Multivariate Volumetric Data Analysis and Visualization through Bottom-Up Subspace Exploration

Kewei Lu*
The Ohio State University

Han-Wei Shen†
The Ohio State University

ABSTRACT

Multivariate volumetric datasets are often encountered in results generated by scientific simulations. Compared to univariate datasets, analysis and visualization of multivariate datasets are much more challenging due to the complex relationships among the variables. As an effective way to visualize and analyze multivariate datasets, volume rendering has been frequently used, although designing good multivariate transfer functions is still non-trivial. In this paper, we present an interactive workflow to allow users to design multivariate transfer functions. To handle large scale datasets, in the preprocessing stage we reduce the number of data points through data binning and aggregation, and then a new set of data points with a much smaller size are generated. The relationship between all pairs of variables is presented in a matrix juxtaposition view, where users can navigate through the different subspaces. An entropy based method is used to help users to choose which subspace to explore. We proposed two weights: scatter weight and size weight that are associated with each projected point in those different subspaces. Based on those two weights, data point filter and kernel density estimation operations are employed to assist users to discover interesting features. For each user-selected feature, a Gaussian function is constructed and updated incrementally. Finally, all those selected features are visualized through multivariate volume rendering to reveal the structure of data. With our system, users can interactively explore different subspaces and specify multivariate transfer functions in an effective way. We demonstrate the effectiveness of our system with several multivariate volumetric datasets.

1 INTRODUCTION

Nowadays, computational fluid dynamics (CFD) and weather model simulations commonly produce multiple attributes/variables associated with each grid point in the data domain. Usually, features in multivariate datasets can be better classified with more than one variable. To display features in a multivariate volumetric data, direct volume rendering has been used to visualize and analyze multivariate data. In direct volume rendering, data values are mapped to optical properties such as color and opacity through transfer functions. A good transfer function can reveal important features in the data more effectively. However, identifying good transfer functions is nontrivial and there have been many previous works on this topic [9, 12, 18, 19, 23]. Designing good multivariate transfer functions is much harder than 1D transfer functions because of the increasing number of variables and the complex relationships among the variables in multivariate datasets. In general, a multivariate transfer function can be represented as a multidimensional lookup table which maps different combinations of data values to different optical properties. However, as pointed out by Kniss et al. [22], a major limitation of the multidimensional lookup table is the increasing memory cost as the number of variables increases. In order to resolve the issues of using a multidimensional lookup table, Kniss et al. [22] proposed to represent a multivariate transfer function as several Gaussian components. In the Gaussian transfer function representation, different features in the dataset are represented as different Gaussian components and each Gaussian component is assigned a unique color. The data point with values that are closer to the center of a Gaussian component will have high opacity, and thus will be shown in the final volume rendering view. The final volume rendering result is visualized by combining the colors and opacities from all Gaussian components. However, it is a nontrivial task to specify those Gaussian components that are potentially corresponding to interesting regions in the data. When users are familiar with the data and know how features are defined based on the data values, users can set those Gaussian components centered around those data values to highlight the corresponding features. However, when users are not familiar with the data, it is difficult for users to specify those Gaussian components to extract salient features.

One way to design a multivariate transfer function is through dimensionality reduction such as Multidimensional Scaling (MDS) [10]. Based on those dimensionality reduction techniques, the high-dimensional points are projected to lower dimensions while preserving the distances between each other. Clusters can be identified and selected from the lower dimensions. Each cluster can then be mapped to a Gaussian component by setting the Gaussian mean around the cluster center to support the design of multivariate Gaussian transfer functions. As identified by Parsons et al. [29], for a high-dimensional data, clustering while considering all dimensions is ineffective due to the following reasons:

- Not all the dimensions are relevant. When lots of irrelevant dimensions are considered, meaningful clusters in subspaces can be destroyed by those irrelevant dimensions;
- Because of the curse of dimensionality problem, distance measurements in high-dimensional space are difficult to interpret [29];

For multivariate scientific data, a feature usually could be well defined with a couple of variables and not necessary all the variables are needed. For example, the Hurricane Isabel dataset which models a strong hurricane in the West Atlantic region contains 13 variables. The feature Hurricane eye in the dataset could be well defined as the low velocity and low pressure region which only involves two variables. Thus, the feature Hurricane eye can be more efficiently identified by exploring the subspace of velocity and pressure. In order to address the above issues, and to efficiently discover features in high-dimensional data, our system employs a bottom-up approach to incrementally construct those Gaussian components during exploration. In the bottom-up approach, users start from a lower dimension and make a selection in the lower dimension. Then, the selection can be further refined by extending to a higher dimension. This kind of bottom-up exploration pipeline has been used in several techniques to explore multivariate volumetric datasets [3, 7, 31, 37, 38]. These previous bottom-up approaches mainly rely on data point brushing on parallel coordinate or scatter plot and usually requires users' prior knowledge about the data, so the users know which data points to brush. In this paper, based on

*e-mail: lu.321@osu.edu
†e-mail: shen.94@osu.edu

2017 IEEE Pacific Visualization Symposium (PacificVis)
18-21 April, Seoul, Korea
978-1-5090-5738-2/17/$31.00 ©2017 IEEE

information derived from the data, we present a systemic approach to guide user exploration, so the users will not blindly brush on parallel coordinates or scatter plots without any guidance.

In our system, a scatter plot matrix is used. Compared with Parallel Coordinates, all pairwise variable relationships can be shown in scatter plot matrices while only the relationship between adjacent coordinates is able to be shown in the parallel coordinate. In our system, we show all single variable and two variables subspaces in a matrix juxtaposition view. Users can navigate through the different subspaces using the matrix widget. Since scatter plots are resulted from projecting high-dimensional data points to lower dimensions, the high-dimensional information can be lost [31]. In [31], the authors proposed a technique to use a combination of 2D and 3D scatter plots to solve this problem. However, when we face datasets with the number of dimensions higher than three, it is nontrivial for us to visualize those high-dimensional data points. In this paper, we proposed two weights: size weight and scatter weight associated with each projected data point as a complementary information for the loss of high-dimensional information due to projection.

In our system, we derived information from the data sets to assist user exploration. In order to guide users to choose the next subspace for exploration, an entropy-based approach is used. Data points filter and weighted kernel density estimation based on size and scatter weights are used to help users to select data points in the current exploring subspace. Compared with previous similar multivariate volume rendering system, our system is unique mainly in two aspects:

- In terms of the final product, Gaussian Transfer Functions, our system allows different gaussian components having a different number of variables. This is unlike other previous approaches [10] which all variables are involved so features existed in subspaces might not be easily discovered.

- Our system employs bottom-up subspace exploration with derived information such as entropy, scatter and size weight to assist user exploration. Based on the derived information, users do not need to blindly brush on parallel coordinates or scatter plots like some other systems. The dense region can be automatically identified through weighted kernel density estimation to help users identify interesting features.

Our system contains several interactive widgets that are linked together to assist user exploration. The layout of our system is shown in Figure 1. The upper triangle of the matrix widget (Figure 1.B) is used to show all single and two variables subspaces in a juxtaposition view. The lower triangle of the matrix widget shows the entropy value computed from the joint histogram defined in the corresponding subspace. Users can navigate through different subspaces by double clicking the corresponding subspace. All the subspaces are linked together and dynamically updated based on users' selection in one subspace. Parallel coordinate plots (PCP) (Figure 1.A) are used to show the Gaussian functions corresponding to the user-identified features. Users can also use the PCP to adjust the variance of the currently specified Gaussian component. A multidimensional scaling plot (Figure 1.F) shows the two-dimensional embedding of all data points by considering all the dimensions. The high-dimensional project view gives users a global view of all the data points and which regions have been covered by the specified Gaussian components so far. The data points that are already covered by a specified Gaussian component are colored by the color assigned to that Gaussian component. A zoom-in view of the user-selected subspace is shown in a window widget (Figure 1.D). Density estimation result is shown in this zoom-in window widget. A 2D color map widget (Figure 1.E) is used to show the color map for volume rendering which allows users to visually link the volume space with the transfer function space. All the 2D images such as density estimation

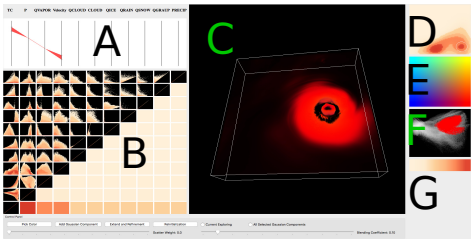


Figure 1: The layout of our system interface.

results, subspace plots, and entropy submatrix use the same color map (Figure 1.G) where white indicates low and red indicates high values. The main contributions of this paper are:

- We presented an effective system for users to interactively explore multivariate volumetric data and identify interesting features existing in different subspaces.

- In order to provide users with the high-dimensional information which is lost due to projection, we proposed two weights: scatter weight and size weight associated with each data point. Then, based on those two weights, users can use data point filter and weighted kernel density estimation to identify features and gradually construct Gaussian Transfer Function.

- Besides the scatter and size weight, we derived additional information such as information entropy from the data to assistant user exploration.

2 Related Works

2.1 Multivariate Data Analysis

Multivariate data usually results in a high-dimensional attribute space. In order to visualize this high-dimensional attribute space and discover interesting features, several approaches have been proposed. One way to explore and visualize the high-dimensional attribute space is to use various dimension reduction techniques. The goal of those dimension reduction techniques is to reduce the number of dimension of the original data while preserving the original high-dimensional data characteristics. Two common dimension reduction techniques are Principle Component Analysis (PCA) and Multidimensional Scaling (MDS). PCA transforms the original high-dimensional data into an orthogonal coordinate system while maximizing variance along the axes. MDS maps the original high-dimensional data into a low dimensional space while maintaining the dissimilarities between data points. PCA [28] and MDS [10] have been utilized in various techniques to visualize and analyze the multivariate data. Parallel coordinate plots [14, 15] is another way to visualize the attribute space and help users to analyze and understand the multivariate data. In a parallel coordinate system, attributes/variables are defined as parallel vertical lines. A data point corresponds to a polyline constructed by connecting vertices on the parallel vertical axes. The position of the i_{th} vertex on the i_{th} axis is defined by the value of the i_{th} attribute. However, parallel coordinate has a major limitation which is when there is a larger number of data points, the visualization becomes clutter and it is ineffective to show the internal structure of the data. A number of techniques have been proposed to deal with this problem [3,13,17,27]. In [17], the authors solved the visual clutter problem by first clustering the polylines into clusters and then using high-precision textures to represent those clusters to make both the visualization and rendering effective. In order to highlight different structural information, the authors also proposed to apply transfer functions on the high-precision textures.

Novotny et al. [27] presented a focus+context parallel coordinate visualization based on binned data representation. Besides visualizing the high-dimensional attribute space, there are techniques focusing on identifying interesting features presented multivariate data. In [8], the authors presented a framework to support interactive specification and identification of features in multivariate data. In [16], the authors introduce local statistical complexity which can be used to detect important features in a multivariate dataset.

2.2 Transfer Function Design

Volume rendering is a popular method to visualize volumetric data. It maps data values to optical properties such as color and opacity to visually reveal the structure of the data. The mapping from data values to optical properties is achieved through transfer functions. A good transfer function can reveal the underlying features in the data in a more effective way. A lot of research effort have been devoted to the design of transfer function in the past. Arens et. al. [2] presented a survey on different types of transfer function used for volume rendering. The transfer function is categorized into six types: 1D data-based, gradient based, curvature based, size based, texture based and distance based. In order to better separate materials and boundaries, many techniques consider gradient magnitude while designing transfer function [18, 20, 21, 25, 34]. The gradient magnitude quantifies how fast the value changes. By incorporating gradient magnitude in the design of transfer function, different materials can be better characterized. Kindlmann et. al. [19] presented a transfer function design approach by utilizing curvatures to enhance the effectiveness of volume rendering. The size of features has also been considered when designing transfer functions. [5]. Correa et. al. [6] observed that the structure of features in volume data could be classified based on the occlusion pattern. Based on this observation, the authors proposed a new transfer function to classify volume data by utilizing the ambient occlusion information of voxels. Several other properties, such as texture [4] and statistical information [11, 30] have also been used to assist the design of transfer function. Besides those different types of transfer function used for volume rendering, techniques that focus on semi-automatic transfer function design have also been proposed [33, 39].

When we are dealing with multivariate data, multidimensional volume rendering can be used to visualize and analyze the data. However, designing an effective multidimensional transfer function for multidimensional volume rendering is even harder and requires lots of researches. Kniss et. al. [22] discussed the issues of using separate transfer functions for each variable and multidimensional transfer function lookup table. Then, the authors proposed to use Gaussian transfer function which overcomes the issues mentioned above. Liu et. al. [24] analyzed high-dimensional data through identifying a set of low-dimensional linear subspaces. Then, the authors presented an approach to assist multivariate transfer function design by animating transitions between different views. Guo et. al. [10] presented an effective and scalable system that combines parallel coordinates plots (PCP) and multidimensional scaling (MDS) projection to assist novel multidimensional transfer function design. Zhou et. al. [38] proposed a transfer function design approach to support intuitive multivariate dataset exploration based on user selected samples. Zhao et. al. [37] presented a new approach which uses parallel coordinates to design multi-dimensional transfer functions.

3 System Overview

As described in Section 1, features represented as clusters in subspaces may not be effectively detected if all variables are considered. In this paper, we propose a system to address the aforementioned problem. Information such as size and scatter weight are derived from the data set to guide user exploration, and multivariate transfer functions are constructed accordingly during exploration. All the user identified clusters can be rendered through multivariate volume rendering to reveal interesting features in the data.

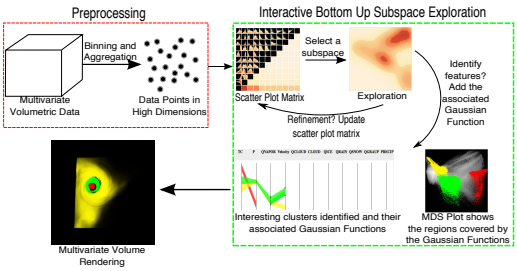


Figure 2: The pipeline of our system.

Figure 2 shows an overview of our system pipeline. Given a multivariate volumetric data set, each grid point corresponds to a multivariate data point. A multivariate volumetric data set with N grid points amounts to N multivariate data points in total. We apply binning and aggregation in value space to quantize and reduce the number of data points, which makes our subsequent interactive exploration more efficient. Given those data points after quantization and aggregation, all single and two variables subspaces are inspected in a juxtaposition view. As a data point in 2-dimensional or 1-dimensional subspace can map to many data points in the original high dimensions due to projection, two weights are assigned to each data point. These two weights severe as the complementary information for the loss of high-dimensional information due to projection, and are used to guide user exploration. The color channel is used to encode the weight information in the scatter plot matrix visualization. The end product of this bottom-up subspace exploration is multiple Gaussian components where each one is corresponding to a user-identified cluster in a subspace. All the selected clusters can be visualized using multivariate volume rendering to depict the features of the data.

4 Preprocessing

4.1 Data Reduction

With the increased computing power, the resolution of data generated from simulation also continues to grow. Taken a moderate size scientific dataset as an example. Hurricane Isabel which models a strong hurricane in the West Atlantic region in September 2003 has a resolution of $500 \times 500 \times 100$. This amounts to $25,000,000$ multivariate data points which poses a challenge for interactive subspace exploration. Given this large number of data points, directly performing our bottom-up subspace exploration can be difficult. In our subspace exploration process, kernel density estimation (KDE) is used to estimate the density to discover potential clusters existing in different subspaces. Given n data points, the time complexity of calculating k evaluation points using KDE is $O(kn)$. Given a constant number of evaluation points, the time complexity is linearly proportional to the number of data points. When the number of multivariate data points is large, it prohibits us to achieve interactive subspace exploration and cluster identification. To resolve this issue, in the preprocessing stage, we use data binning and aggregation to quantize and reduce the number of multivariate data points. In the following section, we discuss in detail the data binning and aggregation, and the reduced set of data points generation in both full dimensions and low subspaces.

4.1.1 Data Binning and Aggregation

Data binning and aggregation are widely used techniques for data preprocessing and reduction. Data binning divides a continuous domain into intervals, and then the original value is mapped to the interval that contains the data value. It is a form of quantization.

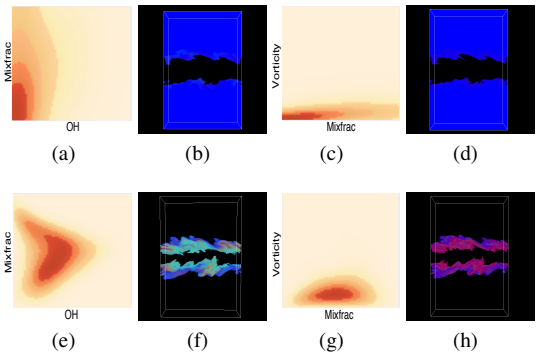


Figure 3: (a) and (c) shows the density estimation results when only size weight is considered and the volume regions corresponding to the dense regions are shown in (b) and (d) respectively. The dense region corresponds to the unimportant background region in this dataset. (e) and (g) shows the density estimation results when both size weight and scatter weight are considered and the volume regions corresponding to the dense regions are shown in (f) and (h). More interesting regions are identified.

In our system, we bin the data in the value space. Suppose the multivariate data set contains n grid points and m variables, then the original multivariate data points denoted as P contains n data points in m dimensional value space. Given the range of the i_{th} variable, denoted as $[a_i, b_i]$, and the width of the bins for the i_{th} variable is h_i, a data point $p_j \in P$ that has values $(p_{j,1}, p_{j,2}, ..., p_{j,m})$ is quantized by using the following equation:

$$Binning(p_j) = (\left\lfloor \frac{p_{j,1} - a_1}{h_1} \right\rfloor, \left\lfloor \frac{p_{j,2} - a_2}{h_2} \right\rfloor, ..., \left\lfloor \frac{p_{j,m} - a_m}{h_m} \right\rfloor) \quad (1)$$

After quantization, different data points can have exactly the same value. Then, we perform data aggregation so that those data points in P which have the same value after quantization are represented as a single point q in the multidimensional value space. The new data point q is defined as: $q = Binning(p_i), i \in U$. where U denotes a set of data points in P that has the same value as q after quantization. We assign a weight w to q:

$$w = |U| \quad (2)$$

the weight represents the number of original multivariate data points in P that have the same value as q after quantization, and we use Q to denote the set of data points after binning and aggregation.

4.2 Subspace Data Points Generation

Given the set of data points Q, a scatter plot matrix that contains all single and two variables subspaces is constructed. Each subspace plot in the scatter plot matrix shows the scatter plot of the data points generated for this subspace. In this section, we discuss subspace data points generation and the two weights: size weight and scatter weight associated with each data point in a subspace. We use 2-variable subspace as the example for illustration, data points in single variable subspaces can be generated accordingly.

Let $S^{x,y}$ denote the projected data points generated for the subspace defined by variable x and y. $S^{x,y}$ can be generated by removing the dimensions other than x and y for all data points in Q, which can be seen as an orthogonal projection in value space. We define two weights associated with each projected data point in $S^{x,y}$. Let $s_k^{x,y} \in S^{x,y}$ be the k_{th} data point. Many data points in Q could be mapped to a single data point in $S^{x,y}$ due to the orthogonal projection. Let U_k denote the set of data points in Q that are mapped to the data point $s_k^{x,y}$ in the x, y subspace, the size weight w_{size} and scatter weight $w_{scatter}$ for this projected data point are defined as:

$$w_{size} = \sum_{i \in U_k} w_i, \qquad w_{scatter} = |U_k|$$

where w_i is the weight associated with the data point $q_i \in U_k$ computed in equation 2.

The size weight w_{size} for a particular data point $s_k^{x,y}$ represents how many data points in the original set of data points P (before binning and aggregation) are mapped to the point $s_k^{x,y}$. A large weight means the data point $s_k^{x,y}$ corresponds to a cluster of a large size. The scatter weight $w_{scatter}$ for a particular data point $s_k^{x,y}$ represents how many data points in Q (after binning and aggregation) are mapped to it. A large weight generally means the cluster defined around the data point $s_k^{x,y}$ is more likely to be further refined in the extended subspace, since its high-dimensional mapping is more scattered.

In our subspace exploration pipeline, weighted kernel density estimation is used to identify possible clusters existing in a subspace. If only the size weight w_{size} is considered in the density estimation, the following problems will arise:

- The density estimation result will bias to the size of cluster, so clusters with relatively smaller size may not be easily discovered;
- When the data contains many unimportant data points such as background that have exactly the same value, the density estimation can not efficiently identify meaningful features that exist in the data. This problem is illustrated in Figure 3.

If only the scatter weight $w_{scatter}$ is considered during density estimation, some clusters that are corresponding to a very few number of data points in Q might be missed during exploration. In our system, we combine the size weight and scatter weight: $w = \alpha w_{size} + (1-\alpha) w_{scatter}$. w is a new weight for each projected data point in the subspaces and is used in density estimation. α is the blending coefficient between 0 and 1 which determines the influence of size weight. Figure 4 shows the effect of using different blending coefficients. As we can see, when α is small such as below 0.2, a small cluster can be easily identified as shown in the leftmost figure. When the value of α increases, a large cluster (usually corresponding to background) is then detected as shown in the rightmost figure.

5 Bottom-Up Subspace Exploration

In this section, we discuss our interactive bottom-up subspace exploration and cluster identification method. Our interactive bottom-up subspace exploration mainly contains the following steps as shown in Figure 5:

Step 1: Subspace Selection: The first step is to pick a subspace to explore. When the number of variables is large, the number of subspaces we have is also large and thus make the selection of which subspace to explore difficult. We provide an entropy-based method to guide users to choose the next subspace to explore. The details of the entropy-based selection are discussed in Section 5.1.

Step 2: Data points filtering: After users select a subspace to explore, the data points with scatter weight larger than a user-defined threshold are filtered, since those data points are more scattered when extending to a larger subspace. After removing those points, density estimation could be applied to the remaining data points to identify clusters that can be well defined in the currently exploring subspace. In our system, users can adjust the threshold with a slider bar.

Step 3: Cluster identification: Weighted kernel density estimation is applied to find the dense regions that are potentially corresponding to interesting clusters. As described in Section 4.2, the weight is decided by blending scatter and size weights. In our

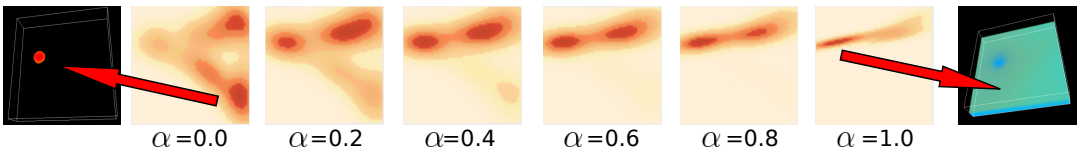


Figure 4: The middle six images show the density estimation results of the Temperature and Pressure subspace with different blending coefficients α after filtering out data points with scatter weights larger than a user-defined threshold. When the value of α is small such as 0.0 and 0.2, the influence of size weight on the density estimation result is small, so dense regions corresponding to some small size clusters can be identified. In this example, a dense region corresponding to the Hurricane center can be easily identified with a small value of α. When the value of α is large, the influence of size weight on the density estimation result is large. In this example, a dense region corresponding to a large size cluster shows up as the value of α increases.

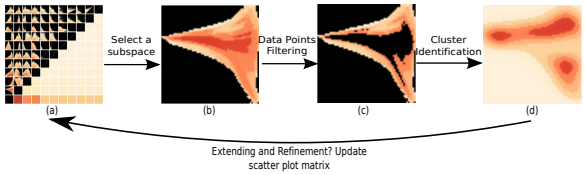


Figure 5: Basic steps of our bottom-up subspace exploration.

system, we provide a slider bar to allow users to easily adjust the value of blending coefficient. Details of the cluster identification are described in Section 5.2.

Step 4: Extending and refinement: Users can further refine the currently identified cluster by extending to a larger subspace. Details of the extending and refinement are described in Section 5.3.

5.1 Entropy-based Subspace Selection

The first step of our exploration is to select the next subspace to explore. Difficulties arise when there are a larger number of subspaces. In our system, for each subspace, we evaluate the information contained in the variables that define the subspace using information entropy. Then, this information is used to guide users' choice of the next subspace to explore. The subspace with a higher entropy tends to contain more information about the data and thus should be explored first. Given a probability distribution $P(X)$ for a random variable X, the Shannon entropy is defined as: $H(X) = -\sum P(X) log P(X)$. When a group of variables $X_1,...,X_n$ are considered, the joint entropy is defined as: $H(X_1,...,X_n) = -\sum P(X_1,...,X_n) log P(X_1,...,X_n)$

In the preprocessing stage, we do data binning and aggregation to quantize the data points and then those subspace data points are generated through orthogonal projection. If we consider the subspace of variable U and V, and suppose the number of bins used for variable U and V during the preprocessing stage is B_u and B_v, then each projected point in the subspace of U and V corresponds to a non-zero entry of the joint histogram of U and V with B_u and B_v bins respectively. The size weight w_{size} of a projected point is the frequency of the corresponding non-zero entry. Given the normalized size weight W_{ns} such that the summation of all projected points' normalized size weight equals to one, then the information contained by a subspace S can be evaluated by using the formula: $H(S) = -\sum W_{ns} log W_{ns}$, which is the entropy of the joint distribution of variable U and V. The entropy values of different subspaces are shown in the lower triangle of the scatter plot matrix subplot as shown in Figure 5.A.

5.2 Cluster Identification

5.2.1 Density Estimation

Once users select the subspace to explore, users can use our system to identify interesting clusters existing in the currently exploring subspace. In our system, we use weighted kernel density estimation to estimate the density and the dense regions are potentially corresponding to interesting clusters. Kernel density estimation (KDE) is a nonparametric technique to estimate the probability density function from a number of data points. Given a set of N data points: $X_1, X_2, ..., X_N$ and a weight w_i associated with each data point X_i, a weighted kernel estimator based on this set of data points is defined as:

$$f(x) = \frac{1}{h} \sum_{i=1}^{N} w_i K(\frac{x - X_i}{h}) \quad (3)$$

where K is the kernel function and h is the kernel bandwidth. The weight is computed by blending size weight and scatter weight as described in Section 4.2. In our system, we use the Gaussian kernel function: $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$. When the data points are in multi-dimensions, multivariate kernel density estimation needs to be used:

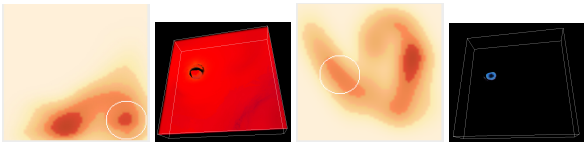
$$f(x) = \sum_{i=1}^{N} w_i \frac{1}{det(H)} K(H^{-1}(x - X_i)) = \sum_{i=1}^{N} w_i K_H(x - X_i) \quad (4)$$

where H is the bandwidth matrix and in our system, we assume H is a diagonal matrix: $H = diag(h_1, h_2, ..., h_d)$. where d is the number of dimensions and h_i is the bandwidth for dimension i.

The bandwidth is an important parameter for KDE which decides the quality of the density estimation. A good bandwidth selection should minimize the mean integrated squared error of the kernel density estimation with the true underlying probability density function. To determine the bandwidth for our kernel density estimation, we employ Silverman's rule of thumb [32].

5.2.2 Cluster Representation

Once users identify a dense region in the density estimation result that is potentially corresponding to an interesting cluster, the users can select the cluster and a Gaussian component g centered at the user-selected cluster is constructed to represent the selected cluster. Users can click on the density estimation subplot to select the corresponding cluster. Let set U denote all the variables that define the current subspace, set W denote all the variables in the dataset and then set $V = \{W - U\}$ contains all the variables that have not been added to g. The Gaussian component g associated with a cluster selected in the currently explored subspace is defined as: $g = \{\vec{m}, H\}$, where $\vec{m}$ is a vector which denotes the means of variables in U and H is a diagonal matrix of which each diagonal element denotes the variance of a variable in U. The mean value is chosen based on users' click position on the density estimation subplot and the variance can be adjusted by dragging the corresponding Gaussian component on the parallel coordinate subplot. To visualize the user-selected cluster using volume rendering, given an original multivariate data point p_i, all the data values for variables from U form a sample data vector $\vec{v}_i$. The multivariate Gaussian transfer function maps the data point p_i



(a) (b) (c) (d)

Figure 6: Bottom-up dense region refinement. We start with an empty Gaussian component g={}. Initially we select the subspace of Temperature and Velocity. (a). The density estimation of the data points in the subspace of Temperature and Velocity. The dense region within the white cycle is selected. (b). The volume rendering highlights the selected region. The Gaussian component is also updated. Temperature and Velocity are added to g. Then, data points are filtered if they are not covered by g. A new set of data points are generated and the subspace matrix is updated. (c). The user selects the Pressure and QVapor subspace in the updated subspace matrix. Now, the current subspace is defined by Temperature, Pressure, QVapor and Velocity. The user explores the subspace by filtering out data points with scatter weights larger than a user-defined threshold and then performs density estimation. The region within the white cycle is selected by the user. (d). The volume rendering highlights the selected region. Pressure and QVapor are added to g and g is extended to a 4 dimensional Gaussian component.

to an opacity value:

$$GTF(\vec{v_i}) = a \times e^{-\frac{(\vec{v_i}-\vec{m})^T H^{-1} (\vec{v_i}-\vec{m})}{2}} \quad (5)$$

where a is a constant to scale the opacity value. The data points that have values closer to the cluster center will have high opacity, and therefore will be highlighted in the volume rendering result. Each Gaussian component is also assigned a unique color. Each Gaussian component covers a subset of data points in Q, a data point is denoted as covered by a Gaussian component g if it is within two standard deviations of the mean of g. Once users decide to add a Gaussian component g to the Gaussian Transfer Function, the data points covered by g are removed from Q since they are already classified and the scatter plot matrix subplot is also updated. In this way, the points that are shown in the scatter plot subplot are always the points which are not covered by our specified Gaussian Components and users can focus on exploring those points.

Suppose the user identifies n interesting clusters through our system, and the i_{th} cluster has a Gaussian component g_i associated with it and is assigned a unique color C_i. All the n clusters can be rendered simultaneously to depict interesting features of the data. The final color C and opacity α are obtained by combining the color and opacity generated by each Gaussian component together:

$$C = \frac{\sum_{i=1}^{n} \alpha_i C_i}{\sum_{i=1}^{n} \alpha_i}, \qquad \alpha = \sum_{i=1}^{n} \alpha_i \quad (6)$$

5.3 Extending and Refinement

Once users identify an interesting cluster in the currently exploring subspace, the users can further refine the cluster by extending the subspace to a larger subspace. Whether a specified cluster needed to be further refined can be decided based on the following two criteria:

- **Scatter:** Scatter indicates how likely a cluster can be further refined. The cluster defined by data points with larger scatter weight usually needs to be further refined.
- **Overlapping:** Given the previously specified Gaussian components $G_1, ..., G_i$ and the currently specifying Gaussian component G_c, if the region covered by G_c has a large overlap with the region covered by all the previously specified Gaussian components, usually, an extending and refinement operation is needed to further refine this region to minimize the overlap or we can simply discard the currently exploring region and explore other regions that have a smaller overlap with the previously specified Gaussian components. This will prevent us from exploring the same region multiple times. A multidimensional Scaling subplot is used here to assist users to identify how much overlap there is between G_c and all those previously specified Gaussian components. The MDS subplot shows a global view of all the data points and which regions have been covered by the specified Gaussian components so far. All the data points covered by those previously specified Gaussian components and the currently specified Gaussian components are highlighted in the MDS subplot, so the users can visually identify how much overlap there is.

The cluster refinement includes three steps: removing data points that are not covered by the selected cluster, updating the subspace matrix, and refining the cluster in an extended subspace.

Removing data points that are not covered by the selected cluster. Once users identify and select a cluster in the current subspace, a Gaussian component is constructed to represent the selected cluster as described in Section 5.2.2. We remove the data points which are not covered by the Gaussian component. As defined in Section 5.2.2, a data point is said to be covered by a Gaussian component g if its value is within two standard deviations of the mean of g. The remaining data points are then used for the subsequent exploration.

Updating the subspace matrix. With the new set of data points, the subspace matrix is updated. The size weight and scatter weight are recalculated for each data point. Suppose the selected cluster is in the subspace defined by variables A and B, then all subspaces involving A and B are also removed from the scatter plot matrix subplot. The entropies for the remaining subspaces are also recalculated.

Refining clusters in the extended subspace. With the updated subspace matrix, the user can navigate through different new subspaces to refine the cluster and identify interesting clusters.

An example of this bottom-up cluster refinement is shown in Figure 6.

6 Results

In this section, we demonstrate the effectiveness of using our system to analyze and visualize several multivariate volumetric datasets. Three datasets were used. For the data binning and aggregation, the number of bins was set to 64 for each variable.

6.1 Hurricane Isabel Data Set

The Hurricane Isabel data set was used for this case study. Eleven variables were used for the experiments: Temperature, Pressure, Wind speed magnitude (Velocity), Water vapor mixing ratio (QVAPOR), Cloud moisture mixing ratio (QCLOUD), Total cloud moisture mixing ratio (CLOOUD), Cloud ice mixing ratio (QICE), Rain mixing ratio (QRAIN), Snow mixing ratio (QSNOW), Graupel mixing ratio (QGRAUP) and Total precipitation mixing ratio (PRECIP). Time step 7 was selected for our experiment.

Figure 7 shows a typical exploration using the Hurricane Isabel dataset. Initially, a scatter plot matrix constructed from all the eleven variables is presented. All the single variable and two variables subspaces are shown in a matrix juxtaposition view, so we can easily compare different subspaces. The entropy values for different subspaces are shown in the lower triangle of the matrix subplot. We can navigate through different subspaces and then a zoom-in view of the selected subspace is shown so that we can investigate the selected subspace in more detail. By investigating the lower entropy triangle, the Pressure and Temperature subspace is selected to explore because it has the highest entropy. Figure 7.(b) shows the exploration process

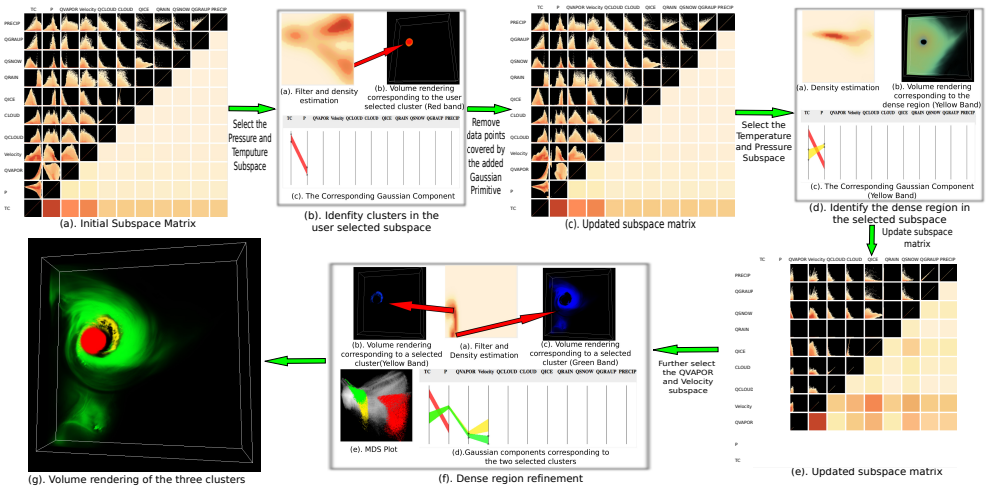


Figure 7: Experiments on the Hurricane Isabel dataset. Details about the exploration process are discussed in Section 6.1.

in the selected subspace. A filter operation is applied first to remove data points with scatter weights larger than 0.6, and then a weighted kernel density estimation is performed on the remaining data points with a blending coefficient 0.1 to find clusters that can be well defined in the current subspace. Figure 7.(b.a) shows the density estimation result. Roughly three clusters can be identified from the density estimation result. We can then click and select different dense regions to see the corresponding volume rendering result. In the example here, the Hurricane eye corresponding to one dense region is identified and we add its associated Gaussian component to the Gaussian Transfer Function. After the specified Gaussian component is added to the Gaussian Transfer Function, the data points that are covered by the Gaussian component are removed and the scatter plot matrix is updated as shown in Figure 7.(c). Then, we begin another exploration from the updated scatter plot matrix. We continue to explore the Temperature and Pressure subspace as this subspace still has the highest entropy. Figure 7.(d) shows the exploration process. Weighted kernel density estimation is directly applied to identify the dense region. Then, a Gaussian component is placed centered at the dense region to select this region. The Gaussian component is shown as the yellow band in the parallel coordinate. The volume region corresponding to the dense region is shown in Figure 7.(d.b). Figure 7.(e) shows the updated scatter plot matrix by removing the data points that are not in the selected dense region. Subspaces which contain Temperature and Pressure are removed. After updating, the lower entropy triangle is also updated. We then navigate to the QVapor and Velocity subspace which has the highest entropy to further refine the selected region in Temperature and Pressure subspace. Now, the subspace is extended to contain Temperature, Pressure, QVapor, and Velocity. The data points with scatter weights larger than 0.6 are filtered out first. Figure 7.(f.a) shows the density estimation result on the remaining data points with a blending coefficient 0.1. Two dense regions are identified from the result. The volume regions corresponding to those two regions are shown in Figure 7.(f.b) and Figure 7.(f.c). Their associated Gaussian components are shown in the parallel coordinate as the yellow and green bands. The two associated Gaussian components are now in four dimensions given the four-dimensional subspace. The regions covered by those three Gaussian components are shown in the MDS subplot as shown in Figure 7.(f.e). Figure 7.(g) shows the final volume rendering result by rendering all the three clusters together.

As we can see from the results, features in a multivariate dataset usually can be well defined with a few number of variables instead of all the variables. In the above example, the Hurricane eye is identified easily in the Pressure and Temperature subspace without considering the other 9 variables.

6.2 Turbulent Combustion Data Set

The Turbulent Combustion simulation data set is produced by Dr. Jackqueline Chen at Sandia Laboratories through US Department of Energy's SciDAC Institute for Ultrascale Visualization. The data has a resolution of $480 \times 720 \times 120$. There are five variables in this data set: Mixture Fraction of Hydroxyl Radical (OH), Mixture Fraction (MIX), Vorticity (VORT), Heat Release Rate (HR) and Scalar Dissipation Rate (CHI). Time step 41 was selected for our experiment.

Figure 8 shows an exploration process to identify interesting features that exist in the data. The initial scatter plot matrix is shown in Figure 8.(a). Initially, the subspace of MIX and OH is selected according to the lower entropy triangle. The data points with scatter weights larger than a user-defined threshold 0.9 are filtered out. Weighted kernel density estimation is then applied to the remaining data points with a blending coefficient 0.1 and Figure 8.(b.a) shows the result. Three dense regions could be identified from the density estimation result. Two dense regions are corresponding to MIX value around 0.5. The Mixture Fraction variable represents the portion of fuel and oxidizer. Its values ranges from 0 to 1 while 0 represents pure oxidizer and 1 represents pure fuel. The value of MIX generally quantifies different characteristics of the flame: a fully burning flame is characterized by the region which has a larger chemical reaction rate than the turbulent mixing rate. Local extinction occurs when the turbulent mixing rate exceeds the chemical reaction rate. As reported in [1], the stoichiometric mixture fraction 0.42 corresponds to the flame. The top two dense regions that are identified are corresponding to the flame area. As can be seen, the flame area is further divided into two regions based on the value of the OH. As described in [1], some areas of the region corresponding to stoichiometric mixture fraction might be only weakly burning, and might reignite independently or with the help of the neighboring burning flame elements. This area can be characterized by low radical, temperature and heat release rate. By taking into account OH, the flame is further divided into two regions: the region with a lower OH value is corresponding to a flat region while the region with a higher OH value is corresponding to a more turbulent region as shown in the volume rendering view. The third cluster is corresponding to low MIX and low OH region. Figure 8.(b.e) shows the Gaussian components for the three clusters. Figure 8.(c) shows the updated scatter plot matrix after adding those three Gaussian

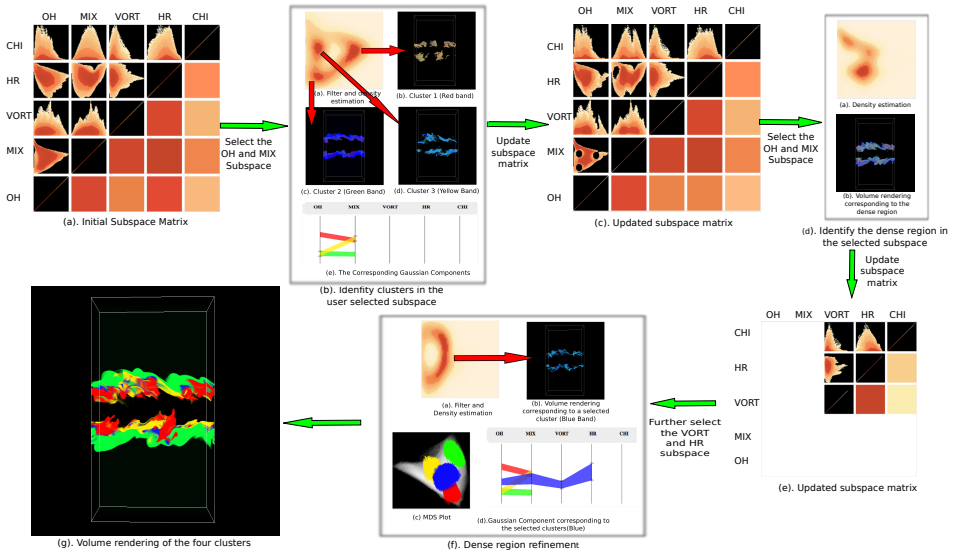


Figure 8: Experiments on the Turbulent Combustion dataset. Details about the exploration process are discussed in Section 6.2.

components to the Gaussian Transfer Function.

Next, we continue exploring the OH and MIX subspace according to the lower entropy triangle. Weighted kernel density estimation is directly applied and the result is shown in Figure 8.(d.a). Figure 8.(d.b) shows the volume region corresponding to the dense region. In order to further refine the dense region in a larger subspace, the scatter plot matrix is updated by removing the data points that are not inside the dense region. Figure 8.(e) shows the updated scatter plot matrix. The subspaces involve MIX and OH are removed. The lower entropy triangle shows the VORT and HR subspace has the highest entropy, so we navigate to the VORT and HR subspace to further refine the selected region in MIX and OH subspace. Now the subspace is extended to four dimensions. The data points with scatter weights larger than 0.7 are filtered out first. Figure 8.(f.a) shows the weighted density estimation result on the remaining data points with a blending coefficient 0.1. The volume region corresponding to the dense region is shown in Figure 8.(f.b). The corresponding Gaussian component is shown in Figure 8.(f.d) as the blue band. Notice that, now the Gaussian component is extended to four dimensions. The regions covered by those four Gaussian components are highlighted in the MDS subplot as shown in Figure 8.(f.c). Figure 8.(g) shows the volume rendering results by rendering all the four clusters together.

6.3 Ionization Front Instability Data Set

In the third case study, we use the Ionization Front Instability dataset. The data was created by Mike Norman and Daniel Whalen which aimed at understanding the effect of instabilities where radiation ionization fronts scatter around primordial gas [35, 36]. Ionization front is the region that separates gas in an ionized state from gas in a neutral state. The resolution of the data is $600 \times 248 \times 248$. It contains ten variables which include particle density, gas temperature and eight chemical species including the mass abundance of H, H^+, He, He^+, He^{++}, H^-, H_2, H_2^+.

To discover interesting features in the data, an exploration process is shown in Figure 9. The initial scatter plot matrix is shown in Figure 9.(a). First, we explore the gas temperature and H_2(gaseous hydrogen) subspace based on the lower entropy triangle. The data points with scatter weights larger than 0.6 are filtered out first. Figure 9.(b.a) shows the weighted density estimation result on the remaining data points with a blending coefficient 0.2. Roughly, two dense regions can be identified. One dense region that is corresponding to relative lower temperature and higher H_2 is shown in Figure 9.(b.b). Its associated gaussian component is shown as the red band in Figure 9.(b.c). This region is corresponding to the shocked state. Figure 9.(c) shows the updated scatter plot matrix after adding the Gaussian component to the Gaussian Transfer Function. Then, we explore the gas temperature and H^+(ionized hydrogen) subspace. The data points with scatter weights larger than a user-defined threshold 0.2 are filtered out. Weighted kernel density estimation is then applied on the remaining data points with a blending coefficient 0.1. Figure 9.(d.a) shows the result. Figure 9.(d.b) shows the volume region corresponding to the dense region. The temperature of this region is around 15,000K which is corresponding to the ionized gas region [26]. This region also has a high value for H^+ as shown in the transfer function view. Figure 9.(d.d) shows the regions that are covered by the two Gaussian components in the MDS view. Figure 9.(e) shows the volume rendering results of all the two clusters.

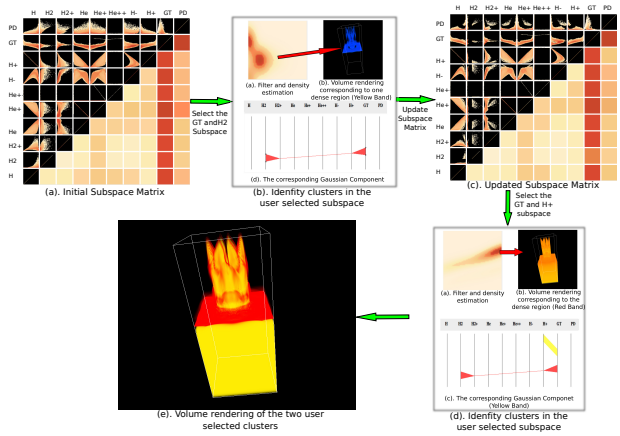


Figure 9: Experiments on the Ionization Front Instability dataset. Details about the exploration process are discussed in Section 6.3

7 Performance

Our system was implemented in C++. The interface was designed using QT and the rendering part was implemented with OpenGL. In

this Section, we measure and report the performance of our system. The machine we used has an Intel Core i7-4870HQ CPU @ 2.5GHz with 16 GB memory. We tested the performance of our system with three datasets: Hurricane Isabel, Combustion, and Ion Front. Table 1 shows the size of the three data sets used and the number of data samples before and after binning and aggregation. Table 2 shows the system setup time for those three datasets. The second column shows the number of samples generated after binning and aggregation. T_l represents the time to load data into memory. T_c denotes the time to do binning and aggregation and T_s represents the time to initialize the scatter plot matrix subplot. For the multidimensional scaling subplot, we precompute the low dimensional embedding and save it as a texture. During system startup, the time to initialize MDS subplot is negligible (less than 0.1 second), so we did not report this time here.

We also measured the system response time during exploration. T_{ds} and T_{pcp} denote the system response time when users brush on the density estimation and parallel coordinate subplot. These two brushing operations will change the mean and variance of the current Gaussian component, and then the points that are covered by the Gaussian component are computed. If more data points are covered, the longer the system will respond. In Table 3, we reported the time when brushing 100% percentage of the data points. T_{extend} denotes the system response time when performing the extending operation to extend to a larger subspace. The response time taken by the extending operation is related to the number of data points that are covered and the number of remaining variables. In our experiments, we report the time when roughly 50% data points are covered. T_{add} denotes the system response time when users add a specified Gaussian component to Gaussian Transfer Function. The time taken by this operation is influenced by the number of data points needed to be removed, the remaining number of data points and the remaining number of variables. In table 3, we reported a range instead of a single number to represent the system response time.

Data Set	Size(MB)	# Samples before binning and agg.	# Samples after binning and agg.
Isabel	1239.8	25,000,000	567,235
Comb.	791	41,472,000	1,669,290
Ion	1407.7	36,902,400	213,824

Table 1: Data Set

Data Set	# Samples	$T_l(sec)$	$T_c(sec)$	$T_s(sec)$
Isabel	567,235	1.38	13.25	1.02
Comb.	1,669,290	1.69	14.53	1.29
Ion	213,824	3.06	19.95	0.37

Table 2: The system setup time for Isabel, Combustion and Ion Front

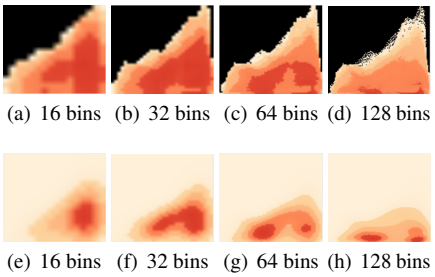
Data Set	$T_{ds}(sec)$	$T_{pcp}(sec)$	$T_{extend}(sec)$	$T_{add}(sec)$
Isabel	0.33	0.32	0.53	0.1 - 1.21
Comb.	0.97	1.09	1.01	0.07 - 3.93
Ion	0.11	0.13	0.18	0.11 - 0.406

Table 3: The system response time when users perform the different brushing operations

8 Discussion

8.1 Influence of Parameters

In our system, there are mainly three parameters to be determined by the users: the scatter weight threshold, the blending coefficient and the number of bins. In this section, we discuss the influence of those parameters on our system.



(a) 16 bins (b) 32 bins (c) 64 bins (d) 128 bins

(e) 16 bins (f) 32 bins (g) 64 bins (h) 128 bins

Figure 10: Weighted KDE results of the Temperature and velocity subspace of Isabel data set under different number of bins.

Scatter Weight Threshold: This parameter is used to filter out data points with too big of a scatter weight. In our interface, we provide a slider bar that can be used by users to change the value of the scatter weight threshold. When users move the slider bar, the filtered result will be shown instantly. Based on the result, users can select appropriate scatter weight threshold.

Blending Coefficient: As described in Section 4.2, the blending coefficient determines the influence of size weight. Users can adjust the value of blending coefficient using a slider bar. While users move the slider bar, the density estimation result will be shown instantly. In order to choose the blending coefficient, users can move the slider bar to see the animation of the density estimation result and observe interesting features.

The Number of Bins: This parameter determines the number of bins used for data binning. When a smaller value is used, more data points are aggregated and we are able to achieve high data reduction rate, but the features may not be well classified. When a larger value is used, features could be well classified but the data reduction rate is low and may impact the interaction ability of the system. Figure 10 shows the density estimation results under a different number of bins for the temperature and velocity subspace of the Isabel data set. The top row shows the subspaces under a different number of bins and the bottom row shows the corresponding density estimation results. As we can see, when the number of bins is very small such as 16, only one dense region is identified. As the number of bins increases, two dense regions are identified. When we choose the value for the number of bins, we need to consider the trade-off between classification ability and data reduction rate. In our experiments, we found 64 is a good value to use for those three data sets.

8.2 Limitation

One limitation of our approach is that as the number of variables increases, the number of subspaces needs to be shown in the scatter plot matrix also increases which causes the problem of visual scalability. When a large number of subspaces need to be shown, the size of the scatter plot matrix could be very large which makes it difficult to be shown in a limited screen size. This limitation prohibits our approach to scale to a large number of variables. One way to solve this limitation is to choose only a subset of subspaces to show in the scatter plot matrix subplot. For example, we could rank subspaces based on their entropy and then only show the top K subspaces.

9 Conclusion

In this paper, we present a novel system for users to analyze and visualize multivariate volumetric data through bottom-up subspace exploration. In our system, we use information derived from the data to assist user exploration. An entropy-based method is used to

help users to identify which subspace to explore. Size weight and scatter weight are proposed as a complementary information for the loss of high dimensional information due to projection. Based on those two weights, we employ weighted kernel density estimation and bottom-up subspace exploration in our system to assist users to explore and identify interesting features in the data. A Gaussian component is constructed to represent each user-selected cluster. Finally, all these selected clusters can be rendered together to reveal the feature of the data. In the future, we plan to extend our system to support time-varying data. It would be interesting to investigate whether the subspace exploration is helpful for feature detection and tracking in time-varying data.

ACKNOWLEDGMENTS

This work was supported in part by NSF grants IIS-1250752, IIS-1065025, and US Department of Energy grants DE-SC0007444, DE-DC0012495, program manager Lucy Nowell.

REFERENCES

[1] H. Akiba, K. l. Ma, J. H. Chen, and E. R. Hawkes. Visualizing multivariate volume data from turbulent combustion simulations. *Computing in Science Engineering*, 9(2):76–83, March 2007.

[2] S. Arens and G. Domik. A survey of transfer functions suitable for volume rendering. In *Proceedings of the 8th IEEE/EG International Conference on Volume Graphics*, VG'10, pp. 77–83. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2010.

[3] J. Blaas, C. Botha, and F. Post. Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1436–1451, Nov 2008.

[4] J. J. Caban and P. Rheingans. Texture-based transfer functions for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1364–1371, Nov 2008.

[5] C. Correa and K. L. Ma. Size-based transfer functions: A new volume exploration technique. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1380–1387, Nov 2008.

[6] C. Correa and K. L. Ma. The occlusion spectrum for volume classification and visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1465–1472, Nov 2009.

[7] H. Doleisch. Simvis: Interactive visual analysis of large and time-dependent 3d simulation data. In *2007 Winter Simulation Conference*, pp. 712–720, Dec 2007.

[8] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the Symposium on Data Visualisation 2003*, 2003.

[9] S. Fang, T. Biddlecome, and M. Tuceryan. Image-based transfer function design for data exploration in volume visualization. In *Visualization '98. Proceedings*, pp. 319–326, Oct 1998.

[10] 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, Sept 2012.

[11] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and M. E. Gröller. Volume visualization based on statistical transfer-function spaces. In *Visualization Symposium (PacificVis), 2010 IEEE Pacific*, pp. 17–24, March 2010.

[12] T. He, L. Hong, A. Kaufman, and H. Pfister. Generation of transfer functions with stochastic search techniques. In *Visualization '96. Proceedings.*, pp. 227–234, Oct 1996.

[13] J. Heinrich and D. Weiskopf. Continuous parallel coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1531–1538, Nov 2009.

[14] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[15] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Visualization, 1990. Visualization '90., Proceedings of the First IEEE Conference on*, Oct 1990.

[16] H. Janicke, A. Wiebel, G. Scheuermann, and W. Kollmann. Multifield visualization using local statistical complexity. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1384–1391, 2007.

[17] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, Oct 2005.

[18] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Volume Visualization, 1998. IEEE Symposium on*, pp. 79–86, Oct 1998.

[19] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: methods and applications. In *Visualization, 2003. VIS 2003. IEEE*, Oct 2003.

[20] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of the Conference on Visualization '01*, 2001.

[21] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, July 2002.

[22] J. Kniss, S. premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In *Visualization, 2003. VIS 2003. IEEE*, pp. 497–504, Oct 2003.

[23] A. H. König and E. M. Gröller. Mastering transfer function specification by using volumepro technology, 1999.

[24] S. Liu, B. Wang, J. J. Thiagarajan, P. T. Bremer, and V. Pascucci. Multivariate volume visualization through dynamic projections. In *Large Data Analysis and Visualization (LDAV), 2014 IEEE 4th Symposium on*, pp. 35–42, Nov 2014.

[25] R. Maciejewski, I. Woo, W. Chen, and D. Ebert. Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1473–1480, Nov 2009.

[26] S. Nagaraj and V. Natarajan. Relation-aware isosurface extraction in multifield data. *IEEE Transactions on Visualization and Computer Graphics*, 17(2):182–191, Feb 2011.

[27] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):893–900, Sept 2006.

[28] S. Oeltze, H. Doleisch, H. Hauser, P. Muigg, and B. Preim. Interactive visual analysis of perfusion data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1392–1399, 2007.

[29] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: A review. *SIGKDD Explor. Newsl.*, June 2004.

[30] D. Patel, M. Haidacher, J. P. Balabanian, and E. M. Gröller. Moment curves. In *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific*, pp. 201–208, April 2009.

[31] H. Piringer, R. Kosara, and H. Hauser. Interactive focus+context visualization with linked 2d/3d scatterplots. In *Coordinated and Multiple Views in Exploratory Visualization, 2004. Proceedings. Second International Conference on*, pp. 49–60, July 2004.

[32] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[33] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma. A novel interface for higher-dimensional classification of volume data. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, VIS '03, pp. 66–. IEEE Computer Society, Washington, DC, USA, 2003.

[34] Y. Wang, W. Chen, J. Zhang, T. Dong, G. Shan, and X. Chi. Efficient volume exploration using the gaussian mixture model. *IEEE Transactions on Visualization and Computer Graphics*, Nov 2011.

[35] D. Whalen and M. L. Norman. Ionization front instabilities in primordial h ii regions. *The Astrophysical Journal*, 673(2):664, 2008.

[36] D. Whalen, B. W. O'Shea, J. Smidt, and M. L. Norman. Photoionization of clustered halos by the first stars. *AIP Conf. Proc.*, 2008.

[37] X. Zhao and A. Kaufman. Multi-dimensional reduction and transfer function design using parallel coordinates. In *Proceedings of the 8th IEEE/EG International Conference on Volume Graphics*, 2010.

[38] L. Zhou and C. Hansen. Transfer function design based on user selected samples for intuitive multivariate volume exploration. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, 2013.

[39] L. Zhou and C. Hansen. Guideme: Slice-guided semiautomatic multivariate exploration of volumes. *Computer Graphics Forum*, (3), 2014.