Eurographics/ IEEE-VGTC Symposium on Visualization (2006)
Thomas Ertl, Ken Joy, and Beatriz Santos (Editors)

Simultaneous Classification of Time-Varying Volume Data Based on the Time Histogram

Hiroshi Akiba†, Nathaniel Fout† and Kwan-Liu Ma†

Department of Computer Science, University of California at Davis, U.S.A

Abstract

An important challenge in the application of direct volume rendering to time-varying data is the specification of transfer functions for all time steps. Very little research has been devoted to this problem, however. To address this issue we propose an approach which allows simultaneous classification of the entire time series. We explore options for transfer function specification that are based, either directly or indirectly, on the time histogram. Furthermore, we consider how to effectively provide feedback for interactive classification by exploring options for simultaneous rendering of the time series, again based on the time histogram. Finally, we apply this approach to several large time-varying data sets where we show that the important features at all times are captured with about the same effort it takes to classify one time step using conventional classification.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Display Algorithm I.3.6 [Computer Graphics]: Methodology and Techniques I.3.8 [Computer Graphics]: Applications

1. Introduction

An important challenge in scientific visualization is the problem of time-varying volume data visualization. Time-varying data sets are typically produced by simulations in which a complex model of some physical phenomena is iterated in time. The data is produced by supercomputers and then stored in depots for subsequent offline examination, which may include visualization. The characteristics which uniquely identify this type of data are the additional dimension of time and the size of the data. Just as visualization of volumes is not a trivial extension of image visualization, 4D time-varying volume visualization is not trivially lifted from 3D volume visualization. There are generally two classes of time-varying visualization: homogeneous, in which all dimensions are treated equivalently, and inhomogeneous, in which the time dimension is considered separately from the spatial dimensions. A practical challenge of time-varying volume data is its size. These datasets are typically large; consider that there may be hundreds of time steps, where each time step consists of a large multivariate volume stored in floating-point format. Interactive exploration of such data sets can easily overwhelm the capabilities of available computing resources.

In terms of visual exploration of time-varying volume data, three areas have been the focus of most research: representation/organization, feature extraction/tracking, and rendering. Research on representation focuses on how the data is stored and accessed, including both organization and reduction. Compression [GS01, LMC01, LPD*02, BCF03], multiresolution analysis [SCM99], and differential encoding [SJ94] are prime examples of research conducted in this area. Feature extraction [SW97, RPS01, PVH*03] takes a higher level view of the data, wherein the data is not just a collection of voxels but rather a set of features evolving over time. These features could be either volumetric or geometric. Once defined and extracted, the evolution of these features can be tracked over time. Research in the area of rendering attempts to display the 4D data with a 2D view. One approach computes projections from 4D to 3D followed by volume rendering [WWS03]. Another alternative is chronovolumes [WS03], where an integration over time with appropriate color mappings is used to show several time steps in one volume. A more comprehensive discussion of research in time-varying volume visualization is given by Ma [Ma03].

One problem often overlooked when rendering time-

† {akiba | fout | ma}@cs.ucdavis.edu

Figure 1: *Histograms are useful in classification and are often displayed as the backgrounds of transfer function editors. Here we see a 1D editor/histogram (left) and a 2D editor/histogram (right).*

varying data sets based on direct volume rendering of individual time steps is that of classification. As opposed to classification of single volumes, classification of time-varying volume data has received very little attention in literature. We are aware of only two works that deal specifically with this problem, namely Jankun-Kelly and Ma [JKM01] and Tzeng and Ma [TM05]. The former describes approaches for computing one or a few transfer functions for an entire time series given a transfer function for each time step, whereas the latter computes transfer functions for all time steps given transfer functions from a few key time steps. In contrast, our approach assumes no transfer functions are given for any time step.

For most time-varying data sets the problem of finding a classification which works for all time steps and is temporally continuous is not trivial. In this work we propose a solution which simultaneously classifies the entire time series. The time histogram is an integral part of this process, both guiding and participating in the creation of temporal transfer functions. Two approaches for building the transfer function are developed, with the most promising being a semi-automatic method which partitions the time histogram into equivalence classes. In order to interactively classify the time series some type of visual feedback is required, and so we explore rendering possibilities which provide a global view of the data set. Our solution allows interactive classification of an entire time-varying data set with marginally more effort than that required to classify a single time step.

2. The Time Histogram

Histograms have proven to be very useful in conjunction with transfer function specification. Traditional 1D histograms showing scalar value versus frequency of occurrence (Figure 1 left) are often displayed as the background of simple 1D transfer function interfaces, where they are useful for guiding the user to populated areas of the data range. Even more dependent on histograms are the 2D transfer functions which depict spatial gradient versus scalar value (Figure 1 right). For volumes in which material boundaries are present these histograms are quite useful in identifying material ranges and boundary ranges in the data [Lev88, KD98]. In some cases they can play a more active role in the classification process as well.

For a time-varying volume data set we can compute a conventional 1D histogram for each time step. If we then concatenate these together then we get a histogram which gives frequency of occurrence for each value and time. This structure, called the time histogram, contains a wealth of information about the entire time series. For one, it provides a concise statistical overview of the data. Second, it offers a global context within which temporal features (i.e. events) can be distinguished. In this work we demonstrate two additional uses of the time histogram: characterization of time series and temporal classification. The time histogram has only recently been investigated in the context of time-varying data visualization. Kosara et al. [KBH04] consider the time histogram from an information visualization perspective, discussing different options for display and interaction, including brushing techniques. Doleisch et al. [DMG*04] make use of the time histogram to analyze the dynamic behavior of a complex diesel exhaust simulation. More recently, Younesy et al. [YMC05] propose a data structure similar to the time histogram called the Differential Time Histogram Table, which is used to provide an encoding that takes advantage of temporal coherence. This structure is used during rendering to minimize the amount of data required during an update.

2.1. Time Histogram Display

As discussed by Kosara et al. [KBH04], the time histogram is a 2D map which can be displayed either in 3D as a height field or in 2D as an image (see Figure 2). The latter method is generally preferred for several reasons. First, in 3D there are occlusion problems, requiring some sort of navigation. Second, color patterns in an image representation are much easier for the human visual system to discriminate than ridges and peaks in a 3D display whose sizes are relative to perspective. Third, the image display is more compact, allowing a more efficient use of screen space.

One aspect of the time histogram not sufficiently explored in the past is the mapping of frequency counts to color in the image representation. This mapping can have a significant impact on the usefulness of the time histogram image. A simple linear mapping is often unacceptable since it allocates image space equally over the entire data range. In fact, many time-varying data sets exhibit value clustering, such that a large portion of the data lies within a small range of values. A potential solution is to perform histogram equal-

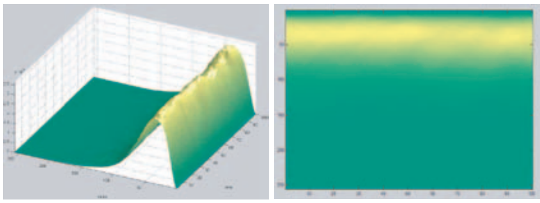

Figure 2: *A 2D time histogram can be displayed as a 3D height field (left) or as a 2D image (right).*

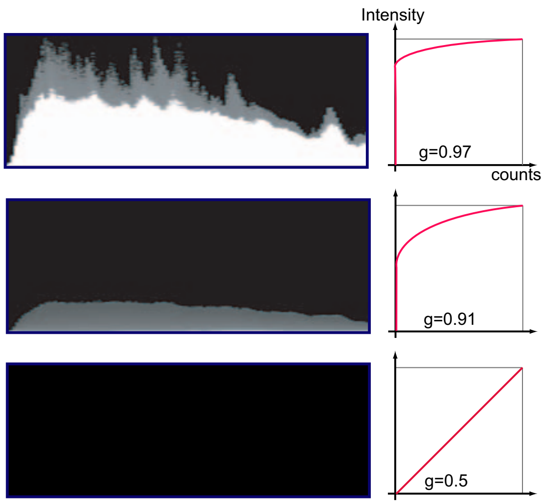

Figure 3: *Display of the time histogram from a combustion simulation using the bias function with different parameter values.*

ization to the histograms of each time step, but then histograms are no longer comparable, as they live on different scales. Clearly it is expedient to allow a variety of mappings in order to best bring out patterns in the time histogram. To this end we propose the use of bias function, as given by Perlin in the context of texture mapping [PH89]. This function is defined as:

$$b(g,f) = f^{\ln g/\ln 0.5}$$

where f is the frequency count (normalized to [0,1]) and g is the bias parameter. Note that when $g = 0.5$ the mapping is linear, whereas for other values it is exponential. By manipulating the single parameter g we can effectively map count to color in a flexible way, as shown in Figure 3.

2.2. Static Versus Dynamic Histograms

Figure 4 shows some typical time histograms. We can immediately observe that in general there are two kinds of time histograms: those in which the histogram changes with time and those in which the histogram is constant with time. Datasets whose time histograms exhibit these traits we call *statistically dynamic* and *statistically static*, respectively. A statistically static data set is usually dominated by diffusion processes, with other processes being in equilibrium. On the other hand, statistically dynamic data sets are characterized by other transient forces, whether they be chemical, thermodynamic, or nuclear.

Characterization of the time histogram as static or dynamic is important because it determines what means must be employed in order to classify the volumes of the time series. In the case of statistically static volumes, a single transfer function can be constructed by examining any time step and then that same transfer function can be used for all other time steps. However, for statistically dynamic data a different transfer function may be needed for every time step, effectively tracking time-evolving features. Finding a way to classify these types of data sets presents a unique challenge.

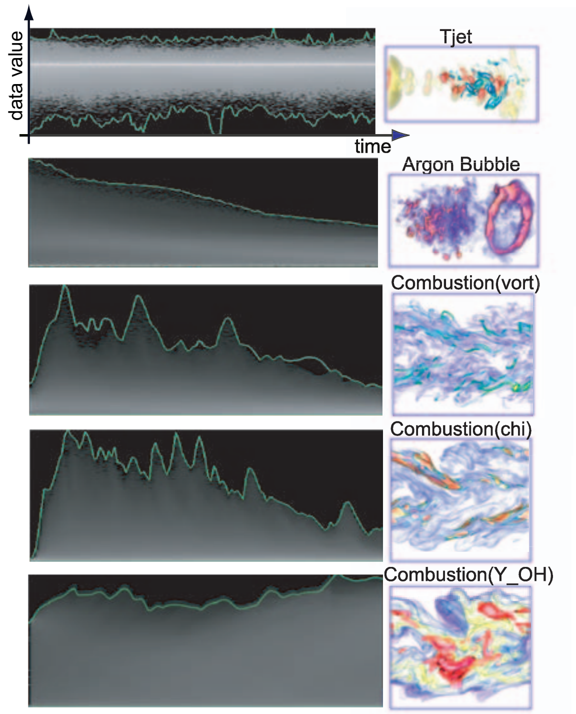

Figure 4: *Display of some typical time histograms. The time histogram can characterize a data set as either statistically static (tjet) or statistically dynamic (all others). Green lines mark maximum and minimum scalar values.*

2.3. Alternative Time Histograms

We can also compute the distribution of other properties of the volume besides scalar value and use these to create a time histogram, such as the spatial gradient magnitude. One property that is particularly interesting for time-varying data is the temporal gradient. By computing the temporal gradient time histogram we can identify and classify highly active regions as they evolve in time, as shown in Figure 5. Often these are the most interesting regions in the data, regions of high energy. By keeping the actual gradient, as opposed to just the magnitude, we can distinguish regions of increasing and decreasing value. As can be observed in Figure 5, images rendered based on classification of the temporal gradient are very informative, but more importantly they are useful to the scientists.

3. Simultaneous Temporal Classification

The problem we address in this work is that of classifying statistically dynamic time-varying volume data. We ex-

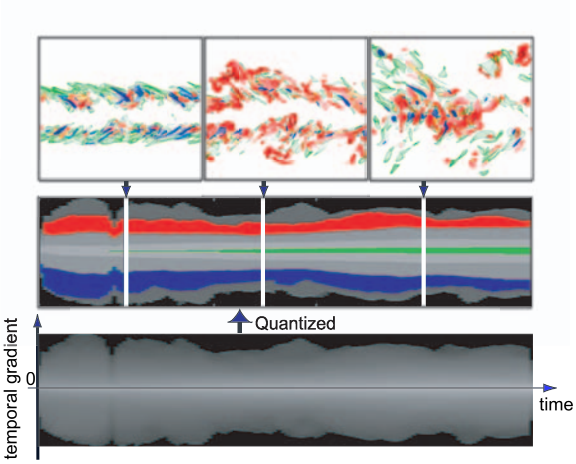

Figure 5: *Time histogram computed from the temporal gradient instead of the scalar value. The images highlight highly active regions, both positive change (red) and negative change (blue).*

plore two different possibilities and identify the one which we believe is the most promising. In both of these methods the time histogram is the common denominator, in one case guiding and in the other case participating in the generation of transfer functions. It is important to realize that in our approach the entire time series is being classified simultaneously. Furthermore, our problem is made more difficult by the fact that we assume no transfer functions are given *a priori*.

A straightforward approach is to simply take the preferred method of classifying a single static volume and then apply this to every time step. For completely automatic methods this is an option, but most practical methods rely at least in part upon user interaction. In these cases it is simply unreasonable to require the user to engage in classification of hundreds of time steps. Even if that were possible, the sequence of transfer functions would likely be discontinuous in time since they would be created independently. This means that animations in time would be incoherent, and also visual comparisons of different time steps would be difficult to interpret. Generally there is a significant amount of coherence between adjacent time steps, and so it makes sense to take advantage of this coherence to make a dependent classification, which will in turn lead to coherence in the transfer functions as well. Furthermore, a dependent classification has the potential to reduce the amount of interaction required from the user. The efficacy of our approach depends strongly on this last point. Since the histogram and the transfer function share the same domain it makes sense to display them together, as is often done in conventional volume rendering systems. Using this approach the time histogram is rendered in the background, whereas the transfer function is displayed in the foreground. A transfer function specified in this domain can be viewed as a single 2D transfer function or

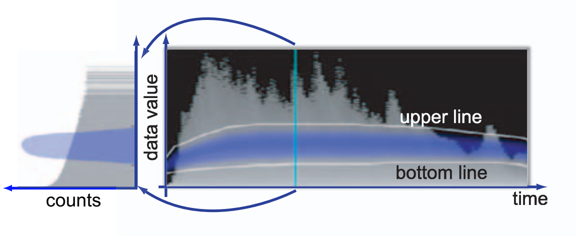

Figure 6: *The constrained freeform interval is a component defined by two horizontal lines. After drawing the lines, the user assigns an opacity profile and color which is used along the vertical axis.*

equivalently a series of conventional 1D transfer functions. We prefer to think of it as a single 2D function in order to emphasize the fact that the individual 1D functions are not independent. Letting time be the horizontal axis means that patterns will predominantly be oriented horizontally. For this reason the structure of the components we use to build the transfer function are likewise oriented horizontally.

We now describe two possible approaches which incorporate the time histogram in order to interactively classify a time series. The first uses the time histogram in a passive sense to guide the user. The second actively uses the time histogram as a selection tool in the classification.

3.1. Constrained Freeform Intervals

In this approach one or more components are used to build the complete transfer function. Each component is defined as an interval that can change arbitrarily in time. The only restriction is that bifurcation is not allowed; that is, the interval cannot split into two intervals. As shown in Figure 6, a component is defined by using a paint brush to draw two horizontal freeform lines. Within these lines the user can specify a vertical profile which is interpolated over the entire length of the component. The vertical profile defines the opacity and color mapping which exists within the interval (see left part of Figure 6). This specification method is intuitive to use and offers great flexibility; however, it is difficult to specify a good transfer function due to the many degrees of freedom available, and it is also quite difficult to be precise.

3.2. Semi-automatic with Equivalence Classes

The second method we propose is unique in that the classification is based directly on the time histogram, unlike the previous approach where it resides passively in the background. Based on the observation that the interesting features of the data are generally value intervals, we define some number of equivalence classes where each class is characterized by a range of scalar values. Within each class we define a common transfer function. The union of the equivalence sets is the entire transfer function domain, and so the union of the

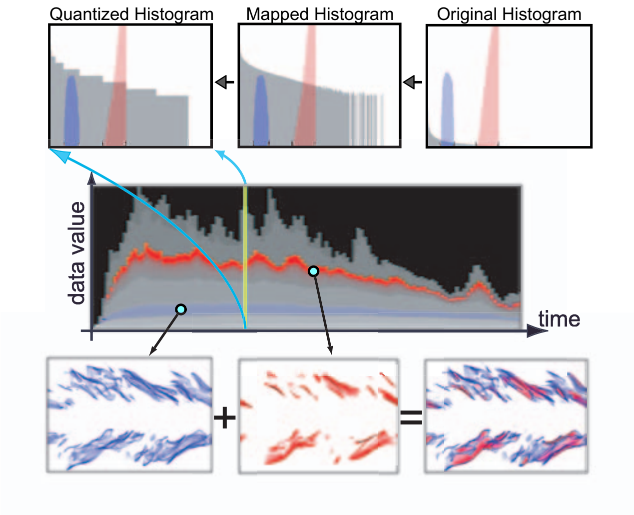

Figure 7: *A quantization of the time histogram creates some number of equivalence classes covering various regions of the histogram domain. In classification these classes are selected and assigned properties by the user.*

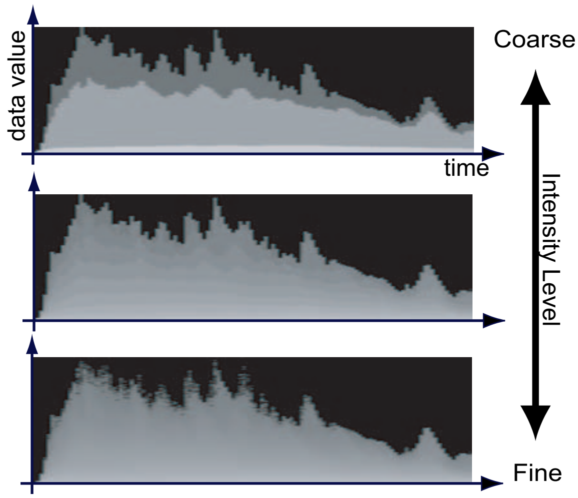

Figure 8: *By adjusting the histogram display mapping and the granularity of the quantization, a very fine to very coarse classification can be obtained.*

transfer functions from each class defines the total transfer function (see Figure 7). In our scheme the equivalence classes are implicitly defined by performing a uniform quantization. In the transfer function domain each class is displayed with a separate greyscale value. The user builds the transfer function by first clicking on a class and then setting the opacity and color profiles for that class in another window.

In some sense this approach is like feature extraction/tracking in the time histogram domain. Although this approach is less flexible than the previous technique it is very simple and intuitive to use. As shown in Figure 8, by adjusting two parameters, the time histogram mapping parameter g and the number of equivalence classes, a very coarse to very fine partitioning of the domain can be achieved. This greatly facilitates exploration of the data at different levels, with a typical scenario being to start with a coarse classification, followed by classification using successively finer transfer functions. We refer to this technique as semi-automatic since the initial classification into equivalence classes is automatic, whereas the choice of which classes to use is manual. Of the two methods presented we believe this one to be the most promising. Specification is very simple, the parameters offer flexibility in terms of feature granularity, and the discrete number of classes (or features) both limit and guide the user towards a meaningful classification.

3.3. Visual Feedback

Interactive classification is a trial-and-error process in which the user relies on visual feedback to modify and refine the transfer function. For a single volume we would simply render the classified volume directly, but this approach does not scale well for time-varying data. Since we are effectively classifying the entire time series at once, it makes sense that in terms of visual feedback we need to "see" the entire time series at once. In general this may not be possible; however, for many data sets we can reduce the problem to viewing a representative subset of the data.

The problem of providing a global view of the data can be simplified if we make two assumptions. The first assumption is that given two identical histograms the desired transfer function will be identical as well, regardless of the data itself. As long as the two histograms are from the same data set then this assumption is reasonable, as features are generally defined according to value ranges. Second, we assume that if two histograms are similar then their desired transfer functions will be similar as well. Based on these assumptions and the observation that most time-varying data sets exhibit significant temporal coherence, we can reduce the data in the time dimension by only considering unique statistical behavior. To this end we propose an algorithm that successively merges time steps into time intervals, thereby reducing the problem of viewing a large number of time steps to that of viewing a small number of time intervals. We then divide the screen space into multiple views, where each interval gets its own view.

The algorithm for temporal reduction takes as input the number of available views, t, and proceeds as follows:

1. Begin with the sequence of histograms from each time step, $\{h_0, h_1, \ldots, h_n\}$.
2. Compute the distance between each consecutive pair of histograms; that is, compute $d_i = \mathbf{dist}(h_i, h_{i+1}) \quad for\ i = 0 \ldots n-1$.
3. Merge the two closest histograms. If the merged repre-

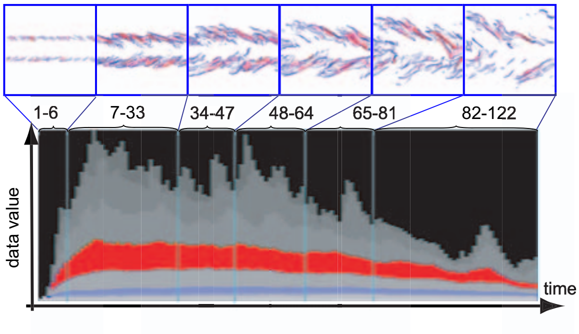

Figure 9: *Time intervals resulting from the histogram merging procedure are displayed in multiple views. The renderings are of the midpoint of each interval.*

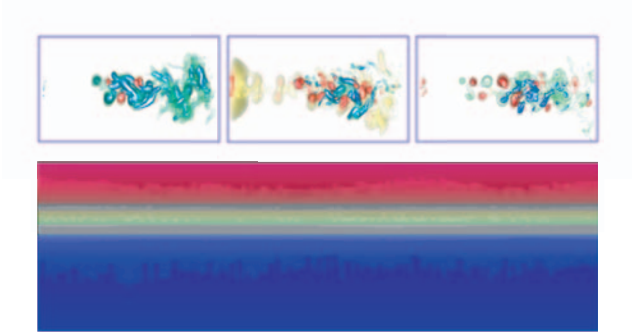

Figure 10: *Since the turbulent jet data set is statistically static, we can define a single 1D transfer function to classify the entire time series.*

sentation is h^* then $h^* = \textbf{merge}\ (h_k, h_{k+1})$, where $d_k = \textbf{min}\ d_i$.

4. Repeat steps 2 and 3 until there are only t time intervals remaining.

The results of this algorithm depend entirely upon the choice of the two functions **dist** and **merge**. The most reasonable choices are the Euclidean distance (based on the L_2 norm) for **dist** and the average for **merge**, although other choices are certainly possible. Once we have reduced the data to some small number of time intervals, the question is how do we render a time interval. This is more of a subjective question and could depend largely on the characteristics of the data set under consideration. Volume rendering of a single time step from the interval is a simple solution. For instance, we could pick the midpoint, a random point, or the point with the highest temporal variance. Another possibility is to compute an average volume for the interval, although interpretation of the resulting rendering could be difficult. Finally, we could use existing techniques for rendering of 4D data, such as the chronovolumes method [WS03].

Figure 9 shows the results of applying the merging algorithm to a time histogram with $t = 6$ (six views). The standard distance metric is used for computing histogram distances, and the merge simply computes the average. For rendering the midpoint of the interval is used.

4. Applications

Combining all the elements of the previous section we developed a time-varying data visualization system capable of temporally simultaneous classification. Three types of transfer functions are available: the usual 1D for statistically static data and the two from the previous section for statistically dynamic data. We use hardware volume rendering with a variable number of views to provide feedback, where all of the views are linked in terms of rendering parameters such as orientation. The user can select to display the scalar value time histogram, temporal gradient time histogram, or spatial gradient magnitude time histogram.

We present results from three different time-varying simulations. We applied our approach to many more data sets with similar results, but space limitations prohibit us from presenting all the cases.

4.1. Turbulent Jet Data Set

The turbulent jet data set is from a simulation consisting of 160 time steps, where each volume is of size 104x129x129. The interesting features of this small data set are the vortices resulting from the turbulent flow. Despite the fact that the flow is turbulent, the statistical behavior of the volume is mostly constant with time, as can be observed by examining the scalar value time histogram. In Figure 10 we show that a single 1D transfer function is sufficient to classify the entire time series.

4.2. Argon Bubble Data Set

The argon bubble data set is from a simulation modeling shock refraction and mixing, wherein the evolution of a shock wave disturbing an argon bubble is observed. The data set consists of 264 time steps, where each time step consists 640x256x256 voxels. The effects of the shock create a 'jellyfish' structure with a head that begins as an amorphous mass and turns into a ring, and a tail that consists of small turbulent structures which disband over time. The primary feature of interest in this data set is the ring structure, however. Since the value range which defines the ring changes over time it is difficult to capture this feature using existing methods. Using our approach this feature is captured in all time steps in only a few minutes. Figure 11 shows a sample from such a classification which took the authors less than 5 minutes to create. The semi-automatic method was used to obtain this result.

4.3. Combustion Data Set

The combustion data set is from a direct numerical simulation of turbulent combustion, in which fuel is injected

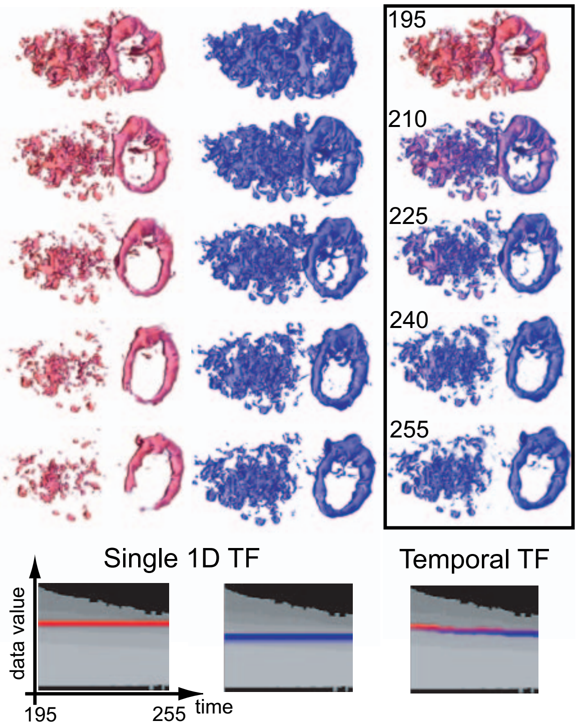

Figure 11: *Classification of the argon bubble data set. The rows show selected timesteps: 195, 210, 225, 240, 255(top to bottom). The two columns on the left show the results of using 1D transfer functions(shown at the bottom). Neither function is able to isolate the feature of interest (the ring) for all the time steps. The right column shows the results of using our approach, which is able to capture the feature in all time steps.*

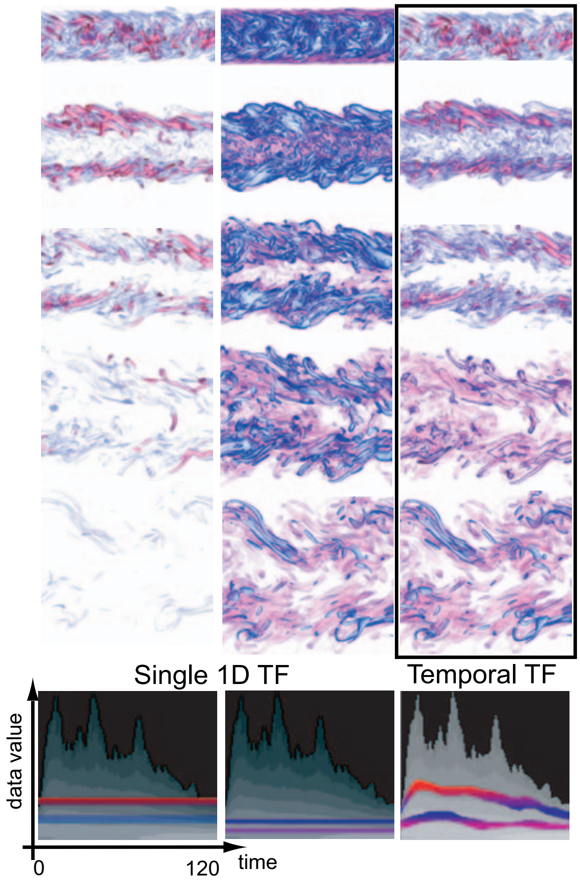

Figure 12: *Classification of the vorticity magnitude field in the combustion data set. The rows show selected timesteps: 5, 35, 60, 90, 120(top to bottom). The two columns on the left show the results of using 1D transfer functions. The right column shows the results of using our approach, which is able to show the vortical structures from all time steps.*

into two countercurrent air streams, and the mixing creates turbulent regions wherein the combustion occurs. The data set consists of 128 time steps, where each volume is of size 480x720x120. This data set is also multivariate, storing several physical properties and chemical concentrations per voxel. Of particular interest to scientists is the vorticity magnitude field. As shown in Figure 12, a 1D transfer function is incapable of capturing the interesting vortical structures over all time steps. With our approach, a classification that works for all time steps is effortlessly constructed in a matter of minutes (less than 5 minutes for the classification actually shown). In order to obtain this result, the semi-automatic method is used to find a initial classification and the constrained free-form interval method is then used for refining the classification.

5. Conclusions and Future Work

In this work we address the problem of finding transfer functions that classify all the time steps of a time-varying volume data set. We offer two methods that simultaneously classify the entire time series at once. The time histogram plays a critical role, guiding and participating in the classification. Visual feedback for interactive trial-and-error transfer function specification is also discussed, and a general algorithm for reducing the problem of viewing the entire time series simultaneously is presented. We finally demonstrate that our approach is able to successfully classify time-variant features from various data sets.

Additional research in the area of time-varying data visualization may uncover even better methods for simultaneous classification. We have developed herein a novel approach that is both simple and powerful, demonstrating its efficacy on several time-varying data sets.

Our proposed solution demonstrates among other things that ideas from information visualization can be applied to help solve problems in scientific visualization. In fact, many visualization tools incorporate components of both types of visualization, sometimes linking the two. We believe that a furtive area of scientific visualization research is the middle ground between these two disciplines.

6. Acknowledgements

This work has been sponsored in part by the U.S. National Science Foundation under contracts ACI 9983641 (PECASE), ACI 0222991, and ACI 0325934 (ITR), and the U.S. Department of Energy under Memorandum Agreement No. DE-FC02-01ER41202 (SciDAC) and DE_FG02-05ER54817 (SciDAC). The argon bubble data set was provided courtesy of John Bell and Vince Beckner of Lawrence Berkeley National Laboratory, and the combustion data set was provided courtesy of Jackie Chen of the Sandia National Laboratory.

References

[BCF03] BINOTTO A. P. D., COMBA J., FREITAS C. M. D. S.: Real-time volume rendering of time-varying data using a fragment-shader compression approach. In *Proceedings of the 2003 IEEE Symposium on Parallel and Large-Data Visualization and Graphics* (2003), pp. 69–76.

[DMG*04] DOLEISCH H., MAYER M., GASSER M., WANKER R., HAUSER H.: Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system. In *Proceedings of the 2004 Eurographics/IEEE TCVG Symposium on Visualization* (2004), pp. 91–96.

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

[JKM01] JANKUN-KELLY T. J., MA K.-L.: A study of transfer functions generation for time-varying volume data. In *Proceedings of the 2001 Eurographics/IEEE TVCG Workshop on Volume Graphics* (June 2001), pp. 51–68.

[KBH04] KOSARA R., BENDIX F., HAUSER H.: Time-histograms for large, time-dependent data. In *Proceedings of the 2004 Eurographics/IEEE TVCG Symposium on Visualization* (2004), pp. 45–54, 340.

[KD98] KINDLMANN G., DURKIN J. W.: Semi-automatic generation of transfer functions for direct volume rendering. In *Proceedings of the 1998 IEEE symposium on Volume Visualization* (1998), pp. 79–86.

[Lev88] LEVOY M.: Display of surfaces from volume data. *IEEE Comput. Graph. Appl. 8*, 3 (1988), 29–37.

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

[LPD*02] LINSEN L., PASCUCCI V., DUCHAINEAU M. A., HAMANN B., JOY K.: Hierarchical representation of time-varying volume data with '4th-root-of-2' subdivision and quadrilinear B-spline wavelets. In *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications - Pacific Graphics 2002* (2002), p. 346.

[Ma03] MA K.-L.: Visualizing time-varying volume data. *Computing in Science and Engineering. 5*, 2 (2003), 34–42.

[PH89] PERLIN K., HOFFERT E. M.: Hypertexture. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques - SIGGRAPH '89* (1989), pp. 253–262.

[PVH*03] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: The state of the art in flow visualisation: Feature extraction and tracking. *Comput. Graph. Forum 22*, 4 (2003), 775–792.

[RPS01] REINDERS F., POST F. H., SPOELDER H. J. W.: Visualization of time-dependent data with feature tracking and event detection. *The Visual Computer 17*, 1 (2001), 55–71.

[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 *Proceedings of IEEE Visualization 1999 Conference* (1999), pp. 371–377.

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

[SW97] SILVER D., WANG X.: Tracking and visualizing turbulent 3d features. *IEEE Transactions on Visualization and Computer Graphics 3*, 2 (June 1997), 129–141.

[TM05] TZENG F.-Y., MA K.-L.: Intelligent feature extraction and tracking for visualizing large-scale 4d flow simulations. In *Proceedings of the 2005 ACM/IEEE conference on Supercomputing* (2005), p. 6.

[WS03] WOODRING J., SHEN H.-W.: Chronovolumes: a direct rendering technique for visualizing time-varying data. In *Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume Graphics* (2003), pp. 27–34.

[WWS03] WOODRING J., WANG C., SHEN H.-W.: High dimensional direct rendering of time-varying volumetric data. In *Proceedings of IEEE Visualization 2003 Conference* (October 2003), pp. 417–424.

[YMC05] YOUNESY H., MÖLLER T., CARR H.: Visualization of time-varying volumetric data using differential time-histogram. In *Proceedings of the 2005 Eurographics/IEEE TVCG Workshop on Volume Graphics* (2005), pp. 21–29,224.