

High Dynamic Range Volume Visualization

Xiaoru Yuan^{†*}

Minh X. Nguyen^{††}

Baoquan Chen^{†‡}

David H. Porter^{§§}

Department of Computer Science and Engineering, University of Minnesota at Twin Cities[†]
The Laboratory for Computational Science and Engineering (LCSE), University of Minnesota at Twin Cities[§]

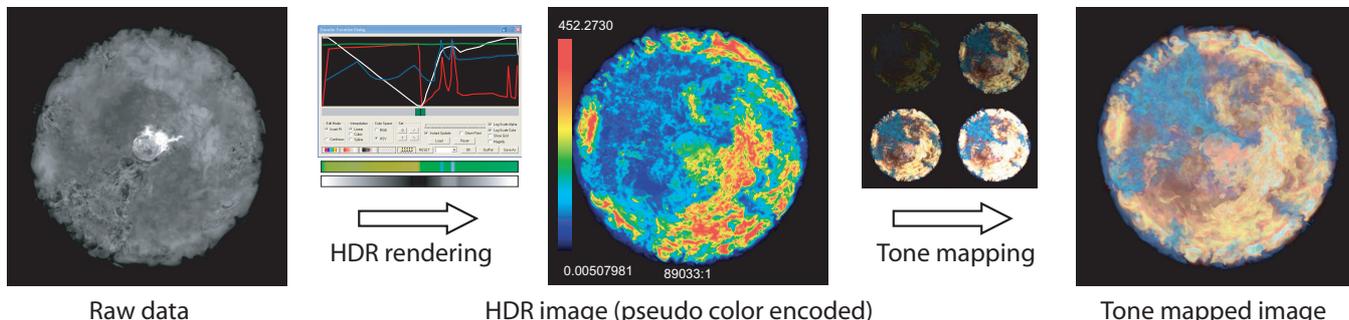


Figure 1: Pipeline of high dynamic range volume visualization. The input is a scalar volume with high precision and/or high resolution (e.g. 2048^3). The user defines a transfer function using a novel non-linear magnification interface. The volume rendering output is in high dynamic range image format. By applying a tone mapping operator, the final result can be displayed on a regular low dynamic range display device.

ABSTRACT

High resolution volumes require high precision compositing to preserve detailed structures. This is even more desirable for volumes with high dynamic range values. After the high precision intermediate image has been computed, simply rounding up pixel values to regular display scales loses the computed details. In this paper, we present a novel high dynamic range volume visualization method for rendering volume data with both high spatial and intensity resolutions. Our method performs high precision volume rendering followed by dynamic tone mapping to preserve details on regular display devices. By leveraging available high dynamic range image display algorithms, this dynamic tone mapping can be automatically adjusted to enhance selected features for the final display. We also present a novel transfer function design interface with non-linear magnification of the density range and logarithmic scaling of the color/opacity range to facilitate high dynamic range volume visualization. By leveraging modern commodity graphics hardware and out-of-core acceleration, our system can produce an effective visualization of huge volume data.

CR Categories: D.2.2 [Software Engineering]: Design Tools and Techniques—User interfaces; I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms I.4.0 [Image Processing and Computer Vision]: General—Image Displays

Keywords: Volume Rendering, High Dynamic Range, Focus + Context Techniques, User Interfaces, Transfer Function Design, Non-linear Magnification

*e-mail: xyuan@cs.umn.edu

†e-mail: mnguyen@cs.umn.edu

‡e-mail: baoquan@cs.umn.edu

§e-mail: dhp@lcse.umn.edu

1 INTRODUCTION

Large fluid dynamical simulations performed on supercomputer systems produce huge data sets with both high resolutions and high precision values. One typical data set we deal with in this paper is from the vorticity volumetric simulation by Mach 1 homogeneous compressible turbulence [20, 21] using the gas dynamics Piecewise-Parabolic Method (PPM) [33, 35] on a grid of 2048^3 cells. This simulation was run on the TeraGrid cluster at the National Center for Supercomputing Applications (NCSA) using a number of dual-CPU machines that varied between 80 and 250 nodes over the course of a 2-month run [34].

Such data sets impose several issues on visualization performed on commodity graphics hardware when a high fidelity visualization is desired. First, the data generated is in 32-bit floating point precision. Quantizing it to 8-bit scalar values to fit the texture-based volume rendering on commodity graphics hardware will severely degrade the data quality and lose subtle, but important details.

Second, the number of sampling slices must be comparable with the spatial resolution of the volume data to ensure sufficient volume sampling. It is necessary to set the opacity of each slice small enough to allow as many slices as possible, hence maximizing the number of phenomena that will be visible to the user. This requires very low opacity assignment for each slice; consequently, the opacity value will be rounded to zero in a low precision rendering system. Therefore, it is desirable to have high precision compositing so that detailed structures residing in the data can be preserved.

The third issue is the transfer function design. To handle high precision data sets, the transfer function should have comparable axis resolutions for both intensity and color/opacity. Displaying and editing transfer functions with up to thousands of entries on regular displays (1K – 2K pixel resolution) presents another challenge.

Lastly, on the output side, with the intermediate image of high precision computed, simply rounding pixel values to regular display scales would lose the computed details. It is preferable to be able to retain as much of the computed high precision (i.e., high dynamic range - HDR) visualization as possible and provide dynamic color intensity mapping (i.e., tone mapping) for regular displays.

In this paper, we present a novel volume visualization method for rendering volume data with both high spatial and high intensity resolutions. Our method performs high precision volume rendering followed by dynamic tone mapping to preserve details on regular display devices. With the input from the user, our system automatically adjusts the tone reproduction for the final display to enhance selected features. Our system is also able to directly output the rendering results to high dynamic range video which can be played by HDR viewing software available in public domain. We also present a novel transfer function design interface with non-linear magnification of intensity range and logarithmic scaling of color/opacity range to facilitate high dynamic volume visualization. By leveraging modern commodity graphics hardware and out-of-core acceleration, our system can produce interactive visualization of huge volume data. The pipeline of our system is illustrated in Figure 1. The main contributions of this paper are twofold:

1. We develop a high dynamic range volume rendering technique to preserve details residing in large volume data;
2. We present a novel transfer function design interface with non-linear magnification for high dynamic range volume data;

The remainder of the paper is organized as follows. In Section 2, we briefly review related work. We discuss technical issues involved in high dynamic range volume visualization in Section 3; this includes high precision alpha compositing (Section 3.1), a new transfer function design interface for high dynamic range volume rendering (Section 3.2), and tone reproduction (Section 3.3). Implementation details are described in Section 4 followed by results and discussions (Section 5). Finally we present our conclusions in Section 6.

2 RELATED WORK

Dynamic range is defined as the ratio between the maximum and the minimum non-zero tonal values in an image. While 8-bits-per-channel image representation is popular, high dynamic range scenes are pervasive in nature. In recent years, obtaining and displaying high dynamic range images have received a lot of attention in the computer graphics community. Algorithms have been developed for capturing both photographs [2] and videos [12] with extended dynamic range. Simultaneously, tone mapping operators [5, 6, 16, 18, 25, 29, 30] have also been developed to reduce the dynamic range so that HDR images can be displayed on normal 8-bits-per-channel displays. In general, two types of tone reproduction operators have been proposed: global (spatially uniform) operators and local (spatially varying) operators [3]. Global operators apply the same transformation to every pixel in an image. A global operator may depend upon the contents of the image as a whole, so long as the same transformation is applied to every pixel. Conversely, local operators apply a different scale to different parts of an image. In addition to reducing the range of luminance, tone reproduction can also be used to mimic perceptual qualities, resulting in an image which provokes the same responses as when viewing the scene in the real world. With the development of commodity graphics hardware, many tone mapping algorithms could be accelerated on graphics hardware [10]. Recently, a high dynamic range display device has been developed [27] based on a combination of an LCD panel and a DLP projector.

The benefits of rendering high dynamic range output, especially its applications to visualization, have not been seriously explored. For direct volume rendering, each voxel contributes to the final image by an alpha compositing operation. For large volume data with highly uneven distribution in intensity, rendered images are likely to have very different brightnesses between highlights and dark regions. High dynamic range rendering techniques are necessary to

preserve all rendering details. Only very recently, Ghosh et al. [9] investigate the use of high dynamic range display technology for volume rendering. Their emphasis is on mapping of the transfer function to a perceptually linear space over the range of intensities that one high dynamic image display can produce. The goal is to reserve several just noticeable difference (JND) steps of intensities for spatial context apart from clearly depicting the main regions of interest. In this paper, we conduct a systematic investigation on high dynamic range volume visualization, especially focussing on visualizing large volumetric data sets with both high resolution and high precision generated from large scale computer simulations. We also target popularly available 8-bit display devices.

3 HIGH DYNAMIC RANGE VOLUME RENDERING AND DISPLAY

In this section we discuss three issues pertaining to high dynamic range volume visualization and display: alpha precision, transfer function design, and tone mapping.

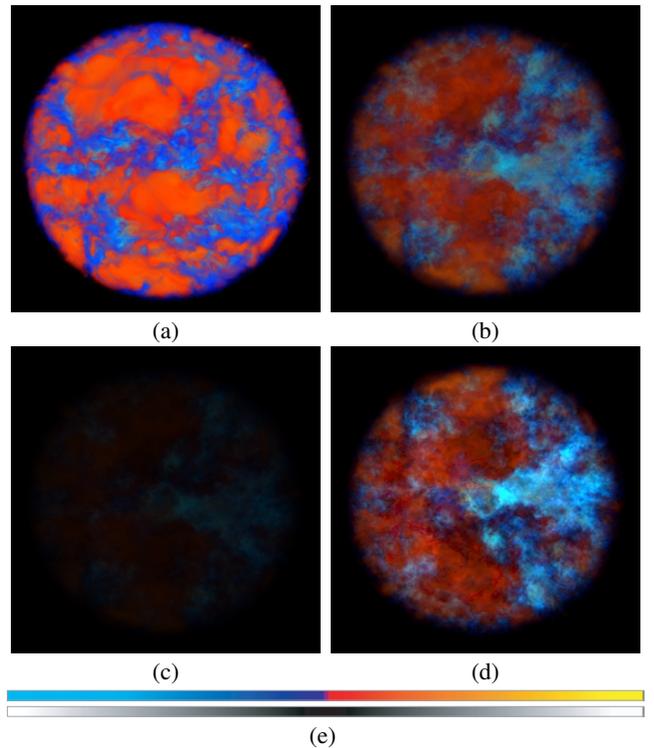


Figure 2: High dynamic range volume rendering using different global alpha α_g : (a) $\alpha_g = 0.5$, (b) $\alpha_g = 0.05$, (c) $\alpha_g = 0.005$. All the images are generated using high precision computation (16 bits), but are linearly mapped to 8 bits for display. (d) Tone reproduction of (c). (e) The corresponding transfer function (top: color, bottom: alpha).

3.1 Alpha Precision for Large Volume Data Visualization

In volume rendering, each sample is assigned an alpha value and alpha blending is performed during compositing on the rendered images in a back to front order according to the formula:

$$\vec{C}_N = (1 - e^{-\tau})\vec{C}_C + e^{-\tau}\vec{C}_O \quad (1)$$

$$\tau = \alpha_g \alpha_C d_x \quad (2)$$

where \vec{C}_N are the red, green, and blue color channels of the composited ($i = O$), new ($i = N$), and current ($i = C$) volume sample at

each pixel. α_C is the alpha value of the current rendering plane at the associated pixel, α_g is the global alpha. The distance between two consecutive samples along the ray is d_x . A simple emission model is applied here without extra shading effects. In this way, every sample contributes to the final rendering. On the other hand, the alpha blending exponentially decreases each sample's contribution. As an example, let us look at a volume data of n slices. In the following discussion, we assume our sampling rate is the same as the data frequency of the volume data, e.g. one sample per voxel. Assuming that every voxel is assigned with the same opacity value α , the contribution of the furthest voxel to the rendered image will be decreased by a factor of $(1 - \alpha)^n$. To guarantee the final contribution to be no less than r , the α value must be set as:

$$\alpha = 1 - e^{\log(r)/n} \quad (3)$$

For data sets with a size of 2048^3 , when r is 0.1, the alpha has to be less than 0.00112; when r is 0.01, alpha has to be smaller than 0.00225. In such cases, the rendering system with only 8 bits per channel will quantize the alpha value to zero and voxel regions with such low alpha values will not contribute to the rendered image. Furthermore, to achieve higher rendering quality, as the number of slices needs to be much higher than the size of volume (at least twice according to the Nyquist-Shannon sampling theorem), the preferred α value would even be smaller. For high dynamic range volume rendering, samples of such low α values must still contribute to the final image. This requires high precision of the alpha values. As an example, for 16-bit float representation, the highest precision is 2^{-24} . This makes it possible to render volumes of 2048^3 resolution.

In addition to the transfer function, which is discussed in the next subsection, we define a global parameter α_g for adjusting the global opacity of the whole volume. This α_g value will be multiplied with the opacity returned by the transfer function to get each sample's final opacity for compositing. As an example, the α_g value is set to 0.5, 0.05, 0.005 in Figures 2(a), (b) and (c), respectively. The opacity value in the base transfer function ranges from 0 to 0.5 (the same transfer function is used for all three images and is not plotted). All the figures are generated using high precision computation (16 bits) and hence are presented using 16 bits per channel. The dynamic ranges are $3.32 \times 10^4 : 1$, $1.19 \times 10^5 : 1$, $2.76 \times 10^4 : 1$ for Figures 2(a), (b), (c), respectively. The images are linearly mapped to 8 bits for display. Figure 2(c) is much darker compared with the other two images due to the lowest global alpha value used. Not shown in the figure, 8-bits-per-channel rendering with the same transfer function and α_g value as those used in Figure 2(c) produces a totally black image. This is because when only 8 bits precision compositing is performed, the smallest α_g value used above will lead to zero alpha values for almost all the samples. With high precision computations, we are able to obtain fine details in the rendering result from Figure 2(c) using non-linear tone mapping instead of linear mapping. The result is shown in Figure 2(d). Figure 2(e) is the transfer function being applied for rendering (top: color specification; bottom: alpha values, dark regions indicates low opacity value). Similar transfer functions are also included in subsequent figures.

3.2 Transfer Function Design for High Dynamic Range Volume Rendering

We develop a novel transfer function design interface for high dynamic range volume visualization. As discussed above, high precision alpha values are desirable. Input data may also require high precision. This introduces new issues in transfer function design. The current display resolution is insufficient to display the full precision of each axis value (intensity and color/opacity). If a low resolution axis is used, different features with close intensity values

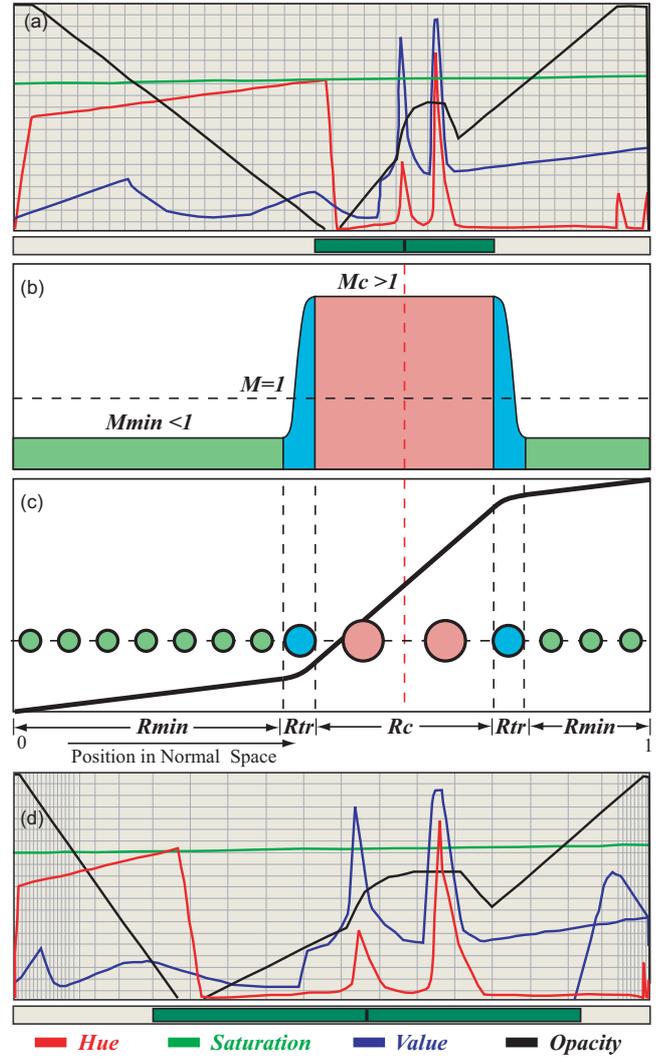


Figure 3: Focus+context transfer function design. (a) Transfer function displayed in normal scale. The green bar indicates the region of interest to be magnified. (b) Magnification curve. The region of interest has magnification value $M_c > 1$. (c) Intergration curve. (d) Transfer function displayed in non-linearly scaled intensity space.

cannot be differentiated. It is useful to be able to magnify one portion of the intensity region while still keeping other regions visible, following the general concept of focus+context.

The focus+context concept allows the simultaneous presentation of global (context) and detail (focus) information in the same display. The fish-eye view technique [7, 8] allows the user to see an object in the region of interest in detail, and other lower resolution objects peripherally. The advantage of the focus+context technique is that the contextual relationship between the focus center and other context regions is preserved, while giving the user a useful level of control over the regions at hand.

As illustrated in Figure 3, we develop a focus+context transfer function design interface which performs 1D non-linear scaling on the intensity axis. In our current implementation, we employ a one-dimensional transfer function which assigns color and opacity to the volume based on the scalar voxel values. Our system allows the user to switch between RGB space and HSV (Hue, Saturation, Value) space [28] during the color specification. Color values are interpolated between user specified knots in HSV space, and then converted to RGB space for later volume rendering. In Figure 3(a),

a transfer function is displayed in linear scale. The horizontal axis represents normalized intensity values. The green bar at the bottom indicates the intensity range to be magnified, which is specified by the user. The desired amount of magnification is also specified by the user. The goal is to non-linearly scale the intensity axis so that the full intensity range still fits in the horizontal window range. The following formalizes the problem and explains our solution.

Figure 3(b) illustrates a magnification curve. The sizes of the circles in Figure 3(c) indicate the magnification factors of the corresponding locations. Through user input, the interested intensity range R_c is magnified by a constant factor of $M_c > 1$. We assume that the rest of the intensity range R_{min} is magnified with a factor of $M_{min} < 1$, which is to be computed. The magnification factors of the transition region R_{tr} are linearly interpolated between M_c and M_{min} . R_{tr} is also pre-defined. The task is to compute the value of M_{min} . Since the entire intensity range should still fit in the horizontal window range, the following equation must be satisfied:

$$\int_{x \in R_{min}} M_{min} dx + \int_{x \in R_c} M_c dx + \int_{x \in R_{tr}} M_{tr} dx = 1 \quad (4)$$

This describes an integral of the magnification curve (illustrated in Figure 3(c)). Since the intensity range is normalized into $[0, 1]$, and M_c and M_{min} are constant, the equation can be expressed as:

$$R_{min} M_{min} + R_c M_c + \int_{x \in R_{tr}} M_{tr} dx = 1 \quad (5)$$

Hence M_{min} can be computed. Non-linear scaling of the intensity axis can then be performed, and the full intensity range is always presented in the view. Figure 3(d) shows the transfer function of Figure 3(a) in a non-linearly scaled intensity space.

As illustrated in Figures 3, we design a widget to indicate the intensity range of interest. This widget is described by a center and a width. During transfer function design, the user can click and drag the center of the widget to specify a region of interest for magnification. By clicking elsewhere, the user can change the width of the interested region to be magnified.

Another transfer function design task is on the vertical axis for alpha value. As discussed in Section 3.1, the values of alpha in the transfer function must be small enough (e.g., 2^{-24}) to visualize detailed features in a large volume data (e.g., 2048^3). One goal of our volume rendering is to be able to display all internal structures simultaneously. A lower range of alpha values is desired because large alpha values will be quickly accumulated to full opacity. Therefore, our transfer function design interface must accommodate the editing of alpha values near zero. The linearly scaled alpha axis provides limited resolution to small alpha values. To address this issue, we use a logarithmically scaled alpha and color axes [23], through which small alpha and color values are magnified.

We combine the logarithmic scaling in the vertical axis and non-linear scaling in the horizontal axis for the transfer function design. Figure 4 illustrates the same transfer function as in Figure 3(d), however with the vertical axis of the interface scaled logarithmically. This makes it more convenient to edit small alpha values. Moreover, with one portion of the intensity scale magnified, the user can fine tune the transfer function in this region. We allow the user to switch between non-linear and linear transfer function axes for convenience.

3.3 Tone Reproduction

While rendered high dynamic range images could leverage the recently developed HDR displays [27], we seek to accommodate popularly available standard display devices. In fact, even an HDR display device has fixed dynamic range. When an HDR image has higher dynamic range than that of the HDR display, tone mapping is still beneficial for display. Our system dynamically maps the

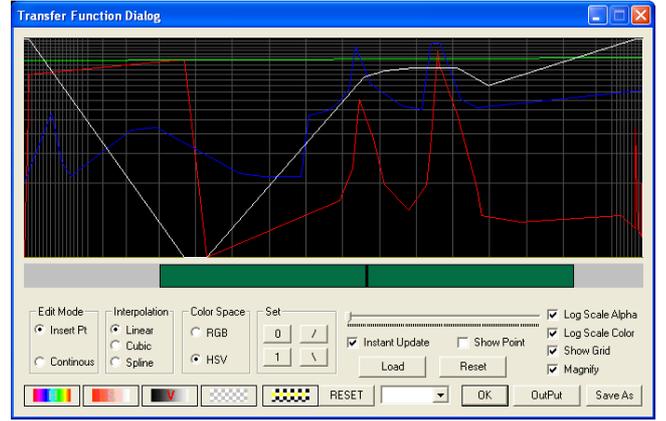


Figure 4: A snapshot of the transfer function interface in our system. The showing curves are displayed in logarithmic scale and with non-linear magnification of the intensity axis.

volume rendered HDR images to regular images of 8 bits per color channel so that they can be displayed on regular 8-bit display devices.

The goal of tone reproduction is to compress the dynamic range and at the same time maintain the contrast of the input high dynamic range image. The tone mapping algorithm we adapt belongs to the category of global operators. We choose this algorithm because it works at interactive speeds. On the other hand, using local tone mapping operators [5, 6, 25] which are relatively computationally expensive may achieve better compression results. The volume rendered images generated by our system can also be saved in HDR format images (e.g. Ward's RGBE format [31]) or in HDR movies, which can be viewed off-line using HDR image viewer [11] or HDR movie player [4] available in public domain. For such off-line viewing, other state-of-the-art local tone mapping methods can be applied. We choose the adaptive logarithmic mapping operator developed by Drago et al. [4]. This method is based on logarithmic compression of luminance values. A bias power function is introduced to adaptively vary logarithmic bases, resulting in good preservation of details and contrast. We integrate this tone mapping algorithm with our high dynamic range volume rendering.

Figure 5(a) shows an HDR image generated from our HDR visualization. In this image, the HDR intensity is pseudo-color encoded. Figure 5(b) shows the regular 8-bits-per-channel image after tone mapping. Figure 5(c) shows 9 tone mapped images with increasing exposure levels. This demonstrates the rich information encoded in HDR visualization. Even a static HDR visualization allows users to interact with it and retrieve details of selected points of interest. Figure 6 further illustrates this concept. Figure 6(a) shows the tone reproduction based on a user specified region of interest. Compared with Figure 6(b), which is based on a different user specification, it can be observed that details and contrasts are retained the best in the regions of interest.

4 IMPLEMENTATION DETAILS

Our high dynamic range volume visualization method is not limited to specific volume rendering implementations. Volume rendering with software implementation approaches can be easily adapted to the high dynamic range paradigm. Special purpose hardware for volume rendering, such as VolumePro [19], can also benefit from the high dynamic range volume by storing results in a high dynamic range format and integrating tone reproduction in the post rendering stage. All experiments have been performed on a Dell Precision 530 workstation with a single Intel Xeon 2.20 GHz CPU, 1GB RAM,

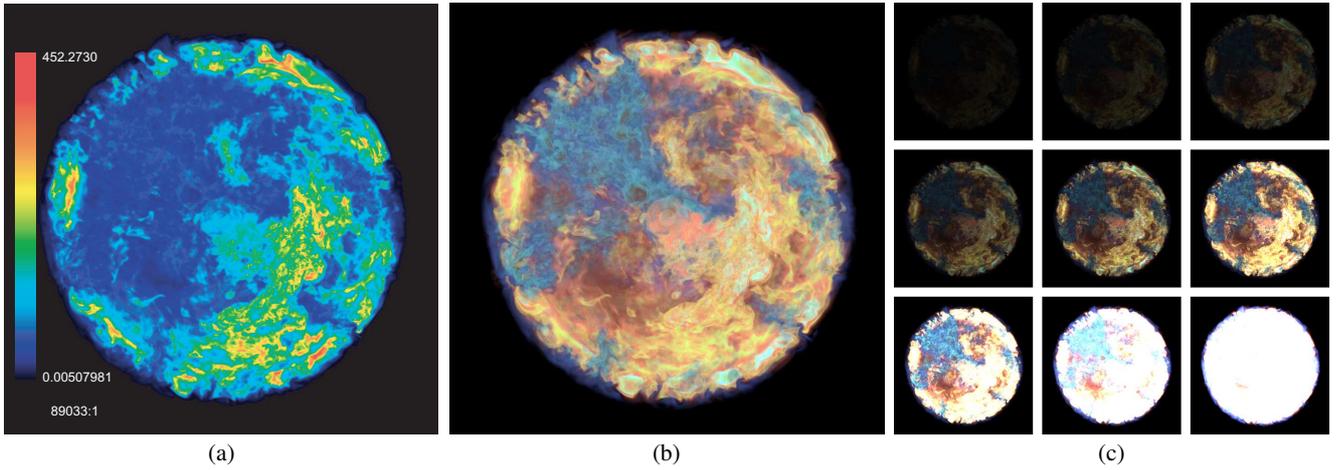


Figure 5: High dynamic range image. (a) The intensity pseudo color encoded. (b) tone mapped image. (c) Images at different exposure levels.

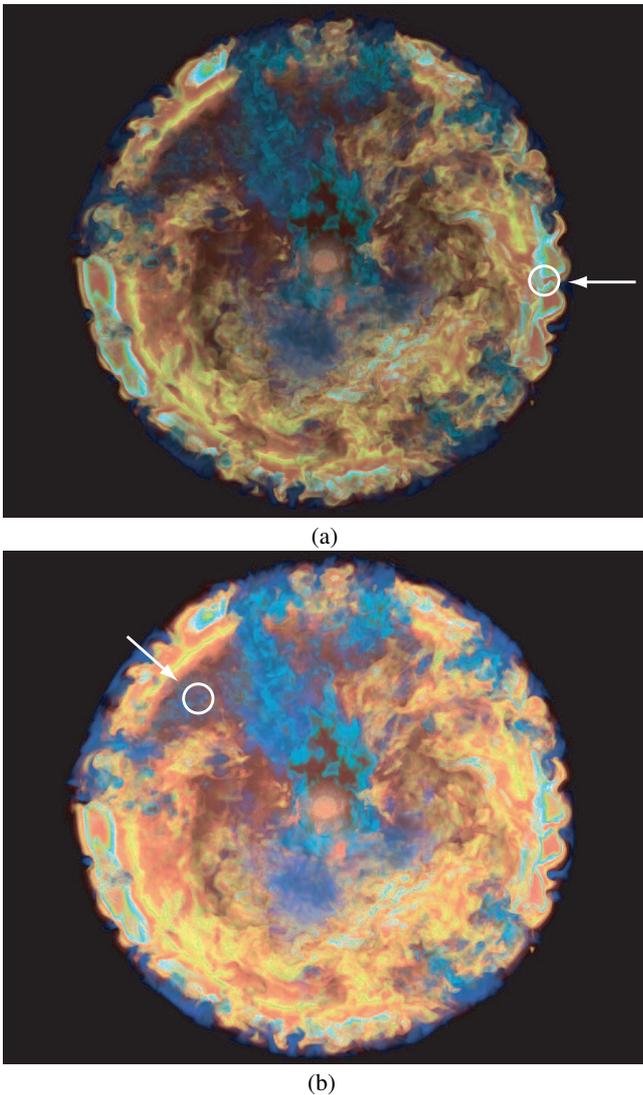


Figure 6: Adaptive logarithmic tone mapping based on center selection. (a) initial tone mapped image based on a user specified region of interest emphasizing highlight parts (white circle); (b) tone mapping based on another user specified region of interest emphasizing parts with low illumination.

and an NVidia 256MB GeForce6800 Ultra graphics card.

In our system, data sets are organized in a hierarchical brick structure. The current implementation of the rendering module is a simple out-of-core method. Two threads, one for rendering, and another for I/O tasks, are executed simultaneously. Data bricks are read from the hard drive into the main memory upon the request of the rendering thread. During the user navigation or transfer function tuning, the system renders the volume in a lower resolution (e.g. 128^3 or 256^3) to achieve interactivity. When the user finishes the interaction, the system automatically loads higher resolutions and render higher quality images.

We leverage modern graphics hardware to achieve high performance HDR visualization. We explain and analyze how to achieve maximum precision within the constraints of current graphics hardware. For conventional computation, we could use vendor extensions such as NVidia extension `GL_NV_float_buffer` and support a float color buffer up to 32 bits per channel. However, for volume rendering, it is important to have alpha blending functionality. For this feature, currently only 16-bit float buffer is supported natively at the hardware level. At 24 bits or higher, the performance is reduced dramatically. Therefore, we use 16-bit float buffers in our volume rendering. As discussed earlier, for 16-bit float representation of alpha, the highest precision is 2^{-24} . This precision can be used to render a volume data of 2048^3 in size while reasonably preserving the details of the entire volume.

We employ a texture mapping-based hardware implementation. Volume rendering is implemented using hardware accelerated 3D texture rendering [1, 26]. Parallel polygon slices perpendicular to the viewing direction are generated and texture mapped with the 3D volume and are composited together in back-to-front order. When rendering a volume of 2048^3 or larger in hardware, an immediate challenge is to cope with the limited graphics card memory. Naturally, we have to develop an out-of-core method so that partial volume data can be efficiently loaded and unloaded to the graphics card memory during the rendering. We also seek time-critical visualization, i.e., we wish to provide instant feedback for user interaction. To achieve these goals, we have developed a multi-resolution volume data structure [14] and level-of-detail (LOD) volume rendering [32] on hardware. The multi-resolution volume data structure is constructed by continuously down-sampling the higher level volume starting from the original volume (as the highest level). For any level volume having size larger than 64^3 , it is subdivided into blocks of 64^3 . When the user interacts with the volume data, volume at a certain level is selected and rendered to keep up with the user interaction. The higher level volume is rendered only when the system enters the idle mode, i.e. no change in rendering pa-

rameters is detected. Specifically, for a volume with sub-blocks, its blocks are sorted to the camera position, then a slice order volume rendering is performed with alpha blending. Through this rendering mechanism, we can quickly navigate the volume data and change the transfer function to achieve the desired visualization. We encode transfer functions into textures. For current commodity graphics hardware, the texture size is limited by 4096 in each dimension. To implement transfer functions beyond this size, we encode 1D transfer function into multiple rows of a 2D texture.

5 RESULTS AND DISCUSSIONS

We have applied our HDR visualization technique to visualize three large fluid dynamics simulations. One simulation data, depicted in Figures 1, 2, 5, and 6, shows temperature fluctuations in the deep convection zone of a red giant star [22]. Large amplitude positive and negative fluctuations are seen near the surface of the star, where they are driven by a nearly constant convective energy flux going through very low density regions. Large negative temperature fluctuations are seen in the large, turbulent, and cool down flowing plume on one side of the star. The warm updraft on the other side of the star is relatively free of turbulence. The temperature field in such a warm updraft is characterized by extremely small amplitude, and small scale, fluctuations around a slightly positive and smoothly varying background temperature. In order to see both large and small amplitude temperature fluctuations in a single visualization, a highly non-linear mapping of temperature to color and intensity is of great importance. Our HDR volume rendering and dynamic tone mapping have made this possible.

The second simulation data shows turbulent mixing of fluids that typically produces a wide range of concentrations. In a turbulent layer between two pure fluids, some regions will be a substantial blend of both, other regions might be completely unmixed (100% one fluid or the other), while still others might have extremely small fractions of either fluid mixed into the other. The precise values of even minuscule amounts of mixing can be very important. Examples include the dilution of toxins in a blood stream, pollution into the Earth's atmosphere, flows with chemical combustion, and nuclear reaction chains in the flame zones of many stars. Here in Figure 7 we show result from a two-fluid PPM simulation where turbulent mixing is driven by shear ($256 \times 512 \times 1024$). Initially, a region of pure air is right next to, and is in relative motion to, a region of pure Sulfur Hexafluoride (SF_6). The interface between these two gasses starts as a plane, and the relative motion corresponds to the two blocks of gas sliding past each other: the initial contact discontinuity is also a slip surface. In addition to the large relative motion there are also small velocity perturbations, which grow with time due to the Kelvin-Helmholtz instability, thereby generating a turbulent mixing layer. The fractional volume of the heavier gas, SF_6 , is followed as a dynamical variable in each computational cell. Fractional volumes range from 0 (pure air) to 1 (pure SF_6). As these visualizations show, very small amounts of the heavier gas can be mixed into the air via turbulent churning of the gas. Similarly, very small amount of air can be mixed into the SF_6 . Hence, if the user wishes to follow these very diluted mixtures, high dynamic range is needed for the fractional volume near both zero and unity.

The third simulation data, depicted in Figure 8, shows magnitude of vorticity from a high-resolution (2048^3) simulation of homogeneous decaying compressible fluid turbulence [20, 21] when the turbulence is fully developed. Figure 8(a) shows a snapshot in time when the turbulence is in the process of developing. Figure 8(b) is rendered from the same simulation after the turbulence is fully developed. Slip surfaces and intense vortex tubes are clearly visible in the images. The goal of this study is to improve our understanding of inertial range turbulence, with applications in astrophysical flows, such as stellar convection, and the development and

testing of subgrid-scale models of turbulence. Since no general theory of fluid turbulence is currently available, volume visualization is important for developing new insights into these flows. Of particular interest is the search for coherent three-dimensional structures, which control the dynamics of the flow. Three-dimensional fluid turbulence naturally produces slip surfaces and intense vortex tubes, which can be seen in the vorticity field here. In systems where a wide range of spatial scales available, such as in this 2048^3 run, values of local vorticity span a wide dynamic range. Since the three-dimensional structure of both weak and strong vorticity is of interest, HDR visualization techniques that can handle such a wide dynamic range of values are very effective for examining these fields. As is evident in these images, one clear advantage for HDR volume visualization is the rich information retained in one single visualization. A wide range of details can be dynamically viewed on a regular low range display. This can be effectively experienced by playing the HDR movies and tuning the tone mapping parameters. Here, even pre-generated animations offer interactive data exploration opportunities. (Videos clips of high resolution HDR volume rendering movies are available at: <http://www.dtc.umn.edu/~xyuan/vis05hdr/>)

As aforementioned in Section 4, we render a volume in a down-sampled resolution during navigation for interactive performance. Starting from the highest resolution (level 1), we downsample the volume by a factor of two for each lower level. In our system, the resolution level corresponding to 128^3 is normally selected for interactive navigation. This way, a framerate of over 30 fps can always be achieved during navigation. When the user stops navigation, volume with higher resolution is progressively loaded and rendered to achieve higher rendering quality; this continues until the highest quality is achieved, i.e., the full resolution volume is rendered. Whenever the user restarts the navigation, the system automatically lowers the rendering resolution to sustain an interactive rendering rate. For a high-resolution (2048^3) simulation of homogeneous decaying compressible fluid turbulence data set (Figure 8), over 15 seconds is required to achieve the highest rendering quality with the window size of 1024^2 . For a smaller data set (red giant star, 512^3), less than 0.5 seconds is required for the full rendering. Note that the rendering time also depends on the number of slices applied per voxel distance. A sampling rate of two slices per voxel distance is used in our experiments.

Without high dynamic range volume rendering, to perform above rendering on commodity graphics hardware, a mapping must be conducted which takes the original scalar values to integers in the range [0-255]. Linear mapping is the simplest approach. A log scaling is effective for positive definite quantities with a large dynamic range, such as mass density in stellar convection. Hyperbolic tangent mappings work well for quantities distributed with wide tails, such as vorticity components. However, such mappings require prior knowledge of the volume data set and substantial effort in the preprocessing. In addition, the loss of information is inevitable during the quantization for such mapping. Our HDR volume rendering provides a direct and flexible way of exploiting full information for volume data sets.

6 CONCLUDING REMARKS

In this paper, we have presented a high dynamic range volume visualization method for rendering volume data with both high spatial and intensity resolutions. Our method performs high precision volume rendering followed by dynamic tone mapping to preserve details on regular display devices. With interactive input from the user, our system automatically adjusts the tone reproduction for the final display to enhance selected features. In this work, we also present a novel transfer function design interface with non-linear scaling of intensity range and logarithmic scaling of color/opacity

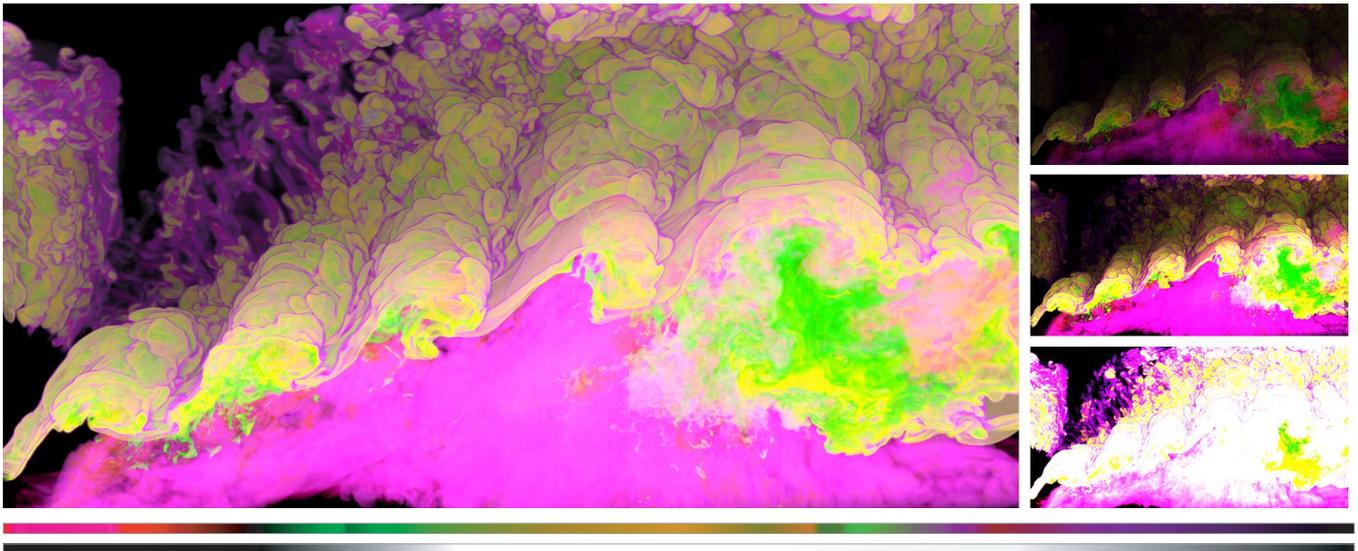


Figure 7: High dynamic range volume rendering result for turbulent mixing of air and Sulfur Hexafluoride(SF_6). Left: tone mapped image from high dynamic range volume rendering; right: images at different exposure levels from the same rendering. The color bars below the rendered images are the corresponding transfer function (top: color, bottom: alpha)

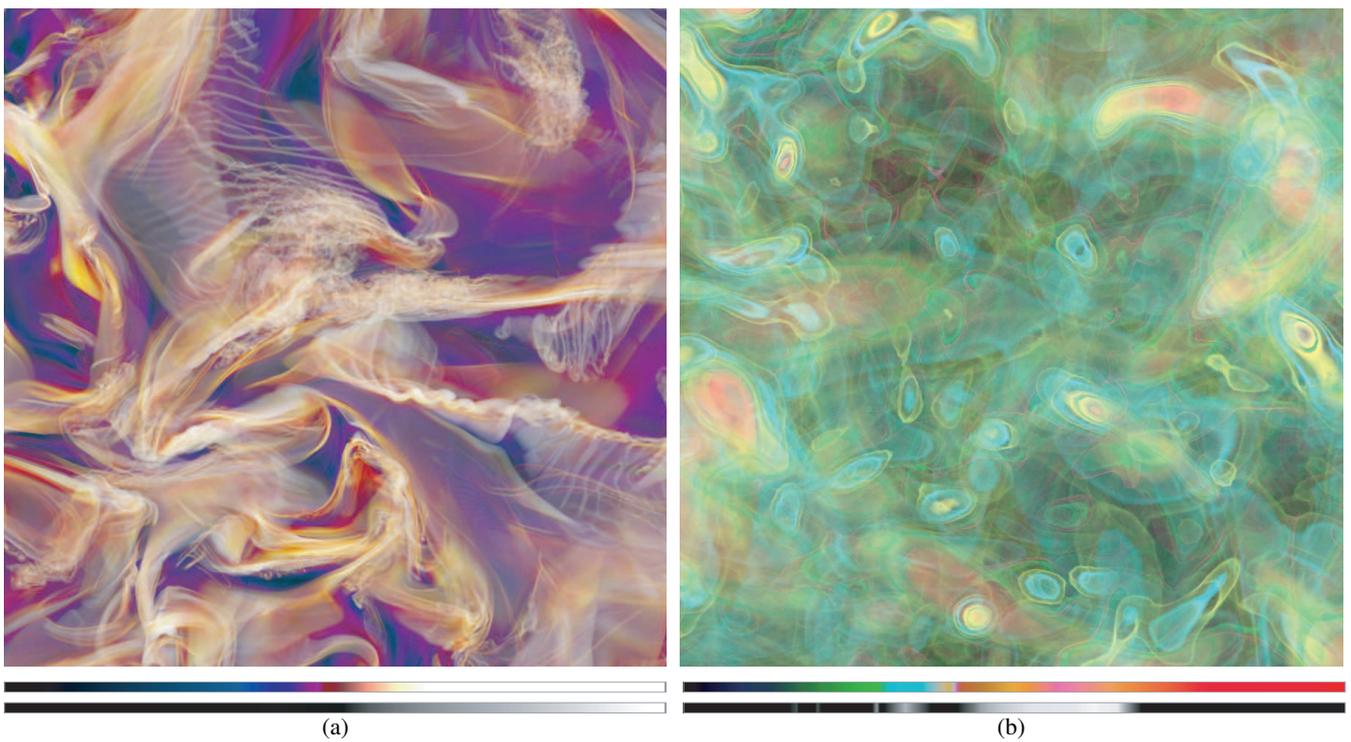


Figure 8: Visualization results by the high dynamic range volume rendering system. Images depict magnitude of vorticity from a high-resolution simulation of homogenous decaying compressible fluid turbulence. (a) is at a time when the turbulence is in the process of developing. (b) is at a time when the turbulence is fully developed. The color bars below each rendered image are the corresponding transfer function (top: color, bottom: alpha)

range to facilitate HDR volume visualization. By leveraging modern commodity graphics hardware and existing out-of-core acceleration, our system can produce interactive visualization of huge volume data. We have demonstrated the suitability of our system in terms of visualizing large simulation data that has wide range of physical properties.

We plan to enhance our transfer function design interface by incorporating some advanced techniques recently developed. For example, we intend to incorporate high dimensional transfer function design [13]. In such cases, techniques of 2D or higher dimensional magnification such as fisheye [24], hyperbolic space [15, 17] are useful. It is also possible to utilize the information of the volume data and analyze topology, then apply semantic magnification. Since most current tone mapping techniques are developed for photo scenes, we are developing new tone mapping operators suitable for our visualization purposes.

7 ACKNOWLEDGMENTS

Support for this work includes University of Minnesota Computer Science Department Start-up funds, University of Minnesota Digital Technology Center Seed Grants 2002-4, NSF ACI-0238486 (CAREER), NSF EIA-0324864 (ITR), and the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAD19-01-2-0014. Its content does not necessarily reflect the position or the policy of this agency, and no official endorsement should be inferred. The last author would like to acknowledge the following supports: DoE grant Nos. DE-FG02-87ER25035 and DE-FG02-94ER25207, NSF PACI at NCSA grant, CDA-950297, and UNISYS Corp hardware gift. We thank Nathan Gossett and Amit Shesh for proofreading of the paper and the anonymous reviewers for helpful suggestions.

REFERENCES

- [1] Brian Cabral, Nancy Cam, and Jim Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *VolVis '94*, pages 91–98. ACM Press, 1994.
- [2] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH '97*, pages 369–378. ACM Press, 1997.
- [3] Kate Devlin, Alan Chalmers, Alexander Wilkie, and Werner Purgathofer. STAR: tone reproduction and physically based spectral rendering. In *Eurographics 2002*, pages 101–123, 2002.
- [4] Frederic Drago, Karol Myszkowski, Thomas Annen, and Norishige Chiba. Adaptive logarithmic mapping for displaying high contrast scenes. *Computer Graphics Forum*, 22(3):419–419, 2003.
- [5] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. In *SIGGRAPH '02*, pages 257–266. ACM Press, 2002.
- [6] Raanan Fattal, Dani Lischinski, and Michael Werman. Gradient domain high dynamic range compression. In *SIGGRAPH '02*, pages 249–256. ACM Press, 2002.
- [7] George W. Furnas. Generalized fisheye views. In *CHI '86*, pages 16–23. ACM Press, 1986.
- [8] George W. Furnas. The FISHEYE view: a new look at structured files. *Readings in information visualization: using vision to think*, pages 312–330, 1999.
- [9] Abhijeet Ghosh, Matthew Trentacoste, and Wolfgang Heidrich. Volume rendering for high dynamic range displays. In *EG/IEEE VGTC Workshop on Volume Graphics 2005*, pages 91–98, 2005.
- [10] Nolan Goodnight, Rui Wang, Cliff Woolley, and Greg Humphreys. Interactive time-dependent tone mapping using programmable graphics hardware. In *EGSR '03: Proceedings of the 14th Eurographics Symposium on Rendering*, pages 26–37, 2003.
- [11] HDRshop. <http://www.ict.usc.edu/graphics/HDRShop/>.
- [12] Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High dynamic range video. *ACM Trans. Graph.*, 22(3):319–325, 2003.
- [13] Joe Kniss, Gordon Kindlmann, and Charles Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Trans. Visual. Comput. Graphics*, 8(3):270–285, 2002.
- [14] Eric LaMar, Bernd Hamann, and Kenneth I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *Proceedings of IEEE Visualization '99*, pages 355–361, 1999.
- [15] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95*, pages 401–408. ACM Press, 1995.
- [16] Gregory Ward Larson, Holly Rushmeier, and Christine Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Trans. Visual. Comput. Graphics*, 3(4):291–306, 1997.
- [17] Tamara Munzner. H3: laying out large directed graphs in 3D hyperbolic space. In *INFOVIS '97*, page 2, 1997.
- [18] Sumanta N. Pattanaik, James A. Ferwerda, Mark D. Fairchild, and Donald P. Greenberg. A multiscale model of adaptation and spatial vision for realistic image display. In *SIGGRAPH '98*, pages 287–298. ACM Press, 1998.
- [19] Hanspeter Pfister, Jan Hardenbergh, Jim Knittel, Hugh Lauer, and Larry Seiler. The volumepro real-time ray-casting system. In *SIGGRAPH '99*, pages 251–260. ACM Press, 1999.
- [20] David Porter, Annick Pouquet, Igor Sytine, and Paul Woodward. Turbulence in compressible flows. *Physica A*, pages 263–270., 1999.
- [21] David Porter, Annick Pouquet, and Paul Woodward. Measures of intermittency in driven supersonic flows. *Physical Review E*, 66, 2002.
- [22] David Porter and Paul Woodward. 3-d simulations of turbulent compressible convection. *The Astrophysical Supplement Series*, 2000.
- [23] Simeon Potts and Torsten Möller. Transfer functions on a logarithmic scale for volume rendering. In *GI '04: Proceedings of the 2004 conference on Graphics interface*, pages 57–63, 2004.
- [24] Uwe Rauschenbach. The rectangular fish eye view as an efficient method for the transmission and display of large images. In *Proceedings of IEEE ICIP '99*, pages 115–119, 1999.
- [25] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. In *SIGGRAPH '02*, pages 267–276. ACM Press, 2002.
- [26] C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume on standard PC graphics hardware using multi-textures and multi-stage rasterization. In *HWWS '00: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware*, pages 109–118. ACM Press, 2000.
- [27] Helge Seetzen, Wolfgang Heidrich, Wolfgang Stuerzlinger, Greg Ward, Lorne Whitehead, Matthew Trentacoste, Abhijeet Ghosh, and Andrejs Vorozcovs. High dynamic range display systems. *ACM Trans. Graph.*, 23(3):760–768, 2004.
- [28] Alvy Ray Smith. Color gamut transform pairs. In *SIGGRAPH '78*, pages 12–19. ACM Press, 1978.
- [29] Jack Tumblin and Holly Rushmeier. Tone reproduction for realistic images. *IEEE Comput. Graph. Appl.*, 13(6):42–48, 1993.
- [30] Jack Tumblin and Greg Turk. LCIS: a boundary hierarchy for detail-preserving contrast reduction. In *SIGGRAPH '99*, pages 83–90. ACM Press, 1999.
- [31] Greg Ward. Real pixels. In James Arvo, editor, *Graphics Gems II*, pages 80–83. Academic Press, 1991.
- [32] Manfred Weiler, Rüdiger Westermann, Chuck Hansen, Kurt Zimmermann, and Thomas Ertl. Level-of-detail volume rendering via 3D textures. In *VolVis '00*, pages 7–13. ACM Press, 2000.
- [33] P. R. Woodward. Numerical methods for astrophysicists. *Astrophysical Radiation Hydrodynamics*, 54:245–326, 1986.
- [34] P. R. Woodward, S. E. Anderson, D. H. Porter, and A. Iyer. Distributed computing in the SHMOD framework on the NSF teragrid. Technical report, LCSE, UMN, Feb 2004.
- [35] P. R. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54:115–173, 1984.