in figure 5, where we see a small and low strength connection between two structures of different shapes. In these cases, structures should be split such that their merits and order of inclusion are computed separately before their distance fields are applied to \mathcal{F} . Practically, the algorithm breaks the components along the shortest cuts that are within a user defined limit κ . This is done by assigning a rank to each vertex equal the minimum distance it has from a boundary vertex. While growing a connected component, if a vertex has a rank less than κ then its neighbors with higher ranks are not added to the component.

3.3 Implementation

One important requirement for the algorithm described here, is the fast computation of the distance field for the different structures fused from the individual fields. This computation can be made more efficient

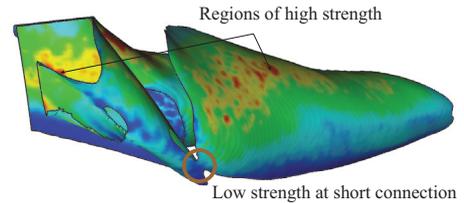


Fig. 5. An example for two components that should be split and added separately to \mathcal{F} based on their individual importance. The components are from a single field in the delta wing dataset.

Algorithm 1: Pseudocode for the computation of \mathcal{F}

Input: set of all feature surfaces \mathbf{S}

Output: scalar field \mathcal{F}

```

 $\mathcal{F} := \text{Inf}$ 
while  $\mathbf{S} \neq \Phi$  do
   $v := \max_x(I(x)), \forall x \in \text{voxels}$ 
   $\mathbf{C} = \text{ConnectedComponents}(\mathbf{S}, v)$ 
  Check the component size is significant
  if  $|\mathbf{C}| < \gamma$  then
     $\mathbf{S} := \mathbf{S} - \mathbf{C}$ 
    continue
  Update the scalar field
   $D_{\mathbf{C}} := \text{DistanceField}(\mathbf{C})$ 
  foreach voxel  $x \in \mathcal{F}$  do
     $\mathcal{F}(x) := \min(\mathcal{F}(x), D_{\mathbf{C}}(x))$ 
  Delete vertices covered by the field
  foreach  $\mathbf{S} \in \mathbf{S}$  do
     $\mathbf{S} := \mathbf{S} - \{x : x \in \mathbf{S} \wedge \mathcal{F}(x) < \alpha\}$ 
   $\mathbf{S} := \mathbf{S} - \mathbf{C}$ 

```

by restricting the maximum distance value to α since any value that exceeds this limit is considered external to the structures. Also, this computation can easily benefit from the parallelism of the GPU or other parallel architectures.

3.4 Feature Tagging

A multifield structure discovered in the volume is formed by the surfaces extracted from one or more of the fields. These surfaces are at a distance of at most α from the structure. Therefore, along with the creation of the multifield structures, the algorithm tags each vertex with information about the surfaces extracted from the individual fields that are present in its neighborhood. This tag is represented by a binary sequence $t = [b_0 b_1 \dots b_{2n_f}]$ where n_f is the number of fields in the dataset. Every field f_i has two corresponding bits b_{2i} and b_{2i+1} . One bit when set indicates the presence of a ridge while the other bit marks the presence of a valley in the makeup of the considered surface.

This tag preserves the information during the merging procedure and provides a compact encoding of the different structures. It summarizes the contribution of the different fields to the existence of every vertex in the final mesh. Hence, the described method can compactly represent a multifield dataset using a single surface mesh with vertex tags per time step. Besides being memory efficient, another significant benefit is that data transfer costs are low.

4 STRUCTURE CORRESPONDENCE

In order to understand the evolution of the different structures over time, we need to match vertices between the structures of subsequent time steps. The information tags associated with the vertices can be used to understand what change in the variables features led to a particular change in shape or topology. This analysis will be covered in details in section 5.3.

4.1 Inference Model

Surfaces in subsequent time steps might differ in translation, rotation, size, shape, topology, or any combination of these. The deformation is best described as a non-rigid deformation in addition to changes in topology. Topological changes in the context of 3D turbulent fluid datasets were discussed previously [30]. These changes include creation, dissipation, bifurcation, and amalgamation. Existing non-rigid correspondence algorithms either depend on a fixed topology [27, 1, 23], or the identification *a priori* of a small subset of knots to use for the correspondence [20, 36]. Most of the existing techniques are limited by mesh size and computational complexity. They also generally assume arbitrary rotation, translation, and scaling which are not necessarily smooth or continuous along the structure (*e.g.*, consider the motion of the different human body parts). The structures in our case have two different distinctive characteristics. First, structure deformations are likely to be spatially smooth. Second, translation is likely to be bounded assuming an appropriate temporal resolution of the simulation output. The mentioned differences limit the applicability of existing techniques to our problem.

Let M be a mesh for which we need to find vertex correspondences on the mesh M' . Each vertex in the mesh M is modeled as a random variable X_i where $i \in \{1, 2, \dots, m\}$ and $m = |M|$. Let x_i denote an arbitrary realization of X_i in M' and let T_i be the translation transformation matrix that maps X_i to x_i . The energy function $H(x)$, where x is a realization of all vertices X , comprises two components:

$$H(x) = H_1(x) + H_2(x)$$

The first component $H_1(x)$ is defined on singletons C_1 and is based on maximum principal curvature difference:

$$H_1(x) = \sum_{X_i \in C_1} h_1(x_i), \text{ with } h_1(x_i) = |\kappa(X_i) - \kappa(x_i)|$$

where κ is the maximum principal curvature function. The second component $H_2(x)$ is defined on 2-cliques C_2 and is based on the pairwise difference in deformation.

$$H_2(x) = \sum_{(X_i, X_j) \in C_2} h_2(x_i, x_j), \text{ with } h_2(x_i, x_j) = D(x_i, x_j)$$

where $D(x_i, x_j)$ represent the L^1 - Norm distance between the transformation matrices T_i and T_j . The second component clearly relies on the assumption of a smooth and continuous variation in the deformation along the surface. Hence, the energy function attempts to minimize the difference between transformations of neighboring vertices. Unlike the commonly used energy function where the second component is based on pairwise distance [1] [35], this definition of the second component allows for topological changes.

The joint distribution of a configuration x in X is $P(x) = \frac{1}{Z} \exp(-H(x))$ where Z is a normalization factor. This leads to:

$$P(x) = \frac{1}{Z} \prod_i \exp(-|\kappa(X_i) - \kappa(x_i)|) \prod_{(i,j)} \exp(-D(x_i, x_j))$$

Substituting $\phi_i(x_i) = \exp(-|\kappa(X_i) - \kappa(x_i)|)$ and $\psi_{ij}(x_i, x_j) = \exp(-D(x_i, x_j))$ gives:

$$P(x) = \frac{1}{Z} \prod_i \phi_i(x_i) \prod_{(i,j)} \psi_{ij}(x_i, x_j)$$

The *max-product* algorithm, a form of belief propagation [3], is used to compute the maximum probability variables assignment. It is applied to a factor graph of random variables. While a precise definition of factor-graphs is beyond the scope of this paper [3], they can be thought of as graphs of random variables with edges representing dependency relations. Edges in the graph represent a dependency relation between the connected variables. The term $\phi_i(x_i)$ can be viewed in the context of the max-product algorithm to represent the *independent*

probability of a certain vertex assignment while the term $\psi_{ij}(x_i, x_j)$ represents the consistency of the assignments of the neighbor nodes X_i and X_j . It is the function to maximize in order to insure proper dependency between the variables of the factor graph. In the surface matching case it indicates how likely the assignments of X_i to x_i and X_j to x_j are to occur simultaneously. For these assignments to have a high probability they should represent almost equal 3D transformations. In the non-rigid matching scenario it is unlikely to find a transformation that leads to identical mappings. Our goal is therefore to minimize the differences between the 3D transformations for these mappings.

The simplest form of the max-product algorithm occurs when the factor graph used for the inference is a tree with edges E such that:

$$\psi_{ij}(x_i, x_j) = \begin{cases} \exp(-D(x_i, x_j)) & \text{if } (x_i, x_j) \in E \\ 0 & \text{if } (x_i, x_j) \notin E \end{cases}$$

and E is a subset of the mesh edges in the matching case¹. In the case of the tree, messages are passed from the leaves toward the root and then propagated from the root down the tree. The message sent from any node i to node k indicates the belief of node i about the status of node k and the formula for the message computation is:

$$m_{ik} = \max_{x_i} \psi_{ik}(x_i, x_k) \phi_i(x_i) \prod_{j \in N(i)/k} m_{ji}(x_i)$$

where $N(i)$ is the set of neighbor nodes to the node i . This equation indicates that the message to be sent from node i to node k depends on all the incoming messages to node i except that of node k . The exact computation of this function is prohibitive due to the maximization over all possible assignments of the node X_i . We instead use only a subset of these possible assignments that are most likely to occur. We apply the max-product algorithm on a tree for multiple iterations. The tree structure is discussed in section 4.2. In each iteration we update the set of candidates at each node based on its belief computation and neighbor suggestions for similar 3D transformations. The larger the set of candidates at each vertex the less likely for the algorithm to get trapped in a local maximum. During the first iteration, each vertex in M is assigned a single candidate corresponding to its nearest neighbor in the mesh M' . The belief for an assignment from X_i to x_i can be written as:

$$b_i(x_i) = k \Phi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i)$$

4.2 Inference Tree

Notice here that the probability of a certain assignment depends on both the local properties of the point and the message propagated from neighbor nodes about the validity of this assignment. The structure of the tree used for the inference must satisfy a set of requirements. First, connected nodes must belong to the same connected component and should be close enough for the assumption of almost equal 3D transformations to be valid. Second, the tree is ideally balanced for the belief at the root node to capture the effect of the different nodes with almost identical weights. Also, this improves the efficiency of the computations since messages for nodes at a certain level in the tree can be computed in parallel and the only dependency is between subsequent levels. There is at least one root on each connected component. Each root is selected to be the furthest point from any boundary or intersection in order to avoid confusion at these nodes. For each node the children are selected as the immediate neighbors that were not yet added to the tree. Hence, we effectively restrict our considerations to a subset of the neighbors of each vertex, thereby limiting the scope of our check on the compatibility of deformations. However, this subset is sufficient for the inference given the assumption of smoothness of the deformations.

¹A graph with cycles leads to a non-exact inference that takes multiple iterations to converge.

Table 1. Mesh sizes and performance numbers for the vertex correspondence experiments.

Experiment	$ M $	$ M' $	Iterations	Computation time (sec.)
6(a)	2392	2702	12	157
6(b)	3305	3428	5	66
6(c)	3260	2667	11	203
6(d)	2869	2538	16	316

4.3 Rigid Transformation

We found that most connected components undergo a rigid transformation in addition to the non-rigid transformation discussed so far. Hence, we extended our approach by adding an additional step after each inference iteration. This step attempts to recover per connected component global translation and global rotation around the mass center of the component. The global translation is computed as the average translation discovered by the inference for that component. The rotation is found by applying the *Principal Component Analysis* technique on two sets of points. The first consists of all the component vertices from the first mesh M . The second set consists of all its corresponding points x in M' . The rotation is computed as the transformation that maps the principal directions of the first set to the principal directions of the second. Notice that the rigid transformation is computed based on only the points of known correspondence after each inference iteration. Hence, we do not make any assumption about the correspondence between any component in the first mesh with any set of components in the second mesh.

We demonstrate in figure 6 the results obtained with our method on four different components extracted from the Turbulent Vortex dataset. The components are matched to their deformed counterparts appearing one or two steps forward in time. We also applied an additional rigid translation and rotation to the components in order to increase the discrepancies between their vertices. Each subfigure shows the original mesh to the left, the initialization with the nearest neighbor assignment in the middle, and the final vertex matching result obtained to the right. The coloring of the middle and right columns (both for M') reflects the correspondence from the original mesh M shown in the first column. The white color is used for points that do not have a correspondence from the first mesh. As clear from the figures, the initialization is often far from the final result. In table 1, we document the performance numbers for the vertex matching experiments.

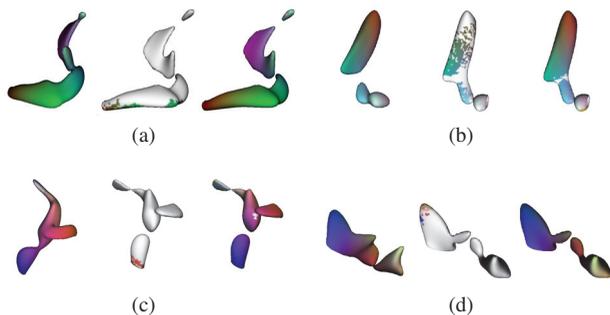


Fig. 6. The first column to the left corresponds to the original mesh M , in the middle we see the mesh M' at initialization, and to the right the mesh M' after completion.

5 DATA ANALYSIS

In this section we present a data analysis approach that naturally complements the previous feature extraction and simplification results to provide different types of information to the observer. This includes information about the interaction patterns (see below) between the different variables, discovery of certain types of relations between a pair

of variables, and finally answers to different types of queries. The approach proposed is fast as it only needs to access the vertices of the multifield structures.

5.1 Interaction Patterns

A tag at any vertex marks the coexistence of surfaces from a set of fields at that point. An interaction pattern is an indication of the structures types and variables that tend to coexist for a group of tags spread across time and space. Hence, to discover the different patterns of interaction between the variables, we need to identify different groups of similar tags. Similarity between tags is evaluated in terms of *Hamming distance* [12]. The pattern is represented as a binary word much similar to how tags are represented.

To find the different patterns, we apply *Recursive Minimal Entropy Partitioning* [8] on the tags. The partitioning is performed top-down by initially assuming all the tags to belong to the same class then iteratively split the classes until all classes have an entropy below a certain threshold. The *entropy* of a class indicates how likely for a tag in the class to be different from the rest of the tags in the same class. Hence, we attempt to maximize the similarity between tags in the same class. The entropy $H(S)$ of a class S can be defined formally using the multifield formula known from information theory [29]:

$$H(S) = - \sum_{b_1} \dots \sum_{b_{2n_f}} P(b_1, \dots, b_{2n_f}) \log_2 [P(b_1, \dots, b_{2n_f})]$$

where b_i is a random variable for the i^{th} bit and n_f is the number of fields. The probability function can be computed based on the tags in the class.

The method iteratively selects the highest entropy class for split. The partitioning is performed based on a single bit at a time. If we are given a class S , and a bit A , the class information entropy of the partition based on the values of A , denoted $H(A; S)$ is given by:

$$H(A; S) = \frac{|S_1|}{|S|} H(S_1) + \frac{|S_2|}{|S|} H(S_2)$$

where the classes S_1 and S_2 correspond to zero and one values of A . The partitioning bit is selected such that it minimizes the entropy function over all possible partition bits:

$$A = \underset{b \in \{b_1, \dots, b_{2n_f}\}}{\text{arg min}} H(b; S)$$

Hence, the classes are picked in such a way as to maximize the information the user gains about a tag knowing the class it belongs to. Splitting the class/pattern results into two new patterns where a zero appears at the bit for one pattern S_1 , and a one appears at the bit location for the other pattern S_2 . After multiple partitioning iterations, the bits not used for split at any level are the ones most likely to be dependent on the ones already used for the partitioning. These bits values are marked with a prime to indicate that their fields are less relevant in distinguishing the pattern, or equally that these bits do not have a common value across all tags in the class even though one value should be more probable than the other.

It is possible to assign a color map to the structures based on their patterns. This coloring can give insight into the spatial relations between the classes and how these classes evolve over time. Note that the suggested analysis using partitioning on the tags to discover the relations between variables is significantly more efficient than any analysis that would require a full inspection of all the 3D fields.

5.2 Query Driven

While pattern partitioning investigates the variables' interactions in a group, we perform additional analysis to understand the pairwise relations among the different variables. Specifically, we attempt to perform a correlation analysis between the different surface types (ridges and valleys) of all variables. Visualizing the correlations will allow the user to attribute different structures to the correlations discovered. This will help scientists understand what structures might form when a

specific pair of variables demonstrate high correlation. The basic idea is to create a correlation graph where the user can color the different links and visualize the associated structures in the 3D data.

The user can query the mesh to find the ridges and valleys of any field. The user can also specify a search tag to be found on the mesh along with *search radius*. The radius indicates how far a vertex tag can be from the search tag to be included in the search result. Since the tags are binary words, it is possible to use the Hamming distance as a measure of distance between tags. The use of the search radius helps overcome possible noise or discontinuities of the tags on a surface. Another form of queries consists in allowing the user to filter structures based on a range query on the different fields. It is also possible to color the structures based on one of the fields in the dataset.

5.3 Temporal Analysis

For the temporal analysis, we suggest three possible visualizations. In the first, the user is presented with a time chart showing the size of each pattern over time. The time chart should give scientists an idea of what patterns are associated with the different structures or events known to appear in the data at different points in time. Also, it can give insight into possible periodic aspects of the data.

The second strategy consists of a transition graph that demonstrates the transition probabilities between the different patterns. There is a single node in the graph for each pattern. The links in the graph reflect the number of vertices that are likely to move from one pattern to another. This graph is built based on the correspondence information between vertices in subsequent time steps. The purpose of this visualization is to give information about the most likely changes in variables on the structures.

The last visualization consists of a graph where each node represent a pattern at a particular time step. The edges between nodes represent the number of vertices that moved from one pattern to the other between time steps. This can be considered a more detailed version of the second visualization mentioned. It provides a better understanding of the different groups of variables affecting the life time of the structures over time. The user can use brushing to assign colors to the different transitions and visualize the corresponding surfaces extracted from the data.

6 RESULTS

The first dataset we consider is a turbulent combustion simulation conducted by scientists at Sandia National Laboratories (SNL) in order to understand the dynamic mechanisms of the combustion process. It contains five fields: stoichiometric dissipation rate (chi), heat-release (hr), mixture fraction (mixfrac), vorticity, and hydroxyl radical (YOH). As a first step we built the correlation graph between creases of the different variables as seen in figure 7(a). It clearly shows the combustion process is dominated by the YOH species and the mixture fraction. The YOH combusts, hence, high YOH correlates with high heat release at the valley like regions of mixture fraction (due to a higher concentration of YOH). Accordingly the scalar dissipation rate also increases due to the combustion. Remarkable is the impact of vorticity: For given concentration of YOH even a slightly higher vorticity is able to enforce the combustion and, therefore, to increase heat release. One reason might be the effect that turbulence models react on shear with higher fluctuation levels, which consequently leads in combustion models to higher combustion rates. The physical explanation behind is that higher fluctuation rates are enforcing a stronger displacement or micro-mixing of species with different densities. This means for a combustion process that the probability to find the right chemical partner for the reaction is increased and the scalar dissipation is reduced. In order to visualize this relation, we preform a query on the combined structures such that we only keep these structures that belong to a good mixing range around 0.42. This mixture condition best describes the burning flame. We then color these structures with both YOH and chi as shown in figure 8. It can be noticed that YOH and chi are exclusive on the burning surface. However, the regions of high dissipation are often in small area. The visualization of the meta features was also able to show small holes and boundaries

with low YOH values close to the burning surface in figure 8(a). Visualizing and tracking these holes is pivotal to the understanding of the reignition process that often occur at the boundaries of these low YOH structures. We also notice that for certain fields there is strong correlation between the ridges and the valleys of the same field. This is an indication that these structures are likely to occur simultaneously at a close proximity, and is clearly a sign of the field instability. One such field is the vorticity as it captures the fine turbulence timescales. Also, this applies to the YOH and hr fields which are highly affected by turbulence and the mixture condition as can be observed from the links in the graph. On the other hand, the mixture fraction structures, ridges and valleys, are only weakly connected, which means that there is an independent source of YOH.

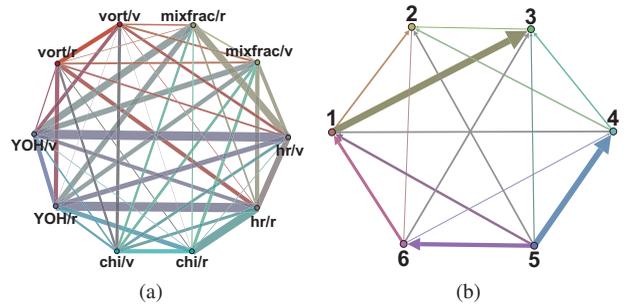


Fig. 7. *Combustion*: (a) Relations between the different types of structures, (b) Particles transition between clusters.

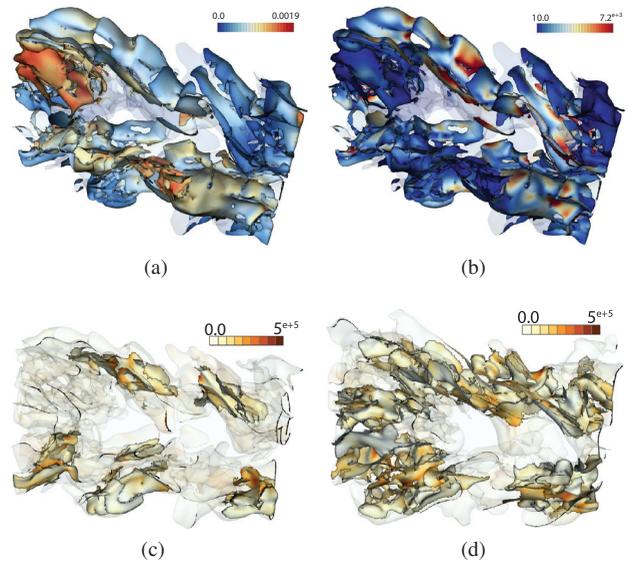


Fig. 8. *Combustion*: The combined structures filtered by the mixfrac field in the range [0.25,0.59] colored by YOH in (a) and by chi in (b). (c) and (d) show vorticity structures from time steps 70 and 80 respectively both colored with vorticity.

The temporal development of the combustion process is shown in figure 7(b): Here, the transition between six derived partitions are shown. Only the bits between circular brackets were actually used for the partitioning. The other bits values are computed based on the most expected value from the partition statistics. First, the combustion is driven nearly purely by the heat at a flame front and the vorticity does not play a significant role, see partition 1 in table 2. Latter the process changes: The influence of vorticity and therefore shear and turbulence gets higher, the combustion structure shifts from partition 1 to partition 3. One might interpret this as (the begin of) a collapse of

Table 2. Partitions in the Combustion dataset.

Partitions	Entropy	YOH	vort.	mixfrac.	hr	Chi
(1)	6.0	11	(0)1	11	(1)1	01
(2)	5.4	(1)1	01	(1)1	(0)1	01
(3)	5.4	11	(1)1	11	(1)1	01
(4)	5.3	(0)1	01	(1)0	(0)1	01
(5)	5.1	00	01	(0)0	(0)(0)	00
(6)	5.1	01	01	(0)0	(0)(1)	01

the flame front. This change is mostly visible at time steps 25, 58, 85 and 115 as shown in figure 9. From the chart, we also notice that the strongest expansion in size was achieved by the third partition. This implies that the burning structures for high vorticity has expanded the most by the end of the simulation. This was confirmed by two visualizations. In the first, we show only the meta features for the time steps 70 and 80 filtered to reveal vorticity structures as in figure 8(c) and 8(d). In the second (figure 10), we show the meta structures from a sequence of steps colored based on the partitions, and assigned an opacity reflecting the match between the local mixture condition and the 0.42 mixfrac value. In both visualization, we notice the expansion of structures and partitions relevant to high vorticity. In addition, we notice that the partitions 1, and 3 have the highest opacity which indicate that these partitions are often close to the flame (burning regions). There is also a strong partition, 5, without YOH and combustion. This region becomes decomposed into a two partitions with different mixture fractions not far from the simulation start. The analysis of this combustion dataset demonstrates that the proposed methodology can significantly support the optimization of both the geometry and the operation mode of burners. In the present case it is clear that the flame front collapses with time. Obviously no external driven swirl has been generated for stabilizing the flame front.

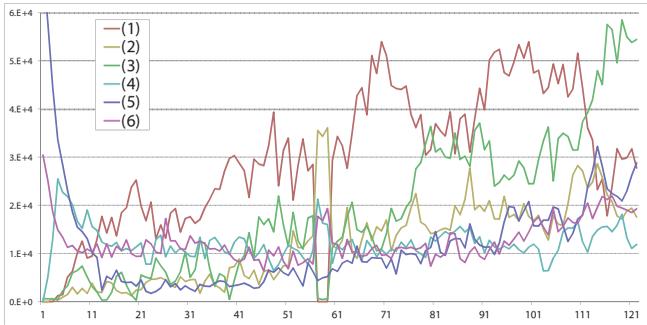


Fig. 9. Time chart for the Combustion dataset partitions.

The Ionization front 3D Scalar dataset is an ionization front instability simulation data set submitted by Mike Norman and Daniel Whalen, and made available through the IEEE Visualization 2008 Contest. It contains 10 different fields including particle density, temperature, and 8 chemical species. The main driver of the process is the temperature dependent density. When the gases enters/leaves region with higher temperatures the density decreases/increases. Looking at the meta features extracted from the the fields at time step 90, we notice the presence of multiple layers of structures around the center. Coloring these structures based on the temperature field (figure 11(a)) revealed a region around the center with high temperature. In order to discover the relation between the density of the chemicals and the temperature we filtered the meta features such that only temperature structures are visible. We then colored these structures based on the density field as in figure 11(b). This visualization revealed that regions of high temperature have low density and *vice versa*. It also revealed that high density is mainly concentrated at the exterior. To understand the role of the different gases in the dataset, we filtered the meta features to reveal structures from each gas individually and used coloring to reveal the locations of ionized and non-ionized gases. For example, in

figure 11(c) we show all structures corresponding to the fields of ionized and non-ionized He . The coloring is based on the non-ionized gas mass. In figure 11(d), however, we color the same structures based on the ionized gas mass fields. This visualization clarified the locations of the different ions and demonstrated the layers of separation between the concentrations of the different gases.

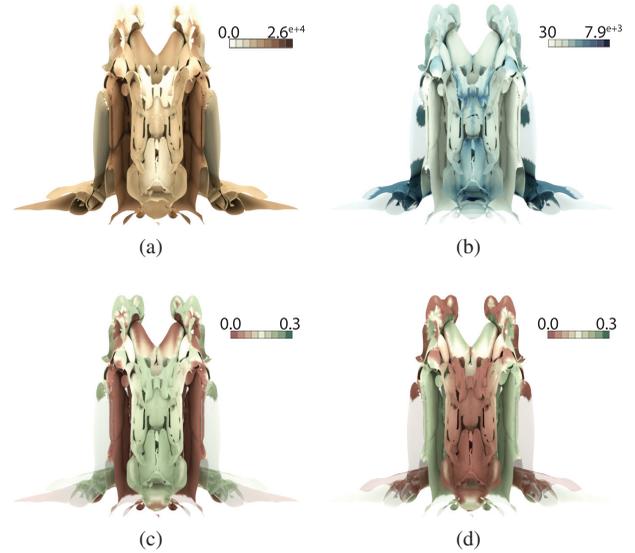


Fig. 11. Ionization front: (a) All structures from time step 90 colored by temperature, (b) structures from the temperature field colored by density, (c) He structures colored by He , and (d) He structures colored by He^+ .

Finding the different partitions of this dataset provides better insight about the variables relations. The different partitions discovered are listed in table 3. Looking carefully, we notice the similarity between the bits for H and He . Similarly, the bits for H^+ and He^+ match exactly for all partitions. In fact there is a kind of “ping-pong” relation visible between H and H^+ and He and He^+ , which can be seen at the flipping bits in table 3. The fields H^- and H^{2+} also match. However, these last two fields also show great correlation between ridges and valleys. This implies that the structures in these fields correspond to a temporary transition in value rather than a stable transition. Looking at the partitioning bits in particular, we notice that most of these bits are shared by three fields only. This further demonstrate the dependency between fields in the dataset.

In figure 12, we show the partitions’ transition graph and the meta features at time steps 80, 90 and 100 colored based on the partitions. The transition from 6 to 5 represents structures moving at the center toward regions of higher density. The transitions from 2 to 4 and from 4 to 1 is for the movement of gases entering the the hot layer with lower density. Finally, the transition from 1 to 3 correspond to material leaving the hot layer either toward the center or toward the exterior. Notice that the gases H , H^+ , He , and He^+ have flipped bits between the partitions 1 and 2. There is a high level of partition fluctuations visible in figure 12, this is also a hint for an underlying highly turbulent jet flow. Due to these fluctuations the ionization process shows the unstable transitional character.

Finally, the well known delta wing dataset is characterized by strong primary and secondary vortices, which occur due to flow separation and attachment. The fields we considered for this test case are the density, Sa viscosity, Eddy viscosity, and λ_2 . During the separation, parts of the boundary layer, coming from the lower part of the wing, is enforced to role up and to form the vortices. Additionally outer fluid is sucked into the vortical system increasing shear. Since dominant vortices consists of both fluid for the boundary layer and from the free flow, footprints of the modeling of turbulence are visible. At wall parallel shearing flows the production and destruction of the turbulent

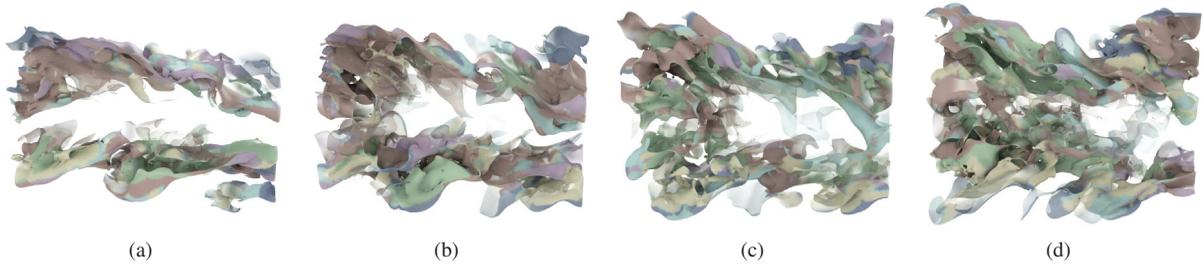


Fig. 10. *Combustion*: (a), (b), (c), and (d) are the meta features extracted for the time steps 60, 70, 80 and 90 respectively.

Table 3. Partitions in the Ionization front dataset.

Partitions	H^{2+}	H^2	H^-	He^{++}	He^+	He	H^+	H	temp.	dens.
(1)	11	0(1)	11	00	(1)0	01	1(0)	01	10	(0)1
(2)	11	1(1)	11	00	(0)1	10	01	10	1(1)	(1)1
(3)	11	0(1)	11	00	(0)0	00	00	(0)0	1(1)	(0)1
(4)	00	0(0)	00	00	00	00	00	00	10	01
(5)	11	1(1)	11	00	(1)1	11	11	11	11	(1)1
(6)	11	1(1)	11	00	(1)1	11	1(1)	11	11	(0)1
(7)	01	0(1)	01	00	(0)0	00	00	00	0(0)	01
(8)	11	0(1)	11	00	(0)1	10	01	(1)0	0(1)	(0)1

Table 4. Partitions in the Delta Wing dataset.

Partitions	λ_2	Eddy visc.	Sa visc.	Density
(1)	01	11	(1)1	(0)0
(2)	01	00	0(0)	(1)0
(3)	01	01	(0)1	(0)0
(4)	01	01	(0)1	(1)0
(5)	01	11	(1)1	(1)0

In the delta wing, we have in principal three phases, the stable vortical flow structure (time step 301), the transitional flow before the onset of vortex breakdown (time step 501) and the onset itself. In regard to Eddy and Sa viscosity, the stable vortical flow structure can be characterized by higher Sa below the primary vortices and at the vortex core and higher Eddy viscosities surrounding the vortices as an outer layer, since the vortex feeding and flow redirecting lead to high shear rates. At shearing regions of attachment and secondary separation, the Eddy viscosity is also less damped. A further effect complicating the flow situation is the advection of once produced Sa viscosity. It is advected away from the wall and mixes with Eddy viscosity. In particular at secondary flow separation Eddy and Sa viscosities are advected and produced in neighboring layers, rolling up. The strong correlation between the Sa viscosity and the Eddy viscosity is demonstrated in the correlation graph in figure 14(a) and in the bits of the partitions as in table 4. In the next phase, the angle of attack has been increased which leads to a weaker feeding of the vortices, lower flow acceleration, in parts deceleration, and shrinking outer shear rates visible in a less dominant Eddy viscosity at the outer layer. However, the vortex core lifts off a little bit from the wing surface, the influence of the damping function is weakening, thus less Sa viscosity is generated. This reduction is Sa viscosity is also supported by the strong transition from partition 1 to partition 3. This effect is reinforced during the last flow phase as seen in the chart in figure 15. The outer radius of the vortices in front of the breakdown is further increased and the breakdown bubble appears as in figure 13(f). At the vortex breakdown the Sa vorticity spreads outwards from the core due the decelerated and reversed flow field. This can be noticed in figure 13(c) as partition 2 dominates the core. The transitions from partitions 1, 3 and 4 to 2 demonstrate this event. Beneath the breakdown the enlarged flow structure has an impact on the wall leading to an increased influence of the damping function and therefore a higher fraction of Sa viscosity (partitions 1 and 5 at the wall and around the bubble). The delta wing dataset does not only exhibit the physical effects of the genesis of vortices, it also exhibits the underlying techniques of modeling with all the strongholds and weaknesses. From this point of view the proposed methodology offers a useful tool for analyzing the interaction of the drivers of the flow structure genesis. It is noticeable that we were able to capture the main vortices, the recirculation bubble, and the outside structure from the fields.

The test datasets and the fields we used in our experiments are listed in table 5. All datasets were produced from scientific simulations and use the floating-point data type. We have benchmarked our algorithm on a server class machine with eight Intel Xeon X7560 CPUs and 110GB of RAM. The timings for the algorithms mentioned in the

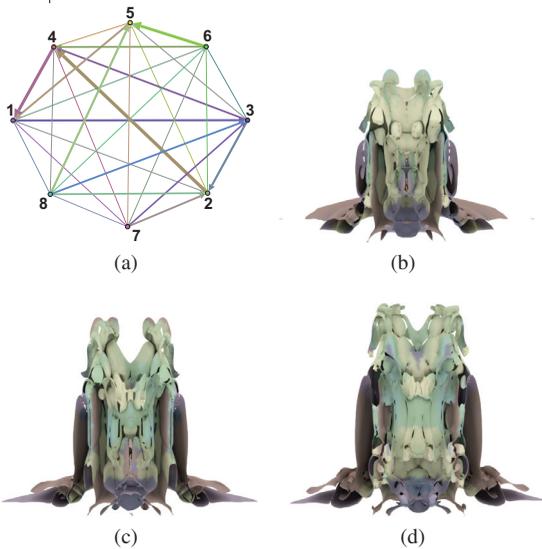


Fig. 12. *Ionization front*: (a) Graph for the transitions between partitions, (b),(c), and (d) show the clusters for the time steps 80, 90, and 100 respectively.

dynamics Eddy viscosity is different in comparison to that in regions of free shear flow. This can be seen in figure 13 by observing the locations of the different partitions. Turbulence modeling with the Spalart-Almaras model considers this effect by introducing a correction factor which shifts the Eddy viscosity in direction of laminar (molecular) viscosity. The correction factor is calculated by a damping function in which the wall distance is one essential parameter. By adjusting the damping function the transition between laminar and turbulent flow can be controlled such that closer to the walls the turbulence level has to be decreased. Additionally in the turbulence model a rotational correction is done. Strong rotation can decrease the turbulence levels as it is at the vortex core. Both effects are now incorporated in Sa viscosity. The comparison of both turbulent viscosities is very important since this allows to adjust the control parameters in the damping function according to data coming from experiments. Therefore, the proposed visualization technique can support the further development of turbulence models.

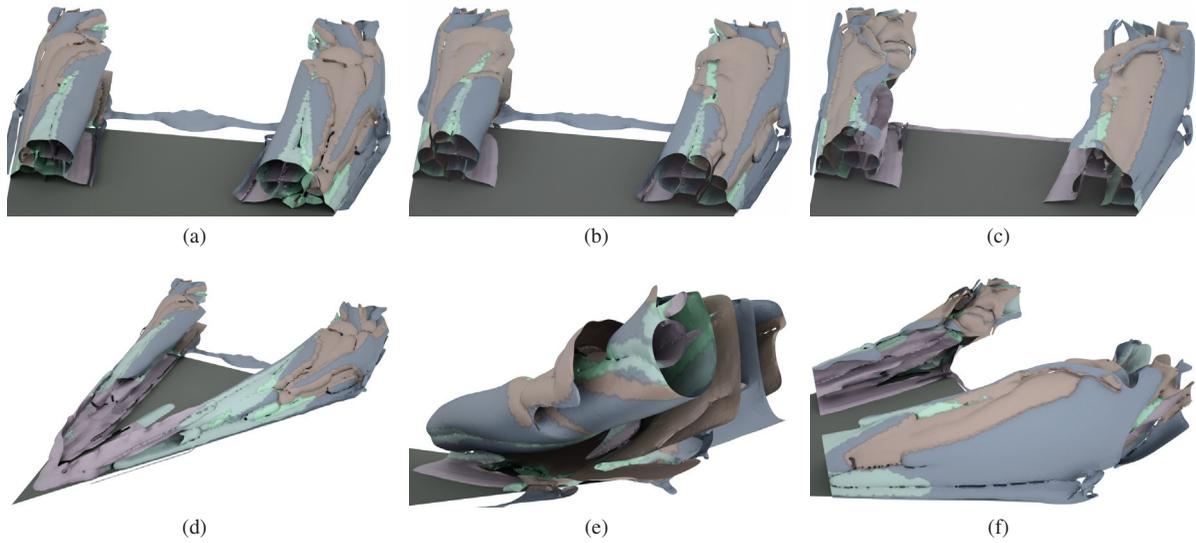


Fig. 13. *Delta Wing*: (a),(d) Initial structures ($\tau_i = 301$); (b),(e) structures before breakdown ($\tau_i = 501$); (c),(f) structures at breakdown point ($\tau_i = 701$).

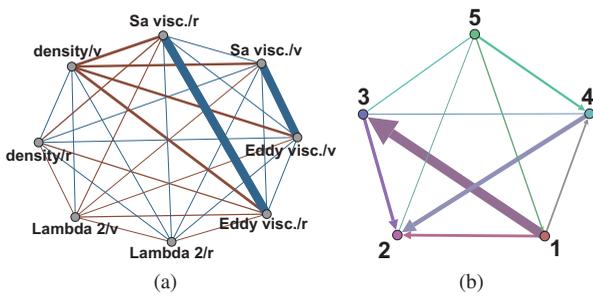


Fig. 14. *Delta Wing*: (a) Graph showing the correlations between different types of structures, (b) Transition graph between partitions.

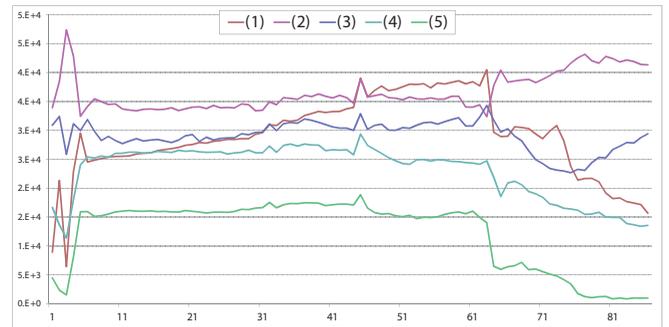


Fig. 15. Time chart for the partitions found in the *Delta Wing* dataset.

addition to the different parameters used are listed in table 5. The parameter β was selected based on a GUI interface that interactively shows the result of the extraction with different values. The parameter γ was selected based on the expected minimum size of an isolated meaningful structure. Small changes in the value for this parameter do not affect the visual result.

7 CONCLUSION AND FUTURE WORK

In this paper, we proposed a geometrical definition for multifield features and we discussed an iterative algorithm to efficiently reduce the structures complexity through hierarchical filtering and fusion of structures discovered in the individual fields. We explained how this model integrates a strategy to filter noise and spurious structures, and we demonstrated the elimination of repetition between structures in the different fields. The purpose of this processing is to enhance the visual perception of structures from multiple fields and permit a visualization of these structures in a single view by computing and visualizing smooth skeletons of the major structures in the data. We also presented a new framework for the spatial and temporal analysis of multifield time-varying 3D data. The analysis is based on the per vertex information about the different fields contributing to the formation of the surfaces. This analysis provides insight into the coherence of features from different variables. We suggested applying data space partitioning in order to discover frequent patterns of feature coherence between variables. The discussed analysis also allows the user to issue different types of queries on the surfaces to visualize structures corresponding to features from any subset of variables. For the temporal

analysis, we proposed a new non-rigid surface matching strategy to find correspondences between surfaces at subsequent time steps. This correspondence information is used to discover variable change associated with structures over time. We applied these ideas to different time-varying multifield datasets and demonstrated different relations and dynamics in these datasets.

For future work, we would like to extend the proposed approach to vector and tensor fields by finding appropriate ways of mapping these fields to surfaces. Another avenue to investigate, is to provide an interactive and adaptive visualization where simplification is linked to the view resolution. This will allow the user to navigate across scales in order to visualize high details in some datasets. Finally, we would like to use our correspondence algorithm in a framework through which the user can select a small portion of any structure and visualize an animation for that structure across time.

ACKNOWLEDGMENTS

The authors wish to thank Dr. Jacqueline Shen and the SciDAC Ultra-vis institute for providing the turbulent combustion dataset. This paper is based in part upon work supported by the National Science Foundation under Grants No. OCI-1150000 and CMMI-1030326 and by a gift by Intel Corporation. Finally, we want to thank the providers of the Teem library <http://teem.sf.net>, the VTK library <http://www.vtk.org/>, and the Octane render engine <http://www.refractivsoftware.com/> for making these tools available to the research community.

Table 5. Dataset Information and performance numbers.

Dataset	Dimensions	Time steps	Number of fields	Extraction time / field (sec)	Simplification time (min)	Matching time (min)	Ave. mesh size $\times 10^3$	α	$\gamma\%$	κ
Combustion	480 x 720 x 120	122	5	44	5.6	7.1	112	9	1	6
Ionization front	600 x 248 x 248	200	10	42	8.6	5.5	69	6	1	3
Delta Wing	517 x 401 x 105	86	4	38	4.4	4.8	141	4	1	3

REFERENCES

- [1] D. Anguelov, P. Srinivasan, H. cheung Pang, and D. Koller. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *In TR-SAIL-2004-100*, at <http://robotics.stanford.edu/drago/cc/tr100.pdf>, pages 33–40, 2004.
- [2] S. Barakat, N. Andryscio, and X. Tricoche. Fast extraction of high-quality crease surfaces for visual analysis. *Computer Graphics Forum*, 30:961970, 2011.
- [3] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [4] G. Borgefors. Distance transformations in arbitrary dimensions. 27(3):321–345, September 1984.
- [5] H. Doleisch and H. Hauser. Smooth brushing for focus+context visualization of simulation data in 3d. In *Journal of WSCG*, pages 147–154, 2001.
- [6] D. Eberly, R. Gardner, B. Morse, and S. Pizer. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4:351–371, 1994.
- [7] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. Ridges for image analysis. *Journal of Mathematical Imaging and Vision*, 4:353–373, 1994. 10.1007/BF01262402.
- [8] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, pages 1022–1029, 1993.
- [9] L. Gosink, J. Anderson, W. Bethel, and K. Joy. Variable interactions in query-driven visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13:1400–1407, November 2007.
- [10] L. J. Gosink, S. Member, J. C. Anderson, S. Member, E. W. Bethel, and K. I. Joy. Variable interactions in query-driven visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 13:2007, 2007.
- [11] J. H. B. M, T. X, and S. G. Automatic detection and visualization of distinctive structures in 3d unsteady multi-fields. *Computer Graphics Forum*, 27:767774, 2008.
- [12] R. W. Hamming. Error detecting and error correcting codes. *Bell System Tech. J.*, 29:147–160, 1950.
- [13] A. K. Jain. *Fundamentals of Digital Image Processing*, chapter 9, page 382. Prentice Hall, 1988.
- [14] H. Janicke, M. Bottinger, and G. Scheuermann. Brushing of attribute clouds for the visualization of multivariate data. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1459–1466, nov.-dec. 2008.
- [15] H. Jnicke, A. Wiebel, G. Scheuermann, and W. Kollmann. Multifield visualization using local statistical complexity, 2007.
- [16] G. Kindlmann, R. San Jose Estepar, S. M. Smith, and C.-F. Westin. Sampling and visualizing creases with scale-space particles. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1415–1424, 2009.
- [17] G. Kindlmann, X. Tricoche, and C.-F. Westin. Delineating white matter structure in diffusion tensor MRI with anisotropy creases. *Medical Image Analysis*, 11(5):492–502, October 2007.
- [18] J. Kniss, P. McCormick, A. McPherson, J. Ahrens, J. Painter, A. Keahey, and C. Hansen. Interactive texture-based volume rendering for large data sets. *IEEE Comput. Graph. Appl.*, 21:52–61, July 2001.
- [19] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. In 1996, pages 465–470, June 1996.
- [20] Y. Lipman and T. Funkhouser. Mbius voting for surface correspondence. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 72:1–72:12, New York, NY, USA, 2009. ACM.
- [21] N. Litke, M. Droske, M. Rumpf, and P. Schroder. An image processing approach to surface matching. In *Proceedings of the third Eurographics symposium on Geometry processing*, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.
- [22] A. R. Martin and M. O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th conference on Visualization '95*, VIS '95, pages 271–, Washington, DC, USA, 1995. IEEE Computer Society.
- [23] A. Myronenko and X. Song. Point set registration: Coherent point drift. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2262–2275, dec. 2010.
- [24] R. Peikert and F. Sadlo. Height Ridge Computation and Filtering for Visualization. In I. Fujishiro, H. Li, and K.-L. Ma, editors, *Proceedings of Pacific Vis 2008*, pages 119–126, March 2008.
- [25] H. Qu, W.-Y. Chan, A. Xu, K.-L. Chung, K.-H. Lau, and P. Guo. Visual analysis of the air pollution problem in hong kong. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1408–1415, nov.-dec. 2007.
- [26] N. Sauber, H. Theisel, and H.-P. Seidel. Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics*, 12:917–924, 2006.
- [27] D. Schneider and P. Eisert. Fast nonrigid mesh registration with a data-driven deformation prior. In *Computer Vision Workshops (ICCV Workshops)*, 2009 *IEEE 12th International Conference on*, pages 304–311, 27 2009-oct. 4 2009.
- [28] T. Schultz, H. Theisel, and H.-P. Seidel. Crease surfaces: From theory to extraction and application to diffusion tensor MRI. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):109–119, 2010.
- [29] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.
- [30] D. Silver and X. Wang. Tracking and visualizing turbulent 3d features. *Visualization and Computer Graphics, IEEE Transactions on*, 3(2):129–141, apr-jun 1997.
- [31] K. Stockinger, J. Shalf, W. Bethel, and K. Wu. Dex: Increasing the capability of scientific data analysis pipelines by using efficient bitmap indices to accelerate scientific visualization. In *In International Conference on Scientific and Statistical Database Management (SSDBM)*, pages 35–44. IEEE Computer Society Press, 2005.
- [32] X. Tricoche, G. Kindlmann, and C.-F. Westin. Invariant crease lines for topological and structural analysis of tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, 2008.
- [33] C. Wang, H. Yu, R. Grout, K.-L. Ma, and J. Chen. Analyzing information transfer in time-varying multivariate data. In *Pacific Visualization Symposium (PacificVis)*, 2011 *IEEE*, pages 99–106, march 2011.
- [34] C. Wang, H. Yu, and K.-L. Ma. Importance-driven time-varying data visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1547–1554, nov.-dec. 2008.
- [35] P. Xiao, N. Barnes, P. Lieby, and T. Caetano. Sparse update for loopy belief propagation: Fast dense registration for large state spaces. In *Digital Image Computing: Techniques and Applications (DICTA)*, 2010 *International Conference on*, pages 546–551, dec. 2010.
- [36] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samaras, and N. Paragios. Dense non-rigid surface registration using high-order graph matching. In *Computer Vision and Pattern Recognition (CVPR)*, 2010 *IEEE Conference on*, pages 382–389, june 2010.