DOI: 10.1111/j.1467-8659.2009.01690.x

Eurographics/ IEEE-VGTC Symposium on Visualization 2010
G. Melançon, T. Munzner, and D. Weiskopf
(Guest Editors)

Volume 29 (2010), Number 3

An Exploratory Technique for Coherent Visualization of Time-varying Volume Data

A. Tikhonova[†1] and C. D. Correa[‡1] and K.-L. Ma[§1]

[1]University of California, Davis, USA

Abstract

The selection of an appropriate global transfer function is essential for visualizing time-varying simulation data. This is especially challenging when the global data range is not known in advance, as is often the case in remote and in-situ visualization settings. Since the data range may vary dramatically as the simulation progresses, volume rendering using local transfer functions may not be coherent for all time steps. We present an exploratory technique that enables coherent classification of time-varying volume data. Unlike previous approaches, which require pre-processing of all time steps, our approach lets the user explore the transfer function space without accessing the original 3D data. This is useful for interactive visualization, and absolutely essential for in-situ visualization, where the entire simulation data range is not known in advance. Our approach generates a compact representation of each time step at rendering time in the form of ray attenuation functions, which are used for subsequent operations on the opacity and color mappings. The presented approach offers interactive exploration of time-varying simulation data that alleviates the cost associated with reloading and caching large data sets.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

One of the main challenges in applying direct volume rendering to time-varying data is the selection of an appropriate global transfer function. One of the desired properties of such a transfer function is *coherence* or consistency. A transfer function is coherent when the same ranges in a data set are assigned the same colors throughout the entire sequence. Coherent visualization of time-varying data is crucial for ensuring correct interpretation of rendered images. As a sequence progresses, however, the data range may vary dramatically between time steps. Thus, renderings generated with local transfer functions may be colored inconsistently. To avoid this problem, it is necessary to know the global data range a priori, or compute it by traversing the sequence in advance. Then, all time steps can be visualized coherently with the same global transfer function. For large time-varying data sets, it may be impractical to pre-process all time steps in advance. Moreover, in remote and in situ visualization scenarios, this is not a plausible solution.

We present a technique for exploring the transfer function space of time-varying volume data in a coherent manner. Using this technique, time-varying volume data can be processed in a single pass. As we render each time step, using a local transfer function, we generate a compact representation of a volume, that allows us to later explore different opacity and color mappings without accessing the original 3D data. This representation is much smaller than the original volume data and can fit into system memory. After the time steps have been loaded and rendered, the user can assign a global transfer function and see it applied coherently to the entire sequence. Consider Fig. 1, where we visualize an argon bubble-shockwave interaction simulation. If we apply a local transfer function to each time step, the results are incoherent. Values with the same hue in an earlier time step do not correspond to values with the same hue in later time steps. For example, we do not see the expected dissipation of gas. In contrast, the global transfer function, as shown in

[†] tikhonov@cs.ucdavis.edu
[‡] correac@cs.ucdavis.edu
[§] ma@cs.ucdavis.edu

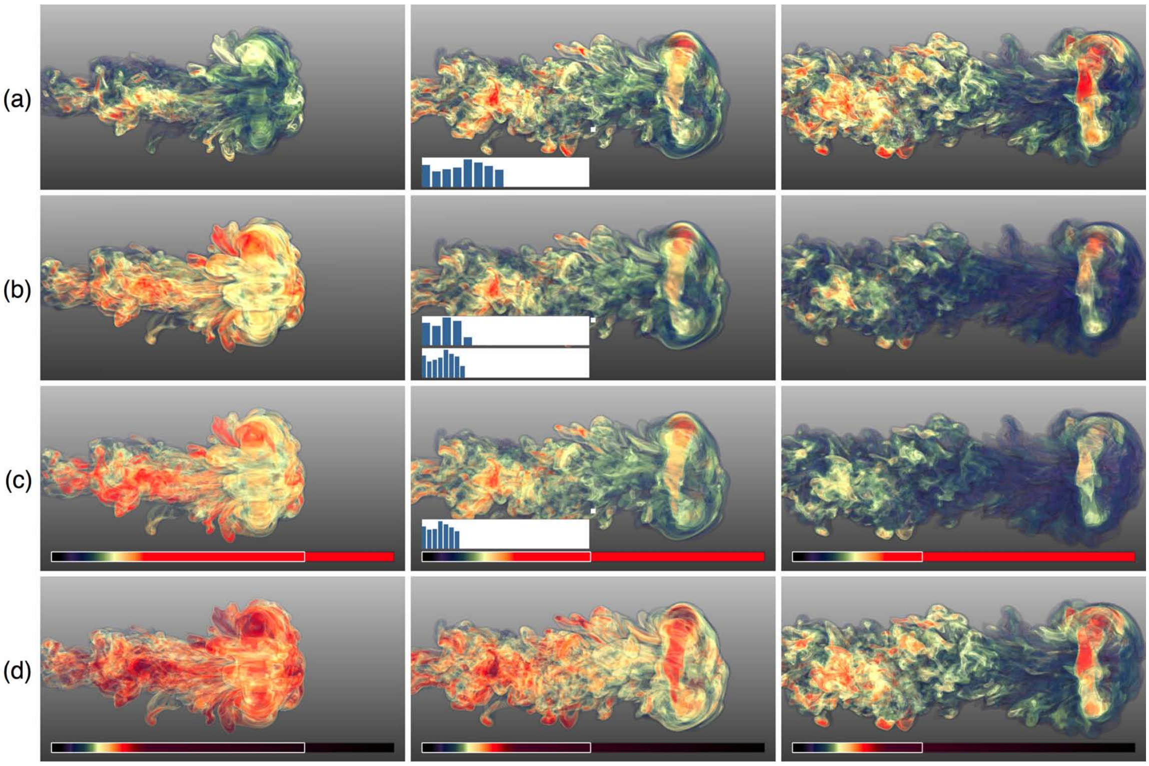


Figure 1: *Coherent Visualization of an argon bubble-shockwave interaction. (a) Three time steps rendered using a local transfer function. Although a local transfer function enhances different structures, they are not coherent, as values depicted in red do not necessarily correspond to the same values in all three time steps. This is misleading to the scientist. (b) The global transfer function provides a coherent visualization, as seen in the apparent dissipation of the gas bubble. Obtaining such result requires another complete traversal of the full simulation sequence. (c) With our approach, we remap the results in (a) to the global transfer function without the need to access the volume data. Compare to (b) for accuracy. (d) In addition, the scientist might adjust the global color map to bring out the ring details in the third time step. Other time steps can be re-adjusted with little cost.* Ray Attenuation Functions : *For the time step in second column, we show a 16-bin ray attenuation function for a given pixel as a logarithmic bar chart. In (b), we see that the attenuation function is a rescaled and biased version of the function in (a). This transformation is more clear when we see the attenuation function discretized in 32 bins. (c) We exploit this behavior to remap the attenuation function from (a) to the global range. Compare this histogram to the actual 32-bin function in (b).*

Fig. 1(b), shows a coherent visualization. Now, hues correspond to the same values throughout the simulation. Obtaining these results, however, requires reloading each time step. In Fig. 1(c), we obtain a coherent visualization using our approach, based on the information gathered locally.

Our technique enables the user to achieve a globally coherent visualization of a time-varying data set after rendering each time step independently using a local transfer function. The user may change the color and opacity mappings to explore the data or to highlight features of interest. None of these operations requires access to the original volume data. Since we can inexpensively apply the changes to all time steps, changes to the color and opacity mappings can be visualized coherently throughout the entire sequence at interactive rates. For example, in Fig. 1(d) we modify the global transfer function to highlight the ring of argon gas. The changes are coherently applied to earlier time steps without accessing the original 3D data.

To be practical for interactive exploration, a volume representation should be as small as possible. One option is to simply cache a volume rendered image for each time step. Ignoring compression, a single image constitutes the smallest unit of information that can be stored. However, such images do not provide the ability to change the color or opacity of features to achieve coherent visualization. Instead, we show that volume rendered images can be decomposed into a linear combination of colors, weighted by the total attenuation of each intensity value. This accumulated attenuation shows the distribution of attenuation along a ray with respect to intensity values, as seen in the bar chart inset in Fig. 1(middle column). In this paper, we show that coherent visualization and transfer function operations such as recoloring or opacity tuning can be achieved as a result of operations on these functions. The accuracy of reconstruction using these functions naturally depends on their resolution. We show that 16-value functions (equivalent to storing four RGBA images) are compact enough for interactive exploration and provide accurate results for complex data sets. Through a visual and quantitative evaluation, we show that our approach is an efficient exploratory technique for visualizing time-varying volume data.

2. Related Work

Time-varying Volume Data. Despite the advances in rendering static volume data sets, the visualization of time-varying volume data remains a difficult challenge. Several strategies have been developed to deal with I/O bottlenecks, including compression [GS01, LMC01, BCF03], multiresolution analysis and partitioning structures [SCM99], and differential encoding [SJ94]. Gao et al. speed up the rendering of time steps using visibility culling and temporal coherence [GSHK04]. Defining coherent transfer functions for time-varying volume data is still a challenge, despite the advances made for time-invariant volume data [PLB*01], which include the spatial derivatives of the data [KD98], the contour spectrum [BPS97], multidimensional spaces [KKH01], and stochastic search in image space [HHKP96, MAB*97]. On one hand, transfer functions must include the temporal behavior of the data, which may be periodic, regular or random. On the other hand, transfer functions must be temporally coherent. Otherwise, colors may be misleading or physically meaningless. These problems are described in detail by Jankun-Kelly and Ma [JKM01]. To address these problems, Kosara et al. suggest the time histogram as a 2D or 3D space to classify time-varying data [KBH04]. Akiba et al. [AFM06] extends a similar space with equivalence classes to handle multiple time steps simultaneously. Younesy et al. proposed the Differential Time-Histogram Table [YMC05], which exploits temporal consistency to extend the time histogram with differential encoding. Wang et al. [WYM08] enhance time-histograms with importance curves that highlight regions with different temporal trends. Woodring et al. [WWS03] consider the time-varying data as a four-dimensional field. Hyperplanes in this space highlight different temporal structures in the data. Park et al. use multidimensional transfer functions to highlight different properties of time-varying flow data [PBL*04]. Other approaches include temporal clustering and sequencing [WS09], and machine learning [TM05]. In all these cases, however, the volume data must be preprocessed in its entirety before classification. In our approach, we provide a technique that does not require the knowledge of the global data range a priori, which is essential for remote and in-situ visualization scenarios.

View-dependent Visualization. Although visualization involves rotation and zooming, operations such as transfer function design are usually performed while the view is fixed. Several exploratory techniques exploit this fact: opacity [RSK06] and feature peeling [MMG07] compute different layers of features from a given viewpoint, depending on where the attenuation reaches a certain value. Visibility histograms [CM09] compute visibility distribution of all image samples to guide the generation of good transfer functions. Other approaches use view-dependent images to cache results [LP03] or composite new views of volume rendered data [RPSH08, WQ07]. Ma et al. propose to cache samples along a ray for repeated use in the transfer function exploration process [MCP91]. To reduce storage complexity, view-dependent compact representations were proposed for volume data [SCM03] and unstructured grids [SLSM06]. These methods, however, cache individual, potentially visible samples. In this paper, we present a view-dependent technique that computes per-ray attenuation functions to allow manipulation of transfer function parameters for time-varying volume data.

3. Ray Attenuation Functions (RAF)

Our coherent visualization technique is based on the computation of per-ray attenuation functions. These functions summarize the attenuation due to each intensity value in a data range. Our key contribution is the ability to cache the accumulated attenuation for a number of data ranges to approximate the volume rendering integral. Therefore, transfer function operations can be approximated by manipulating these functions. In practice, we store these functions using only a small number of bins (16), useful for interactive exploration and visual steering of time-varying volume data. According to the volume rendering integral [Max95], the color that results from compositing volume data is:

$$C = \int_0^D C(t)\tau(t)e^{-\int_0^t \tau(s)ds}dt, \tag{1}$$

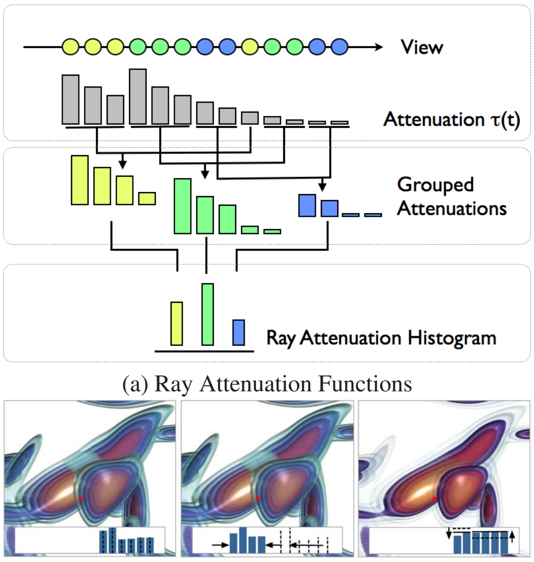
where $C(t)$ is the radiance or color and $\tau(t)$ is the attenuation of a sample t along the view direction. When discretized, the equation becomes:

$$C = \sum_i^M C(i)\alpha(i)\prod_{j=0}^{i}(1-\alpha(j)), \tag{2}$$

where $\alpha(i)$ is the opacity of a given sample along the view direction, and $\prod_{j=0}^{i}(1-\alpha(j))$ is the attenuation due to all sample points in front of sample i, and M is the number of samples along a ray.

It is common to assign similar colors and opacities to values in a particular interval of intensity. Therefore, we can group the intensity values that have the same color, $C(i)$, and opacity, $\alpha(i)$, into discrete bins. Then, the above equation can be approximated as follows:

$$C \approx \sum_{k=1}^{N} C(k) \sum_{\{i|i=0,1,\ldots,M\}\,\mathrm{AND}\,bin(i)=k} \alpha(i)\prod_{j=0}^{i}(1-\alpha(j)),$$



(a) Ray Attenuation Functions

(b) RAF Properties

Figure 2: *(a) Ray Attenuation Functions (RAF) are obtained by adding the attenuation of intensity values grouped into a finite set of bins. (b) RAF Properties. Left: The RAF for a given pixel is visualized as a logarithmic bar chart. Middle: when we simulate a data range change that doubles in size (and use an equivalent transfer function), the resulting RAF is a scaled and biased version of the one on the left. Right: when we decrease the opacity of the two outermost features, the resulting RAF is a modulated version of the original on the left, where some bins decrease in total attenuation (the ones that decrease in opacity), while others increase.*

where $bin(i)$ is a function that assigns an intensity value of a sample i to a bin k and N is the number of resulting bins. Note that this approximation depends on the way the colors are grouped. In the worst case, each bin contains its own sample. In practice, however, the colors can be grouped into a small number of bins. The inner sum describes the distribution of attenuation along a ray with respect to an intensity value. We call it a *Ray Attenuation Function* (RAF). For a given bin k, this function is defined as follows:

$$\mathsf{F}(k) = \sum_{\{i|i=0,1,\ldots,M\}\,\text{AND}\,bin(i)=k} \alpha(i)\prod_{j=0}^{i}(1-\alpha(j)). \quad (3)$$

Therefore, we can approximate the volume integral for N number of bins, discretizing the intensity data range:

$$C \approx \sum_{k=1}^{N} C(k)\mathsf{F}(k), \quad (4)$$

where $C(k)$ is the color of a bin k (from a transfer function) and $\mathsf{F}(k)$ is the corresponding ray attenuation function. The notion of the RAF is depicted in Fig. 2. Similar functions have been proposed for a number of purposes, such as the visual probability histogram by Bordoloi and Shen [BS05] and the visibility histogram by Correa and Ma [CM09]. Since our purpose is to store attenuation distribution for each ray, the notion of visibility does not fit our definition.

3.1. RAF Properties

After examining the RAF for several transfer function operations, we identified two key properties, as shown in Fig 2(b): (1) *Scale and bias*: Fig. 2(b-left) shows the RAF for a single pixel, using a logarithmic scale for the y axis. After simulating a mapping that doubles the data range, we observe a corresponding scale and bias of the RAF, as seen in Fig. 2(b-middle). This suggests that as the data range is scaled and biased, we can estimate the remapped RAF as a re-scaled and biased version of the original RAF. (2) *Modulation*: Fig. 2(b-right) shows the RAF after decreasing opacity of the two outermost features. The new RAF appears modulated; some bins contribute less to attenuation (the ones that decreased in opacity), while others contribute more. For the intensity values that did not change opacity, the modulation is a factor reflecting the new distribution of attenuation. After observing these properties, we can use operations on the RAF to perform remapping, recoloring, and opacity modulation. A more complex example is shown in Fig. 1. In Fig. 1(b), the global attenuation function appears similar to a scaled and biased version of the one in Fig. 1(a). This is more evident when we increase the number of bins (lower bar plot in Fig. 1(b)). Therefore, one can obtain a coherent view of the same time step by scaling and biasing the attenuation function in Fig. 1(a). The result, shown in the bar plot in Fig. 1(c) appears similar to the global attenuation function. As a result, we can approximate the result of the global transfer function (Fig. 1(b)) using the information gathered using a local transfer function, as shown in Fig. 1(c).

3.2. RAF Operations

The compositing equation based on the RAF has a number of advantages for image-space operations. Although it is an approximation, since the number of bins is limited and usually small in comparison to volume depth, perceptually acceptable results can be obtained. The main advantage of this equation is the ability to decouple color and attenuation. On the other hand, opacity and attenuation cannot be decoupled entirely. However, this representation offers a number of operations useful for visual steering and exploration:

Re-coloring. This operation computes a new color after applying a different color mapping to the intensity values. Since color and attenuation are decoupled, it is clear that the result is just the composition of a RAF with new colors:

$$C_{coloring} = \sum_{k=1}^{N} C'(k)\mathsf{F}(k). \quad (5)$$

This operation is trivial if the original volume data is available. If only the resulting volume-rendered image and not the data itself is available, re-coloring of individual data value ranges is not possible. Thus, the RAF provides the same functionality that can be attained if a full 3D volume is available, but with low computational and storage costs that result from using only a small number of 2D images.

Data range remapping. Data range remapping is performed through scaling and biasing the RAF, as observed in our experiments and shown in Fig. 2(b). These operations do not modify the opacities associated with data value ranges. Given a scalar value x in the range $(0,1)$, the corresponding intensity in the intensity range (a,b) is found through a linear mapping and the remapped result at a given pixel is:

$$C_{remap} = \sum_{k}^{N} C(a+k(b-a))\mathsf{F}(k). \tag{6}$$

Opacity Modulation. Opacity modulation computes a new color after a change to the opacity mapping function. Since attenuation and opacity are not decoupled, this cannot be done accurately. However, we show that, under certain common assumptions, it can be reduced to an expression in terms of the RAF in the original image. First, as it is common with transfer function editors, opacity changes often occur for a data interval instead of individual intensity values. In our approach, since we do not rely on the 3D volume but on the compact RAF, we allow opacity changes of entire bins. Let us consider an opacity operation where we change the opacity of all samples that lie in function bin k from α_k to α'_k. As with many natural phenomena, volume data often exhibits spatial consistency, and intensity values do not appear isolated, but rather grouped in regions. When performing volume rendering, this means that samples that lie in bin k form a slab, which we assume has a thickness of K samples. We can re-write the RAF for a bin k as:

$$\begin{aligned}
\mathsf{F}(k) &= \alpha_k \sum_{i=0}^{K} \prod_{j=0}^{i} (1-\alpha_k) T(k) \\
&= \alpha_k \sum_{i=0}^{K} (1-\alpha_k)^i T(k) \\
&= (\alpha_k + 1 - (1-\alpha_k)^{K+1}) T(k),
\end{aligned}$$

where $T(k)$ is the attenuation of opacity of samples that lie in front of the slab. After a change in opacity for that bin, the new attenuation function is:

$$\begin{aligned}
\mathsf{F}'(k) &= (\alpha'_k + 1 - (1-\alpha'_k)^{K+1}) T(k) \\
&= \frac{(\alpha'_k + 1 - (1-\alpha'_k)^{K+1})}{(\alpha_k + 1 - (1-\alpha_k)^{K+1})} \mathsf{F}(k).
\end{aligned} \tag{7}$$

Now, an opacity change also affects the attenuation of samples corresponding to different bins. This, however, depends on the distribution of samples along a ray. To find a suitable approximation, we assume monotonicity of the intensity values along a ray. That is, for two given samples t_1 and t_2 along a ray, $t_1 < t_2$ implies $Volume(t_1) < Volume(t_2)$. Therefore, the attenuation function for the intensity values in a bin l, where $l > k$, depends on the attenuation of all samples before the samples in this bin $T(l)$. According to the monotonicity assumption, this attenuation can be decomposed into two factors, the attenuation due to the samples associated with function bin k and all others, i.e. $T(l) = (1-\alpha_k)^K T_0(l)$. Therefore, the accumulated attenuation is:

$$\begin{aligned}
\mathsf{F}(l) &= \alpha_l \sum_{i=0}^{L} \prod_{j=0}^{i} (1-\alpha_l) T(l) \\
&= \alpha_l \sum_{i=0}^{L} (1-\alpha_l)^i (1-\alpha_k)^K T_0(l) \\
&= (1-\alpha_k)^K \alpha_l \sum_{i=0}^{L} (1-\alpha_l)^i T_0(l),
\end{aligned}$$

where L is the number of samples in the slab containing samples that fall into bin l. After an opacity change for the intensity values in a bin k, new accumulated attenuation is:

$$\begin{aligned}
\mathsf{F}'(l) &= (1-\alpha'_k)^K \alpha_l \sum_{i=0}^{L} (1-\alpha_l)^i T_0(l) \\
&= \frac{(1-\alpha'_k)^K}{(1-\alpha_k)^K} \mathsf{F}(l).
\end{aligned} \tag{8}$$

We can see that, under these assumptions, the resulting attenuation function after an opacity change is a modulation of the original attenuation function. This modulation gets propagated from a bin that changed to the subsequent bins in the intensity range. In general, when the monotonicity assumption does not hold exactly, our solution provides an acceptable approximation, which can be used for exploratory visualization. We believe this is a reasonable assumption, as it appears locally in many scientific data sets. We have conducted a comparative and quantitative analysis that validates our assumptions, as shown in Figs. 4 and 8. Fig. 4 demonstrates attenuation propagation for the turbulent vorticity data set. Fig. 4(a) shows the original volume rendered image and (b) shows the volume rendered image after changing the opacities of the two outermost features. Fig. 4(c) shows the result of reconstruction using the RAF without attenuation propagation, which results in incorrect compositing. Fig. 4(d) shows the result of reconstruction using the RAF with correct attenuation propagation, as described in Eq. 8. Compare to the ground truth image in (b).

4. Coherent Visualization of Time-Varying Data

In this paper, we refer to coherent visualization as one that provides coherent temporal transfer functions. Let us define a temporal transfer function mapping: $TTF : R^2 \mapsto [0,1]^4$, which computes a color $C = (r,g,b,a)$ for a tuple of intensity and time: (s,t). A coherent TTF then guarantees that, if $s_1 = s_2$, then $TTF(s_1,t) = TTF(s_2,t)$, for all t in the domain. Local transfer functions, i.e. those that do not take

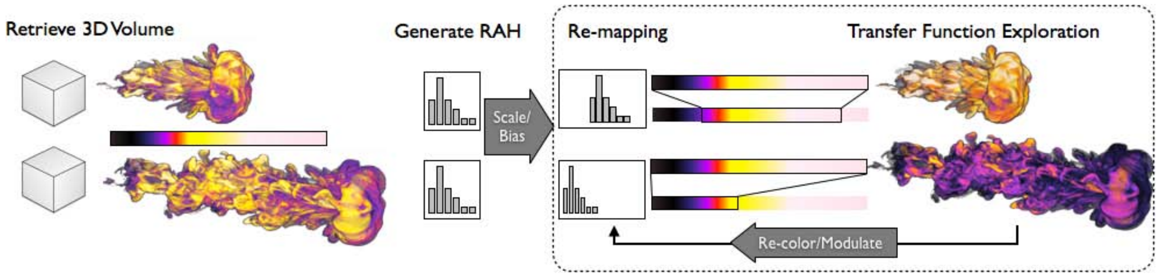
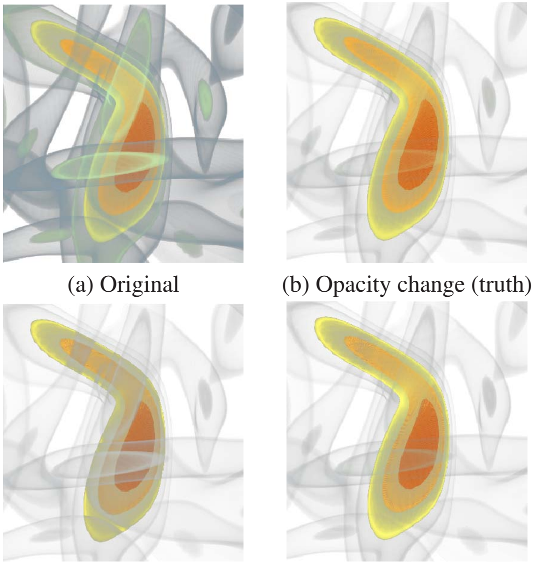


Figure 3: *Coherent visualization process overview. Simulation data is read from disk and classified using local transfer functions. For each time step, we generate a compact representation - ray attenuation functions. We can remap the transfer function by scaling and biasing the function to the global range. The resulting visualization is now coherent. Further operations on the transfer function can be performed on the RAF without the need to retrieve the original 3D volume data.*


(a) Original (b) Opacity change (truth)

(c) No Propagation (d) Correct Propagation

Figure 4: *Attenuation Propagation. (a) Original volume rendered image. (b) Volume rendered image after reducing the opacity of two intensity ranges. (c) Reconstruction using the RAF without attenuation propagation is incorrect. (d) Reconstruction using the RAF with correct attenuation propagation provides accurate results. Compare to (b).*

into account the time dimension, are, in general, not coherent, since intensity values are mapped to the local and not the global range. An example is shown in Fig. 1(a), where similar colors do not correspond to equivalent intervals in the data range. Once the global range is known, we can achieve coherence by manipulating the ray attenuation functions, without requiring access to the 3D volume data. Since the RAF is a compact representation, it can be stored in system and GPU memory for a large number of time steps. Data range remapping can be performed simultaneously for all time steps after they have been loaded into memory, in $O(T)$ time, where T is the number of time steps in a sequence.

Fig. 3 shows the main process for a typical application scenario. In this case, we assume we have several time steps and we do not know the global data range. As we read each individual time step, we apply a local transfer function, which can be provided by the user or generated automatically, based on the data statistics. Instead of generating a color image, we generate a number of images that constitute the ray attenuation function. For a 16-bin function, four RGBA images are sufficient, as we discuss in Section 4.4.

4.1. Local-to-Global Range Remapping

To see the effect of applying a coherent transfer function, we remap the RAF computed with a local transfer function to the global range. This can be achieved by means of data range remapping operations and recoloring, as introduced in the previous section. The remapping can be performed as defined in section 3.2, through two consecutive range remapping operations. First, from the local interval $[min_{local}, max_{local}]$ to the normalized interval $[0,1]$ and then to the global range $[min_{global}, max_{global}]$. This remapping has the advantage of retaining the richness of the local transfer function, which highlights isosurfaces of interest in the local range, while keeping a consistent coloring throughout the simulation. An example is shown in Fig. 1, depicting several time steps of the argon bubble-shockwave interaction simulation. This data set consists of 264 time steps, each containing $640 \times 256 \times 256$ voxels. The figure shows the result of the simulation of a shock wave interacting with a bubble of argon gas surrounded by air. As the simulation progresses, a swirling torus-shaped structure is observed along with smaller turbulent structures. A local transfer function (a) may be applied to capture these structures over time, but since they are not coherent, this may mislead the scientist. For example, the dissipation of the argon gas is not apparent in Fig. 1(a). On the other hand, the global transfer function helps us see the dissipation in (b). In (c), we show the results of remapping using our approach. The color bar shows the

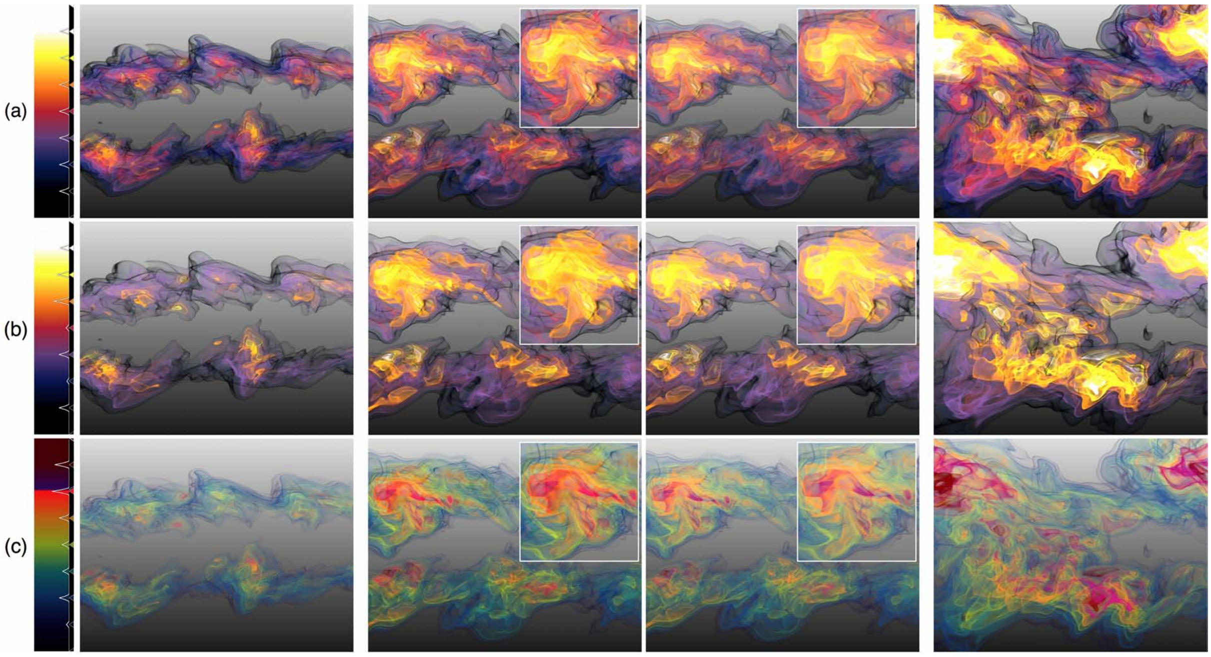


Figure 5: *Coherent transfer function operations performed on a combustion data set (three time steps are shown). (a) Results of reconstruction using attenuation functions. Compare the ground truth volume rendering (second column) to the image reconstructed using attenuation functions (third column). (b) The scientist adjusts the opacity of several features to avoid ambiguity in one of the regions, especially the structures in yellow and orange. (c) Finally, the user applies a new color map to highlight the variation in the interval previously colored yellow, which appeared flat. Now, the scientist can see a more intricate structure.*

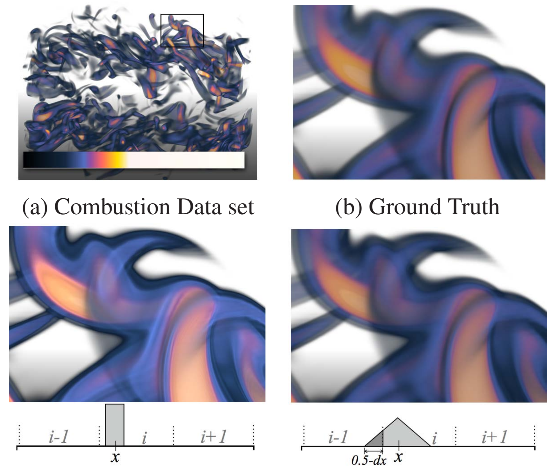
actual range of each local time step within the global range. Compared to (b), we see that the reconstruction is quite accurate. Fig. 1(d) also shows the result of changing the global color map to match the local transfer function for the last time step (right). This new color map brings out the shape of the argon gas ring. Since this change makes previous time steps appear homogeneous, we can add extra colors (dark red) to depict the variation of the scalar field.

4.2. Transfer Function Exploration

Fig. 5 shows results of exploring the color and opacity mappings for a combustion simulation. In particular, we visualize the OH variable, representing the mass fraction of the hydroxyl radical. This data set consists of 122 time steps, each containing 480 × 720 × 120 voxels. Columns $1,3$, and 4 show three reconstructed time steps using different transfer function operations. The second column is the ground truth image obtained using direct volume rendering, for the time step in column 3. We can see a relatively high visual accuracy of reconstruction. Fig. 5(a) shows the result of applying a coherent transfer function via remapping. Fig. 5(b) shows the effect of opacity modulation. Here, we decrease the opacity of features in red and purple to highlight regions with high OH. High OH regions are of interest to scientists, since they serve as re-ignition indicators. Fig. 5(c) shows the effect of re-colorization, applied to add more variation in the high OH regions and to highlight previously hidden structures resulting from low contrast of the yellow and orange regions. Since we can easily cache the compact RAF, we can apply these changes coherently throughout the simulation with little computational cost.

4.3. RAF Adjustment

By definition, each bin in the RAF accumulates the attenuation of all the intensity values mapped to it. This may introduce errors when trying to reconstruct fuzzy boundaries, as seen in Fig. 6 (c). This is due to the nearest neighbor approximation of accumulated attenuation. To alleviate this issue, we adjust the RAF using a smooth interpolation kernel that distributes attenuation between bins. If an intensity value coincides with the center of a bin in the RAF, the accumulated attenuation is assigned completely to that bin. Conversely, if the intensity value lies in the middle of two function bins, we assign half of the attenuation to each of the bins. In general, this adjustment is done as follows: let $T(i)$ denote the accumulated attenuation of a sample i along a ray. Without adjustment, the attenuation function is accumulated as $\mathsf{F}(k) = \mathsf{F}(k) + T(i)$, for $k = bin(i)$. Instead, we compute the offset distance of an intensity value $Volume(i)$ with respect



(a) Combustion Data set (b) Ground Truth

(c) No Function Adjustment (d) Function Adjustment

Figure 6: *Ray Attenuation Function adjustment. (a) A combustion data set and (b) inset view. (c) Without RAF adjustment, the reconstructed volume is not smooth and has sharp color boundaries. (d) With RAF adjustment (using a linear function), the boundaries are smooth. Compare to (b).*

to a bin's centroid: $dx = \lfloor Volume(i)N + 0.5 \rfloor N - Volume(i)$, and update the attenuation function for the bins k and $k-1$: The attenuation function is then updated as:

$$\begin{aligned} \mathsf{F}(k-1) &= \mathsf{F}(k-1) + \kappa(0.5-dx)T(i) \\ \mathsf{F}(k) &= \mathsf{F}(k) + (1-\kappa(0.5-dx))T(i), \end{aligned} \tag{9}$$

where $\kappa(x)$ is a kernel function that describes the contribution of each function bin. In Fig. 6(d), we show a polynomial kernel $\kappa(x) = x^p$ with $p = 1$. Compare to the ground truth image in Fig. 6(b). Examples throughout this paper use a polynomial kernel of degree $p = 3$.

4.4. GPU Implementation and Cost Evaluation

Our approach is fully implemented on the GPU. We use a single pass volume rendering shader to obtain the attenuation functions. We leverage multiple render target functionality of modern GPUs to write a 16-bin RAF in a single pass into four RGBA images. This shader evaluates Eq. 9 as it traverses a ray through the volume. Remapping and transfer function operations are performed in a separate rendering pass. This pass operates on a quad with the same dimensions as the screen, and fetches the attenuation values from the four images computed previously. Since the image is the result of simple 2D operations, this pass is much faster than volume rendering. Fig. 7 evaluates the performance of our approach and compares it to traditional volume rendering. Since our method is pixel-bound, i.e. it depends on the size of the image, we measure the rendering time for each frame vs. the effective pixel area of a given volume. We found an average of 1.7× overhead cost for obtaining the RAF. This cost is expected to approach 1.0 as GPU architectures improve. The cost of the re-compositing stage alone is considerably smaller compared to volume rendering, as seen by the green line running at the bottom of the plot. This is not only useful for remote and in-situ visualization, but proves to be a useful aid for local caching of time-varying data. The accompanying video shows several examples of our technique as a caching mechanism.

5. Limitations and Discussion

Although our technique offers interactive exploration of time-varying data, it is still a view-dependent rendering process. That is, the attenuation functions are only valid for a given viewpoint. However, in a typical exploration scenario, visualization users adjust the opacity and color properties while keeping the view intact. The view is only changed to explore how the classification highlights structures from different view points. In remote and in-situ visualization, viewpoint changes can be handled by pre-fetching different orientations of the data and using image-based techniques. We believe that our approach can be extended in the same way. We can pre-fetch the different RAF corresponding to different orientations of the data. Then, we can change the view direction slightly while still providing coherent transfer functions. Another limitation of our approach is the approximation of opacity transformations. As we have shown in Section 3, adjusting the opacity mapping cannot be reconstructed exactly using attenuation functions, since it depends on the actual distribution of samples along a ray. Fig. 8(left) shows our study of reconstruction accuracy using the RAF as we change the opacity of some intensity ranges for three data sets. We measure accuracy as the sum of square differences (SSD) between the image obtained with our approach and the image obtained with direct volume rendering. We can see a non-linear decrease in accuracy as the opacity modulation decreases (which results in a higher opacity change). The maximum change occurs at opacity modulation 0, which makes the outermost features completely transparent, while opacity modulation 1 means no change in opacity and provides the base reconstruction error. The accuracy also depends on the spatial frequency of the data. For example, for turbulent data sets, the overlapping of structures causes the attenuation saturate rather quickly. Therefore, it is not possible to reconstruct intensity values beyond those visible from a given view point. Finally, we have explored the RAF at 16-bin resolution. This choice was due to the GPU limitations in the number of render targets we could write simultaneously. However, we can extend the number of bins with additional rendering passes. As demonstrated in our results and the accompanying video, the quality of the reconstruction is acceptable for 16-bins. Fig. 8(right) shows the improvement in accuracy as the number of bins increases. For smooth data sets, such as *Vortex*, the accuracy does not improve considerably with additional bins. For turbulent structures, such as the argon bubble, however, we see a less steep

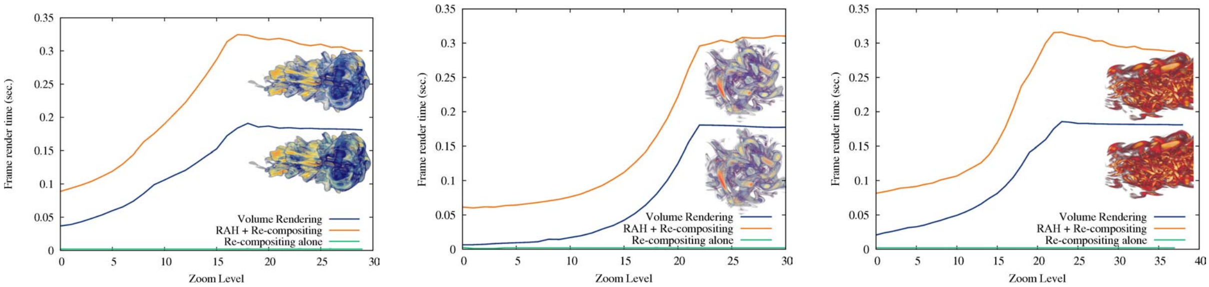


Figure 7: *Timing comparison for three data sets. Since our method is pixel-bound, i.e. it depends on image size, we compare time vs. effective pixel area of the volume. We compare direct volume rendering (blue), computation of the RAF (orange), and compositing of the RAF (green). The cost of computing the attenuation function is well outweighed by the benefit of re-compositing using the RAF (green plot).*

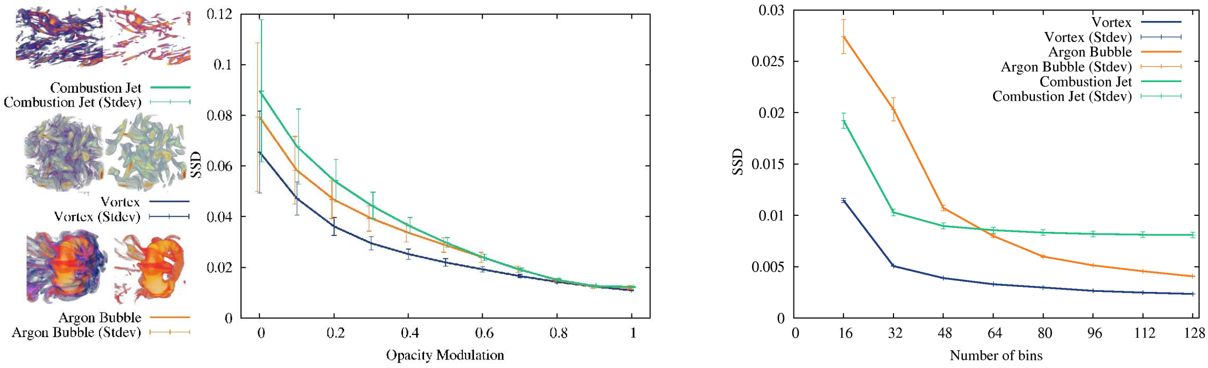


Figure 8: *Left: Accuracy plot for three data sets as we change the opacity of a given interval. We see a smooth increase in error (and in its standard deviation) as modulation goes from* 1 *(meaning no opacity change) to* 0 *(meaning that the outermost features are completely transparent). Right: Accuracy plot as we increase the number of bins in the RAF.*

improvement. These bins are placed uniformly along the intensity data range. In certain cases, data may be distributed unevenly throughout the domain. The function bins can then be adjusted to adapt to the data distribution, so that more bins are assigned to intervals with higher frequency.

6. Conclusion

In this paper, we introduce an exploratory technique for coherent visualization of time-varying data. In the past, coherency required a priori knowledge of the entire simulation, such as the global data range. This prevented scientists from exploring the data in-situ, while the simulation is running, or at interactive rates for large data sets. We have shown that ray attenuation functions, which are compact representations of volume data, can be stored and cached in system memory for a large number of time steps. Their main advantage is enabling coherent exploration in transfer function space without accessing the original time-varying data. Our approach is an approximation of volume rendering and is geared for exploratory/preview visualization. It is a useful mechanism for exploring time-varying volume data that alleviates the costs associated with transferring, reloading, and caching 3D volume data sets. The ability to browse through the simulation data quickly allows scientists to efficiently explore different transfer functions to highlight certain structures of interest. Once the scientist finds a satisfactory transfer function, the system can generate a volume rendering of the entire simulation at full accuracy, if the original simulation data is accessible.

7. Acknowledgements

This research was supported in part by the U.S. National Science Foundation through grants OCI-0325934, OCI-0749217, CNS-0551727, CCF-0811422, OCI-0749227, OCI-0950008, CCF-0938114, and OCI-0850566, and the U.S. Department of Energy through the SciDAC program with Agreement No. DE-FC02-06ER25777 and DE-FG02-08ER54956. Datasets courtesy of Jackie Chen of Sandia National Laboratories, John Bell and Vince Beckner of Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, and Deborah Silver of Rutgers University.

References

[AFM06] AKIBA H., FOUT N., MA K.-L.: Simultaneous classification of time-varying volume data based on the time histogram. In *In Proc. of Eurographics Visualization Symposium* (May 2006), pp. 1–8. 3

[BCF03] BINOTTO A., COMBA J. L. D., FREITAS C. M. D.: Real-time volume rendering of time-varying data using a fragment-shader compression approach. In *In Proc. of IEEE Symposium on Parallel and Large-Data Visualization and Graphics* (2003), p. 10. 3

[BPS97] BAJAJ C. L., PASCUCCI V., SCHIKORE D. R.: The contour spectrum. In *In Proc. of IEEE Visualization* (1997), pp. 167–173. 3

[BS05] BORDOLOI U., SHEN H.-W.: View selection for volume rendering. In *In Proc. of IEEE Visualization* (2005), pp. 487–494. 4

[CM09] CORREA C. D., MA K.-L.: Visibility-driven transfer functions. In *In Proc. of IEEE Pacific Visualization Symposium* (2009), pp. 177–184. 3, 4

[GS01] GUTHE S., STRASSER W.: Real-time decompression and visualization of animated volume data. In *In Proc. of IEEE Visualization* (2001), pp. 349–356. 3

[GSHK04] GAO J., SHEN H.-W., HUANG J., KOHL J. A.: Visibility culling for time-varying volume rendering using temporal occlusion coherence. In *In Proc. of IEEE Visualization* (2004), pp. 147–154. 3

[HHKP96] HE T., HONG L., KAUFMAN A., PFISTER H.: Generation of transfer functions with stochastic search techniques. In *In Proc. of IEEE Visualization* (1996), pp. 227–ff. 3

[JKM01] JANKUN-KELLY T. J., MA K.-L.: A study of transfer function generation for time-varying volume data. In *In Proc. of Volume Graphics* (2001). 3

[KBH04] KOSARA R., BENDIX F., HAUSER H.: Timehistograms for large, time-dependent data. In *In Proc. of Symposium on Data Visualisation* (2004), pp. 45–54, 340. 3

[KD98] KINDLMANN G., DURKIN J. W.: Semi-automatic generation of transfer functions for direct volume rendering. In *In Proc. of Volume Visualization* (1998), vol. 1, pp. 79–86. 3

[KKH01] KNISS J., KINDLMANN G., HANSEN C.: Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *In Proc. of IEEE Visualization* (2001), pp. 255–262. 3

[LMC01] LUM E. B., MA K. L., CLYNE J.: Texture hardware assisted rendering of time-varying volume data. In *In Proc. of IEEE Visualization* (2001), pp. 263–270. 3

[LP03] LAMAR E., PASCUCCI V.: A multi-layered image cache for scientific visualization. In *In Proc. of IEEE Symposium on Parallel and Large-Data Visualization and Graphics* (2003), pp. 61–68. 3

[MAB*97] MARKS J., ANDALMAN B., BEARDSLEY P. A., FREEMAN W., GIBSON S., HODGINS J., KANG T., MIRTICH B., PFISTER H., RUML W., RYALL K., SEIMS J., SHIEBER S.: Design galleries: a general approach to setting parameters for computer graphics and animation. In *In Proc of SIGGRAPH* (1997), pp. 389–400. 3

[Max95] MAX N.: Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics 1*, 2 (1995), 99–108. 3

[MCP91] MA K.-L., COHEN M. F., PAINTER J. S.: Volume seeds: A volume exploration technique. *Visualization and Computer Animation 2* (1991), 135–140. 3

[MMG07] MALIK M. M., MÖLLER T., GRÖLLER M. E.: Feature peeling. In *In Proc. of Graphics Interface* (2007), pp. 273–280. 3

[PBL*04] PARK S. W., BUDGE B., LINSEN L., HAMANN B., JOY K. I.: Multi-dimensional transfer functions for interactive 3d flow visualization. In *In Proc. of Pacific Conference on Computer Graphics and Applications* (2004), vol. 1, pp. 177–185. 3

[PLB*01] PFISTER H., LORENSEN B., BAJAJ C., KINDLMANN G., SCHROEDER W., AVILA L. S., MARTIN K., MACHIRAJU R., LEE J.: The transfer function bake-off. *IEEE Computer Graphics and Applications 21*, 3 (2001), 16–22. 3

[RPSH08] ROPINSKI T., PRASSNI J.-S., STEINICKE F., HINRICHS K. H.: Stroke-based transfer function design. In *In Proc. of IEEE/EG International Symposium on Volume and Point-Based Graphics* (2008), pp. 41–48. 3

[RSK06] REZK-SALAMA C., KOLB A.: Opacity Peeling for Direct Volume Rendering. *In Proc. of Computer Graphics Forum (Proc. Eurographics) 25*, 3 (2006), 597–606. 3

[SCM99] SHEN H.-W., CHIANG L.-J., MA K.-L.: A fast volume rendering algorithm for time-varying fields using a time-space partitioning (tsp) tree. In *In Proc. of IEEE Visualization* (1999), pp. 371–377. 3

[SCM03] SRIVASTAVA V., CHEBROLU U., MUELLER K.: Interactive transfer function modification for volume rendering using compressed sample runs. In *In Proc. of Computer Graphics International Conference* (2003), pp. 8–13. 3

[SJ94] SHEN H.-W., JOHNSON C. R.: Differential volume rendering: a fast volume visualization technique for flow animation. In *In Proc. of IEEE Visualization* (1994), pp. 180–187. 3

[SLSM06] SHAREEF N., LEE T.-Y., SHEN H.-W., MUELLER K.: An image-based modeling approach to gpu-based unstructured grid volume rendering. In *In Proc. of Volume Graphics* (2006), pp. 31–38. 3

[TM05] TZENG F.-Y., MA K.-L.: Intelligent feature extraction and tracking for visualizing large-scale 4d flow simulations. In *In Proc. of Supercomputing* (2005), pp. 6–16. 3

[WQ07] WU Y., QU H.: Interactive transfer function design based on editing direct volume rendered images. *IEEE Transactions on Visualization and Computer Graphics 13*, 5 (2007), 1027–1040. 3

[WS09] WOODRING J., SHEN H.-W.: Semi-automatic time-series transfer functions via temporal clustering and sequencing. *Computer Graphics Forum 28*, 3 (2009), 791–798. 3

[WWS03] WOODRING J., WANG C., SHEN H.-W.: High dimensional direct rendering of time-varying volumetric data. In *In Proc. of IEEE Visualization* (2003), p. 55. 3

[WYM08] WANG C., YU H., MA K.-L.: Importance-driven time-varying data visualization. *IEEE Transactions of Visualization and Computer Graphics 14*, 6 (2008), 1547–1554. 3

[YMC05] YOUNESY H., MÖLLER T., CARR H.: Visualization of time-varying volumetric data using differential time-histogram table. In *In Proc. of Volume Graphics* (2005), pp. 21–29. 3