

Exploring Time-Varying Multivariate Volume Data Using Matrix of Isosurface Similarity Maps

Jun Tao, *Member, IEEE*, Martin Imre, *Student Member, IEEE*, Chaoli Wang, *Senior Member, IEEE*, Nitesh V. Chawla, *Member, IEEE*, Hanqi Guo, *Member, IEEE*, Gökhan Sever, and Seung Hyun Kim

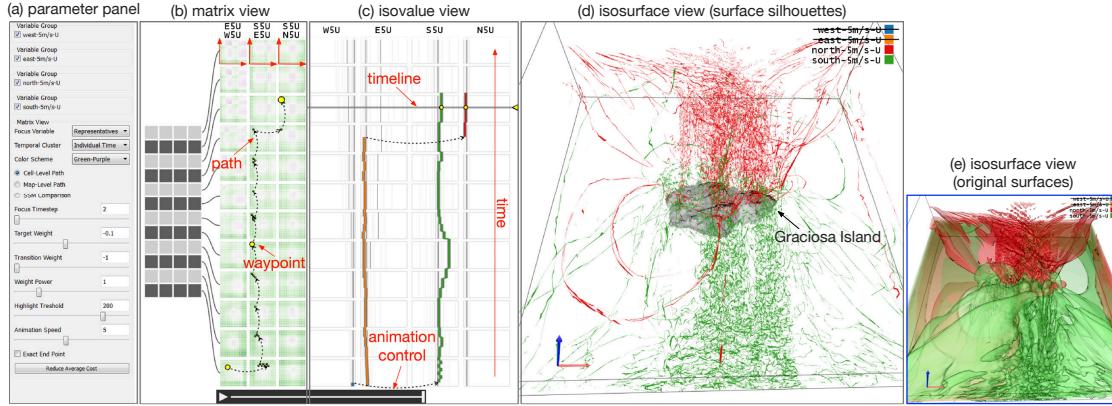


Fig. 1. MISIM interface. The cells in the similarity maps are colored purple/white/green for high/medium/low values. A path is specified to reveal the connections between the isosurfaces of variable U corresponding to four different wind directions using the atmospheric ensemble data set. (a) shows the parameter panel. (b) shows the matrix view with a user-specified path. A summarized view of the matrix is displayed on the left and the representative SSMs and VSMs related to the path are displayed on the right. (c) shows the isovalue view. The blue, orange, green, and red paths correspond to the isovalue of U under the west, east, south, and north wind directions, respectively. (d) and (e) show the isosurface view with surface silhouettes and original surfaces, respectively.

Abstract—We present a novel visual representation and interface named the matrix of isosurface similarity maps (MISM) for effective exploration of large time-varying multivariate volumetric data sets. MISM synthesizes three types of similarity maps (i.e., self, temporal, and variable similarity maps) to capture the essential relationships among isosurfaces of different variables and time steps. Additionally, it serves as the main visual mapping and navigation tool for examining the vast number of isosurfaces and exploring the underlying time-varying multivariate data set. We present temporal clustering, variable grouping, and interactive filtering to reduce the huge exploration space of MISM. In conjunction with the isovalue and isosurface views, MISM allows users to identify important isosurfaces or isosurface pairs and compare them over space, time, and value range. More importantly, we introduce path recommendation that suggests, animates, and compares traversal paths for effectively exploring MISM under varied criteria and at different levels-of-detail. A silhouette-based method is applied to render multiple surfaces of interest in a visually succinct manner. We demonstrate the effectiveness of our approach with case studies of several time-varying multivariate data sets and an ensemble data set, and evaluate our work with two domain experts.

Index Terms—Time-varying multivariate data visualization, isosurface, similarity map, visual interface, path recommendation.

1 INTRODUCTION

Many scientific simulations produce time-varying multivariate volume data that can span hundreds or thousands of time steps and consist of tens of variables. Additionally, ensemble data sets are common nowadays, where a simulation is conducted in multiple runs with different configurations. This leads to the multiplied amount of data to be studied. Understanding the underlying physical phenomena in this kind of data often requires key insights to be discovered through observations. This need places analysis and visualization of time-varying multivariate data

at the heart of scientific visualization. There are two dominant techniques to visualize volumetric data sets: direct volume rendering and isosurface rendering. Direct volume rendering relies on the specification of a transfer function that maps scalar values to colors and opacities for rendering. The visualization can highlight particular values or value ranges with high opacity while deemphasizing non-interesting ranges with low or zero opacity. Isosurface rendering explicitly extracts one or multiple isosurfaces from the volume data and renders these surfaces as the visual data representation. The rendering can reveal important data features or structure if the corresponding isovalue are salient or representative. In this work, we focus on the less-explored surface-based analysis in the context of time-varying multivariate data visualization. Our goal is to investigate volumetric data evolution over space and time and across multiple variables and ensembles by extracting a range of isosurfaces, computing their similarities, identifying representative surfaces, and presenting a visual interface to organize, summarize, and explore the underlying time-varying multivariate data set.

Several key challenges are involved with this approach. First, studying the similarities for all the surfaces across time steps and variables leads to a very large number of similarity maps. For instance, for a data set of 100 time steps and ten variables, the total number of similarity

• J. Tao, M. Imre, C. Wang, and N. V. Chawla are with University of Notre Dame. E-mail: {jtao1, mimre, chaoli.wang, nchawla}@nd.edu.
• H. Guo and G. Sever are with Argonne National Laboratory. Email: {hguo, gsever}@anl.gov.
• S. H. Kim is with The Ohio State University. E-mail: kim.5061@osu.edu.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x.

Date of Publication xx xxx. 201x; date of current version xx xxx. 201x.
For information on obtaining reprints of this article, please send e-mail to:
reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. xx.xxxx/TVCG.201x.xxxxxxx

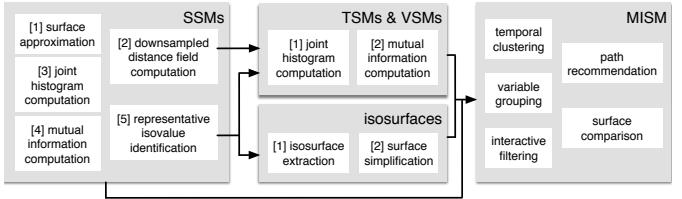
maps amounts to 55,000. These include 1000 self-similarity maps, 49,500 temporal similarity maps, and 4,500 variable similarity maps (refer to Section 1 in the Appendix). Without an effective way to organize and visualize them, examining such a large number of similarity maps would become a daunting task. Second, even though similarity maps can help us see the summarized information, we still need to examine related isosurfaces in the original spatial view to identify where and how they are (dis)similar. However, since a single similarity map typically compares tens or even hundreds of isosurfaces, it is not practical for users to manually select the related isosurfaces for further examination. This task will be even more exhausting when the temporal development of a variable is studied as thousands of isosurfaces may be involved. Third, rendering just a few surfaces may already lead to serious occlusion and clutter which prevents clear observation and comparison. This calls for new techniques that render a number of surfaces with minimal occlusion while preserving their spatial relationships or context. Finally, it could take hours to compute pairwise isosurface similarities for only a single volume if no approximation, downsampling, or acceleration solutions are considered [5]. This makes it impractical to apply the same computation process to generate similarity maps for a typical time-varying multivariate data set, although the process is only done once. Therefore, it is imperative to seek a cost-effective solution to make the similarity computation scalable.

We tackle the above challenges and present the matrix of isosurface similarity maps (MISM), a new approach for comparative visualization of time-varying multivariate volume data. Figure 2 sketches the diagram of our MISM framework. During preprocessing, self-similarity maps are first computed for each volume of the data. For each self-similarity map, we then select a set representative isovalue and compute the corresponding isosurfaces. Finally, each of these isosurfaces is simplified to save storage space and support runtime interaction. Furthermore, the information acquired from self-similarity map computation will be used to derive temporal and variable similarity maps. At runtime, all similarity maps (self, temporal, and variable) will be used to construct MISM. We provide a list of functions for users to explore the collection of similarity maps and compare the underlying isosurfaces. The contributions of our work are the following:

- We extend the isosurface similarity map (ISM) to study the similarity across the temporal and variable domains. We select representative isovalue from the self-similarity maps and use them to efficiently compute the temporal- and variable-similarity map.
- We design MISM, a matrix-based visual interface to organize these similarity maps and present various ways (i.e., temporal clustering, variable grouping, filtering, recommendation, and comparison) for effective relationship overview, level-of-detail exploration, and comparative study.
- We leverage and modify a silhouette-based method to visualize a number of related isosurfaces in a visually succinct manner. This allows users to examine the similarities and differences of a large number of surfaces in the isosurface view, presenting essential information while reducing occlusion and clutter.
- We demonstrate the usefulness of our approach with case studies of several time-varying multivariate data sets and one ensemble data set. We evaluate our system with two experts. The results show that MISM enables users to gain a deep understanding of the data across the spatial, temporal, and variable domains.

2 RELATED WORK

Data analysis and visualization. Analysis and visualization of time-varying multivariate volumetric data is an important yet challenging topic in scientific visualization [22, 27]. For time-varying data, researchers have studied efficient volume rendering based on spatiotemporal coherence [33], transfer function specification [19], and direct rendering of multiple time steps into a single image [40]. They also experimented with illustration-inspired [20] and importance-driven [38] techniques for visualizing time-varying data. For multivariate data, researchers have investigated query-driven visualization using compound range queries [34] and fuzzy queries using textual pattern matching [10]. They also explored variable correlation including point-wise correlation



thumbnail gallery of feature pairs and a graph representation are used to show all relationships between individual contours. Lampe et al. [23] designed a curve-centric volume reformation technique to create compelling comparative visualizations. The technique deforms the volume surrounding a curve and preserves the spatial neighborhood to the curve, thus permitting arc-length parameterized data visualizations in parallel for comparison. Malik et al. [28] presented a comparative visualization technique that uses a multi-image view and an edge explorer for dimensional measurement using 3D X-ray computed tomography. Their generic solution can be applied to other application areas such as parameter study of imaging modalities and detection of artifacts in data sets. Schmidt et al. [31] designed a comparative visualization tool named VAICo for visualizing similarities and differences in large image sets. This tool not only preserves contextual information but also enables detailed analysis of subtle variations. Our purpose is to investigate the evolution of time-varying data in the context of multiple variables using surface-based visual representation. We achieve this goal by clustering and filtering a large number of ISMs organized as a matrix along with comparative surface summarization and rendering techniques that reveal the spatial relationships of the corresponding isosurfaces.

3 DESIGN REQUIREMENTS

To investigate the physical phenomenon of the multifaceted time-varying multivariate data sets, experts need to understand the relationships among time steps, variables, and value ranges, which are often complicated and dynamic. The relationships of value ranges of multiple variables may exhibit various patterns even at a single time step, and these relationships may evolve over time and develop into different patterns. Understanding these relationships can hardly be achieved without an effective visual analytics system. We formulate the design requirements of such a system as follows:

R1. Overview. The system should provide two levels of overview. First, it should allow users to observe the overall relationships between variables and time steps at the *volume* level, helping them answer questions such as *which variables are more similar, which time steps are more similar, and which variables exhibit more frequent changes than the others?* Second, it should allow users to understand the relationships at the *isovalue* level. Similar questions should be answered at this level with more detail: *which ranges of isovales lead to the relationships discovered at the volume level?*

R2. Identification of representatives. The system should quantify the similarities among time steps, variables, and isovales and derive the representative ones. This provides users more quantitative evidence to verify the relationships they observe from the overview and to reduce the search space for detailed exploration.

R3. Relationship-centric exploration. The system should provide multiple modes to investigate various types of relationships, including *the relationships among value ranges for a given variable at a specific time step, the relationships among variables at a specific time step, the relationships of the same variable at different time step, the temporal evolution of the relationships of two variables, and the relationships of a specific feature and others (which may not reside in the same variable)*, etc. The system should be able to quantify each type of relationship and the interface should be optimized to allow exploring each type of relationships efficiently.

R4. Multi-step comparison path. The dynamic relationships among variables may require comparisons of hundreds of surfaces to be demonstrated. The system should decompose these complicated relationships into a *path* of multiple comparison steps. Each step should reveal a facet of the relationships with a reasonable number of surfaces that can be effectively rendered for clear visual comparison. In addition, the system should also maintain the consistency between surfaces in consecutive steps to preserve the user’s mental map.

R5. Path customization. Users should be able to interact with the system to customize paths focusing on certain variables of interest or a specific type of relationship described in **R3**. The customized path should answer questions such as *how does a feature evolve over time, what are the surfaces related to the selected one over time, and what are the most different surfaces from the selected one over time?*

R6. Path animation. The system should produce an animation that concisely describes the comparison path with isosurface rendering. The animation should answer the questions in **R1** with more detail: it not only indicates *which* isosurfaces are (dis)similar, but also describes *how* they are (dis)similar by providing the isosurfaces in the spatial view. For effective visual comparison, the isosurfaces at each animation frame may need to be rendered in different styles, allowing the more important ones to be observed clearly.

4 MATRIX OF ISOSURFACE SIMILARITY MAPS

4.1 Overview

Based on our analysis of requirements in Section 3, we design MISM as the main visual representation and interface for exploring the underlying time-varying multivariate data set. As shown in Figure 1, our interface consists of four components: *parameter panel*, *matrix view*, *isovalue view*, and *isosurface view*. The matrix view is the core component of our system. It displays MISM in multiple modes to allows the observation of both an overview of the data set (**R1**) and different types of relationships among variables (**R3**). The matrix of maps allows the relationships to be understood both at map-level (volumes) and cell-level (isosurfaces). With visual hints provided by the MISM display, users can easily identify the variable/isovalues of interest and interact with the matrix view to navigate the related variables/isovalues. The key function we introduce for navigating MISM is *path recommendation* (**R4** and **R5**): users can select two or more similarity maps in MISM and we recommend a traversal path that maximizes certain criteria (e.g., the smoothest, or most surprising path). In conjunction with the isovalue and isosurface views, animating the path in MISM and displaying the corresponding isosurfaces enable users to conveniently tour through the underlying data set (**R6**).

As described in the Appendix, we define three types of similarity maps in this paper: the *self-similarity map* (SSM) of the isosurfaces from individual volumes, the *temporal similarity map* (TSM) of the isosurfaces from the same variable at different time steps, and the *variable similarity map* (VSM) of the isosurfaces from different variables at the same time step. The definition of similarity follows that of the ISM approach [5]. The matrix view provides four display modes to explore different types of relationships: (1) the *single-variable* mode that displays a single SSM to explore a single volume; (2) the *single-pair* mode that displays a VSM or TSM and the two corresponding SSMs for comparison of two volumes (e.g., Figure 12 (a)); (3) the *all-pairs* mode that displays all SSMs and VSMs for selected variables at a given time step (e.g., Figure 6 (a)); and (4) the *evolution* mode that displays all SSMs and VSMs for selected variables and time steps through clustering and filtering (e.g., Figure 1 (b)). Only the evolution mode displays multiple time steps, where the horizontal direction represents the selected variables and all related variable pairs, and the vertical direction represents representative time steps. SSMs of the selected variables and all related VSMs are displayed but not TSMs.

Interaction and typical workflow. We propose a set of interactive functions for navigating the huge matrix of similarity maps (see Figure 2). We cluster temporal sequences and group variables to reduce the exploration space (**R2**). We also highlight representative cells in a similarity map to attract user attention and guide the exploration (e.g., Figure 5). A typical workflow to explore a data set using MISM is as follows. Users will start by examining the matrix view in the evolution mode, which shows the SSMs and VSMs of all the representative time steps and variables. This provides users an overall understanding of the data set and guides them to discover the time steps and variables of interest. Then, users can simply click and drag to form a path that reveals the temporal development of the selected variables and time steps. Users can further create waypoints to edit the traversal path so that the connections to additional features can be discovered. In the evolution mode, users can easily click an SSM to enter the single-variable mode or click a VSM to enter the single-pair mode for detailed exploration. They can specify a time step to explore the relationship among all variables in the all-pairs mode as well. Finally, they can create paths to investigate the isovales in these detailed modes.

Next, we introduce temporal clustering and variable grouping (Sec-

tion 4.2), path recommendation and graph construction (Section 4.3), and silhouette-based rendering and animation (Section 4.4).

4.2 Clustering and Grouping

We reduce the size of MISIM along the temporal and variable dimensions through clustering temporal sequences and grouping variables, respectively. This not only allows users to quickly identify the representative time steps or variables for exploration, but also produces more compact paths and animations for efficient knowledge discovery. We define a similarity measure between volumes using TSMs or VSMs and use it to derive the similarity between two time steps or two variables. We apply affinity propagation [8] to cluster the temporal sequences based on the similarity measure. Affinity propagation automatically determines the number of clusters, which naturally reflects how frequently the variables change along time in a data set. For variables, we use k -means clustering to group the variables into the desired number of groups. In the following, we only describe our similarity measure for temporal clustering, as the detail for variable grouping is the same.

Similarity measure. We evaluate the similarity of two volumes based on the similarities among their isosurfaces. Two volumes are considered to be similar if for each isosurface in one volume, a similar isosurface can be identified in the other volume. Our similarity measure is analogous to the mean of closest point distances defined on two curves by considering each volume as a curve and each isosurface of the volume as a point on the curve. Formally, the similarity of an isosurface \mathbf{S}' and a volume \mathbf{V} is defined as

$$\mathcal{S}(\mathbf{S}', \mathbf{V}) = \min_{\mathbf{S} \in \mathbf{V}} \mathcal{S}(\mathbf{S}', \mathbf{S}), \quad (1)$$

where the similarity between two isosurfaces, $\mathcal{S}(\mathbf{S}', \mathbf{S})$, can be looked up from TSMs or VSMs. The similarity of two volumes \mathbf{V}_i and \mathbf{V}_j is defined as the weighted average of the similarity of each isosurface in one volume to the other volume, i.e.,

$$\mathcal{S}(\mathbf{V}_i, \mathbf{V}_j) = \frac{\sum_{\mathbf{S}_p \in \mathbf{V}_i} w_{\mathbf{S}_p} \mathcal{S}(\mathbf{S}_p, \mathbf{V}_j)}{\sum_{\mathbf{S}_p \in \mathbf{V}_i} w_{\mathbf{S}_p}} + \frac{\sum_{\mathbf{S}_q \in \mathbf{V}_j} w_{\mathbf{S}_q} \mathcal{S}(\mathbf{S}_q, \mathbf{V}_i)}{\sum_{\mathbf{S}_q \in \mathbf{V}_j} w_{\mathbf{S}_q}}, \quad (2)$$

where the weight $w_{\mathbf{S}_p}$ for an isosurface \mathbf{S}_p is derived based on its representativeness ranking, which is used for selecting the most representative isovales (Section 1 in the Appendix). Note that the similarity between any pair of volumes is in $[0, 1]$, which has the same range as the similarity between any pair of isosurfaces.

Temporal clustering. Temporal clustering evaluates the combined similarity between two time steps as the summation of similarities calculated for each of the variables using Equation 2. Affinity propagation is applied to cluster the time steps based on their combined similarities. For each cluster, the clustering algorithm identifies one exemplar, which is considered as the representative time step for that cluster.

4.3 Path Recommendation

MISM has a two-tiered structure: *maps* at the coarse matrix level and *cells* at the fine map level. The map-level captures the overall relationships among volumes, while the cell-level allows the detailed relationships among isosurfaces to be discovered. Given two user-specified maps (cells) as the start and end points, *path recommendation* identifies a series of intermediate maps (cells) to construct a path for traversal. We create an animation for the generated path, showing the isosurfaces corresponding to a map (cell) at one animation frame. Users can adjust the weights of different terms to define the desired path. They may visit the maps or cells to discover affinity relationships or compare distinct features. The differences between frames can be minimized or maximized for generating a smooth animation or increasing information gain. They can specify the start and end points of a path in the matrix view, and drag any point along the path to add waypoints. We identify a path that minimizes the total cost between the user-specified points.

We introduce the following types of paths: (1) *map-level paths* for the evolution mode, (2) *cell-level paths* for all the four modes, and (3) *variable traversal paths* for only the all-pairs mode. All three types of paths are built with similar constructions. In the following, we discuss map-level paths in detail. For the other two, we only explain their differences with respect to map-level paths.

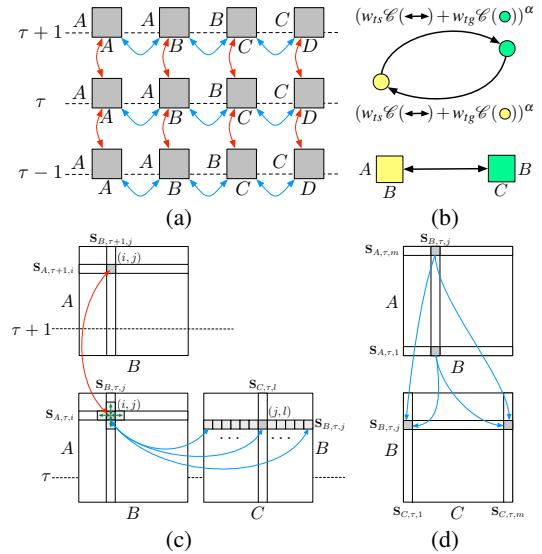


Fig. 3. Top: map-level graph construction. (a) transitions in map-level graphs. (b) edge weights corresponding to two transitions between maps \mathbf{M}_{AB} and \mathbf{M}_{BC} . Bottom: transitions between cells. (c) transitions in cell-level graphs. (d) transitions in variable traversal graphs.

We denote a similarity map between two variables A and B at time step τ as $\mathbf{M}_{AB,\tau}$ and their corresponding volume as $\mathbf{V}_{A,\tau}$ and $\mathbf{V}_{B,\tau}$. Note that $\mathbf{M}_{AB,\tau}$ can refer to an SSM when $A = B$ or a VSM when $A \neq B$. However, it does not refer to a TSM (which is not displayed in the matrix view) as the indices do not specify two different time steps. We use $\mathbf{M}_{AB,\tau}[i, j]$ to denote a cell in $\mathbf{M}_{AB,\tau}$ at the i -th row and the j -th column, and $\mathbf{S}_{A,\tau,i}$ and $\mathbf{S}_{B,\tau,j}$ to denote the two isosurfaces corresponding to the i -th row and j -th column, respectively. We may ignore τ in the notation for simplicity when time is not discussed. We define the difference $\mathcal{D}(,)$ between two elements as $1 - \mathcal{S}(,)$.

Map-level path. We identify the map-level path between two similarity maps as the shortest path in a map-level graph. A *map-level graph* is a directed graph whose nodes are all SSMs and VSMs and whose edges are all the possible transitions from one map to another, as illustrated in Figure 3 (a). Specifically, we consider two kinds of transitions: (1) *variable transition* between similarity maps at the same time step sharing at least one common variable, as shown by the blue arrows; and (2) *temporal transition* between similarity maps at neighboring time steps, as shown by the red arrows.

Weighing the edges appropriately is critical to obtaining a path that shows the desired features. In our approach, the weight of an edge corresponding to a transition is a linear combination of the *target cost* \mathcal{C}_{tg} and the *transition cost* \mathcal{C}_{ts} raised to a user-specified power exponent α : $(w_{tg} \mathcal{C}_{tg} + w_{ts} \mathcal{C}_{ts})^\alpha$, where w_{tg} and w_{ts} are the weights of \mathcal{C}_{tg} and \mathcal{C}_{ts} , respectively. We include α to further distinguish the edge costs, so that the shortest path is less likely to end up with a path with a larger average cost but a smaller number of edges. Consider the transition $\mathbf{M}_{AB} \rightarrow \mathbf{M}_{BC}$ (from the yellow node to the green node) in Figure 3 (b). The target cost $\mathcal{C}(\mathbf{M}_{BC})$ is the difference $\mathcal{D}(\mathbf{V}_B, \mathbf{V}_C)$ between the two corresponding volumes \mathbf{V}_B and \mathbf{V}_C . When the target cost is weighed positively, the shortest path is more likely to visit the maps corresponding to variables with similar structures by minimizing the cost. When the target cost is weighed negatively (we use $1 - \mathcal{C}(\mathbf{M}_{BC})$), the resulting path tends to visit the variables with different behaviors for the most surprise. The transition cost is the penalty when we replace the isosurfaces to display for \mathbf{M}_{AB} to those for \mathbf{M}_{BC} . It can be weighed positively in the linear combination, so that the shortest path minimizes the transition cost and maintains a smooth animation between neighboring frames; otherwise, it can be weighed negatively using $1 - \mathcal{C}(\mathbf{M}_{AB} \rightarrow \mathbf{M}_{BC})$ to visit the maps with diverse information.

A variable transition from a VSM $\mathbf{M}_{AB,\tau}$ to another $\mathbf{M}_{BC,\tau}$ indicates that the focus of analysis shifts from one pair of variables (A and

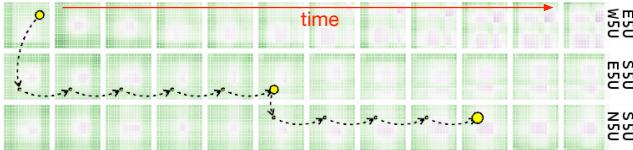


Fig. 4. The path with the same start point and parameters as Figure 1 (b). The example shown here uses the original cell-level graph instead of the reduced weight graph.

B) to another pair (B and C). After the transition, we replace the isosurfaces of A with the ones of C in the isosurface view. Therefore, we weigh the corresponding edge by the difference of their volumes, i.e., $\mathcal{D}(\mathbf{V}_{A,\tau}, \mathbf{V}_{C,\tau})$. The edge weight for transitions between a VSM and an SSM can be defined similarly by letting $A = B$. For coherence, variable transitions should take place only if necessary. To avoid the paths from jumping back and forth among the variable pairs, we include a sufficiently large *jumping cost* in the variable transition.

A temporal transition between two VSMs $\mathbf{M}_{AB,\tau}$ and $\mathbf{M}_{AB,\tau+1}$ describes the evolution of the two variables A and B over time. The corresponding edge weight is defined as $(\mathcal{D}(\mathbf{V}_{A,\tau}, \mathbf{V}_{A,\tau+1}) + \mathcal{D}(\mathbf{V}_{B,\tau}, \mathbf{V}_{B,\tau+1})) / 2$.

Cell-level path. Similar to the map-level path, the cell-level path is the shortest path between two cells in a directed *cell-level graph*. Each node in the cell-level graph represents one cell and each edge represents one transition. For the cell-level graph, we consider three kinds of transitions, as shown in Figure 3 (c): (1) *variable transition* (the blue arrows) between cells in two similarity maps at the same time step sharing at least one common variable; (2) *temporal transition* (the red arrow) between cells in two similarity maps corresponding to the same pair of variables at different time steps; and (3) *isovalue transition* (the green arrows) between cells in the same similarity map. Similarly, the edge weight is a linear combination of transition cost and target cost. The target cost is given by the difference of the two isosurfaces corresponding to the target cell. The transition cost is defined by the difference between the isosurfaces that are not shared by the two cells. For example, for a variable transition $\mathbf{M}_{AB,\tau}[i,j] \rightarrow \mathbf{M}_{BC,\tau}[j,k]$, the transition cost is the difference between $S_{A,\tau,i}$ and $S_{C,\tau,k}$, i.e., $\mathcal{D}(S_{A,\tau,i}, S_{C,\tau,k})$.

We further provide two options for users to improve the cell-level path. First, we allow users to specify the exact endpoint as cells, or simply select a map that contains the endpoint. When a map is specified, the cell with minimum cost is automatically identified on the map. In this case, we create an edge from each cell in the map to a dummy node with zero weight and compute the shortest path. Second, we design a scheme to further reduce the average cost between cells. The shortest path often tends to minimize the total cost along the path by reducing the number of cells. It is possible to identify a path with a minimum number of undesired transitions. Therefore, finding a path with the smallest average weight between nodes is more desired in our scenario. Since this problem is NP-hard, we provide an approximate solution here. We start with the shortest path identified by Dijkstra's algorithm and compute the average weight \bar{w} along the path. Then, we create a cell-level reduced weight graph. This graph contains the same edges as the original cell-level graph, but the weight of each edge is reduced by \bar{w} . The edges with negative weights are eliminated by assigning a small weight (0.001 in our implementation) to them. Note that the edges with weights smaller than \bar{w} have a minimum cost now. This allows a longer path with more desirable transitions to be identified as the shortest path in the reduced weight graph.

Figure 1 (b) and Figure 4 show two cell-level paths produced by the reduced weight and original graphs, respectively. In Figure 4, we can see that without reducing the weights, the path only visits one cell in each map at similar locations, even under the setting of $w_{tg} = 0.1$ and $w_{ts} = -1$, which suppose to show more distinguished isosurfaces. In contrast, with the reduced weight graph, the path usually visits multiple cells in each graph to increase the information gained, as shown in Figure 1 (b). For a quantitative comparison, we evaluate the average

weight of the two paths in the original graph. The path produced by the reduced weight graph can effectively bring down the average weight from 0.227 to 0.172.

Variable traversal path in all-pairs mode. A variables traversal path visits a chain of variable pairs that covers every variable for a complete understanding of relationships among variables at a given time step. For example, this chain can be $\langle A, B \rangle \rightarrow \langle B, C \rangle \rightarrow \langle C, D \rangle \rightarrow \langle D, A \rangle$ for a data set with four variables. Given a user-specified cell (and therefore a pair of variables to start), we determine the order of variables to visit using a greedy strategy: at each step, we select the unvisited variable that is most similar to the current variable.

The variable traversal path is identified in a *variable traversal graph* which is constructed similarly as the cell-level graph with variable transitions and isovalue transitions. The definitions of transition cost and target cost are the same as the ones described for the cell-level graph. The only difference is that variable transitions are constrained to take place between only boundary cells following the order of the maps being visited. For example, consider the two pairs of variables $\langle A, B \rangle$ and $\langle B, C \rangle$ to be neighbors along the path, as shown in Figure 3 (d). We only allow transitions from the boundary cells $\mathbf{M}_{AB,\tau}[1,j]$ and $\mathbf{M}_{AB,\tau}[m,j]$ in map $\mathbf{M}_{AB,\tau}$ to the boundary cells $\mathbf{M}_{BC,\tau}[j,1]$ and $\mathbf{M}_{BC,\tau}[j,m]$ in map $\mathbf{M}_{BC,\tau}$ that correspond to the same isosurface $S_{B,\tau,j}$. This forces the path to enter and leave a similarity map at the boundaries to build a more complete path visiting a large portion of value range for each variable.

4.4 Surface Rendering and Animation

We render the isosurfaces using an approach based on per-pixel link lists [41] for real-time order independent transparency. The rendering is performed in two passes. The first pass generates a link list for each pixel to store the fragments that are rendered to that pixel. The second pass sorts fragments at the same pixel according to their depth and blends the fragments in the sorted order. Inspired by the silhouette-based rendering [7], we only render the silhouettes for the less important isosurfaces to reduce visual clutter. We combine these two approaches by checking whether a fragment belongs to the silhouettes before storing the fragment in the link lists. The fragments of the less important isosurfaces that are not related to the silhouettes are discarded to reduce the storing and sorting effort. Using our approach, rendering some surfaces as silhouettes not only allows clearer observation but also has a lower cost than rendering all surfaces in full.

An animation will be produced when a path is specified in the matrix view. As shown in Figure 1, with the evolution mode of the matrix view, we also display the isovalue view to show the evolution of isovales corresponding to the rendered isosurfaces. We synchronize the animation across the matrix, isovalue, and isosurface views. The path is displayed in both the matrix and isovalue views with the current animation step highlighted. We display a timeline in the isovalue view. Users can drag it to play a specific animation step. They can also drag the highlighted isovales in the isovalue view to create additional waypoints and modify the path. When multiple variables are involved in an animation, the isosurfaces are colored according to their corresponding variables, as indicated by the legends in the isosurface view (e.g., Figure 1 (d)). We create an animated transition for each variable transition on the path to gradually transit from one variable to another. When a single variable is used in an animation, we color the isosurfaces according to their representativeness. The representativeness ranking is indicated by the legends shown in the isosurface view, with the most representative one displayed on the left (e.g., Figure 13 (a)). By default, we render the complete isosurface for the most representative one and the silhouettes for the others. Users may switch the focus to render any representative isosurface as a complete one.

5 RESULTS AND DISCUSSION

5.1 Data Sets and Timing Performance

We used several time-varying multivariate data sets with different sizes and characteristics for our experiment. The data sets and timing performance are reported in Table 1. The timing was collected using a CPU/GPU cluster (8 Quantum TXR231-1000R servers each with a dual Intel E5-2650 12-core CPU @ 2.2 GHz, 128 GB RAM, and 4 NVIDIA

data set	dimension $x \times y \times z \times v \times t$	distance field $x \times y \times z$	SSM	extraction	simplification	reduction	TSM	VSM	total
atmospheric	$360 \times 360 \times 112 \times 32 \times 13$	$45 \times 45 \times 14$	26.85s	8.31s	94.20s	7.56 \times	0.0953s	0.0941s	24.1h
climate	$360 \times 66 \times 27 \times 2 \times 1200$	$120 \times 33 \times 9$	4.85s	0.18s	1.82s	9.39 \times	0.0810s	0.0778s	10.9h
combustion	$480 \times 720 \times 120 \times 5 \times 122$	$60 \times 90 \times 15$	50.66s	8.11s	56.44s	10.72 \times	0.2290s	0.2275s	23.5h
ionization	$600 \times 248 \times 248 \times 8 \times 199$	$75 \times 31 \times 31$	25.99s	4.34s	14.40s	12.42 \times	0.1936s	0.1919s	20.1h

Table 1. The average timing results for computing an SSM, an isosurface and its simplified version, a TSM, and a VSM using a CPU/GPU, and the total computation time for the entire data set using the CPU/GPU cluster.

data set	DS-MC		ORI-APP		DS-APP	
	MSE	SU	MSE	SU	MSE	SU
atmospheric	0.0006	23.11 \times	0.2191	1.06 \times	0.1637	110.67 \times
climate	0.0046	13.13 \times	0.0060	1.02 \times	0.0052	30.50 \times
combustion	0.0001	48.86 \times	0.0584	1.23 \times	0.0474	182.48 \times
ionization	0.0134	43.69 \times	0.0831	1.27 \times	0.0489	299.26 \times

Table 2. Mean squared error (MSE) and speed-up (SU) for computing SSMs using different configurations.

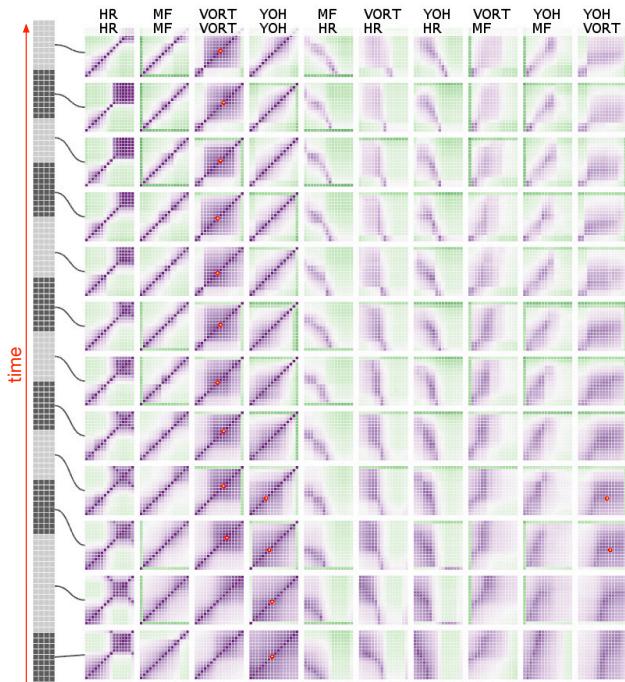


Fig. 5. Overview of the evolution mode of the matrix view with four variables (HR, MF, VORT, and YOH) of the combustion data set.

TITAN X GPUs). The computation of the distance fields and similarity maps (SSMs, TSMs, and VSMs) was performed in the GPU, while isosurface extraction and surface simplification were performed in the CPU. The timing for SSM (256×256) includes the time to compute distance fields, joint histogram, and mutual information. The time costs for surface extraction and simplification mainly depend on the complexity of the surface and the number of triangles involved. The timing for TSM or VSM (16×16) includes the time to compute joint histogram and mutual information. The time cost to compute a TSM or VSM mainly depends on the size of distance fields. In our experiment, since the computation of similarity maps or isosurfaces consists of largely independent tasks, we use 8 CPU cores and 8 GPUs of the CPU/GPU cluster to speed up the computation in parallel, reducing the total computation time for each data set to a day or less.

We further examine the accuracy and speed-up of our similarity map computation. Our solution introduces two major variants to speed up the computation: using approximation (APP) to replace GPU-accelerated marching cubes (MC) for surface generation and using downsampled (DS) distance fields instead of the ones with original (ORI) resolution. We evaluate the performance of different combinations of these two

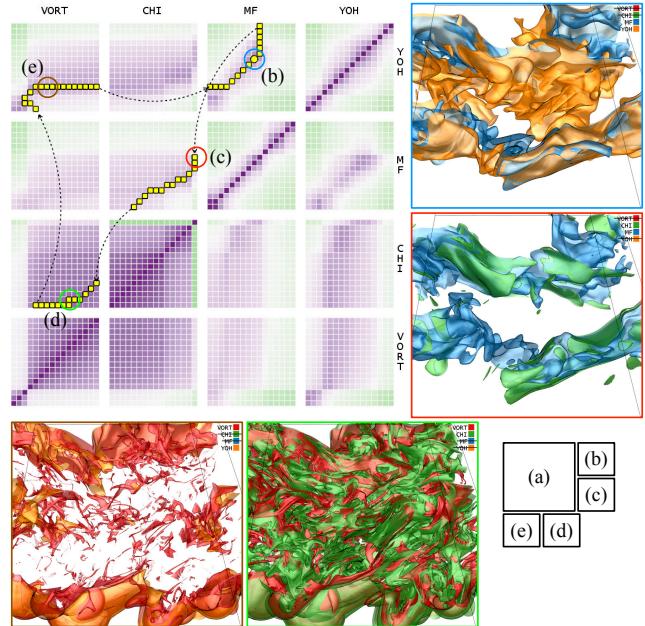


Fig. 6. Variable traversal paths using time step 111 of the combustion data set. (a) shows a path with a positive target weight ($w_{sg} = 1$, $w_{ts} = 0.1$, and $\alpha = 4$). (b) to (e) show four animation frames corresponding to the path in (a).

variants (i.e., DS-MC, ORI-APP, and DS-APP) by comparing them to the ground truth (ORI-MC). The accuracy is evaluated by the mean of squared error (MSE). Giving two sets of values $\mathbf{X} = \{x_1, \dots, x_n\}$ and $\mathbf{X}' = \{x'_1, \dots, x'_n\}$, MSE is computed as

$$\text{MSE}(\mathbf{X}, \mathbf{X}') = \frac{1}{n} \sum_{i=1}^n |x_i - x'_i|^2. \quad (3)$$

For the climate data set, we collect the results using ten sample time steps, and for the others, we use five sample time steps. Note that this already contains 1.3 million to 2 million similarity values per data set so that we can draw a reliable conclusion. In Table 2, we can see that using downsampled distance fields produces mostly the same similarity values, with the largest MSE being 0.0134 for the ionization data set. Surprisingly, we find that DS-APP outperforms ORI-APP in terms of MSE for all the data sets, probably because distance fields of the original resolution may be more sensitive to the errors introduced by the approximation. DS-APP produces reasonable results with MSE smaller than 0.05 except for the atmospheric data set (0.1637). In terms of efficiency, DS itself speeds up the computation by more than 10 \times . APP does not lead to large speed boost with ORI, but it can further increase the performance of DS, resulting in 30 \times to 300 \times speed-up.

5.2 Case Studies with Expert Evaluation

Combustion data set. This data set comes from direct numerical simulation of temporally evolving turbulent non-premixed flames. The simulation generates five variables: heat release (HR), mixture fraction (MF), vorticity (VORT), mass fraction of hydroxyl radical (YOH), and scalar dissipation rate (CHI). Initially, the thin planar scalar layers are placed in the middle of the computational domain. The layers evolve into complex isosurfaces as they interact with the surrounding turbulence. Combustion reactions occur within the scalar layers.

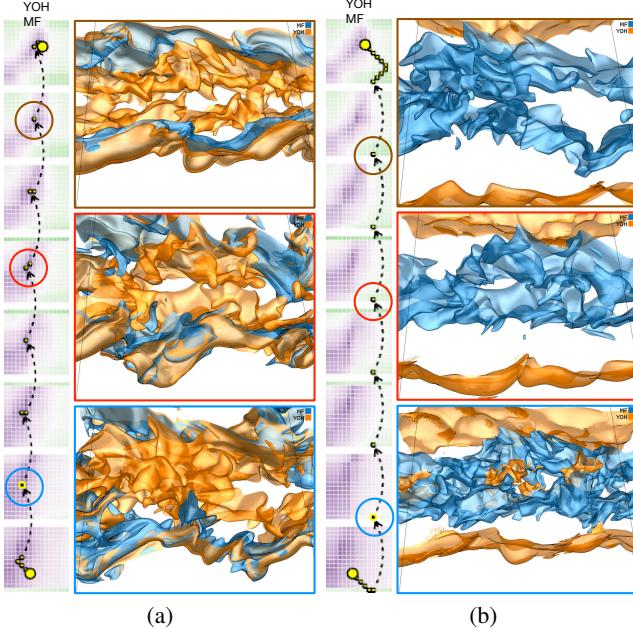


Fig. 7. Cell-level paths in YOH-MF VSMs using the combustion data set. (a) shows a path with a positive target weight ($w_{tg} = 1$, $w_{ts} = 0$, and $\alpha = 4$) and three corresponding animation frames. (b) shows a path with a negative target weight ($w_{tg} = -1$, $w_{ts} = 0$, and $\alpha = 4$) and three corresponding animation frames.

We invited a domain expert with 20 years of experience in turbulent combustion modeling and simulation to explore this data set using MISM. The exploration started with the evolution mode of the matrix view with four variables (HR, MF, VORT, and YOH), as shown in Figure 5. The expert commented that *the similarity maps at different representative time steps are able to summarize the overall data characteristics*. For each individual variable, at earlier time steps, since the scalar layers are planar, all scalars are well correlated, and most regions of the SSMs in the bottom rows are purple. As time goes on, these scalar layers are distorted by turbulence. Meanwhile, the scalar layers are thickened since small-scale turbulence enhances scalar mixing. As the scalar layers are thickened and interact with turbulence, the isosurfaces corresponding to lower and higher values become dissimilar. Therefore, the positive correlations (purple cells) only appear along the diagonals of the SSMs in the top rows. The expert further pointed out that *the matrix view demonstrates the relationship development of YOH and MF over time*. YOH and MF are perfectly correlated, when combustion reaction time scales are much shorter than turbulence time scales. This condition is closely satisfied at later time steps, as the YOH-MF VSMs in the top rows show a strong correlation near the diagonal regions. On the contrary, at earlier time steps, the fast chemistry condition is not satisfied. This leads to no evident correlation of these two variables at earlier time steps. The SSMs in the bottom rows confirm this by exhibiting more uniform correlations. In addition, the expert found that *the relationship between VORT and MF is interesting*, since the VORT-MF VSMs gradually develop into a pattern with a purple upper-left triangle and a green lower-right triangle. The expert stated that *the physical origin of this trend could not be fully understood at this moment, but it was likely related to the effects of HR on turbulence*. This demonstrates the potential of our work for revealing physics behind the data set.

Then, the expert analyzed the relationships among four variables (VORT, CHI, MF, and YOH) at time step 111 using the all-pairs mode of the matrix view. Time step 111 is the first time step of the last temporal cluster. The all-pairs mode facilitates better comparison of the VSMs. In Figure 6 (a), the expert found that *the pattern of the YOH-MF VSM is different from others*. As already observed in the evolution mode, the diagonal of YOH-MF VSM is mostly purple, indicating

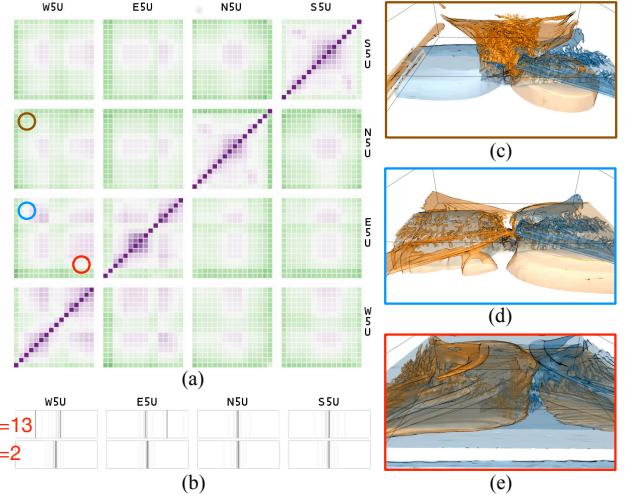


Fig. 8. Comparison of four wind directions with friction speed 5 m/s using the atmospheric ensemble data set. (a) shows the all-pairs mode of the matrix view at time step 13. (b) shows the isovalue view at time steps 2 (bottom) and 13 (top). (c) to (e) show two isosurfaces of variable U corresponding to (c) small absolute wind speeds of the west and north, (d) small absolute wind speeds of the west and east, and (e) large absolute wind speeds of the west and east.

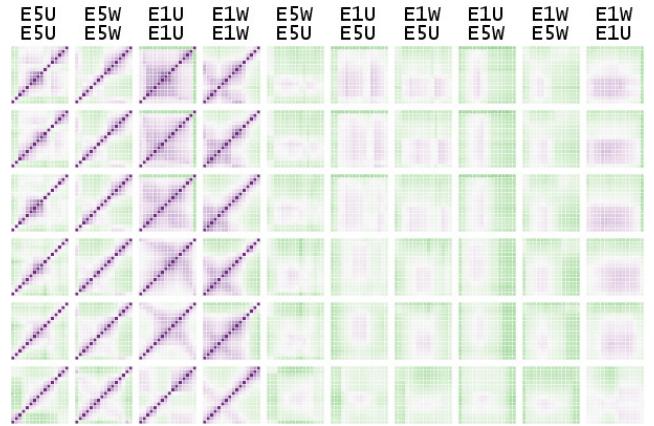


Fig. 9. Overview of the evolution mode of the matrix view with two variables (U and W) produced under two wind speeds (5 m/s and 1 m/s) using the atmospheric ensemble data set.

strong correspondence between the value ranges in YOH and MF. The only exception is the top-right corner corresponding to larger isovales of YOH and MF, which is mostly green. This means that *the larger isovales of these two variables are less relevant to each other*. The expert explained that the value of YOH is the highest in the reaction zone and decreases toward the fuel and oxidizer sides, which correspond to the lower and higher values of MF, respectively. Therefore, there is no evident correlation of YOH and MF at higher values of each variable. The expert selected a purple cell in this VSM, as highlighted in the blue circle, to generate a variable traversal path. A positive target weight ($w_{tg} = 1$) was used to identify a path demonstrating similar isosurfaces for each of the variable pairs: YOH and MF, MF and CHI, CHI and VORT, and VORT and YOH. A small transition weight ($w_{ts} = 0.1$) was used to maintain the minimum smoothness between the consecutive animation frames. Figure 6 (b) to (e) show four animation frames presenting the similar isosurfaces. The similarity of isosurfaces seen in Figure 6 (c) indicates that *the variations of CHI on isosurfaces of MF occur mainly at large scales, which happens mostly for thin scalar layers with high values of CHI*. The regions where the CHI and MF isosurfaces collapse correspond to the thin layers with steep scalar gradients (high values of CHI). In Figure 6 (d), while the isosurfaces

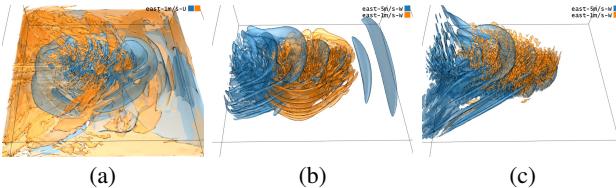


Fig. 10. Isosurface rendering of (a) E1U at time steps 2 and 13, (b) E5W and E1W at time step 2, and (c) E5W and E1W at time step 13.

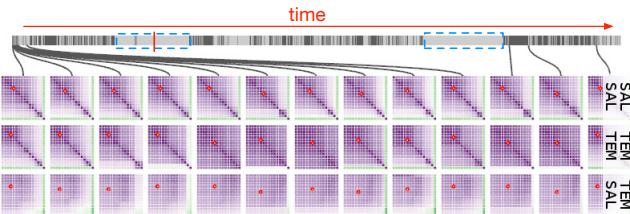


Fig. 11. Overview of the entire climate data set of 1200 time steps.

of CHI and VORT are complex, the expert pointed out that *low CHI regions correspond to high VORT regions, because strong vertical motions tend to mix scalars and reduce their gradients*.

Finally, the expert switched to the evolution mode of the matrix view to further investigate the relationship between YOH and MF. Two purple cells were specified as the starting and ending points in the YOH-MF VSMs. Two paths were generated for comparison: one with a positive target weight ($w_{tg} = 1$) and the other with a negative target weight ($w_{tg} = -1$). The transition weight w_{ts} was set to 0 to eliminate the influence of transition cost, and α was set to 4 to allow more edges with smaller cost. Figure 7 (a) shows the path with the positive target weight and the isosurfaces in three animation steps along the path. The path passes only the purple regions near the centers of the VSMs, corresponding to the middle range isovalues of YOH and MF. In the isosurface view, it can be seen that *the blue isosurfaces occupy similar regions as the corresponding orange ones at all the three time steps*, due to the correlation of YOH and MF. In contrast, the path with the negative target weight goes through the green regions at the bottom-right corners of the VSMs, even if it starts and ends at purple cells, as shown in Figure 7 (b). This path mostly relates to the high MF isosurfaces and the low YOH ones. In the isosurface view, it can be seen that *the blue and orange isosurfaces are disjoint in the space*. As the wrinkling history of isosurfaces due to turbulence is different for isosurfaces present in different regions, the dissimilarity is expected.

Atmospheric data set. A set of 3D numerical experiments is performed to explore the mechanical impacts of Graciosa Island on the dry boundary layer evolution under varying wind speed and surface friction setups. The simulations are conducted using the Cloud Model 1 (CM1 release 19 [6]). The full data set has 32 ensemble simulations which are comprised of four basic wind speeds (1, 5, 10, and 20 m/s), four wind directions (west, east, north, and south), and two bottom boundary conditions (friction/no friction). Two variables, the U and W components of the wind flow, are studied. To distinguish the variables under different simulation setups, we add the initials of the wind directions and speeds to the names of variables. For example, “E5U” stands for variable U with the east wind direction and speed 5 m/s.

We invited a domain expert with ten years of experience in atmospheric sciences to evaluate MISM using this data set. Since friction effects are more pronounced in lower wind speeds, the expert focused on the 5 m/s case to visualize and analyze flow variability. The expert first investigated the evolution of variable U under different wind directions with friction and 5m/s wind setup. Figure 8 (a) shows the matrix view in the all-pairs mode at time step 13. The SSMs at the diagonal show that *the west and east have similar patterns, and so do the SSMs of the north and south*. In Figure 8 (b), the isovalue view shows a similar relationship. The representative isovalues distribute similarly for the four wind directions at time step 2 (bottom). In contrast, at time

step 13 (top), the distributions form two different patterns. The expert explained that this means from time step 2 to 13, *the west and east show much larger flow variability compared to the north and south, which is attributed to the west-east orientation of the island located at the center of the simulation domain*. The representative isovalues spread out in different directions for the west and east, but the representatives of the south and north are still similar to those in the first time step. Figure 8 (c) shows that the isosurface of the north (orange) has quite a different shape from that of the west (blue). Unlike the blue isosurface, the orange one covers both the upper and lower portions of the domain, although the isovalue is small. In contrast, the isosurfaces corresponding to small absolute wind speeds of the west (blue) and east (orange) have similar shapes, as shown in Figure 8 (d). Both isosurfaces reside in the lower portion of the domain, although their orientations are opposite. By selecting the isovalues representing larger wind speeds (as highlighted by the red circle in Figure 8 (a)), the expert found that the corresponding isosurfaces are also similar, as shown in Figure 8 (e). These two isosurfaces occupy mostly the entire domain as the wind gets stronger. The expert commented that *the east wind direction seems to be more interesting. It demonstrates more variation on the surfaces and in the isovalue view, due to a stronger island effect on incoming flow in this configuration*.

Then, the expert compared four variables within the east wind direction: E5U, E5W, E1U, and E1W (i.e., variables U and W with wind speeds 5 m/s and 1 m/s). Only one temporal cluster is produced by affinity propagation, indicating *a consistent pattern over time*. Therefore, the expert used all individual time steps to explore their temporal relationships. Figure 9 shows the overview of the evolution mode of the matrix view. In general, the expert found that the patterns of matrices in different rows are similar, which confirms the temporal clustering results. Note that, for E5U and E1U, the purple regions in the middle value range are growing over time. The expert commented that *this is indicative of the non-steady simulation behavior*. By displaying the isosurfaces, the expert found that the purple region corresponds to surfaces covering the entire domain. This trend indicates that, *at the later time steps, the surfaces of different value ranges of U tend to cover more space and become more similar to each other*. Figure 10 (a) shows two isosurfaces of a larger isovalue of E1U at time step 2 (blue) and time step 13 (orange), respectively. It can be seen that the orange surface covers almost the entire domain, but the blue one is mainly located at the center of the domain.

The temporal development of variable W with wind speeds 5 m/s and 1 m/s seems to be slightly different. While the pattern in column E1W is more stable, the purple region in E5W is shrinking over time. This indicates that, unlike variable U, *the isosurfaces of E5W appear to be similar at the beginning, but tend to be more distinguishable over time*. Although the development trend is different, the expert found that *E5W and E1W become more similar in the larger value range*, since the upper-right corners of the E1W-E5W TSMs gradually transit from green to light green. At time step 2, the isosurfaces corresponding to a larger isovalue of E5W (blue) and E1W (orange) occupy different spatial regions, as shown in Figure 10 (b); but at time step 13, the two isosurfaces reside in the same region, as shown in Figure 10 (c). However, the similarity is still not high, since the distance between different layers of the blue surfaces is much larger.

5.3 Additional Case Studies

Climate data set. This data set was generated from a simulation of salinity (SAL) and temperature (TEM) in the equatorial region from 20°S to 20°N for a period of 100 years. This data set contains 1200 time steps (one month per time step), from which temporal clustering identifies 13 representatives, as shown in Figure 11. From the representatives, we can see that SAL exhibits a more stable pattern than TEM. Interestingly, although our temporal clustering does not consider the order of time steps, nine months in the first year are selected as representatives. Overall, the clusters are consistent with the monthly weather change in each year. But we can also notice that for two long periods of time, most of the time steps are placed in the same cluster, as highlighted in the blue rectangles.

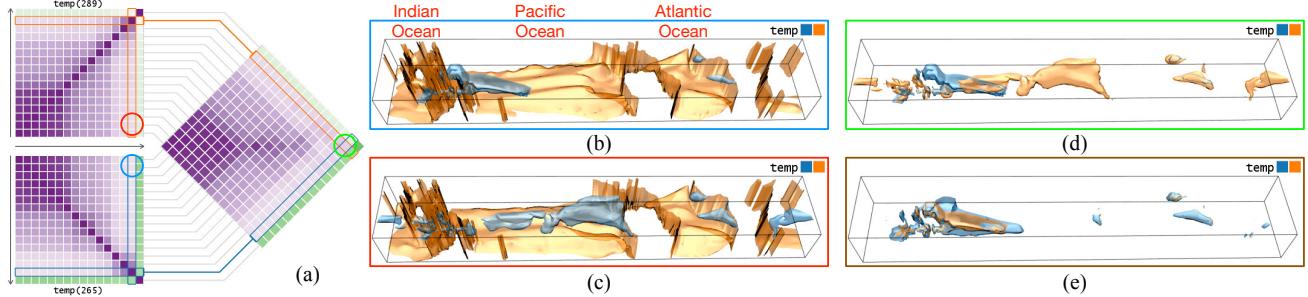


Fig. 12. The single-pair mode of the matrix view using the climate data set. (a) shows the TSM comparing time steps 265 and 289. Time step 289 is associated with the El Niño condition. (b) and (c) show a high TEM isosurface (blue) and a low TEM one (orange) at time steps 265 and 289, respectively. (d) shows two high TEM isosurfaces at time steps 265 (blue) and 289 (orange). (e) shows two high TEM isosurfaces at time steps 241 (blue) and 265 (orange).

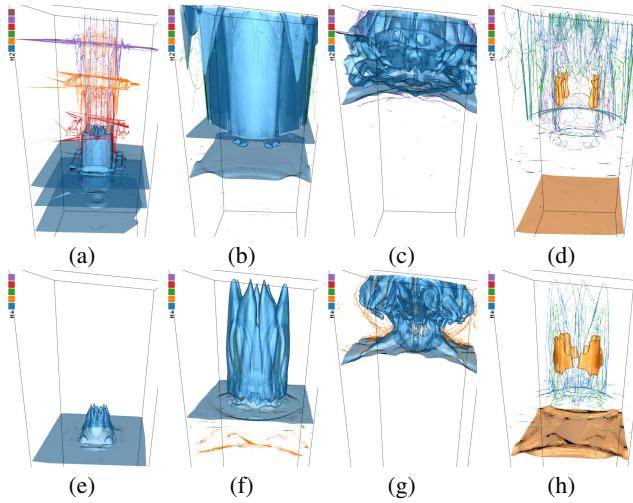


Fig. 13. Isosurfaces corresponding to map-level paths using the ionization data set. (a) to (c) show the representative isosurfaces for H₂ at time steps 31, 98, and 181, respectively. (d) shows the same isosurfaces as (b) but focusing on the orange isosurface. (e) to (g) show the same as (a) to (c) but for H⁺.

Then, we examine the temperature change with the El Niño condition. We select time step 289 (with El Niño), as highlighted by the red bar in Figure 11, and compare it with time step 265 (without). Time step 265 corresponds to the same month as time step 289, but two years before. The single-pair mode of the matrix view displays the two SSMs on the left and the corresponding TSM on the right, as shown in Figure 12 (a). The arrows indicate TEM from low to high. We find that *both SSMs share the same pattern*: the low TEM isosurfaces are very similar to each other, indicated by a dark purple block; the medium TEM isosurfaces are moderately similar to each other and the low TEM ones but not that similar to the high TEM ones; and the high TEM isosurfaces are similar to each other but not that similar to the low and medium TEM ones. Similar patterns can be observed in the TSM as well, except that *the high TEM isosurfaces do not share high similarity*.

By selecting the cell highlighted in the blue circle, the isosurface view renders a high TEM isosurface (blue) and a low TEM one (orange) at time step 265 for comparison, as shown in Figure 12 (b). The high TEM isosurface is associated with a small region close to the ocean surface and the low TEM one covers the entire oceans at the bottom. By selecting the cell highlighted in the red circle, two isosurfaces at time step 289 are displayed, as shown in Figure 12 (c). We can see that *the low TEM isosurface is mostly the same as the corresponding one at time step 265; but the high TEM isosurface is quite different, with the central and eastern equatorial Pacific Ocean covered, confirming the existence of El Niño*. By selecting the cell highlighted in the green circle, we compare the two high TEM isosurfaces, as shown in Figure 12 (d).

The cells related to these two isosurfaces are highlighted by rectangles in the corresponding colors, as shown in Figure 12 (a). Figure 12 (e) shows two high TEM isosurfaces at time steps 265 and 241 (which is exactly two years before 265). Clearly, *without the El Niño condition, the high TEM regions are similar in the same month across years*.

Ionization data set. For this data set, we find that all SSMs show a similar pattern: the purple cells mostly concentrate along the diagonals leaving other regions in green. This indicates *the strongly localized pattern for isosurfaces at different value ranges*. We use map-level paths to trace the movement of the ionization front for each individual variable and show the corresponding isosurfaces in Figure 13. Figure 13 (a) to (c) correspond to H₂ at three representative time steps at the beginning, middle, and ending stages, respectively. Obviously, *the most representative isosurface shown in blue moves from left to right and demonstrates different shape characteristics*. In Figure 13 (a), we can also observe three other representative isosurfaces with a similar shape as the most representative one, displayed as the orange, red, and purple silhouettes. However, in Figure 13 (b), the other representative isosurfaces are occluded by the most representative one shown in blue. Therefore, we shift the focus to the orange one, as shown in Figure 13 (d). The blue, green, and purple silhouettes indicate that *the isosurfaces form multiple layers of the same structure*. Figure 13 (c) shows the isosurfaces of H₂ at the ending stage. Figure 13 (e) to (h) show a similar process for H⁺ but with a different shape of the ionization front.

6 CONCLUDING REMARKS

To provide a convenient mechanism for users to browse through a large time-varying multivariate or ensemble data set in the form of isosurfaces, we present MISM, a visual interface that organizes a huge number of ISMs for navigation and exploration. We design effective solutions to achieve both *computational scalability* (by computing a massive number of self, temporal, and variable similarity maps using GPU-accelerated approximation) and *visualization scalability* (by presenting MISM at different levels of detail via clustering, grouping, and filtering schemes). To the best of our knowledge, both scalabilities have not been demonstrated previously in this context. With advanced features such as path recommendation and surface comparison, MISM is the first of its kind that supports flexible relationship exploration and examination among isosurfaces extracted from the multifaceted data, providing the capability that goes beyond a traditional animation playback could offer.

In the future, we would like to investigate an alternative semantic abstraction of MISM. This would sprout the opportunity to further simplify the relationships to a much condensed form for understanding. We would also like to create a static visualization variant that summarizes the isosurfaces in path animation, so that the relationships among the isosurfaces can be understood using a single image.

ACKNOWLEDGEMENTS

This research was supported in part by NSF grants IIS-1456763, IIS-1455886, and CNS-1629914, DOE grant DE-AC02-06CH11357 and SciDAC program, and NVIDIA GPU grant program. The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

- [1] H. Akiba and K.-L. Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proceedings of Eurographics - IEEE VGTC Symposium on Visualization*, pages 115–122, 2007.
- [2] H. Akiba, C. Wang, and K.-L. Ma. AniViz: A template-based animation tool for volume visualization. *IEEE Computer Graphics and Applications*, 30(5):61–71, 2010.
- [3] D. Angus, A. Smith, and J. Wiles. Conceptual recurrence plots: Revealing patterns in human discourse. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):988–997, 2012.
- [4] A. Biswas, S. Dutta, H.-W. Shen, and J. Woodring. An information-aware framework for exploring multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2683–2692, 2013.
- [5] S. Bruckner and T. Möller. Isosurface similarity maps. *Computer Graphics Forum*, 29(3):773–782, 2010.
- [6] G. H. Bryan and J. M. Fritsch. A benchmark simulation for moist nonhydrostatic numerical models. *Monthly Weather Review*, 130(12):2917–2928, 2002.
- [7] I. Demir, J. Kehrer, and R. Westermann. Screen-space silhouettes for visualizing ensembles of 3D isosurfaces. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 204–208, 2016.
- [8] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.
- [9] S. Frey, F. Sadlo, and T. Ertl. Visualization of temporal similarity in field data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2023–2032, 2012.
- [10] M. Glatter, J. Huang, S. Ahern, J. Daniel, and A. Lu. Visualizing temporal patterns in large multivariate data using textual pattern matching. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1467–1474, 2008.
- [11] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.
- [12] Y. Gu and C. Wang. TransGraph: Hierarchical exploration of transition relationships in time-varying volumetric data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2015–2024, 2011.
- [13] Y. Gu and C. Wang. iTree: Exploring time-varying data using indexable tree. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 137–144, 2013.
- [14] H. Guo, H. Xiao, and X. Yuan. Scalable multivariate volume visualization and analysis based on dimension projection and parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1397–1410, 2012.
- [15] M. Haidacher, S. Bruckner, and E. Gröller. Volume analysis using multimodal surface similarity. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1969–1978, 2011.
- [16] H. Hoppe. Progressive meshes. In *Proceedings of ACM SIGGRAPH Conference*, pages 99–108, 1996.
- [17] M. Imre, J. Tao, and C. Wang. Efficient GPU-accelerated computation of isosurface similarity maps. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 180–184, 2017.
- [18] H. Jänicke, M. Böttiger, and G. Scheuermann. Brushing of attribute clouds for the visualization of multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1459–1466, 2008.
- [19] T. J. Jankun-Kelly and K.-L. Ma. A study of transfer function generation for time-varying volume data. In *Proceedings of Eurographics - IEEE TCVG Workshop on Volume Graphics*, pages 51–66, 2001.
- [20] A. Joshi and P. Rheingans. Illustration-inspired techniques for visualizing time-varying data. In *Proceedings of IEEE Visualization Conference*, pages 679–686, 2005.
- [21] T. Karras. Maximizing parallelism in the construction of BVHs, octrees, and k-d trees. In *Proceedings of ACM SIGGRAPH/Eurographics Conference on High-Performance Graphics*, pages 33–37, 2012.
- [22] J. Kehrer and H. Hauser. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):495–513, 2013.
- [23] O. D. Lampe, C. Correa, K.-L. Ma, and H. Hauser. Curve-centric volume reformation for comparative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1235–1242, 2009.
- [24] X. Liu and H.-W. Shen. Association analysis for visual exploration of multivariate scientific data sets. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):955–964, 2016.
- [25] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of ACM SIGGRAPH Conference*, pages 163–169, 1987.
- [26] A. Lu and H.-W. Shen. Interactive storyboard for overall time-varying data visualization. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 143–150, 2008.
- [27] K.-L. Ma. Visualizing time-varying volume data. *IEEE Computing in Science and Engineering*, 5(2):34–42, 2003.
- [28] M. M. Malik, C. Heinzl, and M. E. Gröller. Comparative visualization for parameter studies of dataset series. *IEEE Transactions on Visualization and Computer Graphics*, 16(5):829–840, 2010.
- [29] N. Marwan, J. Kurths, and P. Saparin. Generalised recurrence plot analysis for spatial data. *Physics Letters A*, 360(4-5):545–551, 2007.
- [30] 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(5):917–924, 2006.
- [31] J. Schmidt, C. Heinzl, and S. Bruckner. VAICo: Visual analysis for image comparison. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2090–2099, 2013.
- [32] D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. Interactive comparison of scalar fields based on largest contours with applications to flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1475–1482, 2008.
- [33] H.-W. Shen, L.-J. Chiang, and K.-L. Ma. A fast volume rendering algorithm for time-varying fields using a time-space partitioning (TSP) tree. In *Proceedings of IEEE Visualization Conference*, pages 371–377, 1999.
- [34] K. Stockinger, J. Shalf, K. Wu, and E. W. Bethel. Query-driven visualization of large data sets. In *Proceedings of IEEE Visualization Conference*, pages 167–174, 2005.
- [35] V. Verma and A. T. Pang. Comparative flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):609–624, 2004.
- [36] C. Wang and J. Tao. Graphs in scientific visualization: A survey. *Computer Graphics Forum*, 36(1):263–287, 2017.
- [37] C. Wang, H. Yu, R. W. Grout, K.-L. Ma, and J. H. Chen. Analyzing information transfer in time-varying multivariate data. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 99–106, 2011.
- [38] C. Wang, H. Yu, and K.-L. Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, 2008.
- [39] J. Woodring and H.-W. Shen. Multi-variate, time-varying, and comparative visualization with contextual cues. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):909–916, 2006.
- [40] J. Woodring, C. Wang, and H.-W. Shen. High dimensional direct rendering of time-varying volumetric data. In *Proceedings of IEEE Visualization Conference*, pages 417–424, 2003.
- [41] J. C. Yang, J. Hensley, H. Grün, and N. Thibieroz. Real-time concurrent linked list construction on the GPU. *Computer Graphics Forum*, 29(4):1297–1304, 2010.

APPENDIX

1 ISOSURFACE SIMILARITY MAPS

To fulfill the design requirement **R3**, we create three types of similarity maps to capture different types of relationships among the isosurfaces.

Self-similarity map. Bruckner and Möller [5] introduced the *isosurface similarity map* (ISM) to evaluate the similarities among isosurfaces generated from a volumetric data set. For a given n sampled isovalue, the ISM is a symmetric $n \times n$ matrix \mathbf{M} where each cell $\mathbf{M}[i, j]$ records the similarity value between isosurfaces S_i and S_j . For every surface S_i , we compute a distance field \mathbf{D}_i which records the distance from each voxel in the volume to the closest point on S_i . The similarity between S_i and S_j is then computed as the mutual information between distance fields \mathbf{D}_i and \mathbf{D}_j , using the joint histogram of their distance distributions. We refer to such an ISM as the *self-similarity map* (SSM) as it is only concerned with isosurfaces generated from a single volume.

For a time-varying multivariate data set of t time steps and v variables, we need to compute $t \times v$ SSMs. Since each SSM may take hours to compute given a reasonable number of sampled isosurfaces (e.g., we use $n = 256$), we apply the GPU-accelerated approximation solution [17] to speed up the computation while maintaining the fidelity of the resulting SSM. The major steps are listed in Figure 2 of the paper. Specifically, we consider the following:

- First, instead of generating the actual isosurface, we record voxels that contain the isosurface and use that information as an approximation for distance field computation. The error of using this approximation to compute the distance from a voxel to a surface is bounded by $\sqrt{3}/2$ (i.e., half the diagonal length of the grid cell). This error is acceptable as downsampled distance fields are often suggested to compute the similarity map.
- Second, for distance field computation, we need to search for each voxel in the volume, the closest point on the isosurface (in our case, the closest voxel containing the isosurface). To reduce the search space, we use the downsampled distance field. We leverage the bounding volume hierarchy (BVH)-trees to speed up the search process. Specifically, we use bounding boxes for a lower construction time and apply Karras's algorithm [21] to build BVH-trees in parallel. This is essential for performance gain since a BVH-tree needs to be built for every isosurface approximation. Since the BVH-trees store approximation points from the original volume, this guarantees that small features can still be preserved for accurate distance field computation.
- Third, the most time-consuming component in the SSM computation is the construction of the joint histogram of two distance fields since it has to be performed for each pair of isovalue. We therefore compute the downsampled distance field and use it for subsequent computation. It has been shown that the resolution can be considerably reduced (e.g., $8 \times$ in each dimension) without substantial changes in the resulting similarity map [5].

For each SSM, we identify m representative isovalue using a greedy strategy that recursively partitions the set of isovalue and selects representative ones based on a priority queue scheme [5]. Typically, m is much smaller than n , e.g., we use $m = 16$. We leverage a GPU to compute isosurface approximations, downsampled distance fields, and joint histograms. With that, we are able to reduce the average time to compute a single SSM from hours to a few minutes using a single GPU.

Isosurface computation. After the representative isovalue are identified for each volume, we implement a GPU version of the marching cubes algorithm [25] to compute the actual isosurfaces and simplify the resulting surfaces [16]. Both steps are performed on the CPU during a one-time preprocessing stage. The simplification can significantly reduce the space for storing isosurfaces without sacrificing much of the surface quality. This is important as we need to generate a total of $m \times t \times v$ representative isosurfaces for the entire time-varying multivariate data set. Using simplified isosurfaces alleviates I/O burden, making it possible for us to achieve interactive visualization and comparison of a number of isosurfaces while still maintaining good visual quality.

As an example, we can see in Figure 1 that our approximation (b) yields a very similar SSM as the exact marching cubes algorithm does in (a). In (d), we show the isosurfaces corresponding to the four most representative isovalue selected in (b). Comparing (a) and (b), although there are slight shifting of the selected isovalue, those are barely visible in the surface rendering, as shown in the difference image (c). Such minor differences are neglectable as the main goal of selecting the most salient isovalue and depicting their surfaces is still achieved. With our CUDA-accelerated solution, the time cost to compute the SSM is improved by a factor of $43 \times$ (72.30s vs. 3,161.70s). The difference image shown in (e) indicates that surface simplification yields a close rendering result with a much less number of triangles (3,519,952 vs. 235,030, or nearly $15 \times$ reduction).

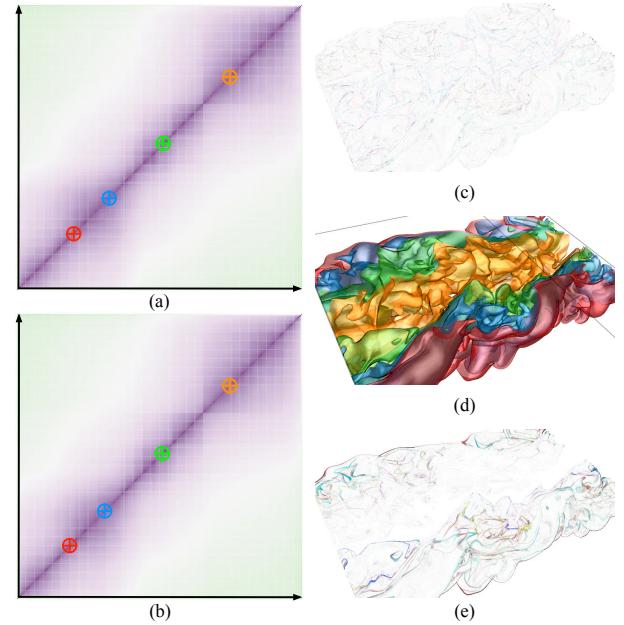


Fig. 1. Comparing SSMs generated using (a) the actual isosurfaces against (b) our approximation with the combustion data set (a time step of the MF variable). The marching cubes algorithm is used to extract the surfaces shown in (d) for the four most representative isovalue selected in (b). (c) shows the difference between the surfaces selected in (a) and (b), and (e) shows the difference between the surfaces selected in (b) and their simplified version.

Temporal and variable similarity maps. Our goal is to investigate not only the SSMs corresponding to individual volumes, but also the similarity maps between the volumes of different time steps and the volumes of different variables. Therefore, we also compute two other kinds of isosurface similarity: *temporal similarity* and *variable similarity*, extending the work of multimodal surface similarity [15] from only a single pair to all pairs of variables, and from steady to time-varying data. These similarities are computed between representative isosurfaces from the same variable at different time steps (temporal similarity), and from different variables at the same time step (variable similarity). We call the resulting similarity maps *temporal similarity maps* (TSMs) and *variable similarity maps* (VSMs), respectively. A TSM or VSM is not symmetric anymore as the representative isovalue come from different volumes. Each TSM or VSM is much smaller in size compared with an SSM ($m \times m$ vs. $n \times n$), but the numbers of TSMs (i.e., $v \times t(t-1)/2$) and VSMs (i.e., $t \times v(v-1)/2$) are much larger than that of SSMs (i.e., $t \times v$) as we need to compute TSMs and VSMs for different pairs of time steps and variables.