

# Structuring Feature Space: A Non-Parametric Method for Volumetric Transfer Function Generation

Ross Maciejewski, *Member, IEEE*, Insoo Woo, Wei Chen, *Member, IEEE*, and David S. Ebert, *Fellow, IEEE*

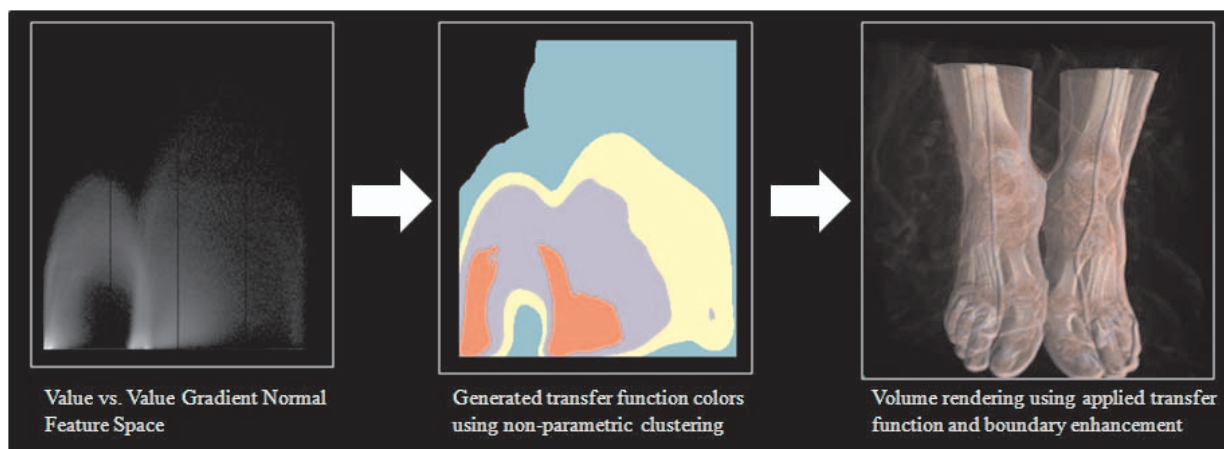


Fig. 1. An application of non-parametric clustering to the value vs. value gradient magnitude feature space using the CT visible woman feet dataset.

**Abstract**—The use of multi-dimensional transfer functions for direct volume rendering has been shown to be an effective means of extracting materials and their boundaries for both scalar and multivariate data. The most common multi-dimensional transfer function consists of a two-dimensional (2D) histogram with axes representing a subset of the feature space (e.g., value vs. value gradient magnitude), with each entry in the 2D histogram being the number of voxels at a given feature space pair. Users then assign color and opacity to the voxel distributions within the given feature space through the use of interactive widgets (e.g., box, circular, triangular selection). Unfortunately, such tools lead users through a trial-and-error approach as they assess which data values within the feature space map to a given area of interest within the volumetric space. In this work, we propose the addition of non-parametric clustering within the transfer function feature space in order to extract patterns and guide transfer function generation. We apply a non-parametric kernel density estimation to group voxels of similar features within the 2D histogram. These groups are then binned and colored based on their estimated density, and the user may interactively grow and shrink the binned regions to explore feature boundaries and extract regions of interest. We also extend this scheme to temporal volumetric data in which time steps of 2D histograms are composited into a histogram volume. A three-dimensional (3D) density estimation is then applied, and users can explore regions within the feature space across time without adjusting the transfer function at each time step. Our work enables users to effectively explore the structures found within a feature space of the volume and provide a context in which the user can understand how these structures relate to their volumetric data. We provide tools for enhanced exploration and manipulation of the transfer function, and we show that the initial transfer function generation serves as a reasonable base for volumetric rendering, reducing the trial-and-error overhead typically found in transfer function design.

**Index Terms**—Volume rendering, kernel density estimation, transfer function design, temporal volume rendering.

## 1 INTRODUCTION

A common method for direct volume rendering is to employ the use of interactive transfer functions as a means of assigning color and opacity to the voxel data. One of the most popular transfer function design tools is the interactive 2D histogram widget introduced by Kniss et al. [10]. In this widget, the user is presented with a 2D histogram (the axes of which represent a feature space of the data) and various selec-

tion tools are used to assign color and opacity to the voxels through an interactive brushing of the feature space. However, the appropriate selection of features in multi-dimensional transfer functions is a difficult task, often requiring the user to have an underlying knowledge of the data set under exploration. Moreover, normal transfer function design widgets typically include rectangular bounding boxes or other linearly defined area specification tools for transfer function definition. Unfortunately, features located in the multi-dimensional space may have complex shapes requiring extensive interaction to extract local boundaries. Also problematic is the fact that within the volume feature space, the features of interest are often not clearly visible as peaks or valleys. Furthermore, in temporal volumetric data sets, the feature space is changing across time, meaning that the structures captured with a transfer function in one time step may not be captured by the same transfer function in the subsequent time step.

In this work, we propose enhancing the conventional 2D histogram transfer function editing scheme through the application of a non-parametric density estimation in order to visualize and extract arbi-

- Ross Maciejewski, Insoo Woo and David S. Ebert are with the Purdue University Rendering and Perceptualization Laboratory, E-mail: {rmacieje|iwoo|ebertd}@purdue.edu respectively.
- Wei Chen is with the State Key Lab of CAD&CG, Zhejiang University E-mail: chenwei@cad.zju.edu.cn.

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.

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

trarily sized and shaped clusters within the given feature space. We apply a nearest neighbor kernel density estimation technique, utilizing the Epanechnikov kernel [21], to the feature space histogram. The estimated density is then used to assign a color and/or opacity to the data by placing each estimated density value into a 1D histogram bin. Each bin corresponds to a given color, and the opacity can then be assigned across the density values. As users manipulate the density binning, regions within the feature space expand and contract, providing contextual information about the structures hidden within the feature space. Users can then interactively select arbitrarily shaped regions within the data feature space and those selected regions form the transfer function used for volume rendering. In this manner, users are able to easily extract regions found within the given feature space that correspond to their regions of interest within the volume. A sample result of our method is shown in Figure 1.

We further extend this scheme to temporal volumetric datasets. In this case, we extract the corresponding feature space histogram of each volumetric dataset from each time step, and create a volume of the feature space, with one axis of the volume representing time and the other two axes representing the feature space dimensions. Kernel density estimation is then applied within this 3D space. At each time step, the user is presented with only the single slice of the resultant histogram volume that corresponds to this time step; however, from the density estimation, temporal coherence is now embedded within this slice. A second volume rendering of the histogram volume is also presented to the user, and as a user selects a feature within the 2D histogram slice, the temporal structure across the feature space volume is revealed. As such, the selection within a slice of the feature space volume propagates a transfer function across the entire temporal feature space, creating a 3D transfer function that will maintain temporal coherence. The user may now step through time and watch the selected feature evolve in both the temporal feature space and the volumetric space. In this manner, once a feature of interest is identified in one time step of the volume, the transfer function will be extended temporally and track this feature as it changes.

## 2 RELATED WORK

Early work by Kindlmann et al. [8] and Kniss et al. [10] applied the idea of a multi-dimensional transfer function [12] to volume rendering. This work identified key problems in transfer function design, noting that many interactive transfer function widgets lack the information needed to guide users to appropriate selections, making the creation of an appropriate transfer function essentially trial-and-error which is further complicated by the large degrees of freedom available in transfer function editing. While many volume rendering systems have adopted multi-dimensional transfer function editing tools, the creation of an appropriate transfer function is still difficult as the user must understand the dimensionalities of the feature space that they are interacting within.

As such, much work has been done on providing users with a simplified means of transfer function creation. Fang et al. [7] defined transfer function design as a set of 3D image processing tools. Takanashi et al. [24] utilized independent component analysis and multi-dimensional histograms to classify areas in the volume domain. By interactively clipping regions in the component space, users were able to effectively generate opacity transfer functions in the data. Work on statistically quantitative volume visualization by Kniss et al. [11] noted that nearly all image data share the characteristic that samples are spatially correlated, and focused on a general statistical classification for feature extraction. Rezk-Salama et al. [17] utilized principal component analysis for transfer function design, and Weber et al. [27] utilized topology-based methods to extract features from volumetric data and applied separate transfer functions for contour viewing. Wu and Qu [29] proposed an intelligent scheme that allows users to create multiple transfer functions and use the resultant volume renderings to add or subtract regions to create enhanced focus plus context views. Similarly, an interactive feature brushing interface was proposed by Ropinski et al. [19] to facilitate automatic generation of a 1D transfer function.

Unfortunately, the projection of the volumetric data properties down to a 2D space will obscure features, making it difficult to separate regions of interest within the volume with only a 2D transfer function. In order to overcome these difficulties, much research has focused on enhancing transfer function design through the addition of other data properties. Examples include work by Kindlmann et al. [9], which employed the use of curvature information to enhance multi-dimensional transfer functions, and Tzeng et al. [25], which focused on higher dimensional transfer functions which use a voxel's scalar value, gradient magnitude, neighborhood information and the voxel's spatial location. Work by Potts et al. [16] suggested visualizing transfer functions on a log scale in order to better enhance feature visibility. Lundstrom et al. introduced the partial range histogram [13] and the  $\alpha$ -histogram [14] as means for incorporating spatial relations into the transfer function design. Correa et al. introduced size based transfer functions [5] which incorporate the magnitude of spatial extents of volume features into the color and opacity channels and visibility based transfer functions [6] where the opacity transfer function is modified to provide better visibility of features.

While such extensions enhance the volume rendering and provide a larger separability of volumetric features, they still fail to provide users with information about the structures within a given feature space. In fact, the addition of more dimensionality into the transfer function is often automatically incorporated into the rendering parameters, obscuring the relationship between the volumetric properties and the volume rendering. However, the addition of more dimensions should provide more coherency between transfer functions, particularly in the case of temporal volumes. Recent work by Akiba et al. [1, 2] utilized parallel coordinate plots to create a volume rendering interface for exploring multivariate time-varying datasets. By means of a prediction-correction process, Muelder and Ma [15] proposed to predict the feature regions in the previous frame, making the feature tracking coherent and easy to extract the actual feature of interest.

## 3 NON-PARAMETRIC TRANSFER FUNCTION GENERATION

Our work focuses on providing the user with information about the volume data within the confines of the transfer function feature space. Given a 2D histogram representing some subset of an  $n$ -dimensional space, not all features can be separated. In fact, the projection of  $n$  dimensions to a 2D space ensures that some features will be obscured. However, one can identify the neighborhood features within this two dimensional projection, providing the users a context in which to explore their data. These identified features can be represented as clusters of color based on the neighborhood properties, and users may then explore the effects that growing or shrinking these neighborhoods (as applied transfer functions) will have on the volume rendering.

### 3.1 Kernel Density Estimation

In order to create these neighborhoods and provide contextual information about the feature space to the user, we employ the use of a variable kernel method [21, 26] formed in Equation 1. A non-parametric clustering approach is chosen in order to remove the need for *a priori* knowledge about the number of clusters within a dataset (as opposed to k-means clustering where the user specifies the number of clusters). Furthermore, we utilize an adaptive kernel which scales the parameter of the density estimation by allowing the kernel radius to vary based upon the distance from each point,  $X_i$ , to the  $k$ th nearest neighbor in the set comprising the  $N - 1$  data points of the histogram feature space.

$$\hat{f}_h(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{d_{i,k}} K\left(\frac{\mathbf{x} - X_i}{d_{i,k}}\right) \quad (1)$$

Here,  $\hat{f}_h(\mathbf{x})$  is the probability density estimate of the histogram ( $h$ ) at a given location,  $\mathbf{x}$ , in the feature space,  $d_{i,k}$  represents the multi-dimensional smoothing parameter and  $N$  is the total number of samples in the histogram (i.e., the number of voxels in the volume). The window width of the kernel placed on the point  $X_i$  is proportional to  $d_{i,k}$  (where  $d_{i,k}$  is the distance from the  $i$ -th sample to the  $k$ -th nearest neighbor) so that data points in regions where the data is sparse will

have flatter kernels. We choose  $k = \lfloor \sqrt{N} \rfloor$  as this tends to approximate the optimal density estimation fitting (this is a rule of thumb approximation [21]). Such a method groups the data based on their neighborhood information, allowing us to visualize the underlying structure of the data.

In order to reduce the calculation time, we have chosen to employ the Epanechnikov kernel, Equation 2.

$$K(\mathbf{u}) = \frac{3}{4}(1 - \mathbf{u}^2)1_{(\|\mathbf{u}\| \leq 1)} \quad (2)$$

The function  $1_{(\|\mathbf{u}\| \leq 1)}$  evaluates to 1 if the inequality is true and zero for all other cases. Other kernels choices (i.e., Gaussian) could be utilized; however, the Epanechnikov kernel is the kernel of choice for fast calculations as it is less computationally expensive than a comparable Gaussian kernel [21]. Future work will allow users to choose from a variety of kernels including uniform, Gaussian, etc.

Currently, For a  $256 \times 256$  histogram, showing the feature space of a  $256^3$  volume (i.e.,  $N = 256^3$ ) interactions with the kernel density feature space tool results in a volume rendering frame rate of approximately 10 fps on an Intel Xeon 3.00 GHz processor with 2.0 GB or ram using an Nvidia 8800 card. Note that the size of the histogram bins will directly affect the kernel density estimation results. As the histogram bins become larger (i.e., the histogram dimensions get smaller), more voxel values will map to the same histogram bin. This results in the adaptive kernel radius tending towards a value of one, and the resultant density estimation becomes more discrete. In general, the smaller the histogram bins in a 2D transfer function window, the easier it will be to separate values as less voxels will map to the same coordinate.

Figure 1 illustrates this process (in the 2D case) using the CT visible woman feet. Here, the user is presented with the typical 2D histogram widget in the value vs. value gradient magnitude feature space. Each entry in the feature space histogram contains information on the number of voxels that map to these features. Given this information, we use the variable kernel method (Equation 1), to create a continuous density distribution function across the entire feature space. This method would produce comparable results to the continuous scatterplots work by Bachthaler and Weiskopf [3] if a logarithmic color scale were to be applied and a finer grid was used in the density estimation; however, our method provides information about clustered structures within the 2D histogram feature space where as the continuous scatterplot method does not. The values obtained in this procedure now represent a probability that a voxel will be found at a location in feature space. We then create a 1D histogram of the probability distribution and color the feature space based on the chosen binning, thereby generating our transfer function. The transfer function is then applied to color the volume, and boundary enhancement (see Section 3.2) is applied for the opacity. Note that the arc-like structures (which represent material boundaries) are preserved, indicating that they share similar distributions of points within the feature space. By creating an automatic coloring of the image, we not only provide information about the underlying feature space structures, but we also provide the user with a reasonable starting transfer function, reducing the trial-and-error exploration time typically required for transfer function creation. The resultant rendering is able to effectively color the various structures within that feature space (note that this method cannot separate overlapping features as it still only uses two-dimensions and the projection of the data down to two-dimensions can result in a combination of features). From there, the user may interact within the clustered transfer function space, refining the number, size and color of the clusters. As such, this initial transfer function generation is able to provide users with a solid starting point for feature space exploration, and users are then able to quickly navigate through the feature space to segment out regions of interest, greatly reducing the time taken to create a transfer function.

Note that the choice of the number of histogram bins for the KDE coloring is not the same as choosing the number of clusters in a k-nearest neighbor approach. In the case of the non-parametric clustering, we have simply mapped a range of density values to a discrete color. Notice that in Figure 2 the red bin corresponds to two unique

clusters. However, this discretization is also a limitation in the system as the user must individually selected each red cluster for visualization in order to determine which feature(s) of the volume the cluster maps too. Examples of this selection and exploration can be seen in Figure 4.

Related work has been done by Shamir [20] where feature-space analysis was applied to unstructured meshes in order to automatically identify structures within the volume. This work utilized a mean shift procedure for clustering which is similar to kernel density estimation. This clustering is performed across a five dimensional space (the  $x$ ,  $y$  and  $z$  components of the volume and the value vs. value gradient magnitude feature space), where as we only perform this on the feature space. However, the goal of Shamir's work was volume segmentation as opposed to transfer function design and interactive feature extraction. Our work provides users with interactive tools to segment the feature space of a volume, and provides a means in which to explore and understand the relationships between a given feature space group and the related volumetric data.

Work by Roettger et al. [18] also generated transfer functions through feature space analysis. Automatic setup of multi-dimensional transfer functions is enabled by adding spatial information to the histogram of the underlying dataset. The approach taken in this paper differs as we focus on the neighborhoods within a feature space to find natural borders of the feature space point densities and project these feature clusters back to the volumetric domain. While our method is currently only applied to the 2D feature space (or 3D temporal feature space), an extension to  $n$ -dimensions is minor, and the incorporation of variable neighborhood information for clustering should improve upon the work by Roettger et al. Furthermore, our procedure also smoothes the noise in the data, interpolating the 2D histogram at all positions through the kernel density estimation procedure. In medical datasets, Roettger et al. noted that their clusters have added noise and segmentation filtering is necessary. In contrast, our interactive widgets allow users to adjust the cluster boundaries, allowing for greater filtering precision as demonstrated in Section 4.

### 3.2 Coloring and Opacity

As previously stated, the coloring of the transfer function is based on 1D histogram binning of the probability values calculated during the kernel density estimation procedure. Figure 2 illustrates the 1D histogram created based on the probability density estimation method defined in Section 3.1. Values on the left of the histogram represent low density areas, while values on the right represent high density areas. Users are able to interactively select the number of bins (and their color). The default color scheme utilizes a qualitative color scheme [4] with a default of twelve color bins (although other schemes could also be applied). In Figure 2, we can observe the effect of increasing or decreasing the number of bins. Users may also increase or decrease the number of bins by entering into a 'join' mode in which they can click on any two histogram bins and they will merge together, with the resultant color being that of the first bin clicked. In joining bins, the effect is to increase the width of a histogram bin, thereby increasing the number of voxels mapped to a given color. This will result in neighboring rings joining together into a single area. Users may also separate bins by right clicking within a bin. The two resultant bins will remain the same color until the user chooses to assign a new color.

Allowing the users to interactively change the number of bins can create more meaningful clusters. For example, in the Feet dataset, one may expect to find four materials, namely, air, bone, skin and soft tissue. By starting with four bins (as opposed to twelve), structures will group together in a more intuitive manner. However, it is important to note that the number of clusters seen can be greater than the number of bins selected. These clusters represent peaks and valleys within the feature space; therefore, clusters of colors appear in various regions across the feature space as the point density changes.

In order to enhance data exploration within the feature space, our histogram widget also allows users to interactively adjust the size of the bins. Users may choose from initial settings of either equally spaced or exponentially sized bins, or users may use the mouse to



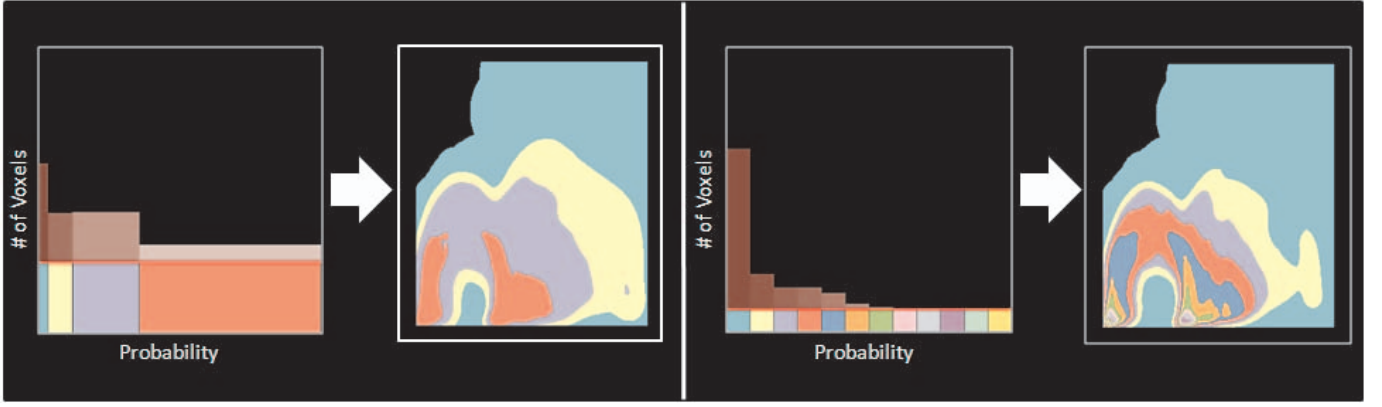


Fig. 2. Two examples of clustering done by binning the probability density estimates of the histogram data. This example uses the value versus the value gradient magnitude feature space of the visible woman feet.

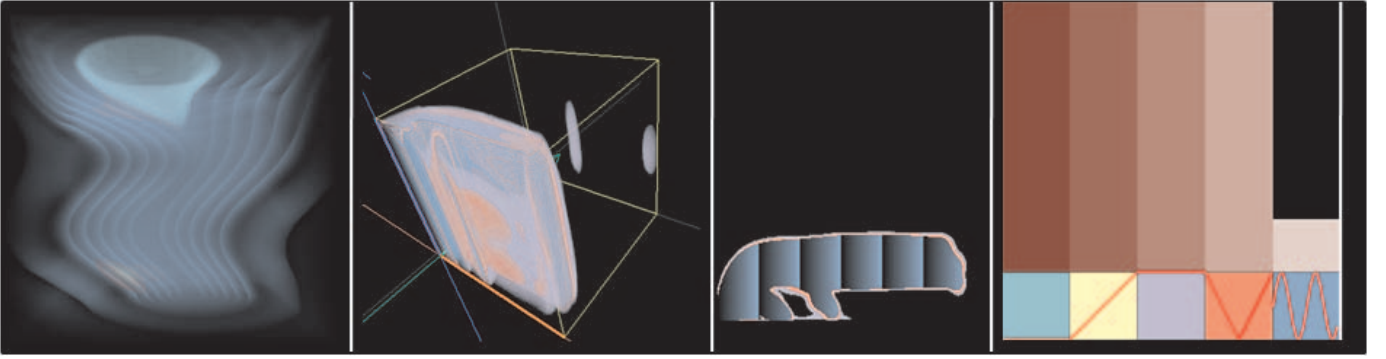


Fig. 3. Temporally coherent transfer function volume: (Left) A single time step rendering of the velocity fields in the tornado volume, (Left-Center) The transfer function volume, (Right-Center) A clustered 2D histogram extracted from the volume at time step  $t$ , (Right) The 1D histogram of the transfer function volume showing various opacity applications in the color bins.

expand the boundary of any histogram bin. As the bin grows or shrinks, regions in the transfer function also grow and shrink, and these changes are automatically reflected in an interactive volume rendering window.

While the color selection is done semi-automatically through the non-parametric clustering, we utilize a boundary enhancement [22] method for modulating the opacity. This choice was made based on the fact that areas with high gradients often represent boundaries between materials. In order to more effectively illustrate these areas, one may simply increase the opacity proportional to the gradient magnitude such that

$$o_e = o_o \cdot |\vec{\nabla}(P)|^\alpha \quad (3)$$

where  $\vec{\nabla}(P)$  is the gradient at the sampled point  $P$ ,  $o_o$  is the original sample opacity,  $o_e$  is the enhanced sample opacity, and  $\alpha$  is the boundary exponent which serves to change the sensitivity of the magnitude of the rendered boundary. The application of our density estimation transfer function coupled with boundary enhancement typically leads to high-quality renderings with no user interaction. To clarify, the rendering in Figure 1 was generated completely automatically. Once an initial rendering is generated, the user may begin exploring their data in the feature space; meanwhile, by providing such a starting location, we have vastly increased the user's ability to understand the relationships between the feature space domain and the volumetric domain.

Along with boundary enhancement, our control widgets also provide several schemes for manually adjusting the opacity values. In the 1D density estimation histogram distribution, the user may enter the opacity selection mode and modify how the opacity is mapped across the color bin. The red line found in the histogram bins indicates the opacity value across the histogram space. The top of each bin represents an opacity of one, and the bottom zero. Users may select from

one of seven default opacity curves (variations of linear, triangular, and sinusoidal) by clicking in the histogram bin. Figure 3 (Right) shows the modified opacity curves in the histogram bin. Future work will contain an interface for user defined opacity curves.

### 3.3 Feature Space Exploration

In order to explore the data feature space, we provide users with the ability to select any color region in the given 2D histogram window. Area selection will effectively create a mask for the transfer function, and only voxels located within the region selected will be rendered. Users may select/deselect a region simply by clicking in the feature space, and if the user chooses to expand these areas through the use of the density estimation histogram controls, the selection will follow this expansion. Users may also combine areas in the feature space by entering a combination mode. In this mode, the user clicks on one area in the feature space, and then another. This will result in the second area being colored the same as the first area. However, such a combination disables the density estimation histogram mapping. For example, Figure 2 has two distinct red areas in the density estimated transfer function. If one of those areas were combined with the purple area, there is no guarantee that such a combination equally splits the red bin. In fact, it is more likely that both red areas share a combination of values across the entire red bin. As such, one histogram color bin does not necessarily represent a single neighborhood in the feature space. These features allow for interactive region exploration and combination, letting the user quickly search for neighborhoods of interest within the feature space. In depth examples of the use of these modes can be found in Section 4.

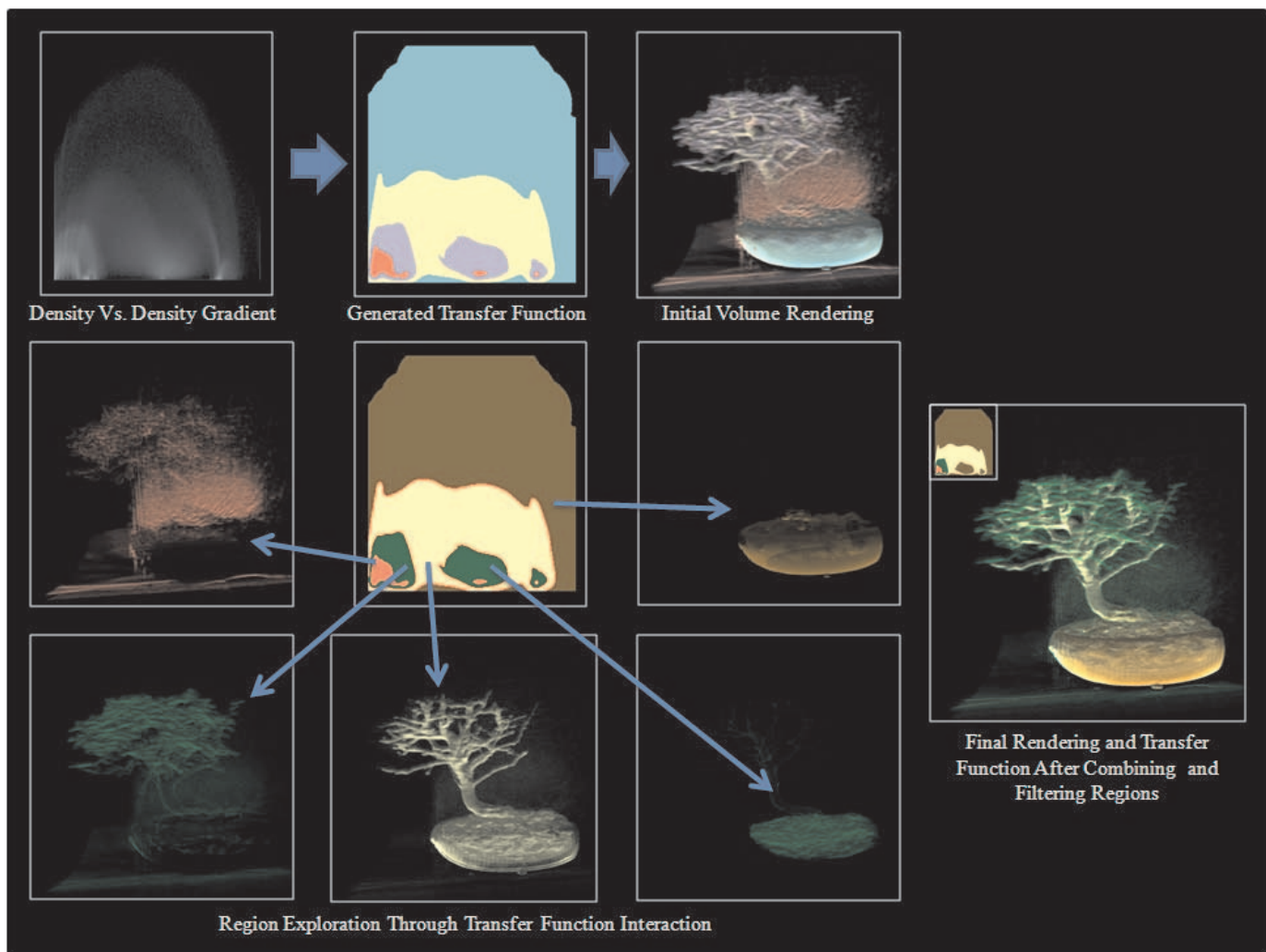


Fig. 4. Feature segmentation and transfer function design in the CT Bonsai dataset.

### 3.4 Temporal Coherency in Transfer Function Design

In this work, we extend our previously described feature space clustering into the temporal dimension. In the case of temporal volumetric data, each time step consists of a volume data set, from which a 2D feature space can be extracted. We create a volume of these feature space histograms (the third dimension being time), and, as in the 2D histogram case, we apply the variable kernel density estimation to extract features within this new volume. The feature space volume is then colored based on the 3D probability density binning (as was done in the 2D case). Now at each time step, the transfer function is the slice of the feature space volume that corresponds to the current time step.

Figure 3 illustrates this procedure on a time step of the tornado dataset. Figure 3 (Left) shows the volume rendering of the tornado data set based on a single transfer function time step (Figure 3 (Left-Center)) extracted from the temporal transfer function volume (Figure 3 (Right-Center)) at time step  $t$  and binned according to the 1D data histogram (Figure 3 (Right)). Here, the user can see the feature space clusters propagated over the entire temporal space. The green-red axis plane of Figure 3 (Left-Center) represents the feature space domain, and the blue axis represents the temporal domain. All previously described widgets also operate in this space; however, modifications to the 1D density histogram now affects the entire transfer function volume, and selections, as described in Section 3.3, also propagate across the transfer function volume.

Such a method is a vast improvement over current transfer function

design as data clusters are able to evolve over time, these data clusters then become part of the transfer function, and features of interest are able to be visualized over time, without manual adjustment of the transfer function. Unfortunately, should features leave and re-enter or somehow become obscured by other features due to the projection into the 2D feature space, our current implementation would be unable to account for these changes. However, as shown in the results of Section 4.4 and 4.5, our method is able to provide high-quality results even under this limitation.

## 4 EXPLORING THE FEATURE SPACE

In this section, we illustrate the benefits and limitations of our system using common volumetric datasets as examples. Here, we discuss the potential for feature segmentation, fast transfer function creation, and temporally coherent transfer function design.

### 4.1 Feature Segmentation

When provided with a 2D histogram of the feature space, novice users have little to no intuition as to what data points in the feature space map to volumetric features. By utilizing our non-parametric clustering technique, we are able to quickly provide the users with information on how the feature space relates to the volumetric space. In Figure 4, the user begins with the value vs. value gradient magnitude feature space, and then automatically applies our method to generate a color transfer function. While the initial volume rendering is less than desirable, the user is able to immediately understand which clusters in

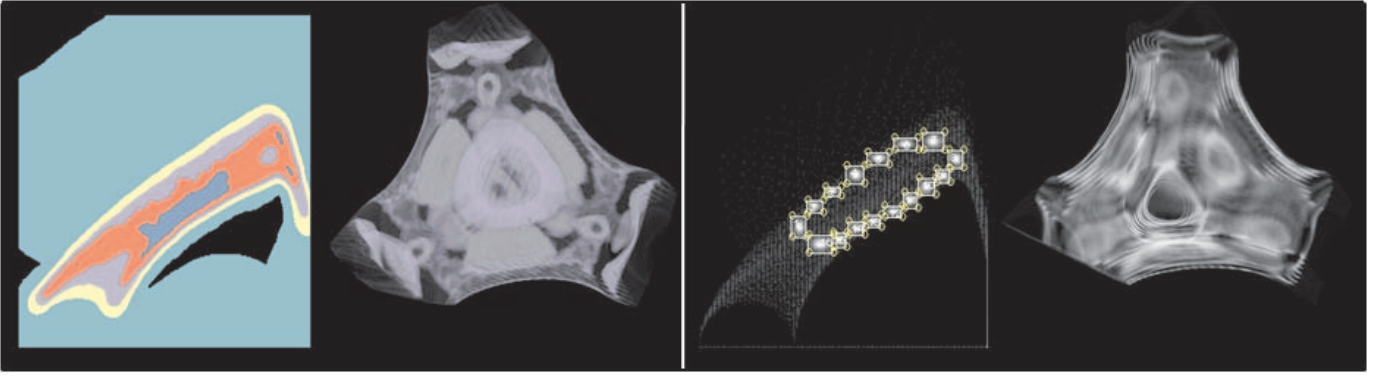


Fig. 5. Feature segmentation and transfer function design comparing traditional transfer function design methods to our non-parametric clustering method in the Cayley's cubic data set using the value versus value gradient magnitude feature space.

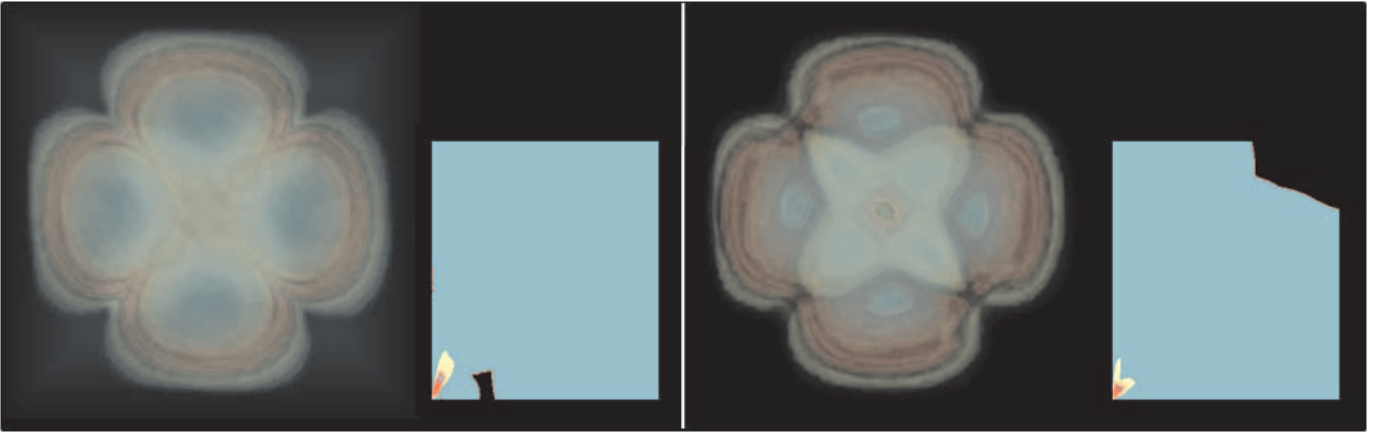


Fig. 6. The electron orbitals of a two million atom Indium-Arsenic quantum dot simulation in the second excited state: (Left) the p orbital rendered using non-parametric clustering of the p-value versus p-value gradient magnitude feature space, (Right) the p orbital rendered using non-parametric clustering of the p-value versus s-value feature space.

the feature space map to volumetric features (for example, the purple color obviously maps to the leaves and the blue color to the earth). Using this knowledge, the user can quickly modify the cluster colors to a more natural set (green for leaves, brown for earth, etc.).

However, while the initial volume rendering is able to provide information about the relation between feature space clusters and volumetric properties, the user may still not be clear on the exact relationship of each feature cluster in the volume rendering. To better explore the feature space, the user may interactively select one (or more) clusters in the feature space using the mouse and a new transfer function is applied to the volumetric data, representing only the selected feature space clusters. In the example of Figure 4, the user finds that the middle green cluster is actually also a portion of the earth. The user can then join that cluster to the brown grouping, and a final rendering is created. Note that fuzzy artifacts still exist in the final rendering as the leaf air boundary is indistinguishable in our technique due to the fact that the 1D histogram binning that is mapping colors to areas of similar density. Notice how in the bottom left cluster of the Generated Transfer Function in Figure 4, the purple cluster encompasses the red cluster. The data nearest the red cluster is most likely the source of the noise; however, even by adjusting the bin width the noise can only be reduced, not eliminated.

#### 4.2 Arbitrarily Shaped Transfer Functions

As shown in the previous examples, our technique provides an intuitive method for quick transfer function generation. However, the argument could be made that were novice users informed that the arc-shaped structures represent material boundaries; they could simply se-

lect those areas with rectangular transfer function widgets and produce a reasonably high-quality image. Such an argument is valid; however, if the user is then provided with a dataset with no discernable arc-shapes, where should exploration begin? Further, as was shown in Figure 4, clusters within feature space do not typically follow simple square clustering.

To illustrate the benefit of arbitrarily shaped clusters, our next example utilizes the Cayley's cubic data [28] shown in Figure 5. The Cayley's cubic is the unique cubic surface satisfying the equation:

$$4(x^3 + y^3 + z^3 + w^3) - (x + y + z + w)^3 = 0 \quad (4)$$

Here, the user has rendered a portion of the Cayley cubic through our transfer function generation widgets by selecting the purple region of Figure 5 (Left). In order to create a similar rendering using common box widgets, the user needs multiple interactions. Furthermore, the knowledge of where to draw such a grouping might never be realized from merely analyzing the distribution within the feature space. As such, it becomes clear that while powerful, rectangular transfer function editing widgets can pose severe limitations on the data exploration; moreover, providing a user with a starting point as to where to draw a transfer function within the feature space is extremely useful.

#### 4.3 Applications Across Arbitrary Feature Spaces

In the previous examples, we applied non-parametric transfer function generation to the value versus value gradient magnitude feature space of data; however, the choice of the feature space dimensions is arbitrary. In fact, by choosing a more meaningful feature space, users may be able to better extract particular features of interest. In Figure 6 we



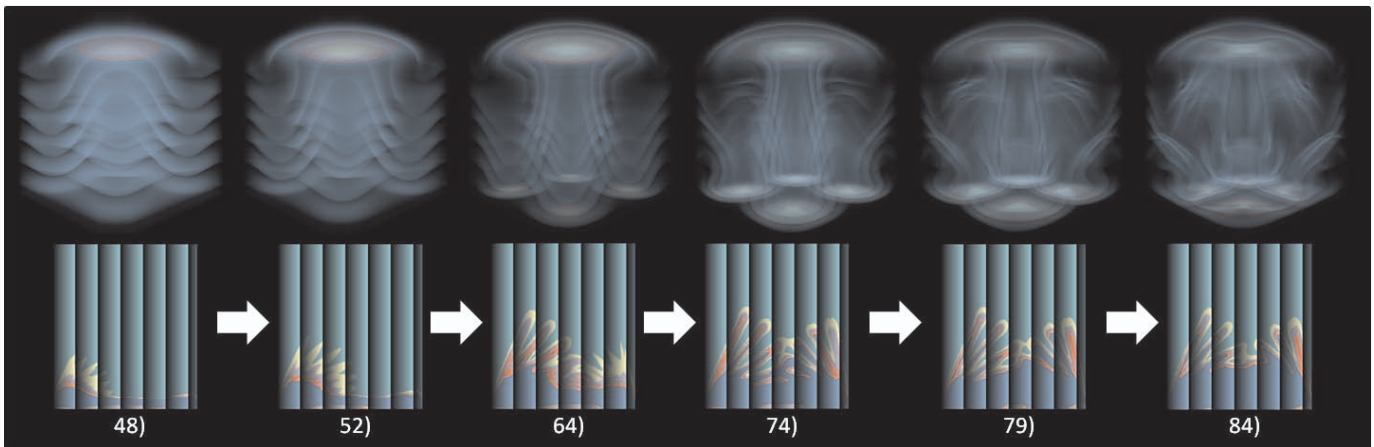


Fig. 7. Oriented structural flow visualizations of temperature advection in a convection dataset with numerical labels corresponding to the time-step. (Top) The volumetric rendering and (Bottom) their associated transfer function.

illustrate the difference that feature space choices make in our methods output by utilizing the s-orbital and p-orbital electron probability density functions from a two million atom Indium-Arsenic quantum dot simulation in the second excited state.

In Figure 6 (Left) we apply non-parametric transfer function generation to the p-value versus p-value gradient magnitude feature space (where p is the orbital), and in Figure 6 (Right) we apply non-parametric transfer function generation to the p-value versus s-value feature space. By comparing the resultant renderings, scientists can gain a better understanding of how various feature space dimensions interact in the physical volumetric space. For instance, Figure 6 (Right) is useful to understand areas and structures within the quantum dot where density relationships between p-orbital and s-orbital electron density are maintained.

#### 4.4 Temporally Coherent Transfer Functions

As previously stated, our transfer function generation method also is able to readily capture structures in  $n$ -dimensional space. By applying non-parametric density estimation across a volume of feature spaces, we are able to create a transfer function at each feature space time step that encodes information about the growth of volumetric features. Figure 7 demonstrates the application of our temporal transfer function generation method across a series of time steps of the convection in a box data set.

We apply a sine function to map the scalar data value to opacity, creating alternating low and high opacity contours in order to create the advection results [23]. In each of the transfer function time steps, we can see the cluster shapes grow tracking regions of like density distributions across time. As the user scrolls through time, the transfer function is extracted from the transfer function volume (described in Section 3.4). As such a new transfer function is automatically applied to the volume at each time interval, eliminating the need for transfer function modification at each step. Here, researchers are able to see the mixing and obtain a feel for the time varying properties in a single image with the transfer function showing both the internal and external structures.

Furthermore, Figure 8 demonstrates how a static transfer function may fail to follow features of the volume. From time step 4380 to 4620 in the cloud simulation, one can observe that the static transfer function seems to lose the upper cloud features which are prevalent when using the temporally coherent transfer function.

#### 4.5 Enhanced Feature Space Knowledge

While the temporal transfer function is a powerful tool for feature tracking, it also is able to enhance a user's knowledge about their data set by providing an analysis in the change of density distributions at each time interval. In many simulation models, researchers want to

look at the frequency in which data variable relationships occur within a data set. For example, in the simulated cloud generation of Figure 8, the transfer function is generated on the turbulent kinetic energy (TKE) versus water content. At each time step, the atmospheric scientist is interested in understanding the density of voxels for a given (TKE, water content) pair and the change of this relationship over time during the cloud formation process. By analyzing the transfer function clusters at each time step, the researcher can see how the voxel distribution changes over time, and future work will track the voxels over time for enhanced analytic capabilities.

### 5 CONCLUSIONS AND FUTURE WORK

In this work, we have shown that the application of non-parametric density estimation for clustering in feature space can provide the basis for semi-automatic transfer function generation. Volumetric structures are quickly separated in the feature space (when possible), and users are able to effectively render individual features by using our interactive widgets. Our work enables users to more effectively explore the volumetric feature space by automatically grouping neighborhoods of similar distributions, and these neighborhoods then tend to correlate to volumetric features. Furthermore, the extension of this scheme to temporal volumetric data provides a means for creating temporally coherent transfer functions, allowing users to track changing volumetric features without the need to redefine the transfer function at each stage of the process. Our scheme has proven to be effective over various types of volumetric data, ranging from scalar data (such as CT and MRI) to flow data.

However, our scheme does have several limitations. First, areas with similar density distributions will map to the same color bin of the KDE histogram, and the resulting clusters will need to be explored in order to determine the feature space cluster, volumetric feature correspondence. Second, the growth of KDE bins will not always allow for 100 There are instances where this could cause features and noise to blend together.

Future work will incorporate advanced edge dragging tools for the clusters to allow for better editing as well as the extension of our scheme to  $n$ -dimensional feature space, as well as enhanced widgets for the exploration of higher data dimensionalities. By incorporating this method across a higher number of dimensions, features that were not separable in the current system should become separable.

### ACKNOWLEDGMENTS

This work has been funded by the US Department of Homeland Security Regional Visualization and Analytics Center (RVAC) Center of Excellence and the US National Science Foundation (NSF) under Grants 0328984 and 0121288 and NSF China (No. 60873123) and 973 program of China (2010CB732504).

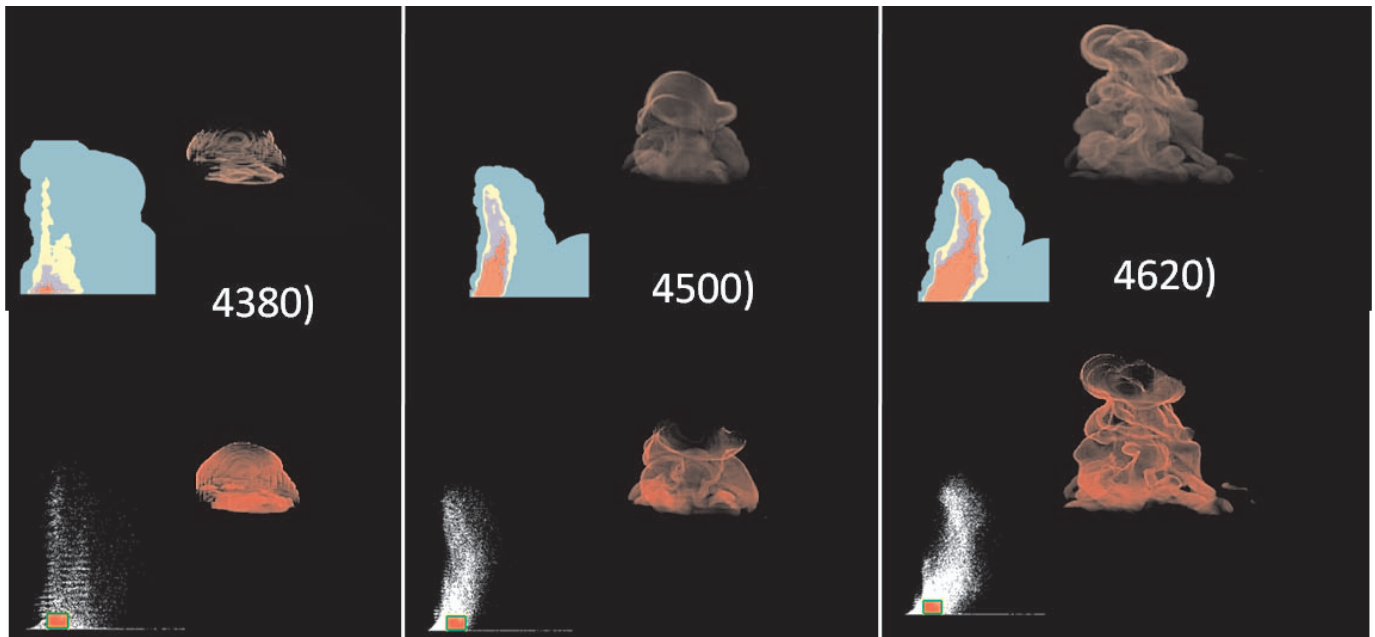


Fig. 8. The use of non-parametric two-dimensional transfer functions to visualize time steps of a simulated cloud formation in the feature space of TKE versus water content. (Top) The volumetric rendering using the red portion of our temporal transfer function. (Bottom) The volumetric rendering using a static 2D transfer function. .

## REFERENCES

- [1] H. Akiba and K.-L. Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proceedings of Eurographics/IEEE VGTC Symposium on Visualization*, pages 115–122, May 2007.
- [2] H. Akiba, K.-L. Ma, J. H. Chen, and E. R. Hawkes. Visualizing multivariate volume data from turbulent combustion simulations. *Computing in Science and Engineering*, 9(2):76–83, 2007.
- [3] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1428–1435, 2008.
- [4] C. A. Brewer. *Designing better Maps: A Guide for GIS users*. ESRI Press, 2005.
- [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, October 2008.
- [6] C. Correa and K.-L. Ma. Visibility-driven transfer functions. In *Proceedings IEEE-VGTC Pacific Visualization Symposium*, Beijing, China, April 2009.
- [7] S. Fang, T. Biddlecome, and M. Tuceryan. Image-based transfer function design for data exploration in volume visualization. In *Proceedings of the IEEE Conference on Visualization*, pages 319–326, 1998.
- [8] G. Kindlmann and J. W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of the IEEE Symposium on Volume Visualization*, pages 79–86, 1998.
- [9] G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of the IEEE Conference on Visualization*, pages 513–520, 2003.
- [10] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of the IEEE Conference on Visualization*, pages 255–262, 2001.
- [11] J. M. Kniss, R. V. Uitert, A. Stephens, G. S. Li, T. Tasdizen, and C. Hansen. Statistically quantitative volume visualization. In *Proceedings of the IEEE Conference on Visualization*, pages 287–294, 2005.
- [12] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics & Applications*, 8(3):29–37, 1988.
- [13] C. L. P. Ljung and A. Ynnerman. Local histograms for design of transfer functions in direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1570–1579, 2006.
- [14] C. Lundström, A. Ynnerman, P. Ljung, A. Persson, and H. Knutsson. The alpha-histogram: Using spatial coherence to enhance histograms and transfer function design. In *Proceedings Eurographics/IEEE-VGTC Symposium on Visualization 2006*, May 2006.
- [15] C. Muelder and K.-L. Ma. Interactive feature extraction and tracking by utilizing region coherency. In *Proceedings of IEEE Pacific Visualization Symposium*, April 2009.
- [16] S. Potts and T. Moller. Transfer functions on a logarithmic scale for volume rendering. In *Proceedings of Graphics Interface*, pages 57–63, 2004.
- [17] C. Rezk-Salama, M. Keller, and P. Kohlmann. High-level user interfaces for transfer function design with semantics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1021–1028, 2006.
- [18] S. Roettger, M. Bauer, and M. Stamminger. Spatialized transfer functions. In *Proceedings Eurographics/IEEE-VGTC Symposium on Visualization 2005*, 2005.
- [19] T. Ropinski, J.-S. Prani, F. Steinicke, and K. H. Hinrichs. Stroke-based transfer function design. In *IEEE/EG International Symposium on Volume and Point-Based Graphics*, pages 41–48, 2008.
- [20] A. Shamir. Feature-space analysis of unstructured meshes. In *Proceedings of the IEEE Conference on Visualization*, pages 185–192, 2003.
- [21] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.
- [22] N. Svakhine and D. S. Ebert. Interactive volume illustration and feature halos. In *Proceedings IEEE-VGTC Pacific Visualization Symposium*, October 2003.
- [23] N. A. Svakhine, Y. Jang, D. S. Ebert, and K. P. Gaither. Illustration and photography inspired visualization of flows and volumes. In *Proceedings of the IEEE Conference on Visualization*, pages 687–694, 2005.
- [24] 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 Graphics*, pages 366–374, 2002.
- [25] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma. A novel interface for higher-dimensional classification of volume data. In *Proceedings of the IEEE Conference on Visualization*, pages 505–512, 2003.
- [26] M. P. Wand. *Kernel Smoothing*. Chapman & Hall/CRC, 1995.
- [27] G. H. Weber, S. E. Dillard, H. Carr, V. Pascucci, and B. Hamann. Topology-controlled volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):330–341, 2007.
- [28] E. W. Weisstein. “The Cayley Cubic.” From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/GreensIdentities.html> .
- [29] Y. Wu and H. Qu. Interactive transfer function design based on editing direct volume rendered images. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1027–1040, 2007.