Data Centric Transfer Functions for High Dynamic Range Volume Data

Amit Chourasia
San Diego Supercomputer Center
University of California San Diego
amit@sdsc.edu

Jürgen P. Schulze
California Institute for Telecommunications and Information Technology
University of California San Diego
jschulze@ucsd.edu

ABSTRACT

Creating effective transfer functions for high dynamic range scalar volume data is a challenging task. For data sets with limited information about their content, deriving transfer functions using mathematical properties (gradient, curvature, etc.) is a difficult trial and error process. Traditional methods use linear binning to map data to integer space for creating the transfer functions. In current methods the transfer functions are typically stored in integer look-up tables, which do not work well when the data range is large. We show how a process of opacity guidance with simple user interface can be used as the basis for transfer function design. Our technique which uses opacity weighted histogram equalization lets users derive transfer functions for HDR floating point easily and quickly. We also present how to adopt these techniques for real-time interactive visualization with minimal pre-processing. We compare our techniques with traditional methods and show examples.

Keywords: Transfer Function, High Dynamic Range, Histogram Equalization, Floating Point Data, Volume Rendering

1 INTRODUCTION

Volumetric visualization lets users present complex, 3D data sets in visual form. An important part of this is the design of transfer functions for color and opacity, to be mapped on the data values. Using a better transfer function is analogous to increasing signal to noise ratio, where the signal is the information desired by the user. As the notion of signal is not defined for volume datasets, there is a need to do this task in a convenient and often exploratory manner. This is where the design of transfer functions plays an important role. Unfortunately, this process is often time consuming and cumbersome.

Transfer function design for medical datasets has been the focus of many studies. In general, this type of data consists of various layers of biological materials (flesh, bone, tissue, etc.). These materials can often be segmented by algorithmically identifying their boundaries.

But what about non-medical datasets, like datasets resulting from computational simulations, like computational fluid dynamics, weather, or simulations of physical processes? With faster processor speeds and the widespread adoption of clustering technology, the use of computational simulation to model physical phenomena is widespread. These simulations can generate enormous amounts of data. The characteristics of these datasets are very difficult to categorize but they often show some common traits like a high dynamic range (HDR), and subtle details in high fidelity. Our work is focused on HDR floating point data sets.

For HDR floating point data sets, transfer function design becomes an increasingly difficult task, primarily because the data range by far exceeds the usual 8 bits of color resolution on the monitor. It is unlikely that a single classifier will provide the desired results. In addition, there is often limited a priori information about the data content which compounds the problem for creating good visualizations.

Data distribution directly affects opacity and color mappings. If interesting data is clustered within small data ranges, then color and opacity need to be tuned specifically for each such range to show it at the maximum possible detail in the visualization result.

In this paper, we present a method which lets users design transfer functions without using derived and hard to comprehend mathematical properties like gradient, curvature, tensor, etc. In our method, we use an extension of histogram equalization, along with opacity guidance provided by the user. Our new algorithms allows the user to minimize the effort of tuning the transfer functions. We show how to utilize data distribution as a basis for transfer function design and how it can be done in a real-time software application. In addition, we compare our results to the popular technique of mapping a color gradient linearly to a data range. The transfer functions our algorithms generates can also be generated manually, but it would take the user considerably more time to do so from the start. Our algorithms are intended to provide a fast way to generate reasonable transfer functions which can then be refined manually.

The main contributions of this paper are:

1. Opacity Weighted Histogram Equalization (OWHE) as the basis for transfer function design.

2. Efficient utilization of the available color space, thus yielding a more detailed visualization of structures in data sets.

3. Real-time interaction with a simple and intuitive user interface to access the algorithm's parameters.

The paper is structured as follows. In Section 2 we review related work. In Section 3 we discuss the underlying theoretical background. Section 4 describes the algorithms we have developed. Our user interface is presented in Section 5. We present the application of our method to three typical high-dynamic range data sets in Section 6. Section 7 concludes the paper and Section 8 suggests areas for future work.

2 RELATED WORK

A multitude of methods have been proposed to generate transfer functions. They can be broadly classified into four categories which are described in detail by Pfister et al. [11]. The first are trial and error methods where the user tweaks parameters to achieve the desired result. Second, data centric approaches work without assuming a data model [1]: the contour spectrum which computes metrics of the data and integrates the results with the user interface used to set iso-values. Third, a data centric approach which assumes an underlying data model [5, 6]: this method semi-automatically generates the transfer function by utilizing the complex relationship between data and its optical properties, the assumption being that the features of interest are boundaries between materials; our approach is not based on material boundaries, gradient values, or multidimensional transfer functions. Fourth, an image centric method based on organized samples; this is similar to the Design Galleries concept [9].

Other approaches, related to the ones presented in this paper are volume visualization using approaches from high dynamic range imaging (HDRI) [16]. The HDR method performs volume rendering in floating-point space, thus preserving precision, and then uses tone mapping to display the result. Practically, insufficient HDR display hardware [4] and difficult interpretation of tone mapped results are remaining problems with this technique. Furthermore, the trial and error effort now moves to tone mapping [3] and exposure selection instead of transfer function design.

Potts and Moeller [12] propose to use transfer functions on a logarithmic data scale. This method provides a mechanism for the user interface for transfer function design. We think that for many users it is more intuitive to see the transfer function on a linear scale and interactively zoom in and out where necessary.

Tzeng and Ma [13] use the floating-point data space during segmentation to identify the various materials present in a data set. However, their approach does not work well if the data set contains continuous densities which do not occur as clusters. Also, defining and searching clusters is a cumbersome task. Lundstorm et. al. [8] use spatial coherence for transfer function design for medical datasets.

For data sets without distinguishable materials and highly inhomogeneous data distribution, none of the above methods sufficiently addresses the dynamic range of the data and its implications. Our method aims to semi-automatically maximize the use of color space for regions of interest, while still allowing the visualization of the entire data set. Most existing volume rendering algorithms can seamlessly incorporate our method into their classification and rendering pipeline as an additional option for transfer function design.

3 HIGH DYNAMIC RANGE TRANSFER FUNCTIONS

This paper addresses the rendering of single-valued floating-point volume data sets. The challenge is that it is often unknown to the user which parts of the floating-point range are of interest, i.e., what data ranges colors and opacities should be mapped to. Many traditional methods allow the user to specify minimum and maximum values to constrain the data range, and then specify color and opacity mappings for the selected range. Histograms can facilitate the process of finding the range of interest to visualize, but they do not help much when the data values are spread out over a very large range. Often, the user wants a quick overview of what is in a data set before making fine adjustments to pull out part of it. A simple way that can achieve this is an opacity function of constant, low opacity. Or a transfer function based on the histogram can be created by mapping higher opacity to higher histogram values, or vice-versa. All these methods pose significant limitations like limited control over the opacity mapping, or a cumbersome trial and error process.

A simple, but frequently used approach to map floating point values in volume data sets to colors is to linearly map the values between minimum and maximum value to a user specified color gradient. This approach is referred to as Linear Binning in the upper half of the box of Existing Techniques in Figure 1. In the recent past, more sophisticated methods have been developed. Some approaches retain the high dynamic data range during the volume rendering step by using a floating-point image buffer. Then, tone mapping is used to convert the HDR image to the color space today's graphics hardware uses, which is usually 24 bits per pixel. Figure 1 illustrates this approach in the lower half of the Existing Techniques box.

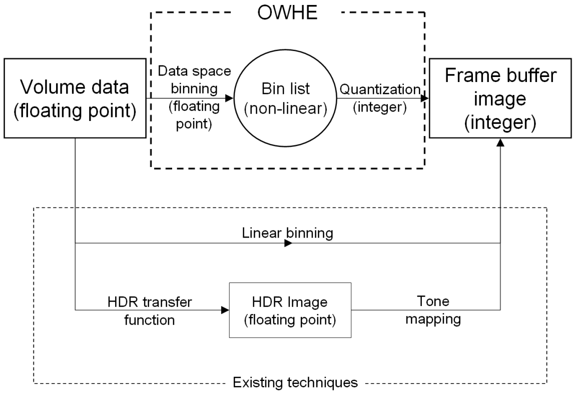

Figure 1: Flowchart showing the Opacity Weighted Histogram Equalization algorithm with the existing techniques of linear binning and HDR with tone mapping. Our approach is shown in the box labeled OWHE, the existing techniques are shown in the dotted box below.

The latest graphics cards support floating-point textures and even floating-point frame buffers. Texture hardware-based volume rendering benefits from these by allowing it to use much higher precision when volume slices are blended with the frame buffer, so that more features of the data set can be preserved during rendering. However, the problem of mapping color and opacity to data over a large data range still persists. Many floating-point volume data sets have sections of high data activity, as well as large sparsely used data ranges. This paper presents an approach to support the user in creating meaningful transfer functions for these kinds of data sets with minimal manual intervention.

3.1 Histogram Equalization

The transfer function generation method we are presenting in this paper is an extension of histogram equalization (HE). In image processing, histogram equalization is used to improve the contrast of images. The idea is that the brightness values in the image are increased or decreased such that the histogram becomes flat. The formula for histogram equalization is:

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{m} \qquad k = 0, 1, 2, ..., L-1$$

s_k are the equalized histogram values, r_k are the original histogram values, L is the number of gray levels in the image, n_j is the number of times each gray level appears in the image, and m is the total number of pixels in the image.

3.2 Opacity Weighted Histogram Equalization

Histogram equalization has been used quite successfully in image processing to improve the image quality by improving contrast. In volume rendering, HE alone can provide a quick way to considerably improve the color mapping; we will show this in Section 6. However, HE alone does not help with the creation of the opacity transfer function. Our transfer function generation approach is a two-step approach. In the first step, the user uses HE to create a first approximation of a useful color mapping for the data set. At this point, a constant, low opacity function or maximum intensity projection is used to show the entire data set. In the second step, the user places a few simple transfer function widgets on the transfer function to pull out specific parts of the data set. Then we apply our Opacity Weighted Histogram Equalization (OWHE) algorithm to refine the color mapping, using the additional hints the software gets from user defined opacity widgets. Our approach is depicted in the upper half of Figure 1.

The basic idea of OWHE is to modulate the number of times a voxel value occurs in the data set by the opacity the user assigned to it. Thus, higher color resolution is created in areas of high opacity, whereas in low opacity regions, fewer colors are used. In our tests, we found that this approach can achieve very good visualization results with much less effort than if the user tried to create the same transfer function manually.

Our modification of the Histogram Equalization function is indicated below. op_j is the user specified opacity of the data value.

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{op_j \times n_j}{m} \qquad k = 0, 1, 2, ..., L-1$$

Figure 2 compares linear color mapping to OWHE. In OWHE, the colors are more compressed the higher the data frequencies in the histogram, and the higher the desired opacity. In areas with fewer data points in the histogram or lower opacity, larger data ranges get mapped to the same color. We use the same color map in both sample images.

For rendering, we are using a standard light emission model for the volume, along with back-to-front alpha blending. We target standard graphics and display hardware, based on a precision of 24 bits per pixel. Our algorithm can potentially improve the visualization results even on true HDR displays which can display more than 24 bits per color.

4 IMPLEMENTATION DETAILS

We implemented the HE algorithm, as well as the OWHE algorithm and integrated them with the DeskVOX [2] volume rendering software. Both

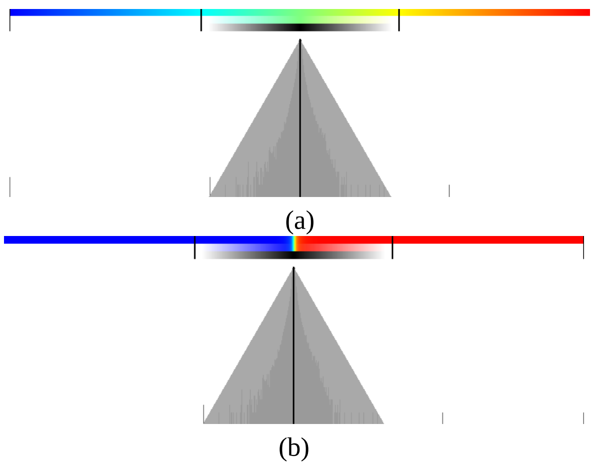

Figure 2: (a) linear binning, (b) OWHE. The data values are located along the horizontal axis. The bars at the top of each image show, from top to bottom: color only, color and opacity, opacity only, bin boundaries. Shown below is the histogram, overlaid by the opacity function.

algorithms have various parameters which can be changed by the user from within the volume rendering application. We implemented these parameters because they turned out to be useful when using the algorithms on real data sets.

Cull Skip regions: This parameter specifies whether regions with transfer function widgets which map zero opacity to a data range (Skip widgets) should be excluded from the binning process. This impacts especially the HE algorithm, because it, by default, uses the entire data range.

Cull Duplicate Values: By default the HE process uses all data values in the volume and distributes them evenly into bins. With this option enabled, the user can specify that duplicate values be removed, so as to evenly distribute different values only. This makes a difference if the data set contains disproportionately large amounts of certain values, which are not of particular interest but should not entirely be excluded from the binning process.

Trim to Range: The data range the algorithm operates on can be constrained to be within a minimum and maximum value. These minimum and maximum values can be the absolute min/max values in the data set, or they can be values in-between. If this option is disabled, all data values are used for the HE algorithm.

Fast Sampling: Using all data values of the volume in the algorithms can be slow if the volume is large. The Fast Sampling option uses only the specified number of samples for the binning process, which speeds up the algorithm, but also makes it less accurate. As Tzeng and Ma have stated in [13], sampling about 1% of the data values often achieves satisfactory results.

5 USER INTERFACE

We developed a user interface for the transfer function editor which integrates our transfer function generation methods with traditional approaches to transfer function design. Figure 3 shows a screen shot of the editor. In the upper half of the window the transfer function is displayed. At the top of it are three colored bars: the upper one shows just the color mapping, the next shows the combined mapping of color and opacity, and the third shows opacity only. The data values are located on the horizontal axis of the graph. The black, vertical lines can be grabbed with the mouse and moved to left and right, so as to place a particular color control point in the color bar. In the same manner the user can use multiple widgets to create the opacity mapping.

Below the color bars is the opacity function. The gray area is the opacity mapping. The opacity function can be created from four different elements (widgets): pyramids, Gaussians, custom functions with control points, and Skip widgets. As with the colors, each opacity widget can be grabbed and moved with the mouse by clicking on a black, vertical line. The user can composite the transfer function with multiple widgets; at overlapping areas of widgets the opacity is determined by the maximum value in one of the widget.

The horizontal axis represents linear data values. Both color and opacity transfer functions are displayed in linear data range. When our algorithms are active, it is more difficult to manipulate the color widgets because the control lines can be much closer together. It is easier to use the control lines when the user zooms in to them. To zoom, the user can use numeric range or use the mouse wheel where the mouse location acts as the center for the region to be zoomed. In pressed state the mouse wheel enters pan mode, where the user can move the transfer function to the left and right.

5.1 Parameter Window

Figure 4 shows the dialog window with the parameters available for our algorithms. When the button labeled *Distribution based data range mapping* is off, the transfer function uses a linear distribution of colors over the range specified by *Start value* and *End value*, and our algorithms are disabled. We used this setting to compare our algorithms to the linear binning approach in Section 6. When *Distribution based data range mapping* is checked, the algorithms described in this paper are applied. The user can choose between HE and OWHE, which have been described in Section 3.

6 RESULTS AND DISCUSSIONS

We have applied our technique to three different scalar floating-point data sets. The data sets have different dynamic range and properties. We chose a representative single time step from the following three temporal

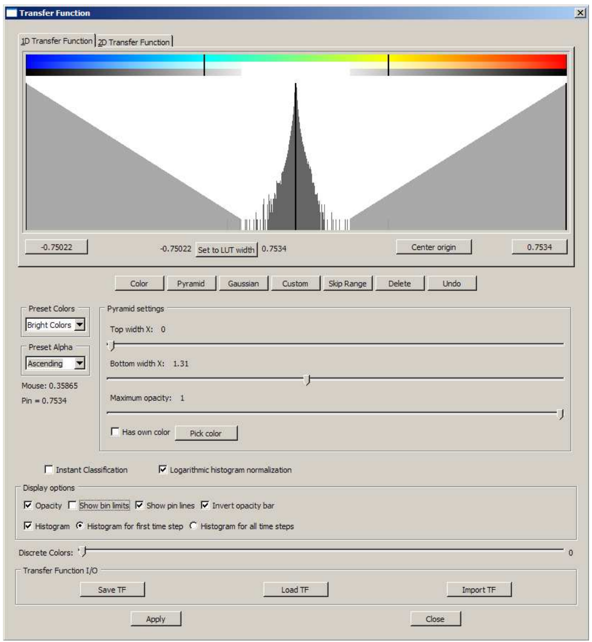

Figure 3: Our widget-based transfer function editor.

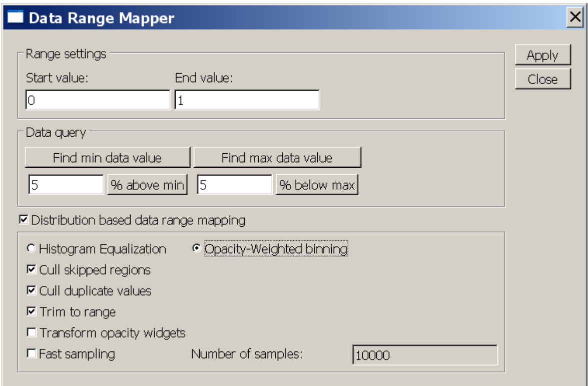

Figure 4: The dialog window to set the new data range mapping parameters.

datasets, all of which are results from simulations on supercomputers.

1) **TeraShake** is an earthquake simulation [10, 15], done to study seismic wave propagation based on kinematic and dynamic rupture models. We use the X component of velocity with a data set size of $750 \times 350 \times 100$ voxels. The data covers a small range but has subtle details in very narrow regions.

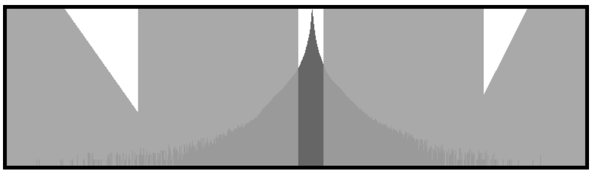

Figure 5: Histogram (logarithmic scale) and opacity function of the TeraShake earthquake simulation data set.

2) **Isabel** is the result of the simulation of a hurricane [14]. We use the snow data with a data set size of $500 \times 500 \times 100$ voxels.

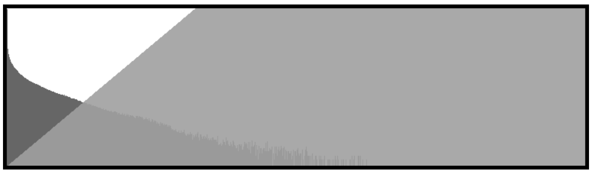

Figure 6: Histogram (logarithmic scale) and opacity function of the hurricane Isabel data set.

3) **Enzo** is a simulation [7] of supersonic compressible turbulence at Mach 6 performed with the Enzo code for cosmology and astrophysics, using 512 CPUs of the DataStar supercomputer at the San Diego Supercomputer Center. It was designed to model a sub-volume (linear size: 5 pc = $1.5 \cdot 10^{17}$ m) within a star forming a molecular cloud. We use a sub-sampled matter density data set with a size of $512 \times 512 \times 128$ voxels.

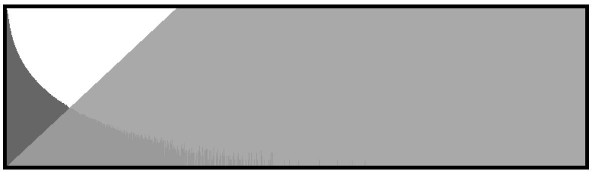

Figure 7: Histogram (logarithmic scale) and opacity function of the cosmology data set Enzo.

More detailed properties of the above data sets are listed in Table 1, and the data histograms are shown in Figures 5, 6, and 7. Figures 8a-c show rendering results of different datasets, generated with our OWHE algorithm.

6.1 Comparison of Linear Color Mapping with HE and OWHE

We compare our HE and OWHE methods to the popular linear color mapping method henceforth referred to as standard method. For the comparisons, we created appropriate opacity transfer functions for each dataset. The opacity functions are shown in figures 5, 6, 7, along with the data histogram.

In the standard method, color map and opacity are applied to a linearly data range. The renderings achieved with the standard method are shown in Figures 8(a),(d) & (g). For HE and OWHE, we use the same opacity transfer functions as for the standard method, but the color mapping gets modified by the algorithms. The processing time for HE and OWHE depends on the number of voxels in the data set and is on the order of several seconds. The corresponding renderings for HE and OWHE are shown in Figures 8(b),(e),(h) & (c),(f),(i).

In the TeraShake data set, most of the data is located around the value 0.0 (see histogram in Figure 5), which represents no ground motion. In the images, the opacity for the data is set such that wave features are visible. A color map has been created to show positive velocity as red-orange-yellow, and negative velocity as blue-cyan-green. Black has been chosen to show a velocity of 0.0 with no opacity. A desirable visualization for this is to show the wave propagation structure in low, mid, and high range all together. With the standard method (see Figure 8(a)) it is difficult to observe the wave propagation features. The HE method (see Figure 8(b)) shows only a minor improvement over the standard method, but our OWHE method (see Figure 8(c)) is able to bring out subtle details of the wave form and its location, which takes a great deal of effort if done manually.

The Isabel data set we chose for this comparison is snow mixing ratio. We created a visualization which lets us see the entire volume, and in addition the coastline of Florida is visible for orientation. The opacity function 6 has been chosen such that higher snow mixing ratio is more opaque. As can be seen in Figure 8(d), visualization with the standard method provides information just for a relatively small region. The HE method is of not much help in this instance but the OWHE method (see Figure 8(e)) is able to show much more detail of the data value distribution by more effectively using the available color space (see Figure 8(f)).

The Enzo volume shows "‘matter density"’ during a star formation process. We chose an opacity function similar to that used for the Isabel data set, where higher density is completely opaque and lowest is transparent. The standard method (see Figure 8(g)) provides little distinction in features of interest. The HE method (see Figure 8(h)) shows some detail. The OWHE method (see Figure 8(i)) brings out more details in the data, even thin filaments can be seen in high fidelity. The V- and U-shaped shocklets or "‘Mach cones"’, characteristic of the supersonic turbulence found in interstellar gas, can clearly be seen.

	Enzo	**Isabel**	**TeraShake**
# Voxels	$33.6 \cdot 10^6$	$25.0 \cdot 10^6$	$28.1 \cdot 10^6$
% Unique	73.2	86.5	71.5
Min	$8.52 \cdot 10^{-3}$	0.00	$-6.68 \cdot 10^{-1}$
Max	$2.23 \cdot 10^2$	$1.35 \cdot 10^{-2}$	$5.97 \cdot 10^{-1}$
Mean	$9.28 \cdot 10^{-1}$	$2.23 \cdot 10^{-6}$	$-1.43 \cdot 10^{-4}$
Std dev	1.58	$1.54 \cdot 10^{-5}$	$8.50 \cdot 10^{-3}$

Table 1: Data statistic for Enzo ($512 \times 512 \times 128$), Isabel ($500 \times 500 \times 100$) and TeraShake ($750 \times 375 \times 100$). % unique is the percentage of duplicate values, removed with our Cull Duplicates parameter. Min/Max are the minimum and maximum data values. Mean is the mean value of all data values, including duplicates. Std dev is the standard deviation over all data values, including duplicates.

6.2 Limitations

Our methods works well for HDR floating point data, but they have some limitations. The methods are not suitable to all datasets. For example in medical datasets with little variation in the data our method will not work well. Additionally, in cases where high color fidelity is not of much importance for visualization our method will not be beneficial.

7 CONCLUDING REMARKS

We presented techniques to facilitate the creation of transfer functions for high dynamic range floating point data sets by taking data distribution and user input into account. We compared our techniques to standard methods and demonstrated how our techniques can be used to create useful transfer functions faster and with less effort. Our experience with this technique indicates that time required to get meaningful images is significantly reduced but a survey would be needed to quantify and test this hypothesis. The color space utilization with our method is efficient leading to high fidelity detail in renderings.

8 FUTURE WORK

In the future, we want to improve our rendering method by supporting higher precision frame buffers (e.g., with Nvidia’s GL_NV_float_buffer extension). We would also like to utilize HDRI color space for transfer functions with our technique. Other existing histogram equalization techniques could be added as comparative exploration methods. In addition, we want to extend the application of this technique to temporal data sets.

9 ACKNOWLEDGEMENTS

We would like to thank Steve Cutchin (SDSC) for helpful comments and ideas. Alexei Kritsuk provided the Enzo data and helped us to explain the visualization. The research was funded through NSF SDSC core grant and CalIT2 research grant.

REFERENCES

[1] C.L. Bajaj, V. Pascucci, and D.R. Schikore. *The Contour Spectrum*. IEEE Visualization ’97 Proceedings, pp. 167–173, 1997.

[2] Deskvox. *Deskvox volume rendering software by Jurgen P. Schulze*. URL: http://www.calit2.net/ jschulze/projects/vox/, 2006.

[3] R. Fattal, D. Lischinski, and M. Werman. *Gradient domain high dynamic range compression*. In Proceedings of ACM SIGGRAPH 2002, Computer Graphics Proceedings, Annual Conference Series. ACM Press / ACM SIGGRAPH, July 2002. 10, 13, 15, 2002.

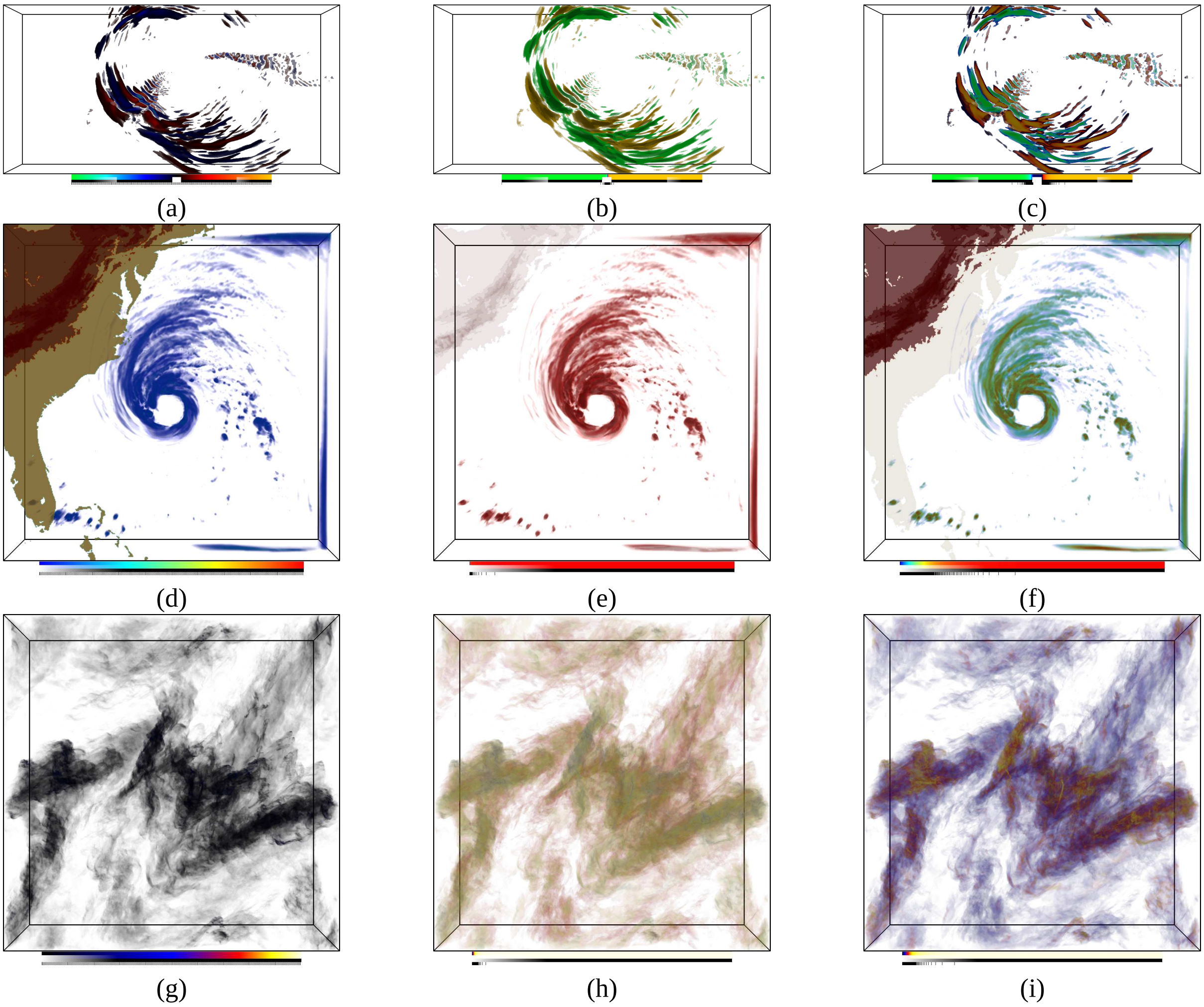

Figure 8: (a),(d),(g) Linear binning; (b),(e),(h) Histogram Equalization; and (c),(f),(i) OWHE renderings

[4] A. Ghosh, M. Trentacoste, , and W. Heidrich. *Volume Rendering for High Dynamic Range Displays*. Proc. EG/IEEE VGTC Workshop Volume Graphics '05 , July 2002. 10, 13, 15, 2005.

[5] G. Kindlmann and J.W. Durkin. *Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering*. Proceedings of the IEEE Symposium on Volume Visualization '98, pp. 79–86, 1998.

[6] J. Kniss, G. Kindlmann, and C. Hansen. *Multi-Dimensional Transfer Functions for Interactive Volume Rendering*. IEEE Transactions on Visualization and Computer Graphics, 8(3), pp. 270–285, 2002.

[7] A. Kritsuk, R. Wagner, M. Norman, and P. Padoan. *High Resolution Simulations of Supersonic Turbulence in Molecular Clouds, in Numerical Modeling of Space Plasma Flows*. Proceedings of CalSpace-IGPP International Conference, Palm Springs, 26-30 March 2006, to appear in Publ. Astron. Soc. Pacific., 2006.

[8] Claes Lundström, Anders Ynnerman, Patric Ljung, Anders Persson, and Hans Knutsson. *The alpha -histogram: Using Spatial Coherence to Enhance Histograms and Transfer Function Design*. EUROVIS - Eurographics /IEEE VGTC Symposium on Visualization 2006, 2006.

[9] 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*. ACM SIGGRAPH '97 Proceedings, pp. 389–400, 1997.

[10] K.B. Olsen, S. Day, J.B. Minster, Y. Cui, A. Chourasia, M. Faerman, R. Moore, P. Maechling, and T. Jordan. *Strong Shaking in Los Angeles Expected From Southern San Andreas Earthquake*. Geophys. Res. Lett., 33, L07305, doi:10.1029/2005GL025472, 2006.

[11] H. Pfister, B. Lorensen, C. Bajaj, G. Kindlmann, W. Schroeder, L. Sobierajski Avila, K. Martin, R. Machiraju, and J. Lee. *Visualization View-*

points: The Transfer Function Bake-Off. IEEE Computer Graphics and Applications, May/June 2001 (Vol. 21, No. 3), pp. 16–22, 2001.

[12] S. Potts and T. Moeller. *Transfer Functions on a Logarithmic Scale for Volume Rendering*. Proceedings of Graphics Interface '04, 2004.

[13] Fan-Yin Tzeng and Kwan-Liu Ma. *A Cluster-Space Visual Interface for Arbitrary Dimensional Classification of Volume Data*. Proceedings of the Joint Eurographics-IEEE TVCG Symposium on Visualization '04, May, 2004.

[14] VisContest. *Visualization Contest at IEEE Visualization Conference '04*. URL: http://vis.computer.org/vis2004contest/data.html, 2004.

[15] VisContest. *Visualization Contest at IEEE Visualization Conference '06*. URL: http://vis.computer.org/vis2006/contest_c.html, 2006.

[16] X. Yuan, M.X. Nguyen, B. Chen, and D.H. Porter. *High Dynamic Range Volume Visualization*. IEEE Visualization '05 Proceedings, pp. 327–334, 2005.