Multi-Dimensional Transfer Function Design based on Flexible Dimension Projection Embedded in Parallel Coordinates

Hanqi Guo [1,2] He Xiao [1] Xiaoru Yuan [1,2] *

1) Key Laboratory of Machine Perception (Ministry of Education), and School of EECS, Peking University, Beijing, P.R. China
2) Center for Computational Science and Engineering, Peking University, Beijing, P.R. China

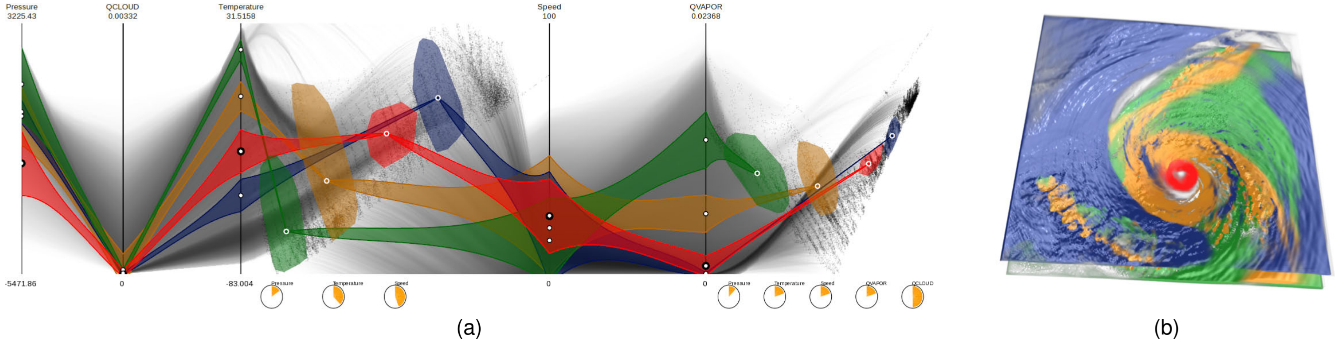


Figure 1: Volume visualization of multivariate data (Hurricane Isabel) with our proposed interface of transfer function design. (a) Multidimensional scaling plots are embedded in the parallel coordinates to facilitate feature selection without context switching. (b) Corresponding rendering result with the TF specified in (a).

Abstract

In this paper, we present an effective transfer function (TF) design for multivariate volume, providing tightly coupled views of parallel coordinates plot (PCP), MDS-based dimension projection plots, and volume rendered image space. In our design, the PCP showing the data distribution of each variate dimension and the MDS showing reduced dimensional features are integrated seamlessly to provide flexible feature classification for the user without context switching between different data presentations. Our proposed interface enables users to identify interested clusters and assign optical properties with lassos, magic wand and other tools. Furthermore, sketching directly on the volume rendered images has been implemented to probe and edit features. To achieve interactivity, octree partitioning with Gaussian Mixture Model (GMM), and other data reduction techniques are applied. Our experiments show that the proposed method is effective for multidimensional TF design and data exploration.

Keywords: Multivariate volume rendering, transfer function, parallel coordinates, dimension projection, user interface design.

Index Terms: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms I.4.10 [Image Processing and Computer Vision]: Image Representation—Multidimensional H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI)

1 Introduction

Transfer function (TF), which maps voxel values to colors and opacities, is the key to obtain informative and insightful visualization of volumetric data. Starting from the beginning of volume visualization, the community has contributed numerous works on TF design. With recent advances in methods, multidimensional TF design is more and more important. For scalar volumetric data, e.g. CT scan, adding derivative dimensions into TF design can enhance the classification results [18]. Features of interest can be selected by various widgets in 1D, 2D or even 3D feature space by trial-and-error [17, 18].

Multidimensional TFs are also necessary for multivariate volume data. From meteorological simulations to fMRI medical scans, domain experts need to take consideration of multiple dimensions and modalities. TFs of such multivariate volume data demand the power to explore correlation and identify features in high dimensional data space. However, the corresponding feature space is inaccessible to users without effective high dimensional feature space exploration tools.

For discrete multidimensional data, there are many well studied visualization and interactive exploration techniques in the domain of information visualization, including parallel coordinates plots (PCP), dimension embedding and reduction, and others. PCP [13, 14] transforms points from high dimensional space to 2D space in the form of polylines, keeping the information on each individual dimension, while simultaneously visualizing the correlations between the neighborhood axes. Dimension projection and reduction algorithms directly map high dimensional points into 2D. Among the dimension reduction techniques, Multi-Dimensional Scaling (MDS) [35], which keeps the distance metric of the high dimensional space, is widely applied for cluster identification and selection.

Many multidimensional TF designs have utilized traditional visualization methods, especially PCP [1, 2, 4, 39] and dimension projection [28, 9]. However, the direct extensions have limitations and drawbacks on TF design. For example, when PCP is used as TF design interface, users have to repeatedly adjust parameters on each dimension axis on the PCP alone. Additionally, some complicated features are not clear enough to be observed on the PCP. It seems to be easier to define a feature on the MDS plot according to

*e-mail:{hanqi.guo,xiaohe.pku,xiaoru.yuan}@pku.edu.cn

the local density of point clouds. However, it is hard to tell physical meanings from the clusters in a MDS plot, due to the information loss during the dimension embedding.

Inspired by SPPC (Scattering points into parallel coordinates) [36], which smoothly embeds MDS plots into PCP for multidimensional data visualization, we present a new multidimensional TF design leveraging the combined power of PCP and MDS. The proposed design flexibly integrates multiple interactive exploration space, including tone-mapped continuous PCP, weights-adjustable MDS plots, volume rendering view with sketch feedback, and more. TFs can be defined and modified in different views by various interactions, and all changes are reflected in other linked views. Compared with previous work, there are some significant differences. Zhao and Kaufman [39] utilized PCP for TF design, and show the classification result on the LLE projection plot in the same time. In our framework, we further closely integrate the linked view the PCP and MDS.

In our framework, we achieve high performance on multidimensional visualization for PCP, by exploiting various techniques. For example, octree decomposition, which reduces the data scale for real-time interaction and computation on the PCP and the MDS. Meanwhile, continuous PCP [11], which is superior to discrete PCP when dealing with volume data, can be generated very fast by leveraging octree representation of the volume data.

The contributions of this paper are two-fold. First, we present an efficient multidimensional TF design with tightly linked views of continuous PCP, metric adjustable MDS plots, and volume rendering view with feedback. Second, the adaptive rendering of hybrid continuous PCP and discrete MDS method is proposed.

The remainder of this paper is organized as follows. We summarize related works in Section 2, and the overview of the system in Section 3. Then we describe detailed algorithms in Section 4. Sample usage of the tool and the corresponding results are shown in Section 5. Conclusions are drawn in Section 6.

2 Related Works

Multi-dimensional TF design has become increasingly important in various applications [25]. For multi-modal volume data, one single meaningful rendering result can be obtained by multidimensional TF, instead of navigating the results of all the dimensions. A number of approaches have been proposed for better classification results by bringing derivative dimensions for TF design. For example, gradient magnitude can enhance the boundary information [20]. Similarly, curvatures [16], relative size [6], and ambient occlusion [7] can also be exploited to visualize more specific features. Statistical properties can also be extracted as new dimensions for TF design [24, 10]. By exploiting multi-dimensional classifications to combine derivative dimensions, insightful volume rendering results can be generated.

However, it is non-trivial to design multi-dimensional TFs without effective user interfaces. Some widgets are designed for 1D, 2D or 3D TFs under the guidance of histograms in feature space[17, 18]. For even higher dimensionality, the community has developed many alternative methods, including cluster space-based design, machine learning based algorithms, projection space-based methods and PCP-based user interfaces.

Cluster space-based TF design interface [31] enables users to work in cluster space, rather than in the TF space. Tzeng and Ma [31] utilized ISODATA clustering to classify the feature space. Furthermore, automatic TF design based on hierarchical clustering was proposed [27]. The cluster space-based method makes it convenient to design multidimensional TFs, but their performance relies heavily on the results of clustering, which involve very limited user participation. In addition to cluster space-based methods, machine learning is considered as an important technique for TF design [21]. Artificial Neural Networks (ANN) and Support Vector Machine (SVM) algorithms are applied to the TF design for multimodal volume data sets [29, 30]. Sketches on volume slices act as the sample input of the learning algorithms, and the output of the algorithm is the complicated fitting of multidimensional feature space. Although machine learning based methods are effective and straightforward, it is difficult to tell the physical meaning of the visualization results.

Dimension reduction and projection provide a similarity-based layout for the data. The distance in high-dimensional space is embedded into a lower-dimensional space. Utilizing dimension reduction and projection for TF design has been investigated in previous works [28, 9, 39]. The results of the projection in the form of point clouds are much easier to comprehend and select. The local clustering of the point clouds are likely to represent the features in the volume data. Various projection and dimension reduction methods have been exploited to facilitate multidimensional TF design, including ICA [28], PCA [26], SOM [9], LLE [39] etc. In our work, Pivot MDS is employed, which is very fast and stable, and it is suitable for interactive exploration.

Parallel Coordinates is widely used to visualize multivariate discrete data [13, 14], and some extensions on assisting TF design were proposed in recent years [1, 2, 4, 39]. In previous works, TFs are defined on the dimension axes by assigning numerical ranges [1, 4]. PCP are also used for feature selection in particle simulations [15], which is a similar process to TF design. Works by Zhao and Kaufman [39] utilized PCP for TF design, meanwhile showing the transferred voxel points on the dimension reduction plot. In our design, users can also pick up features on the projection plot, and reflect the TFs on the PCP, providing multiple linked explorative spaces for TF design. The simple integration of PCP and MDS plots has been applied to colormap design of multi-modal remote sensing images [38]. However, due to the inherent complexity and the huge amount of voxel data, improvements in several aspects are required before an extension to the TF design.

3 Overview

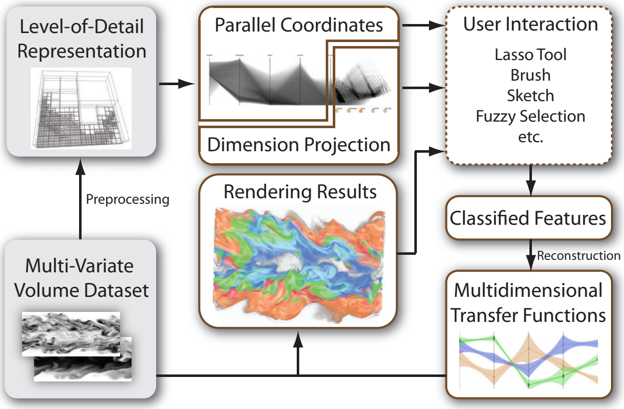


Figure 2: Algorithm overview of the proposed TF design system for multivariate volume data sets. A multivariate volumetric data set is first preprocessed with level-of-detail representation before rendered in the PCP and MDS plots. Users interact with the plots and classify features of the volume data. TFs then can be constructed based on the user clustering results. Further interaction can also be performed on the rendered images, and fed back to the TF design.

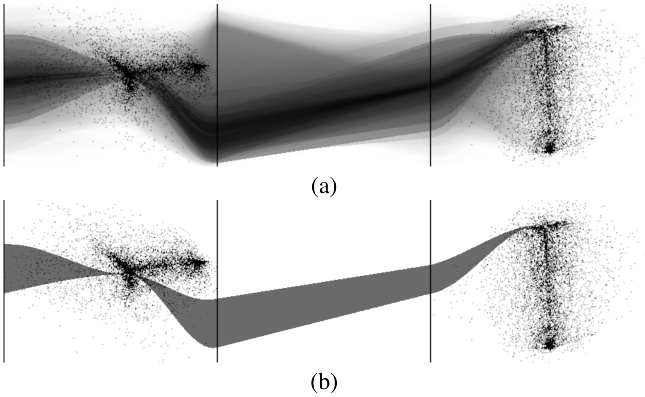
The main design philosophy we employ in this system is to provide multiple flexible explorative spaces to assist users to perform classification, while reducing context switching as much as possible during the operation. Specifically, the proposed system facilitates effective user interaction by providing closely integrated PCP and MDS plots for feature space exploration. Direct feedback from interacting with rendered volume images further enables insightful data investigation during the TF design.

The algorithm pipeline of our system is illustrated in Fig. 2. After the pre-processing of the input multivariate volumetric data sets, level-of-detail representation of the volumes is rendered as the continuous PCP with dimension projection plots. Based on the user classified features (clusters), TFs are automatically reconstructed. Meanwhile the classification results are also updated in all linked views.

3.1 Transfer Function Widget

The TF widget (Fig. 1 (a)) is composed of a TF layer (in color) and a PCP / MDS layer. In the top layer, the TFs are presented with ribbon metaphors in an illustrative manner [23], which is spatially more compact and informative. The illustrative PCP is also extended to the MDS plots, by linking the central points. Each ribbon represents one component of the TF. Its center and range is encoded by the position and the ribbon width. Users can adjust the ribbon centers by dragging the corresponding control points, and tune the width on each dimension directly. PCP and MDS plots are linked. Operations on TFs in any view are reflected on other plots in real time.

The bottom layer is the PCP embedded with MDS plots. Both PCP and MDS plots provide data clustering and data distribution information. Multiple MDS plots with different dimension weights can be embedded between neighboring axes of the PCP, or on the side of the PCP, visually linked by the curve bonds (Fig. 3). The clusters can be visually traced between the MDS plots and the PCP, without significant context switching.


(a)

(b)

Figure 3: Smooth linkages between the continuous PCP and the MDS plots: (a) the linkage of the whole domain; (b) the local shape of the linkage when a small set of samples is shown.

Various user interactions have been enabled in our TF widget. On the PCP, traditional brushing such as axis brushing and angular brushing are supported. Axis brushing helps select a certain range on one dimension, and angular brushing is suitable for filtering features with high correlations shown on the PCP by selecting line segments with similar slopes between two neighboring axes. In the MDS plot, features can be selected either by the lasso tool or by the magic wand tool, which picks up similar neighborhood features from the given point.

3.2 Volume Rendering View

In addition to the operations of the TF widget, direct sketching interactions are implemented on the volume rendering view (Fig. 4). Users can pick the query tool from the pie menu, and then sketch on the desired region of the rendered volume image. The corresponding TF component then selected and highlighted on the TF widget. The removal tool helps to quickly drop the component that is sketched on. The sketch interaction in our system can give instant feedbacks when reviewing the rendered results. Algorithm details are described in the next section.

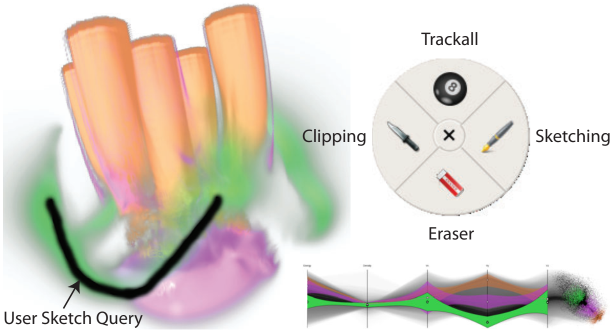


Figure 4: User interactions on direct volume rendering view. The green colored TF component is highlighted after a sketch on the rendering view.

4 Algorithm Details

In this section, we explain the key parts of the algorithms that sustain our interactive TF design, including the generation of PCP and MDS, the TF construction, and the feature selection on multiple coupled views.

4.1 Adaptive Continuous Parallel Coordinates

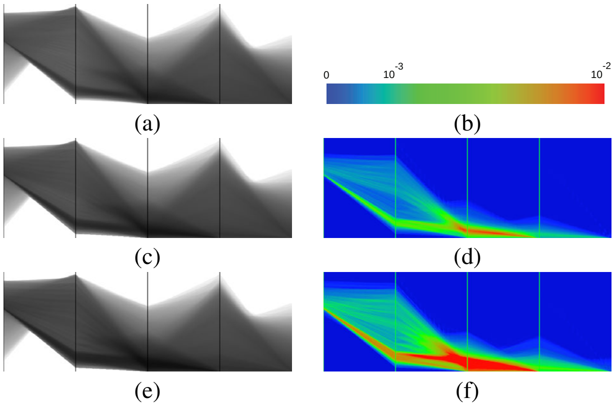
Traditional PCP maps discrete multi-variate data points as polylines intersecting parallel axes. However, for volumetric data which are defined over a continuous domain, we may not obtain correct or effective visualization if inappropriate sampling strategies are applied. The continuous PCP [11], which generates a dense presentation of PCP, is suitable as a representation for volumetric data. It gives users better insight into the multivariate volumetric data by filling the blank regions in the discrete scatterplot. However, there is no direct correspondence between the density map and the original discretely stored data when users brush regions on the continuous PCP. Other linked components in our framework, especially the MDS plots, still need discrete representation of the volume data. To solve this problem, an hybrid strategy based on octree is applied to generate continuous PCP, making it practical for operations in both continuous and discrete spaces.

The adaptive hierarchical PCP rendering strategy is similar to the adaptive continuous scatterplot rendering method [3]. The PCP is generated by accumulating the height map from quadrilateral strip primitives, and the rendering error is controllable. At the same time, the hierarchical structure largely reduces the input data amount by merging isotropic regions in the data set, which makes interactive exploration pissible. The sample points are also used for the generation of MDS plots (Section 4.2), as well as for TF construction (Section 4.5).

4.1.1 Hierarchical Data Partition and Adaptive Rendering of the Continuous PCP

In our system, we utilize octree to partition the raw data. In each octree leaf, the numerical properties can be seen as identical. When users want to pay attention to the overall distributions only, fast visualization results with less details can be provided. When exploring detailed features, more detailed sampling points can be added on demand. Hierarchical data sampling strategies provides flexibility and interactivity during data exploration with a controllable error level.

There are two steps to generate the adaptive continuous PCP. First, the octree leafs are filtered out according to the given error threshold during the traversal. Second, the filtered octree leafs are splatted and then accumulated onto a 2D plane as quadrilateral strips. The intensity of the quadrilaterals is decided by the volume of the corresponding octree leafs. The rendering error of the method is controlled by choosing different thresholds (Fig. 5). If the whole data is partitioned small enough, we can roughly expect that the result closely approaches to the accurate continuous PCPs.



(a) (b) (c) (d) (e) (f)

Figure 5: The comparisons of the adaptive PCP rendering at different error level: (a) 5%, (c) 10%, (e) 15%. The error images of (c) to (a), (e) to (a) are shown in (d) and (f) respectively, with pseudo colormap (b).

4.1.2 Tone-Mapping of the PCP

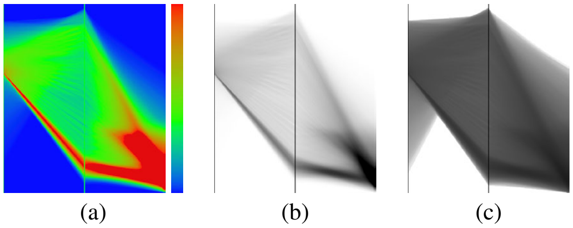
The accumulation result of the continuous PCP is a height map, with float precisions and very high dynamic ranges. If we employ a naive linear mapping to quantify the height map to 256 luminance levels for display on a scrren, many subtle yet important features will lost in the final image. Similar situations often occur in high dynamic range visualization applications [37]. To maintain and enhance the feature presentation on the PCP, a non-linear tone-mapping operator (TMO) is applied on the PCP height map. Because the range of the accumulated continuous PCP is always in $[0,1]$, we do not need to normalize the image before tone-mapping. The logarithmic transform that we utilize can be written as:

$$I' = 1 - \frac{ln((1-e^{-\alpha})I + e^{-\alpha})}{-\alpha}, \tag{1}$$

Where I' is the tone-mapped intensity, and I is the original intensity. Users can interactively change the parameter α to achieve different visual effects by scrolling the mouse wheel. The comparisons of linear and logarithmic tone-mapping are shown in Fig. 6. In Fig. 6 (c), the data distribution on the lower half of the leftest dimension is clearly visible, while such a cluster would be totally invisible if linear mapping was applied, as shown in Fig. 6 (b). All steps for adaptive rendering of continuous PCP are done with GPU acceleration, including the accumulation and the tone-mapping.

4.2 The MDS plots

As we discussed in previous sections, the MDS plots facilitate feature identification and feature selection based on the local density of the point clouds. Furthermore, the distance metrics of the MDS plots are adjustable. It enables the domain experts to define the different importance factors of the dimensions, supporting sub-dimensional space classification.



(a) (b) (c)

Figure 6: Result comparisons with linear and logarithmic tone-mapping: (a) pseudo color and the colormap on $[0,1]$; (b) linear; (c) logarithmic.

4.2.1 Pivot MDS Projection

An MDS algorithm with low storage and low computational complexity called Pivot MDS [5] is utilized in our system. The method of Pivot MDS is suitable for data sets with large amount of homogeneous items. We use the median values of each octree leaf as the sample points for the projection. The Pivot MDS algorithm can be summarized as follows:

- Randomly pick k pivot item from input data set;
- Construct a double-centered dissimilarity matrix $\mathbf{C}(c_{ij})$ between pivot items and all input items, whose elements are defined as

$$c_{ij} = -0.5(\delta_{ij}^2 - \frac{1}{n}\sum_{r=1}^{n}\delta_{rj}^2 - \frac{1}{k}\sum_{s=1}^{k}\delta_{is}^2 + \frac{1}{nk}\sum_{r=1}^{n}\sum_{s=1}^{k}\delta_{rs}^2), \tag{2}$$

- Calculate the eigenvalues and eigenvectors of the matrix $\mathbf{C}^T\mathbf{C}$
- Pick up the largest d eigenvectors $\{\mathbf{v}\}$, the low dimension embedding is achieved by

$$\mathbf{x}_i = C\mathbf{v}_i,\ i \in \{0,\ 1,\ 2,\ \ldots,\ d\} \tag{3}$$

where k is the number of pivot points, and n is the total number of data points. δ_{ij} stands for the dissimilarity of the ith item of input data with the jth pivot item.

4.2.2 Rotation Correction for MDS Plots

Due to the numerical instability and random factors in Pivot MDS, a rotation correction algorithm must be applied to ensure frame coherence during the tuning of the dimension weights. There are two steps in the algorithm. First, the same set of the pivots is used whenever the MDS plot updates, in order to avoid random factors related to the first step of the Pivot MDS algorithm. Secondly, a rigid transformation is applied to every new result, in order to minimize the rotations between the new result and the former one. We follow the least-squares registration algorithm [32]. More specifically, we search for the zoom factor c, rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ that satisfy following condition:

$$\arg\min E = \sum_{i=0}^{n} ||\mathbf{p}_i' - (c\mathbf{R}\mathbf{p}_i + \mathbf{t})||^2, \tag{4}$$

where $\{\mathbf{p}_i'\}$ and $\{\mathbf{p}_i\}$ are the newly generated and the original point cloud. Notice that we must use rigid transformation instead of affine transformation, because non-rigid transformation cannot keep the relative distances of the projected points. After finding the parameters by optimization, the transformed points $\{\mathbf{p}_i''\}$ are:

$$\mathbf{p}_i'' = c^{-1}\mathbf{R}^{-1}(\mathbf{p}_i' - \mathbf{t}). \tag{5}$$

Fig. 7 shows an example in which MDS plot in (b) and (c) flipped randomly when the weights of one dimension varied from 30% to 35% and 40% respectively. Whe we apply our rotation correction algorithm, the MDS plots are stable as shown in Fig. 7 (d) and (e). The whole process, including the MDS and rotation correction, can reach an interactive speed (Table 1).

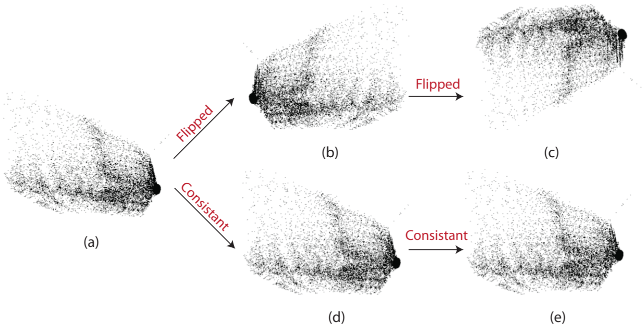


Figure 7: Rotation correction for the Pivot MDS results. The weightings modify the distance metric of the MDS, thus adjusting the MDS layout. The weighting of dimension "Temperature" is set to 30% in (a), and then it rises to 35% in (b) and (d), 40% in (c) and (e). MDS layout is flipped randomly with traditional methods as shown in (b) and (c). Sequence (a), (d), (e) keeps coherence with our rotation correction algorithm.

4.3 Embedding MDS Plots into PCP

SPPC [36] is adopted in our system, in order to keep visual continuity and avoid context jumps. We substitute the spline curves from SPPC with the quadrilateral strips. In this way, the accumulated intensity of the spline strips keeps coherence to the strips from the PCP, so the data can be tracked continuously from axis to axis. On PCP axes, the width of the spline strip of each octree leaf is set to be the range on the specific dimension, as described in Section 4.1.1. On the MDS plot, the strip length passing through the corresponding point is set to zero. Cubic splines connect the upper and lower boundaries on the axis to the projected point on the MDS plots. Several cases of smooth linkages are enumerated in Figure 3. Notice that the MDS plots can also be embedded on the side of the PCP, which is better for occlusion reduction, but not intuitive to present correlations. Users can freely decide and combine the components.

The embedding of MDS plots into the PCP brings multiple benefits. First, the combination provides strong visual hints of data clusters [36, 12]. Furthermore, the tool supports coordinated selection between the PCP and the MDS plots without the cost of context switching.

4.4 Feature Selection on the Linked Views

Our proposed framework provides several linked spaces for exploration, including the PCP, the MDS plots, and the volume rendering view. Each component has a distinctive way for feature selection. The interactions on the PCP is similar to previous work, e.g. axis brushing and angular brushing, so we mainly introduce here the details for the MDS plot and the volume rendering view.

4.4.1 Interactions on MDS Plots

As we mentioned in Section 3, there are two interaction methods available on the MDS plots, including the lasso and the magic wand tool. The lasso tool picks up the sample points inside the specified polygon. The magic wand tool automatically selects the neighborhood samples from a given seed point according to the local distribution as illustrated in Fig. 8.

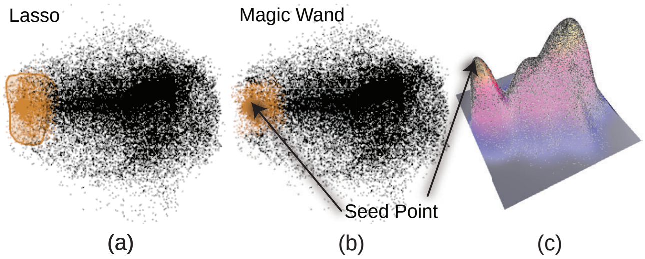


Figure 8: Two methods of user selection on projection plot: (a) the lasso tool, circling a group of points explicitly; (b) the magic wand tool, clicking on a seed position, and the selection is made by automatic diffusion based on the point cloud distribution as the corresponding height map shown in (c).

The computation of the magic wand selection is made up of two phases, including the height map construction and the diffusion process. The height map is essentially the construction of the feature space [22], which can be generated by GMM fitting or point splatting. A example result for height map construction is shown in Figure 8(c). For the diffusion process, we first find the nearest data point from the user defined position as the diffusion seed, and then utilize a BFS (breadth-first search) process that spreads the selection from the seed point to the final selection set by searching the neighborhoods. At every single step, the algorithm gives a fuzzy weighting depending on the similarity value with the seed point.

4.4.2 Interactions on Volume Rendering View

One of the key issues of direct operations on the rendering view is the intelligent speculation of user intention. In our design, users can only pick up what they see from the results, which aims to find out the closest Gaussian blob from the TF set. A ray can be traced from a certain sketch point, and we can calculate the visibility of each sample point along the ray. The visibility at position $\mathbf{x}$ can be written as [8]:

$$V(\mathbf{x}) = (1 - \alpha(\mathbf{s}(\mathbf{x})))O(\mathbf{x}), \tag{6}$$

where $\mathbf{s}(\mathbf{x})$ is the multidimensional sample at point $\mathbf{x}$, and α is the opacity TF. $O(\mathbf{x})$ is the occlusion value at $\mathbf{x}$. We assume that the user intention is very likely to fall on the local maxima of the visibility. We can also derive the ambiguity of the visibility local maxima A_i. Finally, the Gaussian blob selection i is decided by searching for the optimized value of A_iV_i. In order to accelerate the ray traversal, the octree traversal along the ray is implemented.

4.5 Transfer Function Construction

The construction process converts user selected features and assigned colors and opacities into TFs. The major algorithm for the construction in our system is the GMM, which approximates the input features. A series of Ellipsoid Gaussian TFs [19, 33] are generated as the construction results.

Suppose that the numerical distribution in every octree leaf is Gaussian. Ideally, we can then directly use the Gaussian blobs from selected leafs as TFs. However, due to limited computational capability, the sum of huge numbers of Gaussian function values for each sample point cannot be evaluated in real-time. We are able to find more efficient fittings for the user desired feature space with a relatively small numbers of mixtures, since many of the leafs selected will have very similar numerical distributions. We employ the GMM (Gaussian Mixture Model) to fit the feature space:

$$\sum_{k=1}^{m} \mu_k G_k(\mathbf{v}) = \sum_{i=1}^{n} \omega_i m_i G_i(\vec{\mathbf{v}}) + \varepsilon(\mathbf{v}), \tag{7}$$

where G_i, m_i, ω_i are the estimated Gaussian distribution of octree leaf i, the volume of the leaf, and the fuzzy selection coefficient, with the range of $[0,1]$. The left term of the equation is the Gaussian mixture, where m is the number of Gaussian blobs, and μ_k is the weighting for the Gassian blob G_k. The standard Expectation Maximization (EM) algorithm determines the parameters of the GMM model through iterations. The Gaussian mixtures can be transformed to Gaussian TF [19] by multiplying the opacity value α_{max}:

$$GTF(\mathbf{v}) = \alpha_{max} \sum_{k=0}^{m} \mu_k G_k(\mathbf{v}), \tag{8}$$

where $\mathbf{v}$ is the input data vector, and μ_k is the maximum opacity value for scaling each Gaussian bolb, which is obtained in Eq. 7. The final color $\mathbf{C}$ and the opacity α values are:

$$\mathbf{C} = \frac{\sum \alpha_i \mathbf{C}_i}{\sum \alpha_i}, \alpha = \sum \alpha_i, \alpha_i = \alpha_{max}\mu_i \tag{9}$$

where $\mathbf{C}_i$ and α_i are the color and opacity of each Gaussian component. Gaussian TFs are applied on-the-fly to each sample point in the standard GPU volume raycasting process.

5 Case Study and Analysis

In this section, we demonstrate the effectiveness of our multidimensional TF interface design through applications to several representative data sets as listed in Table 1.

Data Set	Size	D	T_{pre}	Leafs	T_{proj}	T_c	T_r
			(s)		(ms)	(ms)	(ms)
Isabel	500x500x100	5	224.0	55273	150	725.0	233
Multifield	300x124x124	12	90.6	40321	148	1045.5	833
Combustion	480x720x120	4	311.6	87176	230	188.8	227

Table 1: Data descriptions and timings. D is the number of dimensions in use; T_{pre}, T_{proj}, T_c, T_r are timings of pre-processing, projection, TF construction and frame update time respectively.

5.1 Atmospheric data set (Hurricane Isabel)

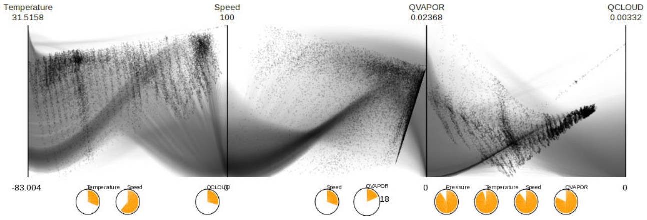


Figure 9: Different MDS layouts of Isabel data embedded in the PCP. The impact of each dimension is indicated by the small round widget at the bottom of each layout.

We apply the proposed TF design to an atmospheric data set from IEEE Visualization contest 2004 [1]. In our test, five dimensions are considered including pressure, temperature, wind speed, the water vapor mixing ratio (QVAPOR), and the cloud moisture mixing ratio (QCLOUD). The PCP and the MDS plots give several perspectives of overview of the data clusters, which are easy to be identified and selected by domain users. The MDS layouts can be adjusted by changing the dimension weightings, in order to strengthen or weaken the influences of the specific dimensions. Sub-dimensional space can also be navigated by selecting a small set of dimensions. Color and opacity properties of TF can be assigned to the desired features or clusters by brushing, lasso or magic wand tools on the TF widget. After further tuning the Gaussian blobs in the PCP view, insightful results can be generated. One rendering result of the Isabel data with our TF design is shown in Fig. 1. The TF consists of 4 TF components. Rendering results of each individual component are shown in Fig. 10. The red part of the result is the side region of the hurricane eye, where the pressure is low, and the values of wind speed and QCLOUD are medium; The outside feature with yellow color has a much higher pressure value but lower wind speed than the eye region. We can also see how different features are mixed. Many other patterns can be recognized from the visualization results.

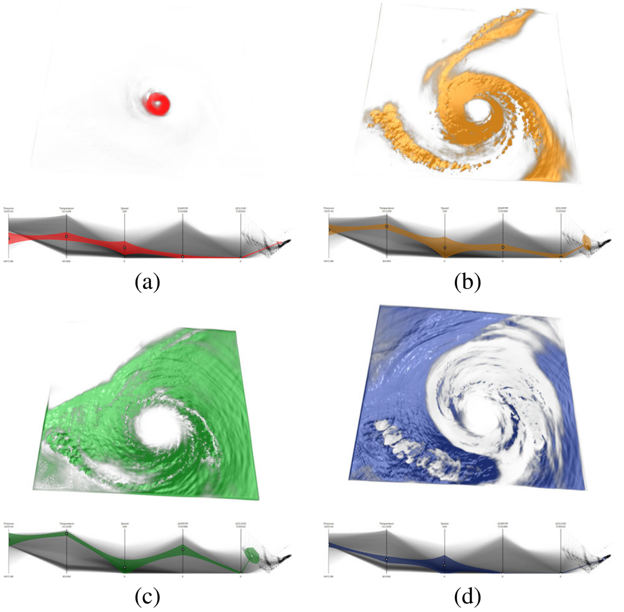


Figure 10: Atmospheric data visualization results with decomposed TFs in Figure 1(a). Each color represent a specific feature cluster.

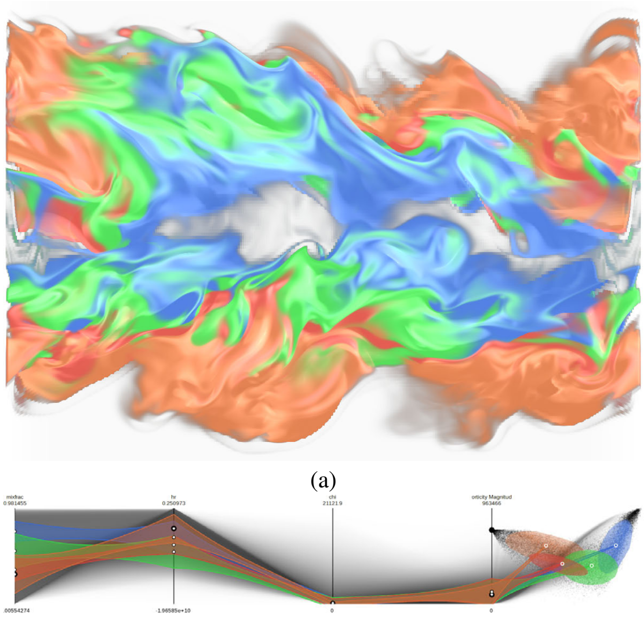
5.2 Turbulent Combustion Simulation

The second data set is from a turbulent combustion simulation [2] (Fig. 11). The four dimensions utilized in our case are stoichiometric mixture fraction (mixfrac), hydroperoxy radical (hr), dissipation rate (chi), and vorticity magnitude respectively. The mixfrac dimension depicts the mixture condition of two gases in the combustion process, and the hr gives hints of burning stage. By magic wand selection on the MDS plots, we exclude meaningless sample points of air and burned gas from the data, and generate a TF showing different parts and stages of the combustion. After adjusting the range of clusters on different axes in the PCP, the direct volume rendered image clearly shows meaningful surfaces in the region of interest.

5.3 Multifield 3D Scalar data

The multifield 3D scalar data set is from IEEE Visualization contest 2008 [34], which aims at understanding the ionization front instability. We construct a 12 dimensional TF, taking consideration of particle density, gas temperature, speed, vorticity magnitude, and mass abundance of various ions. The visualization results (Fig. 12) bring out some insights into the data. The green ionization front

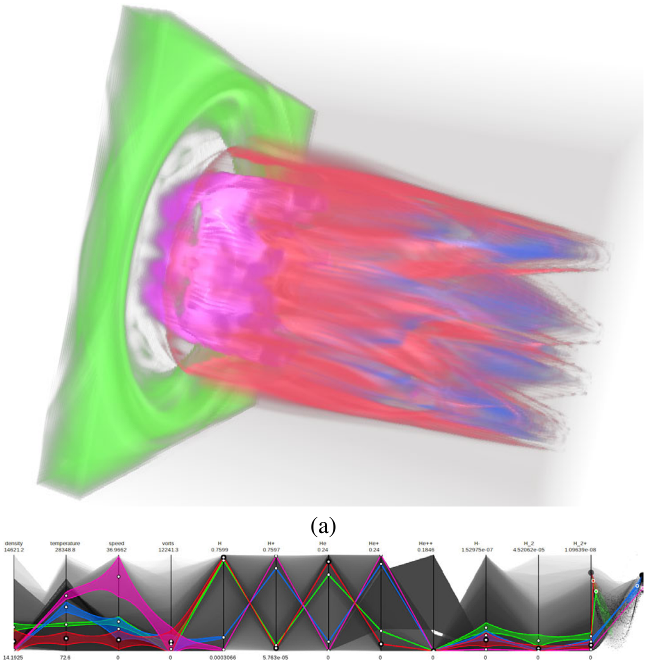
[1] http://vis.computer.org/vis2004contest


(a)

(b)

Figure 11: Combustion simulation data (4 dimensions): (a) the volume rendering result; (b) the corresponding TF.

has medium mass abundance of He and He^+, where users can conclude that there are some transformation process from He to He^+. This region has higher density, medium speed and vorticities. In the central area of the green region, there is a purple structure, where a much more intense process is happening. Scientists can conveniently define more insightful features using our TF design interface.


(a)

(b)

Figure 12: Multifield Data: (a) the volume rendering result; (b) the corresponding TF.

5.4 Discussions

We tested our method on three different multivariate data sets, including an atmospheric simulation, a combustion simulation and multifield scalar data. From the above results, we can see that our method is very flexible for multidimensional TF design. The data distribution is well presented in the TF view in multiple ways with insightful meanings. The Data distribution on each individual dimension can be tracked between the MDS plots and the PCP without significant context switching. The proposed method is very suitable for higher dimensionalities, which is shown in Section 5.3. If we use a traditional PCP-based TF design tool only, it is very hard to define meaningful features by adjusting ranges on each individual dimension axis. By embedding the MDS plots into the PCP, meaningful clusters can be quickly identified and highlighted only by a few steps of operations.

5.5 Implementation and Performance

The TF design system is tested on a Dell T3400 workstation, with a 2.66MHz CPU, 4GB Memory, and an Nvidia GTX 470 graphics card with 1280MB video memory. The system is implemented with OpenGL and the Cg shading language. The parameters of the Gaussian TFs are transferred to the shader programs in the form of textures. In the raycasting process, the values of Gaussian functions can be looked up from a pre-computed 1D texture, which is about 4 times faster than the on-the-fly evaluation of the exponential function. Now that the Pivot MDS and the TF construction are performed on CPU, we believe that performance will be further boosted by exploiting GPU acceleration for more sophisticated data in the future. The timings for the algorithms and the rendering are shown in Table 1. The raycasting step size used in the timing is 1 voxel. All the rendered image sizes are 640x480, and all the timings are tested with 4 Gaussian blobs. Although the pre-processing takes relatively longer time, it only requires running once. The run-time performance of the system is fast for user interactions.

6 Conclusions and Future Works

In this work, we present a novel user interface for multidimensional TF design, which integrates multiple data exploratory methods, including the PCP embedded with MDS plots, and the volume rendering view. The proposed method takes advantage of both PCP and MDS plots by providing numerical correlation and cluster layout simultaneously. Besides the high performance of the TF generation algorithms and other convenient user interaction techniques, domain experts can quickly make feature selections on any of the components, in order to generate insightful visualization results by undertaking only a few steps, meanwhile becoming aware of the distribution information about the data.

There are a few limitations in our work. First, a hybrid strategy of continuous and discrete data processing is applied, instead of a real continuous method. Continuous dimension projection is probably a future improvement, which would allow the data presentation to be continuous. Second, our tool relies on manual selection on features. More automatic feature detection techniques can be integrated into the systems to generate meaningful results for the users. Third, the illustrative presentation of Gaussian TFs cannot convey the covariance of Gaussian blobs on the PCP.

A few applications and extensions for this work can be developed in the future. First, our system has the potential capabilities on large scale time-variant data sets. Parallelism can largely boost the efficiency and scalability of our algorithm for larger and more complicated data in practice domain applications, by leveraging super computing and parallel IO techniques. For time-varying data, insightful results could be generated by exploiting some feature tracking and interpolation algorithms. The TF construction can utilize other base functions instead of the Gaussian TF. There are also many substitution data partition and reduction strategies other than octree. We also believe that some effective projection algorithms can be integrated into our framework for specific data sets, e.g. LLE or SOM, depending on the data features. Exploiting different distance metrics e.g. L^1-norm for dimension embedding could also be taken advantage of to reveal more meaningful features.

Acknowledgements

The authors wish to thank the anonymous reviewers for their comments. The authors would also like to thank the data providers, UCAR (Hurricane Isabel), SciDAC (Combustion Simulation), and SDSC (Multifield Scalar data). This work is supported by National Natural Science Foundation of China Project No. 60903062, Beijing Natural Science Foundation Project No. 4092021, 863 Program Project 2010AA012400, Chinese Ministry of Education Key Project No. 109001 and IIPL-09-016.

References

[1] H. Akiba and K.-L. Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proceedings of EuroVis 07: Joint Eurographics - IEEE VGTC Symposium on Visualization*, pages 115–122, 2007.

[2] H. Akiba, K.-L. Ma, J. H. Chen, and E. R. Hawkes. Visualizing multivariate volume data from turbulent combustion simulations. *IEEE Computing in Science and Engineering*, 9(2):86–93, 2007.

[3] S. Bachthaler and D. Weiskopf. Efficient and adaptive rendering of 2-d continuous scatterplots. *Computer Graphics Forum*, 28(3):743 – 750, 2009.

[4] J. Blaas, C. P. Botha, and F. H. Post. Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1436–1451, 2008.

[5] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *GD'06: Proceedings of the 14th international conference on Graph drawing*, pages 42–53, 2007.

[6] C. Correa and K.-L. Ma. Size-based transfer functions: A new volume exploration technique. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1380–1387, 2008.

[7] C. Correa and K.-L. Ma. The occlusion spectrum for volume classification and visualization. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1465–1472, 2009.

[8] C. Correa and K.-L. Ma. Visibility driven transfer functions. In *Proceedings of IEEE Pacific Visualization 2009*, pages 177–184, 2009.

[9] F. de Moura Pinto and C. M. D. S. Freitas. Design of multidimensional transfer functions using dimensional reduction. In *EuroVis07: Proceedings of IEEE VGTC Symposium on Visualization 2007*, pages 131–138, 2007.

[10] M. Haidacher, D. Patel, S. Bruckner, A. Kanitsar, and E. Gröller. Volume visualization based on statistical transfer-function spaces. In *Proceedings of IEEE Pacific Visualization 2010*, pages 17–24, 2010.

[11] J. Heinrich and D. Weiskopf. Continuous parallel coordinates. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1531–1538, 2009.

[12] D. Holten and J. J. V. Wijk. Evaluation of cluster identification performance for different PCP variants. *Computer Graphics Forum*, 29(3):793–802, 2010.

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

[14] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *Proceedings of IEEE Visualization 1990*, pages 361–378, 1990.

[15] C. Jones, K.-L. Ma, S. Ethier, and W.-L. Lee. An integrated exploration approach to visualizing multivariate particle data. *Computing in Science & Engineering*, 10(4):20–29, 2008.

[16] G. L. Kindlmann, R. T. Whitaker, T. Tasdizen, and T. Möller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of IEEE Visualization 2003*, pages 513–520, 2003.

[17] J. Kniss, G. L. Kindlmann, and C. D. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization 2001*, pages 255–262, 2001.

[18] J. Kniss, G. L. Kindlmann, and C. D. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Trans. Vis. Comput. Graph.*, 8(3):270–285, 2002.

[19] J. Kniss, S. Premoze, M. Ikits, A. E. Lefohn, C. D. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In *Proceedings of IEEE Visualization 2003*, pages 497–504, 2003.

[20] M. Levoy. Display of surfaces from volume data. *IEEE Comput. Graph. Appl.*, 8(3):29–37, 1988.

[21] K.-L. Ma. Machine learning to boost the next generation of visualization technology. *IEEE Comput. Graph. Appl.*, 27(5):6–9, 2007.

[22] R. Maciejewski, I. Woo, W. Chen, and D. S. Ebert. Structuring feature space: A non-parametric method for volumetric transfer function generation. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1473–1480, 2009.

[23] K. T. McDonnell and K. Mueller. Illustrative parallel coordinates. *Comput. Graph. Forum*, 27(3):1031–1038, 2008.

[24] D. Patel, M. Haidacher, J.-P. Balabanian, and M. E. Gröller. Moment curves. In *Proceedings of IEEE Pacific Visualization 2009*, pages 201–208, 2009.

[25] H. Pfister, W. E. Lorensen, C. L. Bajaj, G. L. Kindlmann, W. J. Schroeder, L. S. Avila, K. Martin, R. Machiraju, and J. Lee. The transfer function bake-off. *IEEE Comput. Graph. Appl.*, 21(3):16–22, 2001.

[26] C. Rezk-Salama, M. Keller, and P. Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Trans. Vis. Comput. Graph.*, 12(5):1021–1028, 2006.

[27] P. Sereda, A. Vilanova, and F. A. Gerritsen. Automating transfer function design for volume rendering using hierarchical clustering of material boundaries. In *Proceedings of EuroVis06: Joint Eurographics - IEEE VGTC Symposium on Visualization*, pages 243–250, 2006.

[28] I. Takanashi, E. B. Lum, K.-L. Ma, and S. Muraki. ISpace: Interactive volume data classification techniques using independent component analysis. In *Proceedings of Pacific Conference on Computer Graphics and Applications 2002*, pages 366–374, 2002.

[29] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma. A novel interface for higher-dimensional classification of volume data. In *Proceedings of IEEE Visualization 2003*, pages 505–512, 2003.

[30] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Trans. Vis. Comput. Graph.*, 11(3):273–284, 2005.

[31] F.-Y. Tzeng and K.-L. Ma. A cluster-space visual interface for arbitrary dimensional classification of volume data. In *VisSym 2004: Proceedings on Symposium on Visualization 2004*, pages 17–24, 338, 2004.

[32] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376 –380, 1991.

[33] Y. Wang, W. Chen, G. Shan, T. Dong, and X. Chi. Volume exploration using ellipsoidal gaussian transfer functions. In *Proceedings of IEEE Pacific Visualization 2010*, pages 25–32, 2010.

[34] D. Whalen and M. L. Norman. Competition data set and description, in 2008 IEEE Visualization Design Contest, 2008, http://vis.computer.org/VisWeek2008/vis/contests.html.

[35] P. C. Wong and R. D. Bergeron. Multiresolution multidimensional wavelet brushing. In *Proceedings of IEEE Visualization 1996*, pages 141–148, 1996.

[36] X. Yuan, P. Guo, H. Xiao, H. Zhou, and H. Qu. Scattering points in parallel coordinates. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1001–1008, 2009.

[37] X. Yuan, M. X. Nguyen, B. Chen, and D. H. Porter. HDR VolVis: High dynamic range volume visualization. *IEEE Trans. Vis. Comput. Graph.*, 12(4):433–445, 2006.

[38] X. Yuan, H. Xiao, H. Guo, P. Guo, W. Kendall, J. Huang, and Y. Zhang. Scalable multi-variate analytics of seismic and satellite-based observational data. *IEEE Trans. Vis. Comput. Graph.*, 16(6):1431–1420, 2010.

[39] X. Zhao and A. Kaufman. Multi-dimensional reduction and transfer function design using parallel coordinates. In *Proceedings of IEEE/EG International Symposium on Volume Graphics 2010*, pages 69–76, 2010.