SCI INSTITUTE
TECHNICAL REPORT

Interactive Transfer Function Specification for Direct Volume Rendering of Disparate Volumes

Fábio F. Bernardon, Linh K. Ha, Steven P. Callahan, João L. D. Comba, and Cláudio T. Silva, Member, IEEE

UUSCI-2007-007

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

June 5, 2007

Abstract:

Transfer functions play a critical role in feature detection through direct volume rendering in volumetric scalar fields. Because of the inherent difficulties of exploratory visualization, assisting the user in transfer function specification is still an important area of research. In particular, the disparate nature of simulated and measured volumetric data necessitates more flexibility in the specification process than existing tools provide. In this paper, we introduce a framework that combines elements of existing techniques as well as introduces new features to handle the diversity of volumetric datasets. The contributions of our work include a range mapping technique for assisting in transfer function specification for high dynamic range data, an algorithm for combining multiple transfer functions into ensembles that allow multiple specification techniques to be used collectively, and an intuitive method for specifying transfer functions for multiple classes of time-varying data.

THE UNIVERSITY OF UTAH

Interactive Transfer Function Specification for Direct Volume Rendering of Disparate Volumes

Fábio F. Bernardon, Linh K. Ha, Steven P. Callahan, João L. D. Comba,
and Cláudio T. Silva, *Member, IEEE*

Abstract—Transfer functions play a critical role in feature detection through direct volume rendering in volumetric scalar fields. Because of the inherent difficulties of exploratory visualization, assisting the user in transfer function specification is still an important area of research. In particular, the disparate nature of simulated and measured volumetric data necessitates more flexibility in the specification process than existing tools provide. In this paper, we introduce a framework that combines elements of existing techniques as well as introduces new features to handle the diversity of volumetric datasets. The contributions of our work include a range mapping technique for assisting in transfer function specification for high dynamic range data, an algorithm for combining multiple transfer functions into ensembles that allow multiple specification techniques to be used collectively, and an intuitive method for specifying transfer functions for multiple classes of time-varying data.

Index Terms—Transfer functions, direct volume rendering, exploratory visualization

1 Introduction

Creating insightful visualizations from both simulated and measured data is an important problem for the visualization community. For scalar volumes, direct volume rendering has proved to be a useful tool for data exploration. With the use of a transfer function, scalar values can be mapped to colors and opacities to identify and enhance important features. Though some automatic techniques have been developed for transfer function specification [9, 13], the exploration process still involves tuning the parameters manually until the desired visualization is produced. A great deal of research has recently been performed to assist the user in this specification task with interactive widgets [15, 17]. These tools generally assist the user by allowing them to create and manipulate widgets over one or more dimensions of histogram information of the data.

Even with current tools, the specification of transfer functions is not a trivial task [26]. The primary obstacle is the diversity of datasets to be rendered. A tool that excels at extracting features from a structured grid of scanned medical data may have difficulty finding relevant features in an unstructured grid of simulation data. The use of multidimensional transfer functions may be helpful for some datasets, but may just complicate the specification in others. This problem is further compounded when different features in a dataset are enhanced by different histograms. Another difficulty is that the datasets may contain a high dynamic range of floating point scalar values that vary over time. General tools to efficiently handle the diversity of data that can be encountered in both medical and scientific domains are not currently available.

As an example, consider the ubiquitous Utah Torso dataset—an unstructured grid containing simulated electric potential in the human torso. In our example, we use a version of the dataset consisting of about 50 thousand tetrahedra and 360 floating point time steps of a rotating dipole that emphasizes the simulation results. Figure 1 illustrates several visualizations of this data through direct volume rendering. There are several issues that are encountered when attempting to explore this data using transfer functions. First, for each time step of the data, the scalars in the volume are concentrated in one peak in the histogram (*ie.,* 83% of the scalars fall into 1% of the scalar range). Thus, with traditional specification tools such as a polyline defined over the histogram, feature finding may be difficult because much of the data maps to few entries in the corresponding color and opacity lookup table (see Figure 1(a)). A zooming interface on the color map (*eg.,* Yuan *et al.* [40]) will facilitate the placement of the specification widget, but will not show additional features due to the static resolution of the lookup table. The second issue that occurs in the specification is that some features of the data can only be found using one type of histogram. On the Utah Torso, positive and negative potentials can be determined with a polyline or rectangle widgets on the scalar histogram (see Figure 1(a)), but the amount of change in the scalars over time requires a time-sensitive 2D histogram (see Figure 1(b)). Available techniques that merge transfer functions (*eg.,* Wu *et al.* [38]) are not designed to provide user control of the blending of 1D and 2D transfer functions. The final issue that is encountered during transfer function specification is that limited control is currently available for specifying transfer functions for time-varying data. The ability to transition between user-defined transfer functions temporally is essential for tracking features through time or to provide temporal focus and context animations.

The goal of our research is to create a tool that facilitates feature extraction through transfer function specification for disparate data types by addressing each of these issues. Our system builds on more than a decade of previous work by unifying existing methods as well as introducing new techniques for specifications that can be used independently or in unison for volume visualization of complex data. In this paper, we focus on transfer function generation for unstructured volumes due to their complexity and disparity; but the algorithms described are general enough to be applied to structured data as well. In particular, our contributions include:

- we provide an interactive system that facilitates data exploration of diverse data types by combining existing techniques with new ones;
- we describe an algorithm for simplifying feature extraction in high dynamic range datasets by allowing the interactive, nonlinear remapping of the scalar range in the histogram;
- we introduce the notion of transfer function *ensembles* that are a user-controlled blending of multiple transfer functions defined on different histograms—each representing different features;
- we discuss transfer function specification across multiple steps of time-varying data and introduce a tool for blending ensembles of transfer functions over time through keyframing;
- we provide the results of a user study of our system in the form of an expert review.

Figure 1(c) and (d) demonstrate these contributions on our example dataset. Figure 1(c) provides the results of blending transfer functions

- *Fábio F. Bernardon and João L. D. Comba are with the Instituto de Informática, UFRGS, Brazil, E-mail: [fabiofb,comba]@inf.ufrgs.br.*
- *Linh K. Ha, Steven P. Callahan, and Cláudio T. Silva are with the Scientific Computing and Imaging Institute, University of Utah, USA, E-mail: [lha,stevec,csilva]@sci.utah.edu.*

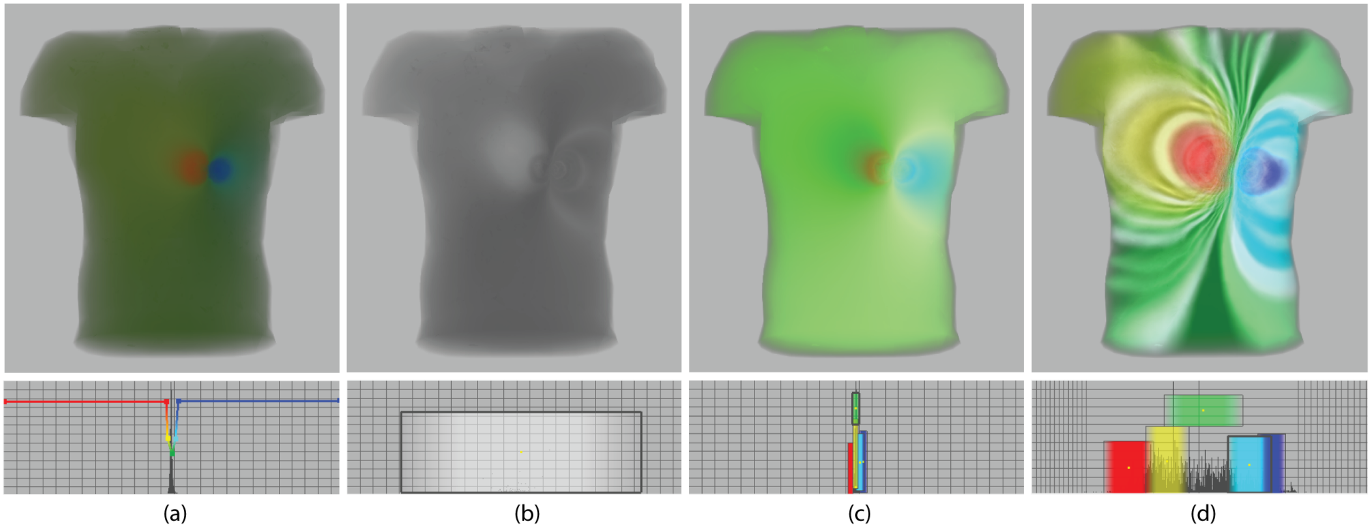

Fig. 1. An example demonstrating some of the capabilities of our system using the Utah Torso dataset. (a) Traditional transfer function specification using a simple polyline widget over the scalar histogram shows positive and negative potentials in the simulation. (b) Other features such as the variation of the scalars over time can only be expressed using more advanced histograms such as this 2D histogram of the magnitude of the variation. (c) Our transfer function *ensembles* provide user-specified blending operators between histograms defined on in both 1D and 2D, in this case a 1D scalar histogram and a 2D time histogram. (d) By remapping the scalar range from the previous ensemble, high dynamic range details can easily be explored, resulting in a more insightful visualization.

defined on different histograms. In this case, it uses an additive blending function to combine a 1D scalar histogram defined using rectangle widgets (Figure 1(c) bottom) with a 2D time histogram that uses the variation of scalars over time to emphasize the changing regions in the volume. Thus multiple, distinct features are visible in a single visualization. Figure 1(d) shows the results of distributing the scalars more evenly through the color and opacity lookup table by dynamically remapping them. Previously unseen features (such as the abrupt changes potentials) become visible by taking advantage of a non-linear mapping of the scalars. Finally, using our keyframing interface, ensembles of transfer functions (such as the one shown in Figure 1(d)) can be assigned to specific time steps and smoothly transitioned during animations using user-defined step functions. The overall result is a tool that allows insight into the data that is not otherwise available with established tools and techniques.

The rest of the paper is outlined as follows. Section 2 summarizes related work. Section 3 provides an outline of our system and tools for transfer function specification. In Section 4 we introduce range mapping for high dynamic range volumes. In Section 5 we discuss the merging of multiple transfer functions. Section 6 outlines our algorithm for blending transfer function ensembles over time. An evaluation of our methods is described in Section 7 followed by a discussion in Section 8. We summarize our work and relate future work in Sections 9.

2 Related Work

Scalar volume visualization through direct volume rendering has received much attention in the research community (for a recent survey, see [12]). Yet the specification of transfer functions is still a challenging task. There have been many attempts to automate the process of specification. Levoy [19] described how boundaries can be visualized using the computed gradient of the scalar field to generate a transfer function. Along similar lines, Kindlmann and Durkin [13] proposed the use of histograms of the first and second derivatives to automatically generate a transfer function which emphasizes boundaries around homogeneous regions in the volume. An alternate approach was proposed by Fujishiro *et al.* [9] which uses a hyper Reeb graph to distinguish features of the dataset topologically.

Automatic techniques are good at extracting important boundary features from a volume, however they may not always give the user the desired visualization. Therefore, many systems have been developed that push more of the specification burden to the user. Marks *et al.* [24] introduced Design Galleries which allow the user to explore the transfer function parameter space by iteratively picking images from collections of visualizations. More recently, a parallel coordinate interface for parameter exploration was described by Tory *et al.* [34]. Tzeng *et al.* [35] introduced a system that learns regions of the volume by allowing the user to paint on slices of the volume. Along similar lines, Roettger *et al.* [30] include spatial information in standard 2D histograms to allow selection by region. Another high-level specification system was proposed by Rezk-Salama [29] that provides semantics for different visualization tasks and hides much of the underlying specification with the use of simple user interfaces for parameter exploration.

The most common approach to user-assisted transfer function specification is to incorporate histograms of the scalar field that classify the volume into different materials [8]. Bajaj *et al.* [2] proposed a system that analyzes the volume to extract isocontour information that is plotted for the user as a collection of 1D histograms for static datasets and 2D histograms for time-varying datasets. Kniss *et al.* [15] introduced widgets to facilitate transfer function specification on the multidimensional histograms introduced in Kindlmann and Durkin's work. Lum and Ma [20] modified the 2D joint histogram by showing the scalar pair along the gradient direction to introduce the Lighting Transfer Function that selectively enhances the boundary surface of interest. Another recent approach by Sereda *et al.* [31] uses Gaussian kernels to determine material transitions in CT scans and displays their L and H parameters as a 2D histogram that facilitates boundary extraction.

Focus and context approaches have received a lot of attention recently due to the recognition that features deserve different levels of focus. Hadwiger *et al.* [10] introduced a volume rendering system for segmented data that selects a different transfer function based on segmented voxel IDs. Similarly, Viola *et al.* [36] use importance compositing to assign higher opacities to more important features in segmented data. Svakhine *et al.* [32] extend this work to perform different rendering techniques for the unique materials in the data. For multiple volumes, Bruckner and Gröller [4] introduce a system that controls the compositing of inter-penetrating objects by using an opacity weighted average of two dimensional intersection transfer functions. Ma [21] and later König and Gröller [17] recognized that feature extraction would be simplified by defining transfer functions separately then consolidating them into one image with additive blending. Wu *et al.* [38] extended this idea with the use of a genetic algorithm for non-linear

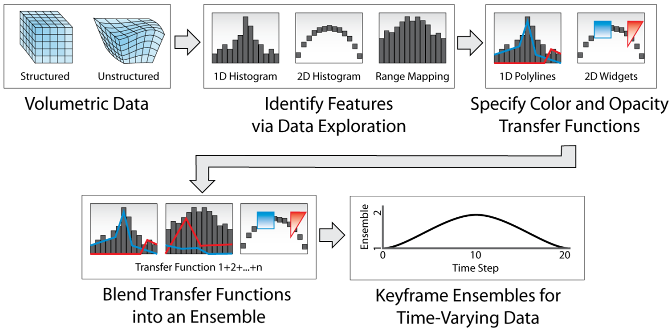

Fig. 2. The visualization process provided by our system.

combinations of transfer functions. Our approach for blending transfer functions is similar to the additive approaches. However, unlike these methods, we allow the user to blend transfer functions defined on different histograms in 1D and 2D with user-controlled blending operators.

More specific techniques for transfer function specification have also been developed to handle high dynamic range data. By using a Gaussian transfer function, Kniss *et al.* [16] avoid the inaccuracies that are present with a low resolution lookup table. Another approach was introduced by Potts *et al.* [27], which uses a logarithmic scaling on their transfer function. Kraus *et al.* [18] use a similar logarithmic approach for lookup tables that was later used by Qiao *et al.* [28] to render large simulation data. These latter approaches based on logarithmic scaling assume data centered near zero. Recognizing that this is not always the case, a recent system by Yuan *et al.* [40] provides a high precision lookup table and the ability to non-linearly zoom into regions of the transfer function for detailed specification. Instead of focusing on a non-linear mapping of the lookup table, we take a different approach by remapping the scalars through equalization or user-controlled range mapping. This allows more control over the aspects of the data that deserve focus than the previous approaches provide.

Techniques have also been developed to handle time-varying datasets. These techniques must consider the data for the entire time series when specifying transfer functions. The work of Jankun-Kelly and Ma [11] analyzes the generation of a single transfer function that works globally for a time-varying dataset. Doleisch *et al.* [7] presented a framework using time histograms to analyze unsteady flow data from computational fluid dynamics (CFD) simulations. Younesy *et al.* [39] proposed the Differential Time Histogram Table, using temporal coherence to minimize the amount of data required from disk to accelerate the rendering process. Usually a transfer function is designed to capture features that have a regular or periodic behavior. If a dataset presents different behavior, a complex transfer function is required to capture all features at once [22]. One recent paper has attempted to address the problem of aperiodic time sequences. Akiba *et al.* [1] extended the time varying transfer function framework to handle statistically dynamic time-varying volume data by performing temporal reduction and visualization feedback to find suitable time classified intervals. Wu *et al.* [37] extend the idea of transfer function fusion to create animations of static data by keyframing focus and context visualizations. We are not aware of any techniques that have been developed to assist with specification of multiple transfer functions for time-varying data. In this paper, we introduce techniques for transitioning between multiple transfer functions with the use of keyframing and user-controlled transitions.

3 System Overview

In this section we summarize our system for transfer function specification. It comprises several tools that aim to provide useful insights about the underlying data and help the user during the creation of transfer functions. An overview of the visualization process, as provided by our system, is shown in Figure 2. First, the volumetric data is loaded and the appropriate visualization algorithm is selected depending on the data type. We currently support unstructured, time-varying grids. Next, the task of data exploration is performed by finding features in the data with the assistance of histograms and specifying color and opacity using interactive widgets. A variety of specialized 1D and 2D histograms and corresponding color and opacity widgets are available to highlight regions of interest in the volume. In addition, fine details in high-dynamic range volumes can be further explored by directly manipulating the spacing of the scalars in the histograms through range-mapping. This process of data exploration through transfer function specification continues until a desired set of features is distinguished in the volume through one or more transfer functions (potentially defined on different histograms). The next step of our visualization process allows the user to combine these transfer functions into ensembles using user-defined blending operators. Finally, these ensembles can then be keyframed with user-specified transitions for time-varying data.

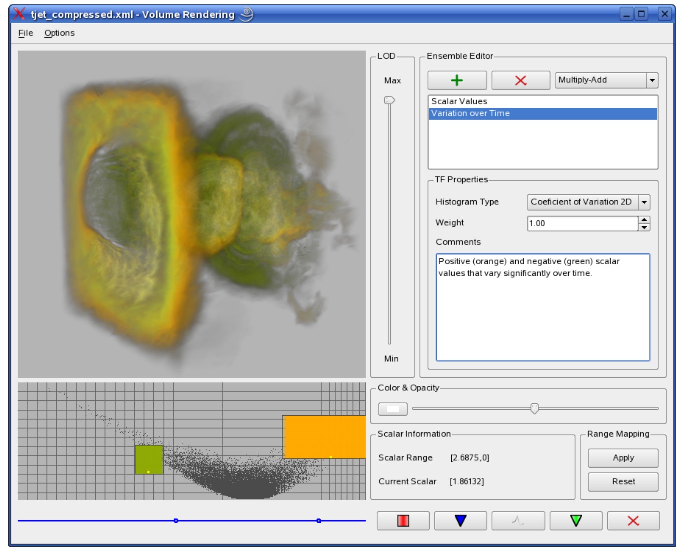

Fig. 3. The user interface for interactive transfer function specification is shown for the time-varying Turbulent Jet dataset using a 2D time histogram.

The user interface for our system is shown in Figure 3. The upper left window of our interface displays the volume rendering described by the current transfer function specification. Though the transfer function tools we describe are general enough for any data type, we developed our tools primarily for unstructured grids. We perform our volume rendering using the Hardware-Assisted Visibility Sorting (HAVS) algorithm for unstructured grids [6] with dynamic level-of-detail [5] and support for time-varying data [3]. Partial pre-integration is used to reduce the overhead of dynamic updates to the transfer function [25].

Statistical analysis is important to capture hidden aspects of the data, thus our system offers several histogram options. In addition to a basic 1D scalar histogram, we also provide a 2D histogram that includes the gradient magnitude. For time-varying data, we also provide 1D and 2D histograms based on the coefficient of variation calculated from the scalar time steps [11]. Additional histograms can also be incorporated into the framework. The active histogram is displayed in the lower left window of the interface as shown in Figure 3.

Similar to Kniss *et al.* [15], interaction with the scalar histograms is performed by defining widgets that represent color and opacity on the histogram. We provide four types of widgets to the user, some of which are shown in Figures 1 and 3. The first widget is a rectangle defining a single color and opacity ramp and is used in both 1D and 2D histograms. In 1D, the vertical position does not change the transfer function but is useful for manipulating overlapping widgets. The second widget is a triangle defining a single color used for 2D histograms. The third widget is a triangle widget with a fall-off, similar to the one

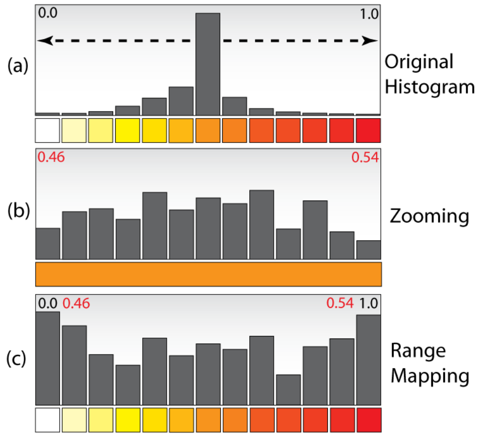

Fig. 4. (a) A high-dynamic range histogram (above) is shown with a corresponding lookup table (below). (b) Zooming into the dense region of the histogram does not change the resulting image due to the static resolution of the lookup table. (c) By range mapping the scalar values, the high-dynamic range elements of the mesh can be spread more evenly across the static lookup table, enhancing hidden features in the data.

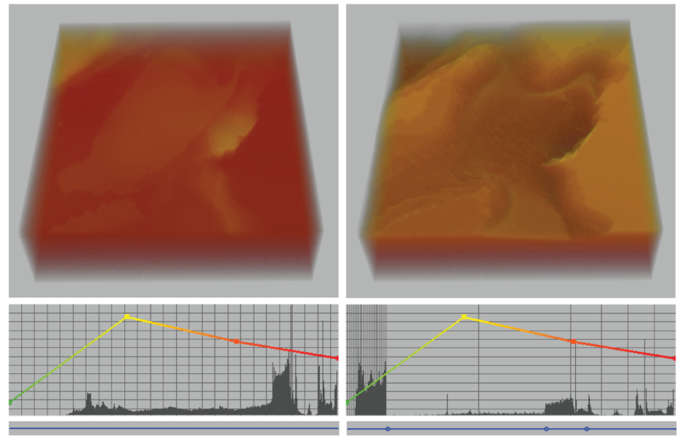

Fig. 5. Volume exploration on the San Fernando earthquake simulation through range mapping. A predefined transfer function (left) is used to explore the data by remapping the scalars (right). Only a non-linear remapping can enhance features that are hidden in multiple spikes of the data.

described by Kniss *et al.* [15], to emphasize boundaries. This widget is primarily used in 2D gradient magnitude histograms. Finally, the last widget is a polyline, where the control points denote colors and the vertical placement of the points controls opacity. This final widget is used only in 1D histograms.

4 Scalar Range Mapping

Volume data produced from scientific simulation typically contains a high dynamic range (HDR) of floating point scalar values. In addition, a high percentage of the scalar values are often contained in a small range of the histogram (see Figure 4(a)). Consequently, to expose details that may be contained in these small regions, a large number of control points and a high resolution lookup table are required. There are two main issues with traditional transfer function design when dealing with HDR data. First, the narrow range of values makes specification difficult due to the low resolution of the features on the histogram interface. Second, the limited resolution of the color and opacity lookup table in graphics hardware is not sufficient to fully represent all the unique scalar values in the data.

To overcome the resolution limitations of the histogram interface, tools like ParaView [14] incorporate user-controlled zooming widgets to assist with transfer function specification over small regions of the data. Yuan *et al.* [40] recently introduced a 1D fish-eye visualization of the histogram based on a *focus and context* concept, which allows simultaneous representation of global (context) and detail (focus) information on the same histogram display. These approaches, based on magnifying the range of interest in the user interface, greatly assist the user with HDR transfer function design. However, the second issue is still a challenge. Ideally, the number of entries in the color and opacity lookup table should correspond to the number of unique scalar values in the volume. Yuan *et al.* [40] leverage tone mapping and specialized high-precision graphics hardware to handle the high precision of texture based volume rendering. With limits in texture size, this is not always sufficient and may result in many scalar values being assigned to one entry in the table (see Figure 4(b)). Instead, we propose *range mapping*, which redistributes the scalar range non-linearly to spread the regions of interest more evenly across the lookup table (see Figure 4(c)). Range mapping is related to histogram equalization, a common approach in image processing for handling low contrast images. This feature facilitates the design process by allowing focus and context zooming effects, while avoiding resolution issues of a fixed-size lookup table. The result is a tool naturally capable of extracting detailed features in the data, as shown in Figure 5.

Based on the observation that the transfer function design difficulties of HDR data are mainly due to the non-uniform distribution of scalar data, our solution is to redistribute the scalar range. This can be done automatically by performing histogram equalization, which spreads out the clustered regions. Mathematically, histogram equalization is performed by introducing a cumulative density function (CDF) as a sum of probability density functions (PDFs) over normalized scalar inputs:

$$CDF(x_i) = \sum_{x_j < x_i} PDF(x_j).$$

Then, a simple mapping is performed on the normalized scalar input value x that yields a new uniformly distributed normalized output y:

$$y = CDF(x).$$

Due to its speed and simplicity, it is common to use a discrete histogram equalization and perform this mapping with a lookup table. This approach is automatic, but gives the user very little control over the redistribution process and tends to break the continuities of the scalar range. Range mapping, a generalization of histogram equalization, is based on piecewise linear mapping functions and provides more control while maintaining the continuity of the scalar range. The range mapping functions that map the input scalars $[x_0 \dots x_n]$ to a new scalar range $[y_0 \dots y_n]$ are a class of piecewise continuous functions f over the input range that satisfy the following conditions: f is a monotonically increasing function, $f(x_0) = y_0$, and $f(x_n) = y_n$. Similar to histogram equalization, the new scalar value y is computed as:

$$y = f(x).$$

Given this definition, the function can arbitrarily redistribute the scalar range while maintaining the order and continuity.

In practice, we use the linear range mapping functions that combine many line segments, each of which performs mapping from a specific range $[x_i \dots x_{i+1}]$ to $[y_i \dots y_{i+1}]$ by applying the linear mapping equation:

$$y = \frac{x - x_i}{x_{i+1} - x_i}(y_{i+1} - y_i) + y_i.$$

These linear functions are sufficient to represent all range mappings since any function can be approximated using many piecewise linear functions.

Because the cost of the linear interpolation is relatively low, we can perform range mapping interactively while the user is manipulating control points for the range. The remapping process is performed in hardware by storing a 1D texture that contains one entry for every control point of the remapping. When the mapping changes, to minimize CPU to GPU transfer, only the new mapping texture is sent to the GPU.

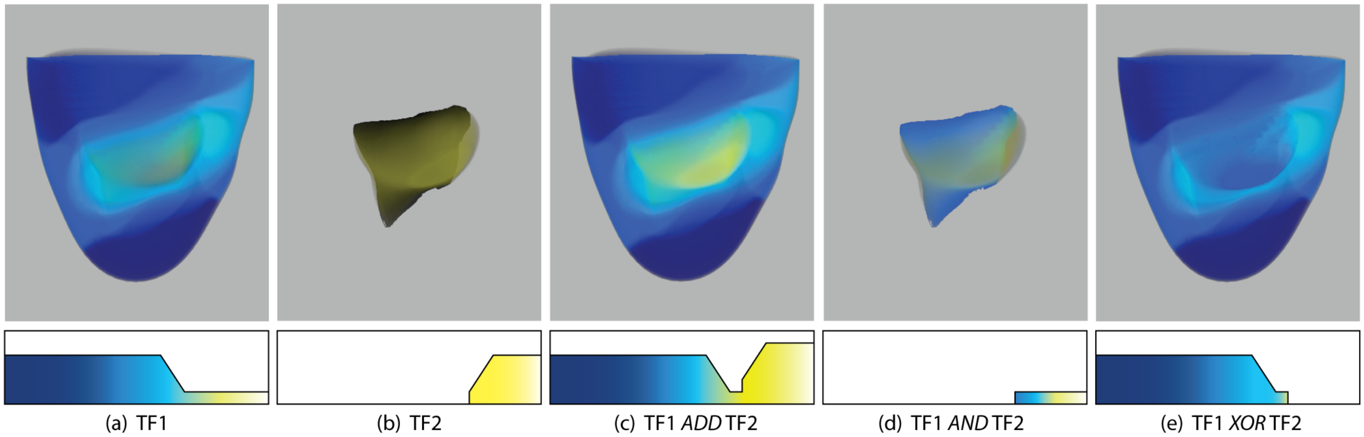

Fig. 6. An ensemble is used to combine multiple transfer functions with different blending strategies for an unstructured dataset of the heart (top: volume rendering result; bottom: illustration of the operation in transfer function space). (a) A single transfer function shows all features of the volume. (b) A more specific transfer function focuses on one detail. (c) Adding both transfer functions combines the results. (d) The *AND* operation emphasizes common features of both transfer functions. (e) The *XOR* operation removes common features from the two transfer functions.

During rendering, the normalized scalar values can then be remapped to normalized scalar values using a single texture lookup with linear interpolation enabled. Thus, this extra remapping step minimally impacts the rendering performance and is flexible enough to be used in a variety of volume rendering algorithms.

As illustrated in Figure 4, range mapping yields a magnification that is different from a normal zooming effect, since the actual shape of the histogram changes non-linearly. This helps the user exploit the real data distribution in narrow clusters of the scalar range. Even without transfer function widgets, range mapping can be a powerful exploration tool. Figure 5 shows how range mapping can be used to explore the data using a simple, predefined transfer function.

Creating a user interface that can fully exploit the power of range mapping is a challenge. Our solution is a simple, intuitive interface that allows the user to choose the range by adding control points and extend the range by dragging two control points away from each other. This allows the user to continue adding control points between the previous points to further probe important regions. The histogram and volume rendering change interactively during this control point manipulation to provide visual feedback of the remapping. In addition, the range and scalar value under the cursor are displayed to the user to facilitate specification when there is *a priori* knowledge of the data. Figures 3 and 5 show snapshots of this interface.

5 Transfer Function Ensembles

Designing transfer functions that effectively enhance volume characteristics is a difficult task because the exploration of the transfer function space can be unintuitive [15]. Volume statistics may provide meaningful insights about data and aid users during the specification process, but a single statistical measure may not reveal all the important features in the data. We advocate the use of multiple histograms for transfer function specification that highlight different features in the data. These transfer functions are then merged using weighted blending operations to produce a single, complex transfer function that we call an *ensemble*. Ensembles provide the ability to derive completely new transfer functions from a collection of existing ones.

The idea of combining transfer functions is not new [17, 21, 38]. Previous work in the area has concentrated on merging transfer functions defined on the same histogram space using both linear and nonlinear combinations of transfer functions. Our approach with ensembles is to allow more flexibility in the combination process by providing boolean-like blending operations. The only limitation to these transfer functions is that they share the same space (*ie.,* scalar value). Thus, an ensemble represents a new transfer function that is created by aggregating a collection of transfer functions using different blending operations.

For 1D histograms that define a transfer function on the scalar value blending is straightforward. The transfer function widgets directly map to an RGBA lookup table that is used for rendering. Blending multiple transfer functions is then just a texture compositing of RGBA values. Extending this to other spaces (*ie.,* gradient magnitude or coefficient of variation) requires an additional dimension for each space. Unfortunately, it is not feasible to extend beyond three dimensions due to constraints on textures in hardware. Thus, we limit these combinations to histograms that have three unique dimensions that do not share a common space. We have found that this is adequate for the volumes used in our experiments.

Another approach for combining multiple transfer functions defined in multiple spaces is to map them to a single common space. The extra dimension in the 2D case is then used to modulate the intensity of the color. This is done by determining the number of scalar values within a 2D widget and the number outside and using this fraction to reduce the intensity accordingly. Though not nearly as powerful as true multi-dimensional transfer functions, this heuristic tends to produce acceptable results in practice and allows all the histograms to be reduced to a common space. For instance, the combination of transfer functions in 1D and 2D shown in Figure 1 were performed using this heuristic.

Through the use of ensembles, users can intuitively explore features enhanced by different transfer functions. Complex mappings can be generated by aggregating several simpler transfer functions. Another application is the contextualization of the entire volume through one transfer function, while different transfer functions focus only on important features.

Different strategies can be interactively swapped, providing a fast tool for exploring volume datasets. Our system also allows users to load custom transfer functions and combine them with new transfer functions designed with our tool. In our framework, we provide the ability to specify multiple ensembles of transfer functions and manage different visualizations for a volume.

5.1 Blending Strategies

We provide several weighted blending strategies for combining transfer functions. These blending functions are illustrated in Figure 6. We provide three types of operations, though others are easily incorporated. Each transfer function is assigned a weight that corresponds to its intensity contribution in the resulting ensemble. This weight is user-controlled and can be used to provide emphasis in the most important regions. Each transfer function is multiplied by its weight prior to blending with other transfer functions. The examples given below describe the blending process for two weighted transfer functions. If more than two transfer functions are provided then the compositions are performed sequentially. We provide the following principle blending operations:

1. **ADD**: This is the most common blending operation in which transfer functions are summed together. The result r for color C and opacity α of a lookup table entry i when combining transfer function 1 and 2 can be expressed as the following:

$$\begin{aligned} C_r(i) &= C_1(i) + C_2(i) \\ \alpha_r(i) &= \alpha_1(i) + \alpha_2(i) \end{aligned}$$

   This mode is useful for combining features that are unique to two transfer functions, producing a single transfer function that emphasizes all features of both transfer functions. Figure 6(c) shows the result of adding two transfer functions to combine their enhanced characteristics into a single image.

2. **AND**: This blending operation enhances features that are shared by two transfer functions. The result of this combination of transfer functions can be expressed using the same notation as above:

$$\begin{aligned} C_r(i) &= Max(C_1(i), C_2(i)) \\ \alpha_r(i) &= Min(\alpha_1(i), \alpha_2(i)) \end{aligned}$$

   We use the maximum of the RGB channels (independently) to maintain the color intensity and the minimum alpha to remove regions that are not common. Figure 6(d) shows the result of an *AND* operation in the heart dataset. This feature is useful for determining common properties between regions of interest, in this case it emphasizes the region that contains a sensor.

3. **XOR**: This blending operation removes common areas from two transfer functions. The result of the *XOR* operation on two transfer functions can be expressed using the same notation provided above:

$$\begin{aligned} C_r(i) &= (C_1(i) \wedge \overline{C_2(i)}) \vee (\overline{C_1(i)} \wedge C_2(i)) \\ \alpha_r(i) &= (\alpha_1(i) \wedge \overline{\alpha_2(i)}) \vee (\overline{\alpha_1(i)} \wedge \alpha_2(i)) \end{aligned}$$

   In other words, a color and opacity contribution defined by a transfer function is preserved only if no other transfer function assigns a color and opacity value to it. This is a good strategy to remove features from existing transfer functions. Figure 6(e) shows an example of removing the region of the heart that contains a sensor.

Combining different transfer functions is a straightforward procedure for creating complex visualizations. For instance, in medical imaging data, one can create individual transfer functions that enhance different organs (heart, lung, liver, *etc.*) and combine them all in different ways using the *ADD* to provide a series of visualizations. The subtraction capabilities of *AND* and *XOR* also allow the user to create unique transfer functions from existing ones.

6 Time Varying Datasets

Transfer function design and generation for time-varying datasets has received little attention in the research community. The available literature focuses on defining transfer functions that are applied globally to all time steps [1, 22]. However, these global approaches may not be sufficient for all types of volumes. Datasets like the Utah Torso, shown in Figure 1, contain regular time-varying patterns that can be readily inspected with the use of a single ensemble of transfer functions. This regularity is demonstrated by the relatively static appearance of the histogram as the time steps progress. Unfortunately, all volumes do not demonstrate this same temporal regularity or periodicity. For example, the Five Jets dataset, shown in Figure 8, has a large variance in the histogram as the time steps progress. This results in moving features that cannot be enhanced with a single transfer function.

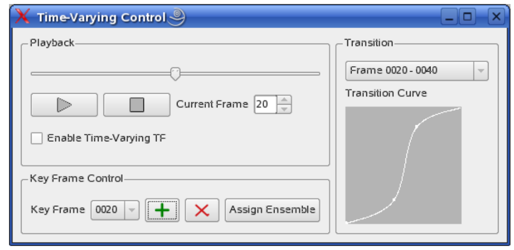

Fig. 7. Our time-varying interface: playback control (upper left), keyframes control (lower left) and interpolation control (right).

Our work concentrates on easing the burden of specifying transfer functions that change across time. Our system allows manipulation of raw or compressed datasets and provides controls to play, pause, and stop animations (see Figure 7). As with previous techniques, we provide temporal histograms for capturing global information about the changing data. In addition, the histograms in other spaces change interactively to reflect the current time step. This feedback gives the user greater understanding of the underlying structure of the temporal data. It also allows the user to specify transfer function ensembles that are unique to specific time steps.

To handle temporally irregular volumes, we introduce a keyframing approach for transitioning between ensembles of transfer functions. Our interface provides the user with the ability to identify frames in the animation that require changes, specify a new ensemble of transfer functions, and assign the ensemble to the frame. To avoid the visual discontinuities that occur when changing from one keyframe to the next, we provide an editable transition curve that specifies the blending between adjacent keyframes. An ensemble is represented in hardware as a color and opacity lookup table that was composed using one or more transfer functions (see Section 5). Blending between multiple ensembles for time-varying data is performed in a similar manner. For every color C and opacity α entry i in the new lookup table, the resulting interpolation r between two ensembles (1 and 2) can be determined from the transition curve $f(t)$ as follows:

$$\begin{aligned} C_r(i) &= C_1(i)(1 - f(t)) + C_2(i) f(t) \\ \alpha_r(i) &= \alpha_1(i)(1 - f(t)) + \alpha_2(i) f(t) \end{aligned}$$

The horizontal axis of the transition curve represents the time interval between the two keyframes. The weight of the interpolation between ensembles is described by the vertical axis.

Combining keyframed ensembles with transition curves provides an effective solution for capturing moving features in dynamic datasets while maintaining smooth animations. Keyframing can also be used as a temporal focus and context method for exploring different aspects of the data during different time intervals, similar to an existing approach for static volumes [37]. Our detailed control of the temporal aspect of transfer function specification can greatly benefit researchers who have been performing time-varying visualization in a brute-force manner with tools that are currently available.

7 Evaluation

An important consideration for a transfer function specification tool is that it does not introduce additional computational overhead and thus adversely impact interactivity. There is no measurable performance penalty when using our transfer function specification tools. Thus, the interactivity of the rendering remains the same as the original volume renderer.

To evaluate the usefulness of our proposed techniques, we performed an informal expert evaluation of the system. Expert reviews have been shown to be a useful means of evaluation, and require fewer reviewers than standard user studies [33]. We collected comments and suggestions from four experts with different relevant backgrounds. The first expert develops open source visualization software. The second performs research in the area of volume visualization. The third expert has used existing volume visualization software in a medical setting. Finally, the fourth expert is a specialist in bioengineer-

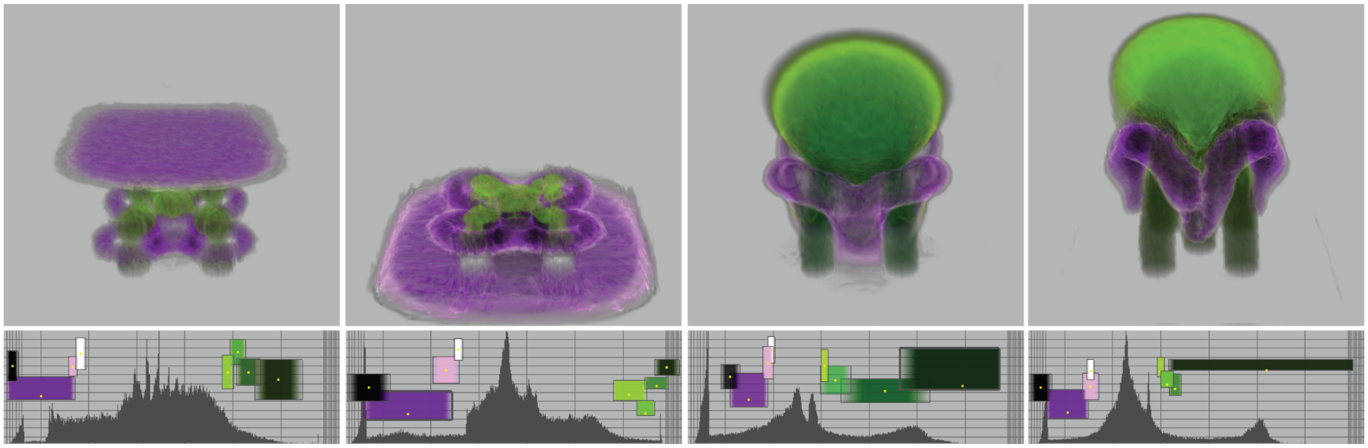

Fig. 8. Multiple transfer functions are used during different time steps of a time-varying simulation of five jets. Keyframing can be used for temporal focus and context visualization as well as a means of highlighting features that may move through the scalar space over time (as shown above).

ing and concentrates mostly on biomedical computing. These experts were given a demonstration of our system along with ParaView [14], a freely available system that has some basic transfer function specification abilities such as a zooming interface. The experts were then given the opportunity to perform their own explorations using both systems and asked a series of questions about their experience with the system compared to ParaView and other systems that they have used in the past.

Overall, the feedback was very positive and the reviewers feel that our system is useful for fast data exploration and that existing visualization systems would benefit from some of the components that we introduce in our system. The reviewers also provided many suggestions that we plan on incorporating into our system. Here we summarize some of the main advantages and disadvantages that the reviewers pointed out.

Advantages:

- The system provides fine-grain control of the data due to the resolution control that range mapping provides.
- Interactive histogram information significantly improves the ability to place widgets and to explore the volume.
- The histograms that update over time provide more information about the volume than other systems provide.
- The ability to interpolate between transfer functions over time is very useful for contextualizing the data.

Disadvantages:

- The interface could use some work to consolidate the concept of ensembles and make it more intuitive.
- Though some of the features are more powerful, they may require longer to learn to use if the user is unfamiliar with transfer function specification.
- Currently there is no undo for operations.

Beyond these general comments, we received some interesting comments about the usefulness of the system from the reviewers unique perspectives. The first reviewer mentioned that when the application he develops moved from 8-bit data to higher precision, he noticed problems associated with limits in the lookup table precision, though he had not found a reasonable solution for the problem. After the evaluation, his plan is now to add range mapping to his system. He was also impressed with the idea of creating transfer functions using combinations of other transfer functions and is evaluating this addition to his system as well. The third reviewer worked extensively with time-varying data that changes substantially over time. He commented that the ability to define different transfer functions for time steps and interpolate between them through keyframing would have saved him enormous amounts of time. He also mentioned that the ability to keyframe the range mapping would be a useful feature for these datasets. We plan to add this feature to our system as soon as possible. The fourth reviewer stated that he would like to see the features presented in our system incorporated into the biomedical simulation software that he and his collaborators use because it would facilitate the process of analysis for time-varying volumes.

8 Discussion

Many automatic methods for transfer function specification proposed in the literature focus on identifying and enhancing single features, such as boundary transitions with a 2D gradient magnitude histogram. The transfer function ensembles proposed in this work provides a way for combining different features of several transfer functions. For high dynamic range data, our range mapping proves very efficient for finding features that otherwise are impossible to distinguish. A current limitation of our method is that the range mapping effects all transfer functions in an ensemble. In addition, as pointed out in our evaluation, range mappings effect all time steps. These issues could be resolved with multiple or multidimensional range mapping tables, respectively. An interface similar to the transition curves for time-varying transfer functions could then be used to give the user temporal control over the range mappings.

Our transfer function ensembles are blended in texture space due to the simplicity of the approach as well as the performance implications. The disadvantage of performing the blending there is that the number of dimensions that can be combined is limited to three. Another approach to overcome this limitation is to perform the blending at the fragment level as in multivolume approaches (*eg.,* [23]). This type of blending is more robust for combining transfer functions in different spaces, but is more complex and will effect the rendering performance because it requires multiple lookup tables.

Although our tool was designed with the goal of specifying transfer functions for unstructured data, there is no limitations that prevents it from being applied to structured grids. For example, the datasets shown in Figures 3 and 8 originate from simulations over structured grids. In addition, our framework can easily be extended with new histograms, widgets, and blending methods. In addition, the new features described in this paper can be easily integrated into existing visualization systems because of their independence from the rendering algorithm.

9 Conclusions and Future Work

In this paper, we proposed a framework for transfer function specification based on several new features. First, we extract statistical information using multiple histogram types, including time-varying data, and allow the user to specify widgets containing color and opacity on the histograms. For histograms with high dynamic range, we proposed a range mapping equalization that focuses the analysis into smaller regions of the histogram, providing a more powerful tool than existing zooming techniques. We introduced the concept of ensembles for combining multiple transfer functions with user-controlled blending functions. Finally, we introduced a technique for transfer function specification on time-varying data that uses keyframing and transition curves to change ensembles over time. The resulting system has been demonstrated to introduce new insights in data by facilitating transfer function specification for disparate data types.

For future work, we would like overcome the issue of combining transfer functions defined on different range mapping spaces. This will extend the possibilities offered while designing the transfer functions. We would also like to try the multivolume approach for blending transfer functions. We also plan to explore automatic and semi-automatic approaches to find good keyframes positions and their respective transfer functions. Finally, we would like to incorporate comments from our evaluation and prepare the code for open source release.

Acknowledgments

The authors acknowledge Chems Touati for video production, Neely and Batina (NASA) for the Langley Fighter dataset, the SCI Institute for the Torso dataset, and Kwan-Liu Ma (UC Davis) for the Turbulent Jet and Five Jets datasets. The work of Fábio Bernardon is supported by a CNPq scholarship. The work of João Comba is supported by a CNPq grant 478445/2004-0. Linh K. Ha, Steven P. Callahan, and Cláudio T. Silva are funded by the National Science Foundation (grants CCF-0401498, EIA-0323604, OISE-0405402, IIS-0513692, and CCF-0528201), the Department of Energy, an IBM Faculty Award, Sandia National Laboratories, Lawrence Livermore National Laboratory, the Army Research Office, and a University of Utah Seed Grant.

References

[1] H. Akiba, N. Fout, and K.-L. Ma. Simultaneous classification of time-varying volume data based on the time histogram. In *IEEE VGTC/Eurographics Symposium on Visualization*, 2006.

[2] C. L. Bajaj, V. Pascucci, and D. R. Schikore. The contour spectrum. In *IEEE Visualization*, pages 167–ff., 1997.

[3] F. Bernardon, S. Callahan, J. Comba, and C. Silva. Interactive volume rendering of unstructured grids with time-varying scalar fields. In *Eurographics Symposium on Parallel Graphics and Visualization*, pages 51–58, 2006.

[4] S. Bruckner and E. Gröller. Volume shop: An interactive system for direct volume rendering. In *IEEE Visualization*, pages 671–678, 2005.

[5] S. Callahan, J. Comba, P. Shirley, and C. Silva. Interactive rendering of large unstructured grids using dynamic level-of-detail. In *IEEE Visualization*, pages 199–206, 2005.

[6] S. Callahan, M. Ikits, J. Comba, and C. Silva. Hardware-assisted visibility sorting for unstructured volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):285–295, 2005.

[7] H. Doleisch, M. Mayer, M. Gasser, P. Priesching, and H. Hauser. Interactive feature specification for simulation data on time-varying grids. In *Conference on Simulation and Visualization (SimVis)*, pages 291–304, 2005.

[8] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. In *ACM SIGGRAPH*, volume 22, pages 65–74, 1988.

[9] I. Fujishiro, T. Azuma, and Y. Takeshima. Automating transfer function design for comprehensible volume rendering based on 3d field topology analysis. In *IEEE Visualization*, pages 467–470, 1999.

[10] M. Hadwiger, C. Berger, and H. Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *IEEE Visualization*, pages 301–308, 2003.

[11] T. J. Jankun-Kelly and K.-L. Ma. A study of transfer functions generation for time-varying volume data. In *IEEE TCVG/Eurographics Workshop on Volume Graphics*, pages 51–68, 2001.

[12] A. Kaufman and K. Mueller. *Overview of volume rendering*, chapter 7, pages 127–174. Academic Press, 2005.

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

[14] Kitware. Paraview. `http://www.paraview.org`.

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

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

[17] A. König and E. Gröller. Mastering transfer function specification by using VolumePro technology. *Spring Confererence on Computer Graphics*, 17:279–286, 2001.

[18] M. Kraus, W. Qiao, and D. S. Ebert. Projecting tetrahedra without rendering artifacts. In *IEEE Visualization*, pages 27–34, 2004.

[19] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1998.

[20] E. B. Lum and K.-L. Ma. Lighting transfer functions using gradient aligned sampling. In *IEEE Visualization*, pages 289–296, 2004.

[21] K.-L. Ma. Image graphs - a novel approach to visual data exploration. In *IEEE Visualization*, pages 81–88, 1999.

[22] K.-L. Ma. Visualizing time-varying volume data. *Computing in Science and Engineering*, 5(2):34–42, 2003.

[23] R. Managuli, Y. M. Yoo, and Y. Kim. Multi-volume rendering for three-dimensional power doppler imaging. *IEEE Ultrasonics Symposium*, 4:2046–2049, 2005.

[24] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: a general approach to setting parameters for computer graphics and animation. *Computer Graphics*, 31:389–400, 1997.

[25] K. Moreland and E. Angel. A fast high accuracy volume renderer for unstructured data. In *IEEE Symposium on Volume Visualization and Graphics*, pages 9–16, 2004.

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

[27] S. Potts and T. Möller. Transfer functions on a logarithmic scale for volume rendering. In *Graphics Interface*, pages 57–63, 2004.

[28] W. Qiao, D. S. Ebert, A. Entezari, M. Korkusinski, and G. Klimeck. Volqd: Direct volume rendering of multi-million atom quantum dot simulations. In *IEEE Visualization*, pages 319–326, 2005.

[29] 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), 2006.

[30] S. Roettger, M. Bauer, and M. Stamminger. Spatialized transfer functions. In *IEEE VGTC/Eurographics Symposium on Visualization*, pages 271–278, 2005.

[31] P. Sereda, A. V. Bartroli, I. W. O. Serlie, and F. A. Gerritsen. Visualization of boundaries in volumetric datasets using LH histograms. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):208–218, March/April 2006.

[32] N. Svakhine, D. S. Ebert, and D. Stredney. Illustration motifs for effective medical volume illustration. *IEEE Computer Graphics and Applications*, 25(3):31–39, 2005.

[33] M. Tory and T. Möller. Evaluating visualizations: Do expert reviews work? *IEEE Computer Graphics and Applications*, 25(5):8–11, 2005.

[34] M. Tory, S. Potts, and T. Möller. A parallel coordinates style interface for exploratory volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11(1):71–80, 2005.

[35] F.-Y. Tzeng, E. B. Lum, and K.-L. Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):273–284, 2005.

[36] I. Viola, A. Kanitsar, and E. Gröller. Importance-driven volume rendering. In *IEEE Visualization*, pages 139–145, 2004.

[37] Y. Wu, H. Qu, H. Zhou, and M.-Y. Chan. Focus + context visualization with animations. In *IEEE Pacific-Rim Symposium on Image and Video Technology*, pages 1293–1302, 2006.

[38] Y. Wu, H. Qu, H. Zhou, and M.-Y. Chan. Fusing features in direct volume rendering images. In *International Symposium on Visual Computing*, pages 273–282, 2006.

[39] H. Younesy, T. Möller, and H. Carr. Visualization of time-varying volumetric data using differential time-histogram table. In *IEEE VGTC/Eurographics Workshop on Volume Graphics*, pages 21–30, 2005.

[40] X. Yuan, M. X. Nguyen, B. Chen, and D. H. Porter. HDR VolVis: High dynamic range volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):433–455, July/August 2006.