

Visual Analytics for Enabling Extreme-Scale Scientific Discovery

Han-Wei Shen, Teng-Yok Lee, Abon Chaudhuri, and Boonthanome Nouanesengsey

Department of Computer Science and Engineering
The Ohio State University

E-mail: hwshen@cse.ohio-state.edu

Abstract.

The growing power of supercomputers has substantially enhanced our capability to simulate complex problems at greater fidelity, leading to high-impact scientific and engineering breakthroughs. Over the years, visualization has become instrumental for analyzing data generated from numerical simulations in many science disciplines. To fully understand such vast amounts of data, scientists need scalable solutions that can perform detailed data analysis at different levels of detail. In this paper, we present techniques that address two difficulties faced by computational scientists. The first is deciding what data are the most essential for analysis, given that only a small fraction can be retained. The second is transforming these data into visual representations that rapidly convey the most insight to the viewer. The techniques presented in the paper utilize information theory, fractal analysis, and time-varying data analysis to facilitate effective data summarization.

1. Introduction

Visualization has become an essential means to analyze data generated from a variety of applications. Numerical simulations for fluid flows, climate modeling, astrophysics, and astronomy are some examples that routinely produce large amounts of data. Visualization, as a fast-maturing discipline, can now offer many standard techniques to satisfy the user's basic data analysis needs. Well-known techniques such as isosurfaces, particle tracing, and direct volume rendering are now being used by many scientists on a regular basis. In the meantime, the complexity of data generated from multiphysics, multiscale simulations is increasing at an astonishing rate, a rate that is often on par with the data growth rate. Not only are those datasets multivariate time-varying, containing scalar, vector, and tensor quantities, they are also defined on meshes of highly complex geometry. In addition, features often manifest in a wide range of spatial and temporal scales, and a complete development of a phenomenon often involves multiple stages.

As scientists eagerly anticipate the benefits of extreme-scale computing, roadblocks to science discovery at scale threaten to impede their progress. The disparity between computing and storing information and the gap between stored information and the understanding derived from it become the main barriers to success. Considering that data movement is becoming the greatest limiting factor for extreme-scale computing, it is crucial to address two major difficulties faced by computational scientists. The first is deciding what data are the most essential for analysis, given that only a small fraction can be retained. The second is transforming these data into visual representations that rapidly convey the most insight to the viewer.

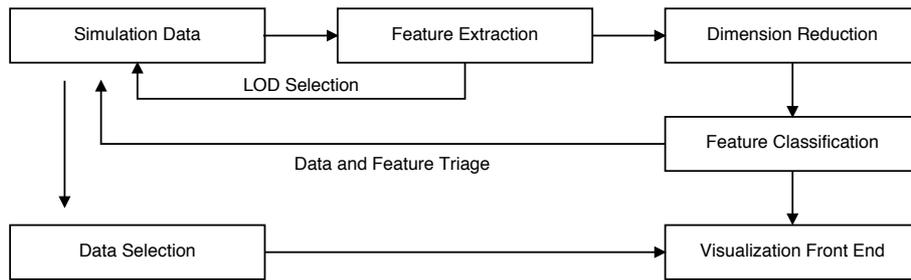


Figure 1. Analytics-driven visualization pipeline.

The traditional workflow for scientific data visualization mostly treats the entire simulation output as sequences of flat files from which visualizations are produced. As we move into the era of exascale computing, the “save the data first, think about it later” mentality needs to be completely changed and replaced with more aggressive data reduction and data triage whenever possible. To answer the questions of what data are the most essential for analysis, summaries of data need to be created to indicate the existence and nature of features—topological, geometrical, statistical, and time-varying—and to quantify the features’ degree of interest. Effective data summaries can assist one in performing the following tasks more effectively:

- Data reduction: the resolution of data can be adjusted based on the existence of features and its degree of interest.
- Data triage: data can be prioritized and organized based on the features’ characteristics. The storage of data can also be optimized accordingly.
- Data characterization and classification: feature descriptors, and data indices can be created so that offline user queries can be performed efficiently.

With effective data reduction, the amount of data that is required to be moved in and out of the disk will be significantly reduced. With data triage, scientists can start data analysis by looking at the most salient portion of the data. With data characterization and classification, scientists can obtain a quick overview about the distribution of features, which can facilitate more effective offline query.

Figure 1 shows a simple workflow that highlights the data reduction, triage, and classification process. As data are output from the simulation, selected feature extractors are invoked to capture the essence of the data. The result of feature extraction can be looped back to the data to assist data reduction; for example, data blocks that do not contain interesting features can be discarded or stored in a lower resolution. The extracted features can also be used to construct feature descriptors to assist feature indexing and data classification. Since the lengths of the feature descriptors can be long, it may be necessary to invoke dimension reduction techniques to project the descriptors to the most salient principal dimensions. This is to both reduce the storage cost and the computation time for feature classification, and to make it possible for the user to visualize the distribution of features in 2D display. At the offline analysis stage, the user can query the relevant data based on the features of interest, taking the available computation and storage resources into account.

Generally speaking, besides domain-specific requirements, the degree of interest for scientific datasets and their associated features can be classified based on their geometrical and topological complexity, statistical complexity, and time-dependent data complexity. In this paper, we present three visual analytics techniques that can effectively summarize scientific data based on these factors. These techniques allow us to perform effective data reduction, feature selection, and summarization. In the following, we present each of the techniques in detail.

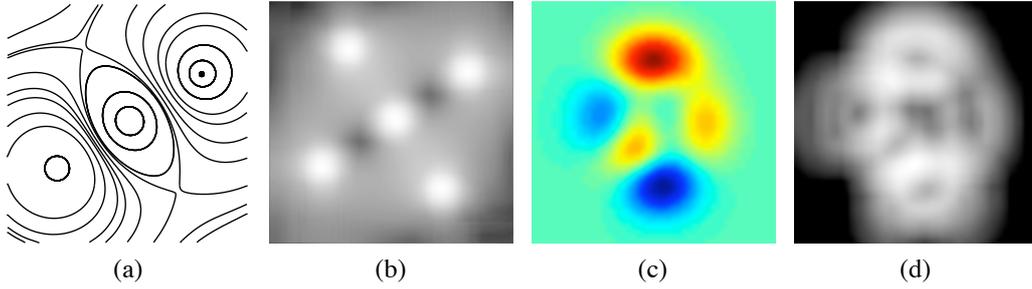


Figure 2. (a) A vector field, (c) a scalar field and their respective entropy measures in (b) and (d), respectively. In (c), blue represents low values, and red represents high values. The image intensity in (b) and (d) represents the value of local entropy.

2. Measuring Statistical Complexity via Information Theory

In this section, we describe how information theory can be used to measure the statistical complexity of data and assist in data reduction. Information theory provides a complete theoretical framework to quantify the uncertainty, or the information content, of a random variable. Formally, let X be a discrete random variable with alphabet \mathcal{X} and probability mass function $p(x)$, $x \in \mathcal{X}$. Shannon’s *entropy* of X is defined as

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log p(x). \quad (1)$$

The entropy is a measure of the average uncertainty in X . It is the number of bits on average required to describe the random variable. An important property of the entropy is that $H(X)$ is convex and reaches its maximum when $p(x)$ is equal for all x .

To apply Shannon’s entropy, we can treat a scientific dataset as a discrete random variable where each data point in the domain carries a value as an outcome. The probability mass function $p(x)$ of the random variable can be estimated using histograms. One issue when using histograms, however, is that the result can be sensitive to the level of discretization, that is, the number of bins. This problem can be remedied by using various probability density estimation techniques [1–3].

Figure 2 shows examples of entropy computed from a two-dimensional vector field and a scalar field. At each grid point we estimate the entropy in the local neighborhood. One can see that areas near critical points in the vector field, and regions that have higher variation in the scalar field exhibit higher entropy. Depending on the application, certain derived quantities such as velocity magnitude, gradients of scalars and velocities may also be used to compute entropy and help understand the characteristics of data.

2.1. Data reduction via information theory

Common strategies for data reduction in visual data analysis, such as importance-based and multiresolution rendering, will likely continue to play an important role in the future. The major benefit of such techniques is that they reduce data to a size suitable for interactive exploration. Traditionally, the amount of information loss is not considered when choosing what level of detail to use. Consequently, the scientist often has to go back and forth between different data resolutions to find a balance between quality and speed. Since data movement is the major limiting factor for analyzing data at scale, such trial-and-error methods will incur a large amount of unnecessary data movement, which is cost-prohibitive.

When analyzing a scientific dataset, we are often interested in looking at regions that exhibit a higher degree of “surprise,” or unpredictability. From this point of view, the importance of data can be linked to whether the data have an equal probability to exhibit all the possible outcomes. This property can be fully captured by Shannon’s entropy, which is also a measure of the information content in the data. Take a vector field as an example, regions near critical points will have vectors with an equal probability

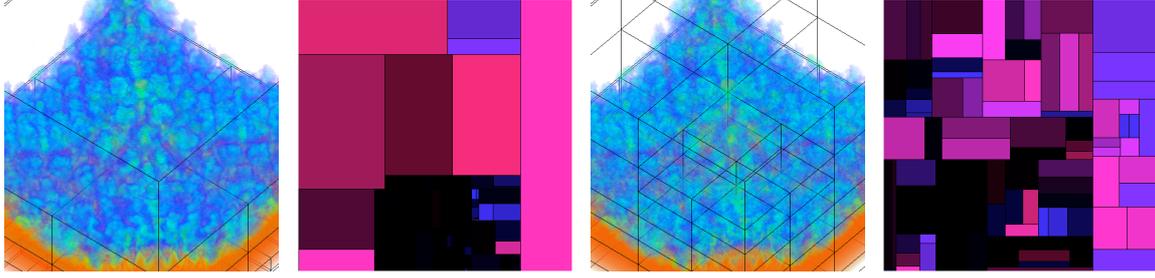


Figure 3. LOD adjustment on the Richtmyer-Meshkov instability dataset. From left to right: the LOD selected based on the mean square error (MSE) and its corresponding LOD map; the LOD after adjustment and its corresponding LOD map. The number of blocks remains the same while more details of the data are revealed after the adjustment.

pointing to all directions, an indication of high entropy. Similar principles can be applied to a scalar field, where regions that are less homogeneous exhibit higher entropy.

Based on this idea, we use Shannon’s entropy to measure the data importance, hereafter denoted as I . To measure the entropy, we need to model the dataset as a discrete random variable where each data point in the domain carries a value as an outcome. The probability mass function $p(x)$ of the random variable can be estimated from the histogram of data values, and the entropy is computed from Equation 1.

The concept of entropy can be extended to study the relationship between data at two different resolutions. The underlying idea is to measure how much information loss there will be if we use data at a lower resolution to represent the original data. To implement this idea, we will model a multiresolution data hierarchy as a sequence of random variables $X_i, i \in [0..k - 1]$ in a k -level hierarchy, where X_0 represents the data at the original resolution, and X_i is the lower-resolution data at the i_{th} level. To measure how much information from the original data is lost by using the low-resolution data X_i , we can use the concept of *conditional entropy*, $H(X_0|X_i)$. The conditional entropy models the remaining uncertainty in the random variable X_0 after the value of the random variable X_i is known. Compared with the conventional metrics such as squared error, the conditional entropy places a special emphasis on analyzing the correlation between the original and the lower resolution data, which has been shown previously to play a major role in assessing the data distortion rate [4–6].

To strike a balance between storage saving using data of lower resolution, and minimizing information losses, we can evaluate the overall quality of an LOD using Shannon’s entropy measure. First, we define the probability of a multiresolution data block b_i as

$$p_i = \frac{I_i \cdot E_i}{\sum_{i=1}^M I_i \cdot E_i}, \quad (2)$$

where I_i and E_i are the importance and error of b_i , respectively, and M is the total number of blocks in the multiresolution data hierarchy. The summation is taken over all data blocks, and the division is required to make all probabilities add up to unity. The entropy of an LOD then follows the definition in Equation 1. Shannon’s entropy measure is a convex function and reaches its maximum when all probabilities are equal. Because the probability of a data block is defined as the multiplication of its importance and error, if a region has a higher importance value, then choosing a block of lower error (i.e., higher data resolution) will give a higher entropy score. Conversely, if a region is of smaller importance, then we can afford to increase its error by choosing a lower-resolution data block. Equation 2 thus complies with our intuition: we want to use high-resolution data for regions that are important and use lower resolution data otherwise.

The above information measure for LOD can be used to create a novel user interface too, so that the user can more easily navigate through the immense data and parameter spaces. One example of user interface is the *LOD map* [6], a visual interface for navigating multiresolution volume data. In the

interface, we calculate the error and importance of multiresolution data blocks, and provide a quantitative LOD quality measurement using entropy. The LOD map is constructed by mapping the LOD quality (error and importance of data blocks) to a 2D treemap (color, size, and opacity of rectangles) from the information visualization literature. Through the prominent visual features, the LOD map effectively shows the tradeoff between computation cost and information gain, as well as the completeness of the visualization results. Figure 3 shows an example of LOD adjustment using the LOD map. The LOD map after adjustment gives a more balanced result in terms of the size and color of rectangles, which indicates a better LOD quality. In this manner, the users are informed not only of what they have seen (i.e., visible data blocks) but also what they have not yet seen (i.e., occluded data blocks). This interface facilitates LOD selection and comparison and hence increases the effectiveness and productivity of visual analysis.

3. Measuring Geometric Complexity via Fractal Dimensions

Fluid flow simulations play an important role in many scientific disciplines. In order to visualize vector data generated from these simulations, one popular method is to display flow lines such as streamlines or pathlines computed from numerical integration. The primary challenge of visualizing flow lines is to reveal regions with salient flow patterns of interest to the scientists. Previously, researchers tackled this problem by controlling the placement of flowline seeds [7–10]. For extreme-scale applications, applying those methods is difficult and cost-prohibitive since it often involves multiple iterations and refinements.

In this section, we present a method that can classify a large number of streamlines based on their geometric complexity. Feature descriptors can be constructed from this complexity measure and allow for more effective visual analysis and data selection. We adopt a metric called *box counting ratio* (originally used in computation of fractal dimension) as an effective tool for geometric analysis of streamlines. By definition, it measures the compactness of a geometry at a given scale without having to know its exact shape. It works because most of the interesting flow features like vortex and turbulence are represented by streamlines with complex curvilinear paths confined within a region, leading to a high value of the metric. On the other hand, fairly straight, or winding but sparse, streamlines correspond to a lower value of this metric. The nature of the metric suggests that its use is not limited to the detection of any particular type of feature. It can be useful when an exact definition of a feature is not available. Moreover, streamlines often pass through multiple regions of interest and contain more than one feature, possibly of different sizes. Box counting ratio can be employed to detect features of varying sizes. Also, its close relation to fractal dimension lends specific interpretations to its value. For instance, a value close to 1 indicates that the geometry of the streamline is close to linear. A value close to 2 means the streamline densely fills up a 2D subspace of the 3D space and resembles a surface. Or, it may also represent a relatively sparse 3D subspace. A value close to 3 indicates that the streamline forms a really dense 3D structure (Figure 4).

3.1. Box counting ratio

The formal definition of fractal dimension is as follows: If a fractal F is measured at a scale of measurement δ , then the measurement, denoted by $N_\delta(F)$, and the unit of measurement have the relation $N_\delta(F) \simeq \delta^{-D}$, where the constant D is known as fractal dimension [11]. Ideally, D is a logarithmic rate that can be estimated by the gradient of a line through $\log N_\delta(F)$ plotted against $-\log \delta$ (Equation 3). However, a practical way to estimate this is box counting dimension, which estimates $N_\delta(F)$ by counting the number of sets of diameter at most δ , which can cover F . For an object embedded in a 3D rectilinear grid, such as a streamline, the number of grid cubes that the object intersects with provides an estimate of $N_\delta(F)$ for that grid resolution. Now, if this measurement of box counting is taken at two different scales, say δ_1 and δ_2 , that are sufficiently small, their logarithmic ratio, or *box counting ratio*, denoted by B , should be close to the limiting value for that object. A streamline is not expected to be a true fractal, yet its box counting ratio is a good measure of its complexity. The count of grid cubes of a fixed scale that intersects with the streamline indicates how spread out or irregular the geometry of the streamline is at that scale. This value is first computed in a grid of cubes having length δ and then

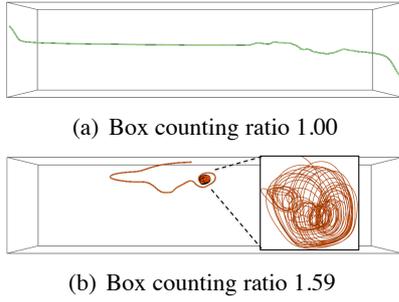


Figure 4. Box counting ratio 3D streamlines: **(a)** a relatively straight streamline with box counting ratio 1; **(b)** a streamline with a complex feature and higher box counting ratio; **Inset** zoomed-in view of the complex segment.

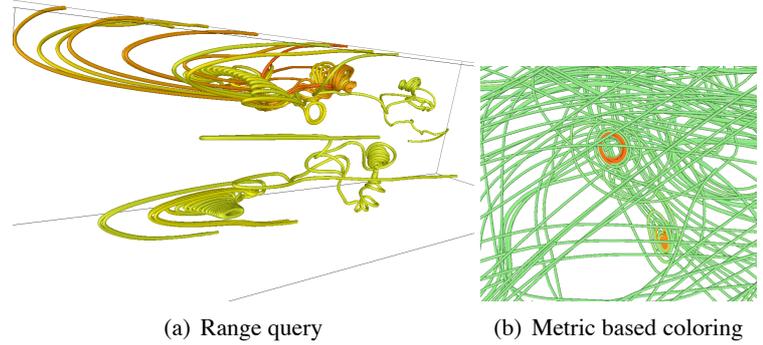


Figure 5. Simple uses of box counting ratio: **(a)** range query on streamlines with high box counting ratio reveals the complex structures in Solar plume dataset; **(b)** Each streamline from an MJO simulation dataset is divided into a series of fixed-length segments. Computation of box counting ratio of each segment followed by value-based coloring reveals the vortex-like structures.

in a grid of cubes with length $2 \times \delta$. The logarithmic rate (Equation 4) of these two counts gives the box counting ratio of the streamline. A streamline is not expected to be a true fractal, yet its box counting ratio is a good measure of its complexity. The count of grid cubes of a fixed scale which intersects with the streamline indicates how spread out or irregular the geometry of the streamline is at that scale. This value is first computed in a grid of cubes having length δ and then in a grid of cubes with length $2 \times \delta$. The logarithmic rate (Equation 4) of these two counts gives the *box counting ratio* of the streamline. should be close to the limiting value for that object. *box counting ratio*, denoted by B .

$$D = \lim_{\delta \rightarrow 0} \frac{\log(N_{\delta}(F))}{-\log \delta} \quad (3)$$

$$B = \log \frac{N_{\delta_1}(F)}{N_{\delta_2}(F)} \quad (4)$$

3.2. Geometry-based visual analysis

In addition to showing many simple but useful applications of this metric, such as range query and transfer function (Figure 5), we have exploited its power to develop a complete visual analytic framework. The goal of this framework is to quantify the geometric complexity of a large collection of streamlines and present the potential features to the user, encouraging interactive exploration. The framework comprises the following steps: compute box counting ratio of each streamline segment, apply a high-pass filter to retain only the potentially important segments, construct high-dimensional feature vectors from each retained segment, and project the feature vectors to a 2D space using a suitable dimensionality reduction technique. The projection provides a visual space, which meaningfully places streamlines and allows the user to select regions from the space to locate corresponding streamlines on a linked display. This allows the user to navigate the vector field without necessarily going through the 3D space, which is less friendly to interaction. Each step of the framework is summarized below.

Feature localization and segmentation: Features of different sizes can be located at different positions along the trajectory of a streamline. Since it is not computationally tractable to examine streamline segments of all possible lengths from all possible points, we have employed a hierarchical two-way subdivision to recursively partition a streamline into two smaller halves until a segment shorter than a threshold is found. Box counting ratio of each segment thus formed is computed for different grid resolution pairs. Hence, for each streamline we have obtained a *feature map* that is a hierarchy of

segments along with the corresponding metric values. We have also developed an algorithm [12] that can automatically process a series of feature maps, computed using different grid resolution pairs, to generate a disjoint segmentation for each streamline.

Feature vector construction: Since the number of segments is much higher than the number of streamlines, box counting ratio can be used to filter segments with higher importance and discard the rest. Feature vectors are constructed from each retained segment. Since box counting ratio alone does not contain information about the spatial location and size of the segment, we have also included the center and the diagonal span of the bounding box of a segment into its feature vector to guarantee close proximity of similar segments in the feature space. In essence, the feature vector constructed this way describes a segment by its location, shape, and space-filling capacity.

Projection using dimensionality reduction: To visually understand the relationship among the feature segments in a high-dimensional space, we have employed *principal component analysis* (PCA), a widely used dimensionality reduction technique, to project the features on to the two dimensions with maximum variance. The main advantage of this projection is that the user can easily select regions from the projected space to see the resulting streamlines in the original spatial domain. This feature-guided interaction helps reveal flow patterns otherwise intractable because of clutter and occlusion.

3.3. Case study

Plume: Figure 6 presents results of PCA on the feature segments created from the **Solar Plume** dataset ($126 \times 126 \times 512$). The left column displays the output of PCA where each point represents a segment. Segments enclosed in a blue box are the ones selected by the user. The right column represents the streamlines containing the selected segments. The actual segment(s) that have caused the corresponding streamline to be visible is painted in red. We can see for this dataset that a cluster of points in the feature space tends to form a cluster of streamlines going through a common complex region.

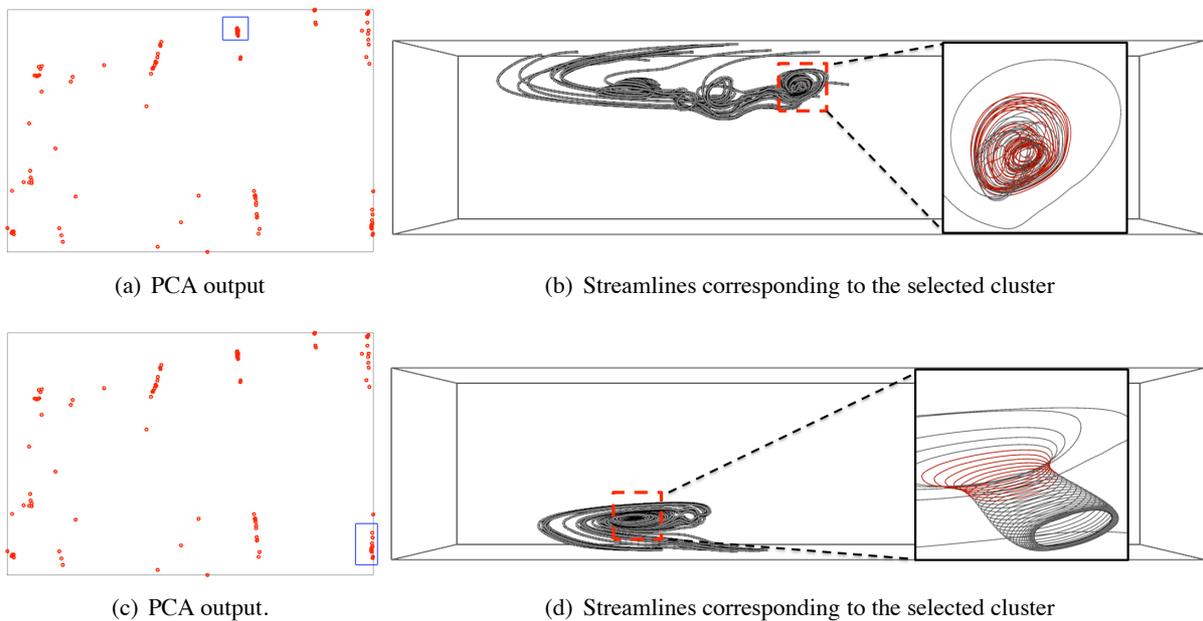


Figure 6. Geometry-based organization of features applied to the Plume dataset.

MJO simulation data: We have experimented with streamlines computed from the wind velocity fields ($2599 \times 599 \times 50$) found in a large scale climate dataset that simulates a phenomenon called MJO. This dataset does not contain many vortices or complex structures, but the ones present neatly appear on the PCA output (Figure 7(a)). The corresponding features in the streamline space are shown in inset.

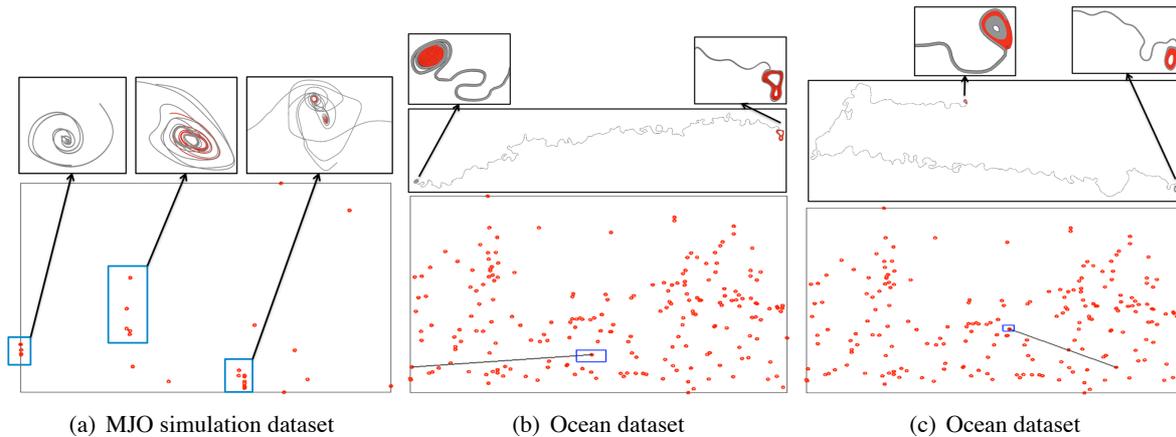


Figure 7. *Feature-based organization of streamlines from climate datasets.*

Ocean: We have tested our framework on a $3600 \times 2400 \times 40$ vector field which is the result of an eddy resolving simulation with $1/10$ horizontal spacing at the equator [13]. The dataset is courtesy of Mathew Maltrud of Los Alamos National Laboratory. Dense streamline generation is needed to capture all the numerous vortices and complex patterns present in this vector field. However, visualizing all the streamlines together is not useful because of clutter and the vector field’s large x-y span. On the other hand, our technique singles out the potentially interesting regions on the PCA space and allows the user to individually explore streamlines that contain some interesting segment (Figure 7(b) and Figure 7(c)). We have provided a visual enhancement to the PCA space by linking two points if they come from the same streamline. This *feature linking* helps to track flow pattern in general and turns out to be useful especially for data with very long streamlines such as this one. Two such links have been explored in the figures. In general, an unusually long link on the feature space may reveal a connection between two distant features in the vector field. Without the help of these links, revealing such connection from a static streamline-oriented visualization wouldn’t be easy at all.

4. Measuring Time-Varying Data Complexity via Temporal Activity Curves

Traditionally, time-varying datasets are viewed with animations. While animations can depict how some phenomena evolve over time, it is difficult to infer precisely the changes of data numerically by viewing animations alone. This is because of our limited short term visual memory and because the process is further complicated by the need to mentally map from pixel colors to data values. It is even more difficult when trying to identify and compare multiple temporal trends simultaneously from different regions. For extreme-scale science applications, simply generating an animation using all the data without selecting specific features is not only ineffective but also wasteful since the cost of computation and I/O will be much higher. In fact, many scientific phenomena such as seismic waves or seasonal temperature averages are analyzed as time series. Organizing scientific data based on temporal trends and then enabling the scientist to retrieve a subset of data exhibiting one or more trends simultaneously will not only allow for more effective analyses; it will make the task of visualizing exascale time-varying data more tractable because the amount of data movement is reduced.

In this research, we model time-varying data at each spatial point as a time series, also called a *time activity curve* (TAC). TACs are common as feature representations in medical data, such as electrical signals of the heart via electrocardiography, brain signals via positron emission tomography imaging or functional magnetic resonance imaging, and electromagnetic radiation from dynamic SPECT. Researchers have also proposed several analysis and visualization methods using TACs [14] [15] [16]. In the following, we first briefly introduce the concept of Dynamic Time Warping (DTW) that can be used to characterize time-varying scientific data represented as TACs. We then show examples of how

our temporal trend analysis can assist analysis and visualization of time-varying data.

4.1. Dynamic Time Warping

The purpose of dynamic time warping (DTW) is to measure the similarity of TACs at two spatial locations. When measuring the dissimilarity between two TACs, several issues need to be considered. First, because a feature travels through space at a finite speed, different spatial locations will exhibit the feature at different times. This phenomenon can be observed as a shift of phase in the data points' respective TACs. The width of the feature on the TAC can also be stretched/compressed because the feature may travel at different speeds in different regions. Another point of consideration is that as the feature travels in space, its property (e.g., the magnitude of the earthquake wave) may also gradually change over time. This will cause the shape of the feature TAC to deform along its motion path. As a result, two time series may contain a similar TAC but with different starting time and length. Previously, L_1 and L_2 distance metrics and cross-correlation were used to compare TACs [14] [15] [16], but these distance metrics cannot adequately address the issues mentioned above. In our approach, DTW is used as the distance metric to characterize time varying features [17] [18]. DTW is widely applied in speech recognition and data mining, which is mainly used to align two time series with the smallest distortion. With DTW, time series of similar shapes with different temporal shifts and time spans can be matched.

In essence, DTW aims to match one time series to the other with least distortion. To this end, three constraints are imposed. First, the first and last time steps in one time series are always mapped to the first and last time steps in the other time series, respectively. Second, the mapping should preserve the temporal order in the time series. Third, two adjacent time steps in the same time series cannot be mapped to nonadjacent ones. Based on these constraints, DTW can be modeled as an optimization problem solvable by a sequence of subproblems.

DTW is used to compare the two *entire* time series. If one time series matches only a subsequence of the other time series, DTW may fail to identify the similarity. To address this issue, we devise an improved DTW algorithm called *SUBDTW* [18]. The main benefit of SUBDTW is that with the same time complexity of DTW, SUBDTW can match a pattern within a subsequence of a time series, which cannot be done by DTW.

4.2. TAC-based time-varying data analysis

In this section we show how DTW and TACs can be used to assist analysis of time-varying datasets.

For univariate time-varying data, the user can specify a target TAC to search as a feature. Then, DTW is used to compute the distance between the time series from every spatial point to the feature TAC and create a distance field called the *TAC-based distance field* [17]. Along with the TAC-based distance field, each spatial point is associated with a *feature time step*, which indicates when the feature of interest occurs at that point. By considering both the distance and the feature time step, it becomes much clearer where and when the feature occurs.

An example of using the TAC-based distance field is demonstrated here using the TeraShake 2.1 [19] dataset. This dataset was generated from an earthquake simulation for the Southern San Andreas Fault. When analyzing the simulation output, the scientist was interested in knowing how the seismic energy propagated through different areas and the corresponding time. To understand this, we used the magnitude of a seismic wave to create the TACs. Figure 8(a) presents three of the TACs obtained by grouping the TACs in the dataset via K-mean clustering. The magnitudes of the TACs are normalized to the same scale. One can see that all three mean TACs contain a peak, at different time steps, followed by a tailing signal of different lengths. This difference suggests that using DTW to calculate the dissimilarity will obtain a smaller distance than using L_1 distance, L_2 distance, or the cross-correlation metrics.

In our study, we chose the TAC plotted in blue color in Figure 8(a) as the feature TAC. Figure 8(b) shows the joint histogram of the feature time step and distance in the obtained TAC-based distance field. The DTW distance and the feature time step are represented as the horizontal and vertical coordinates, respectively. The frequency of each bin is normalized to [0, 1] and then mapped to color according to the

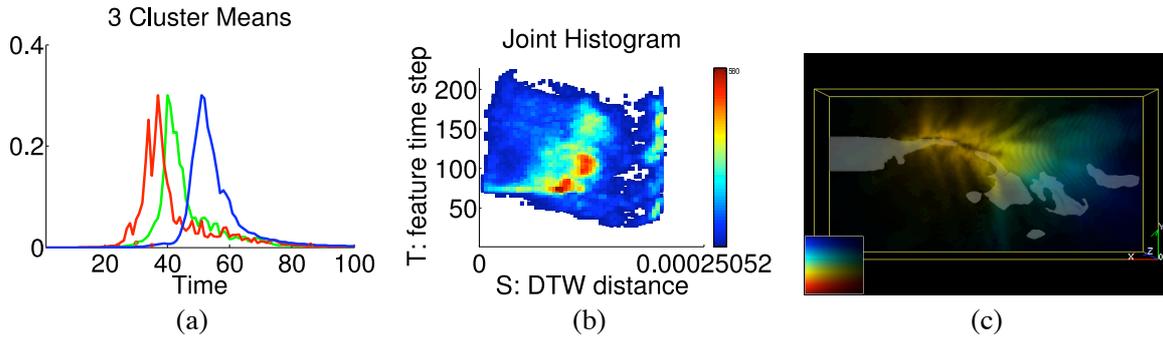


Figure 8. Visualization of the magnitude field of the dataset TeraShake 2.1. (a) The mean TACs from three clusters. The selected feature TAC is plotted in blue. The magnitude of the three TACs are normalized to match that of the selected feature TAC. (b) The joint histogram of distances and feature time steps of the TAC-based distance field. (c) Volume rendering of the TAC-based distance field.

color bar displayed on the side. In the joint histogram, it can be seen that after time step 75, the horizontal distance between the t -axis and the first nonempty bin is increasing as the t coordinate increases, which means the number of voxels that contain wave magnitudes closer to the maximum magnitude of the feature TAC is decreasing over time, indicating that the strength of the earthquake is reducing after time step 75. This result is consistent with the simulation setting that the fault rupture was terminated at the 60th second or equivalently the time step 55.

Figure 8(c) presents a volume-rendering image from the TAC-based distance field. Here the transition of color from yellow to green indicates the propagation paths of the earthquake energy. The wavefronts of the reflected waves are also depicted. Figure 8(c) also plots the basin surfaces to reveal the correlation between the wave propagation and the basin structure. Note that the yellow region shown in the image exhibits semi-transparent strips, which is consistent with the structure of the basin.

5. Conclusions

Feature-based data summarization can facilitate effective data reduction, triage, and classification. In this paper, we present three visual analytics techniques to perform data summarization by measuring the dataset’s statistical, geometric, and time-varying data complexity. To measure the information complexity of a dataset, we use Shannon’s entropy in information theory to define both the importance and error measures for multiresolution data blocks. This analysis of information complexity can help one choose data at appropriate levels of detail with a balanced quality and cost tradeoff. To visualize three-dimensional flow fields, we use the concept of box counting ratio to measure the geometric complexity of streamlines. The box counting ratio allows one to approximate the fractal dimensions of streamlines and identify regions with more salient geometric features such as swirls/vortices. To analyze time-varying data, we propose to model the data as time activity curves (TAC) and use dynamic time warping (DTW) to cluster data into groups of different temporal trends. With the DTW measure, a novel representation of time-varying fields, called the TAC-based distance field, can be used to produce visualizations that highlight both the spatial and temporal characteristics of user-selected features.

References

- [1] Silverman B 1986 *Density estimation for statistics and data analysis* (Chapman & Hall/CRC)
- [2] Scott D 1992 *Multivariate density estimation: theory, practice, and visualization* (Wiley-Interscience)
- [3] Duda R, Hart P and Stork D 2001 *Pattern classification* (Wiley-interscience)
- [4] Wang Z, Wu G, Sheikh H R, Yang E H and Bovik A C 2006 *IEEE Transactions on Image Processing* **15** 1680–1689
- [5] Wang C, Garcia A and Shen H W 2007 *IEEE Transactions on Visualization and Computer Graphics* **13** 122–134
- [6] Wang C and Shen H W 2006 *IEEE Transactions on Visualization and Computer Graphics* **12** 1029–1036
- [7] Turk G and Banks D 1996 Image-guided streamline placement *Proceedings of ACM SIGGRAPH Conference* pp 453–460

- [8] Jobard B and Lefer W 1997 Creating evenly-spaced streamlines of arbitrary density *Visualization in Scientific Computing* pp 43–56
- [9] Verma V, Kao D and Pang A 2000 A flow-guided streamline seeding strategy *Proceedings of IEEE Visualization Conference* pp 163–170
- [10] Mebarki A, Alliez P and Devillers O 2005 Farthest point seeding for efficient placement of streamlines *Proceedings of IEEE Visualization Conference* pp 479–486
- [11] Falconer K 2003 *Fractal Geometry: Mathematical Foundations and Applications* 2nd ed (John Wiley & Sons)
- [12] Chaudhuri A, Lee T Y, Shen H W, Khoury M and Wenger R 2011 Exploring flow fields using fractal analysis of field lines Tech. Rep. OSU-CISRC-4/11-TR15 Department of Computer Science & Engineering, Ohio State University Columbus, OH
- [13] Maltrud M E and McClean J L 2005 *Ocean Modelling* **8** 31
- [14] Fang Z, Möller T, Hamarneh G and Celler A 2007 Visualization and exploration of time-varying medical image data sets *GI '07: Proceedings of Graphics Interface 2007* pp 281–288 ISBN 978-1-56881-337-0
- [15] Guo H, Renaut R, Chen K and Reiman E 2003 *BioSystems* **71** 81–92
- [16] Wong K P, Feng D, Meikle S and Fulham M 2002 *IEEE Transactions on Nuclear Science* **49** 200–207 ISSN 0018-9499
- [17] Lee T Y and Shen H W 2009 Visualizing time-varying features with tac-based distance fields *PV'09: Proceeding of IEEE Pacific Visualization Symposium 2009* (Los Alamitos, CA: IEEE Computer Society) pp 1–8 ISBN 978-1-4244-4404-5 URL <http://dx.doi.org/10.1109/PACIFICVIS.2009.4906831>
- [18] Lee T Y and Shen H W 2009 *IEEE Transactions on Visualization and Computer Graphics* **15** 1359–1366
- [19] Olsen K B, Day S M, Minster J B, Cui Y, Chourasia A, Faerman M, Moore R, Maechling P and Jordan T 2006 *Geophysical Research Letters* **33** L07305