Temporal Styles for Time-Varying Volume Data

Jean-Paul Balabanian[†] Ivan Viola[†] Torsten Möller Eduard Gröller[†]
[†]University of Bergen Simon Fraser University Vienna University of Technology

Abstract

This paper introduces interaction mechanisms for conveying temporal characteristics of time-varying volume data based on temporal styles. We demonstrate the flexibility of the new concept through different temporal style transfer function types and we define a set of temporal compositors as operators on them. The data is rendered by a multi-volume GPU raycaster that does not require any grid alignment over the individual time-steps of our data nor a rectilinear grid structure. The paper presents the applicability of the new concept on different data sets from partial to full voxel alignment with rectilinear and curvilinear grid layout.

1. Introduction

Many areas of science, industry, and medicine are nowadays increasingly using time-varying volumetric data sets in their daily routine. Such data sets are usually discretized forms of real-world measurements or results of physical simulations. The nature and usage of time-varying data strongly depends on the given application domain.

A typical application area where time-varying data sets are studied on a daily basis is meteorology. Such data is usually organized on a 3D regular lattice with dozens of characteristic values per sample point. One time-step represents one moment in time and the overall data contains the development over time (e.g., one time-step per hour of a total of 48 time-steps) [12]. Especially for the exploration of hurricane behavior, these simulation results are studied to increase the understanding of how individual properties influence the global behavior of this weather phenomenon.

Probably one of the youngest domains where time-varying data sets have been acquired is marine research. New sea-surveillance technology based on acoustic echo is designed to enable sustainable harvesting of fish stocks and studying fish school behavior [2]. The time-varying data acquired from this technology are partially overlapping pyramids on not aligned curvilinear grids.

The time-varying data sets significantly differ in the number of time-steps from very few to hundreds. The data values might be scalar values or vectors. There might be several data values per single sample, data sets might overlap, or the data values can be organized on a variety of different grids. Despite these differences we can state that effective handling of time-varying volume data is a complex and challenging task. We differentiate between two basic challenges.

The first challenge relates to the computational complexity and memory requirements when processing the time-varying volumes. For example interactive rendering of mid-size volume data organized in a time series is difficult as only few time-steps can be loaded to the dedicated memory of high performance processing units (e.g., GPUs) at once.

The second challenge concerns how to effectively represent a temporal data set visually to enable a particular exploration task. In this paper we aim at addressing this later challenge, in particular how to interact with and how to effectively convey the temporal characteristics of the time-varying volumetric data set.

The focus of this paper is on the easy interaction with visually classified temporal data. The visual classification is carried out through temporal styles that create an intuitive way of condensing information from a series of volumes into a single image. We propose new interaction techniques for selection of visual representations for single image synthesis and animation sequences. Our visualization technique requires to process many volumetric time-steps at once. We propose a multi-volume GPU raycaster utilizing the latest graphics hardware capabilities that can perform ray compositing from multiple time-steps. The multi-volume ray caster features a volume overlap management that handles data with partial to full voxel alignment and defined on rectilinear and curvilinear grid layouts.

Condensing a time-series of bouncing super-ellipsoid data by using temporal styles is indicated in Figure 1. This figure illustrates the difference between rendering of time-steps separately and rendering them as one time space domain in which the image synthesis is carried out.

We provide a brief review on existing visualization approaches for time-varying data in the following Section 2 and a high-level description of our approach in Section 3.

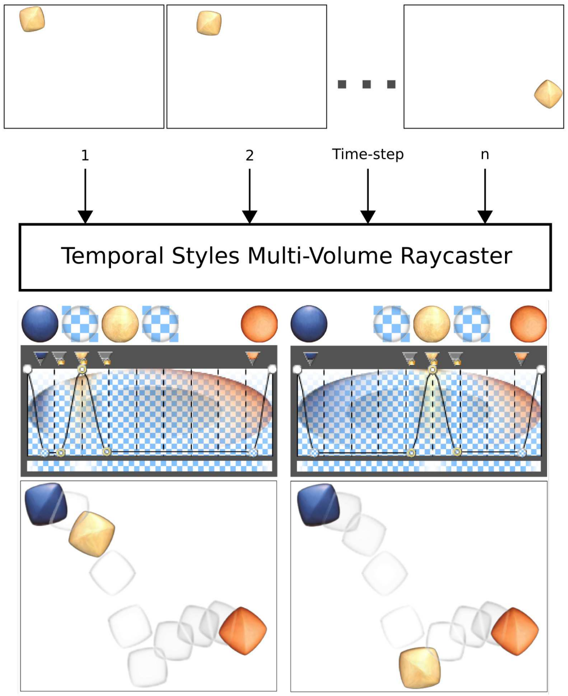

Figure 1. Condensing volumetric time-series using temporal transfer functions. The images show how changing the temporal style generates a new view of the temporal behaviour.

In Section 4 we describe the interaction metaphors that we designed for temporal styles and in Section 5 we give a detailed description of our proposed technique. Finally we present our results in Section 6 and conclude with Section 7.

2. Related Work

An often used depiction of volumetric data is visualization of a selected single time-step. Such a visualization can be effective to show temporally invariant characteristics. A straightforward approach for visualizing temporal characteristics of time-varying data is time series *playback* as a passive animation or with interactive viewpoint changes. Such visualizations can give a notion of structured movement, however they will be less useful for most precise analytic and quantitative tasks. In the case when the time series consists of a small number of time-steps, it is possible to use the *fanning in time* approach that shows all time-steps next to each other [8]. However these approaches do not specifically address visual emphasis of temporal characteristics in a time varying data set.

In medical visualization time-varying data has for example been used to plot the temporal development of contrasted blood perfusion as a one dimensional polyline [5, 7]. A similar concept of interactive visual analysis and exploration has been used in data originating from the automotive industry [6]. Other techniques let the user link multi-variate time-varying data with temporal histograms to simplify the exploration and design of traditional transfer functions for this type of data [1]. In the past, techniques have been proposed on automatic generation of transfer-functions by incorporating statistical analysis and coherency [9, 16]. All the above techniques that use automatic or interactive data analysis for exploration and transfer function design employ as visualization single-time-step renderers. This means that information on temporal characteristics is not represented in a single image.

Visualizations mostly related to our approach attempt to visually represent the temporal characteristics of the time-series directly in the physical space. They condense the visual representation so the visual overload is reduced. Some approaches have been inspired by illustration, cartoon or comic drawing techniques where helper graphics like arrows and lines indicate temporal developments such as movement or size changes [10]. The final image consists of several individual time-steps and the helper graphics convey the temporal information. Another approach that we share temporal compositors with, has been inspired by the idea of chronophotography [14]. This technique integrates a time-varying 3D data set into a single 3D volume, called a chronovolume, using different integration functions. When a change of the integration algorithm is requested by the user, the chronovolume has to be recalculated. In contrast to this method the proposed concept of temporal style transfer functions allows interactive visual feedback during the design of the visual representation. Chronovolumes have later been generalized [15]. The technique creates 3D volumes of the time-varying data by slicing the 4D space spanned by the data with hyperplanes and integration operators. The main difference between our rendering technique and theirs, is that we have access to all volumes in real-time. This leads to greater flexibility in interactive design of compositing operators. The visual result of a compositing operator can be easily changed by newly proposed interaction techniques which are the main focus of this paper. The difference between our new and their approach is analogous to the difference between pre- and post classification.

Previous work on visualization of 3D sonar data [3] modifies the standard volume ray-caster to support rendering of parameterized curvilinear volumetric time-steps. The framework gives the user the opportunity to visualize and perform semi-automatic segmentation of fish-schools. The segmentation mask from a single time-step is propagated to neighboring time-steps as the initial segmentation mask. It is then automatically adjusted to the new time-step. The temporal aspect of the data, however, can only be visualized in a single time-step at a time.

The concept of style transfer functions [4] describes how images of lit spheres can be used to apply color and texture to volume rendering.

3. Temporal Compositing of Time-Varying Data

The basic idea of temporal style specification stems from the challenge of showing the temporal development of features in time-varying volume data within a single image. Some parallels to this can be drawn from traditional photography where the concept of multiple exposures is well known. The technique creates images that show, for example, where objects move from exposure to exposure in a single image. In Figure 1 we can see the start and stop positions of the bouncing object in addition to the traversed path. In the Figure one can also observe that we are able to change the visual representation of the photographed object, something which normal photography is incapable of doing. Simply by interacting with a widget a user is able to reproduce many of the results generated by long- and multiple exposure techniques. Generating several images where the exposure changes it is possible to create animations that higlights the change.

Traditional volume raycasting of time-varying data creates images that represent individual time-steps without any temporal information. Even if images from several time-steps are created it is still difficult to compare them spatially and temporally. It is also difficult to identify areas where there is a temporal change. Our aim is to condense several time-steps into one image.

A transfer function is a function that takes density values and converts them to a visual representation. This conversion is usually $R \rightarrow R^4$ and results in a color with opacity.

Style transfer functions [4] are transfer functions that in addition to opacity also define lit-sphere styles for densities instead of colors. Using a density value an opacity and a style is calculated by interpolation. A color with shading is retrieved from a texture of a lit-sphere using additionally the density gradient vector.

The temporal style at a position p is a color and opacity derived from n density values, this can be described as a mapping $R^n \rightarrow R^4$. Our task is to take a density vector in R^n, where n is the number of spatially overlapping voxels from distinct time-steps, and assign a visual representation to it. Our visualization framework calculates values in the temporal domain using a Temporal Compositor (TC) depicted as central blue box in the schematic description of our pipeline in Figure 2. This module processes the spatially overlapping voxels and enables operations on the temporal data. These include temporal average, temporal change, or temporal gradient calculations. The TC takes the n values and converts them to a so-called temporal characteristic ($R^n \rightarrow R$). The result of this conversion using temporal operations can then be applied to a Temporal Style Transfer Function (TSTF), which generates a visual representation $R \rightarrow R^4$.

A temporal style is the visual representation that is generated by the temporal compositor. Depending on the TC the visual representation generated can solely be based on the TSTF or a modulation between the style transfer function and the TSTF. This is usually dependent on the type of information the TC is conveying. In Section 5.3 we describe different TCs that we have implemented.

Our system allows the use of both partially and fully overlapping volumes and volumes defined on regular or curvilinear grids. In Section 5 we give a detailed description of our framework and a detailed explanation of TCs and TSTFs.

4. User Interaction with Temporal Styles

The user has several ways of interacting with the time-varying data. First of all the user can define a time-invariant style transfer function that applies to all volumes and basically defines the visual representation of the content of the volumes. This is similar to single time-step volume rendering. The user can also interact with a temporal style transfer function which defines the visual representation that the currently selected temporal compositor will use to produce its results. We have implemented two different TSTFs. The first one lets the user supply a modulation for the visual representation for every time-step. The TSTF is divided into sections equal to the number of time-steps. Figure 1 illustrates this metaphore. On the left we have selected a blue style for the first time-step and a light yellow style for the third time-step. In between we have specified a style that only shows the contour. We have also implemented another type where the TSTF represents temporal gradient magnitude and defines a range $[0, 1]$. The value calculated by the TC is then used directly to retrieve a visual representation from the TSTF.

A new interaction type is how the user works with the TSTF. The TSTF contains several styles and nodes that define color and opacity. The styles and nodes can be grouped together and all of the members of the group can be moved simultaneously. This technique is especially applicable for the time-step index TSTF 5.4 where moving a group would be analogous to moving the focus in time. Figure 1 shows this concept on the bouncing super-ellipsoid data set and in Figure 5 the concept has been applied to the hurricane Isabel data set. The styles and nodes that are grouped together are indicated with yellow circles or padlocks.

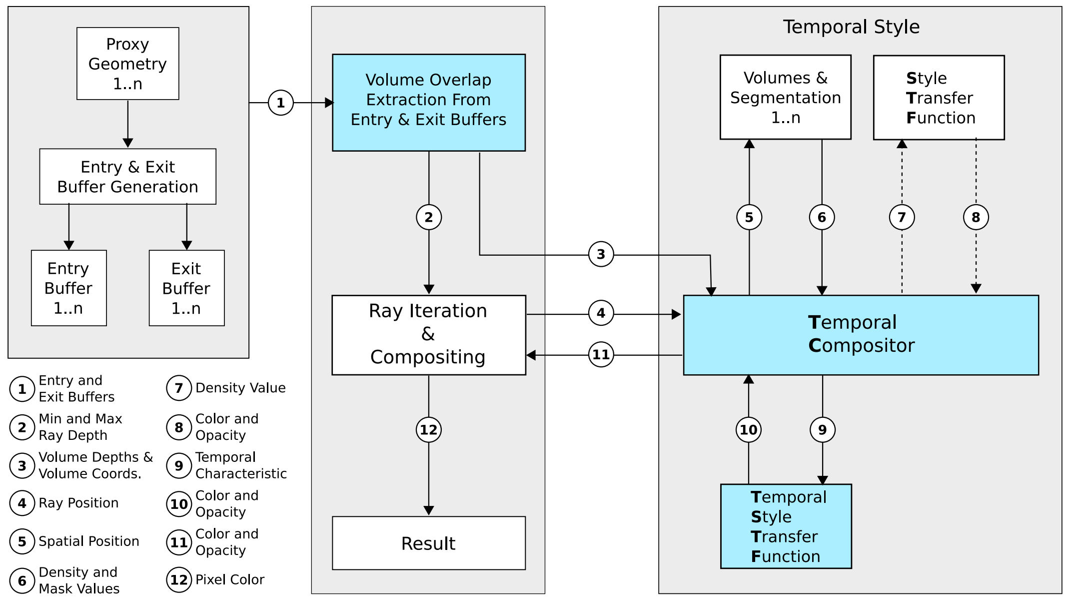

Figure 2. Schematic overview of the TSTF based multi-volume raycaster. Blue indicates our extensions to the standard raycaster. n is the number of volumes. The encircled numbers indicate the processing order. The dashed lines are an optional part of the pipeline.

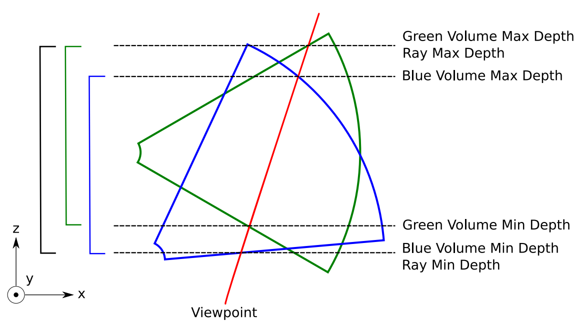

Figure 3. Overview of the minimum and maximum depth values.

5. TSTF Based Multi-Volume Raycaster

The temporal style transfer function based multi-volume raycaster can be realized by extending a standard single volume raycaster. The extensions include the temporal compositor (TC) which handles spatially overlapping voxels and the TSTF that provides a visual representation for the overlap. Figure 2 gives an overview of the system that we have developed. The first step, i.e., the entry and exit buffer generation, is similar to a single volume raycaster, as described by Krüger and Westerman [11]. The difference is that it is repeated for all the volumes that need to be included. To encompass all the steps necessary for multi-volume rendering of time-varying data, various additional steps have been added. First of all, overlapping volumes along a ray have to be identified. In the schematic depiction of our framework (Figure 2) this is done in the module specified by the blue box in the upper left corner. The central issue that has to be addressed is how to represent the temporal information in an intuitive way. In Figure 2 our solution is indicated by the two blue boxes in the lower right corner named Temporal Compositor and Temporal Style Transfer Function. In this part of the process we analyze the temporal behavior and assign a visual representation based on temporal compositing operators and temporal style transfer functions.

5.1. Volume Overlap Extraction

To extract the information about volume overlap we generate entry and exit buffer pairs for every time-step. If the position and orientation of the proxy geometry is static for all time-steps only one entry and exit buffer needs to be generated. This process is indicated by the leftmost box in Figure 2. The volume raycasting uses the entry and exit buffer technique for the GPU as previously described [11].

Every pixel corresponds to a single viewing ray. Corresponding pixels in the entry and exit buffers for the various volumes belong to the same ray. This means that the depth information available in the entry and exit buffers describes the amount of overlap for every volume along a ray. By taking the minimum and maximum depths from all vol-

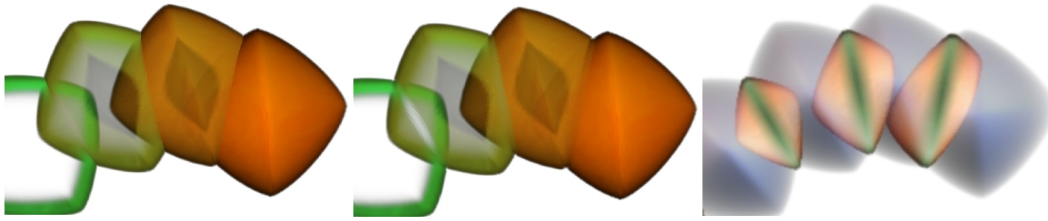

Figure 4. Left: Temporal maximum intensity projection, Center: Temporal average, Right: Temporal maximum density change

umes we determine where a ray should start and end. This is illustrated in Figure 3. The process of extracting this information is represented in the pipeline by the blue box in the upper left corner in Figure 2. The Volume Overlap Extraction iterates through every pixel in an entry buffer and checks if the depth is less than the maximum depth of the depth buffer. If this condition holds the volume associated with this entry and exit buffer is intersected with the ray at this position. Every volume is checked and a minimum depth and maximum depth is stored, which is called the ray depth. The ray depth is forwarded to the ray iteration process (path 2 in Figure 2). The volume data coordinates and volume depth values for each volume are forwarded to the temporal compositor (path 3).

5.2. Ray Iteration & Compositing

The ray iteration process takes the ray depth values and, using a suitable step size, samples the ray in regular intervals. For each step the current ray position is forwarded to the Temporal Compositor through path 4 and a temporal style is returned through path 11. The color values are composited, front-to-back along the ray. Using early ray termination, the ray is stopped if the composite opacity is close to 100%.

5.3. Temporal Compositor

The temporal compositor (TC) takes every ray position and checks if it is within the range of the particular volume depth. If the ray position is inside a volume then the sample value, and optionally the segmentation mask value, is included in the temporal compositing (paths 5 and 6). Optionally the TC can also fetch a color and opacity from the time-invariant style transfer function (paths 7 and 8). At this stage the TC has the following values available for multiple volumes: sample value and segmentation mask value, and color and opacity from the style transfer function (STF). The task now is to combine these values in a way that leads to insight into the data. We consider the following strategies:

Temporal maximum intensity projection TC We use the value from the time-step with the highest opacity.

Temporal average TC We average the colors and opacities from all non transparent overlapping time-steps.

Temporal maximum density change TC We calculate the difference between the minimum and maximum density value of all time-steps.

These operators straightforwardly process the underlying data values. However, we support their applicability on pre-classified data by opacity values using the STF as well. This allows for rapid windowing of interesting data values prior to the temporal compositing stage. The opacity defined by the user in the STF represents visual significance for non-temporal characteristics. It may highlight information that the user is most interested in. The resulting values from these operators, i.e., temporal characteristics, are then applied to a TSTF (paths 9 and 10 in Figure 2).

Figure 4 shows results of using the different TCs on the bouncing super-ellipsoid data set. The image on the left shows the temporal maximum intensity projection TC. The image in the center shows the temporal average TC. The results from these two operators are very similar in this case. The visual difference is that the temporal average has smooth transitions between volumes while the temporal maximum intensity projection has hard edges. The image on the right is rendered with the temporal maximum density change TC and indicates the change from time-step to time-step in the overlapping regions. The green areas are where the change is close to zero while the orange areas depict larger changes.

5.4. Temporal Style Transfer Function

TSTFs are functions that assign a visual representation to results calculated in the TC. We have developed two different TSTFs that correspond to the different outputs generated by the TC. These are:

Time-step index TSTF This TSTF lets the user define time-step visibility and visual representation. The TC provides an index for a single time-step and requests the color and opacity that the TSTF defines for this time-step. Images that use time-step index based TSTF can be seen in Figures 1, 4 (left and center), 5, and 6 (center).

Gradient magnitude TSTF The calculated value received from the TC is the gradient magnitude that is scaled to the range $[0, 1]$. The TSTF represents color and opacity from a style dependent on gradient magnitude. Images that use gradient magnitude based TSTF can be seen in Figures 4 (right) and 6 (left and right).

5.5. Temporal Compositing

The last step for the TC is to take the color and opacity obtained from the TSTF, perform a final processing and then return them to the Ray Iteration & Compositing process (path 11 in Figure 2). The final processing usually consists of deciding whether to return the STF value, the TSTF value or the modulation of those values. If the data contains segmentation information which defines regions of interest the temporal compositing can take this into consideration and only apply the TSTF to the masked region. Then it applies the time-invariant style transfer function for unmasked areas. This technique of highlighting the region of interest has been used on all images showing sonar data in this paper. See Figure 6 (center and right) for examples.

6. Results

We have applied our temporal styles concept to three different time-varying data sets. We have applied the temporal average TC and temporal maximum density change TC to the data sets and defined time-invariant style transfer functions and temporal style transfer functions.

The bouncing super-ellipsoid has, throughout the paper, been used to illustrate discussed concepts. The volume data has a resolution of 64^3 and consists of 10 time-steps. The time-steps are partially overlapping and the density values have been calculated from an implicit super-ellipsoid function stored in a rectilinear grid. The bouncing motion has been simulated using a physics library. We also applied the temporal maximum intensity projection TC to this dataset. In the bottom part of Figure 1 we have chosen a blue style for the first time-step and an orange style for the last time-step. In the renderings we can see that the super-ellipsoid at the end-points have fixed positions. We would like to focus on the location of time-step 3. This is achieved by setting the opacity to opaque and a light yellow style for time-step 3. Additionally we set the immediate surroundings of the focused time-step to a low opacity and the silhouette style. The result of this can be seen on the left side as a series of semi transparent objects and an opaque super-ellipsoid at time-step 3. Grouping together nodes and styles of the focused time-step and moving the group to time-step 6 changes the resulting image. Now the focused super-ellipsoid is located at time-step 6 and the semi transparent super-ellipsoids create paths backwards and forwards in time.

We have also applied the temporal maximum density change TC to the bouncing super-ellipsoid data set (Figure 4). Regions with a low density change have been assigned a green style that turns into an orange style when changes increase. Parts of the super-ellipsoids that do not overlap do not have a defined change and we apply the time-invariant STF.

The hurricane Isabel is a fully overlapping rectilinear data set. We have resampled the *QCLOUD* volume data attribute to the resolution of $125 \times 125 \times 25$ and selected 10 of the original 48 time-steps uniformly distributed over the time-series. In Figure 5 we have applied similar temporal visual settings as to the bouncing super-ellipsoid data set. In the left image we can see where the hurricane starts (the blue region) and where it ends (the red region). It is also easy to see the path of the hurricane as a suppressed contour rendering is set for all time steps and locations of the hurricane. The lower left part of the left image shows the propagating hurricane front for the current time-step (in green). In the center image we have moved the focus to 20 hours from the beginning of the simulated sequence and in the last image it is at 40 hours.

Applying the temporal maximum density change TC to the hurricane Isabel data set we get the image on the left in Figure 6. From this image we can immediately recognize that the areas of highest change are in the hurricane eye towards the direction of the movement.

The final data set that we have applied our technique on is the sonar data. This data is a partially overlapping curvilinear data set. Each volume has a pyramid shape with a resolution of $25 \times 20 \times 1319$. The sequence consists of 75 time-steps (described by Balabanian et al. [3]). We have selected a sequence of 10 time-steps that contain fish-school data and a per time-step segmentation of that school. In the center image of Figure 6 we have set an orange style for the first time-step and a blue silhouette style for the last time-step.

We have also applied the temporal maximum density change TC to the sonar data. The result of this is the right image in Figure 6. The blue style indicates areas of low change and the red style shows high change areas. The center of the school has the highest change which seems reasonable since the density of a school decreases at the edges.

Table 1 shows the performance of the different TC operators. We rendered to a viewport with dimensions 512×512.

By using 3D textures for entry and exit buffers instead of several 2D textures and merging several volumes into a single 3D texture the number of time-steps that we are able to process is not limited to the number of available texture

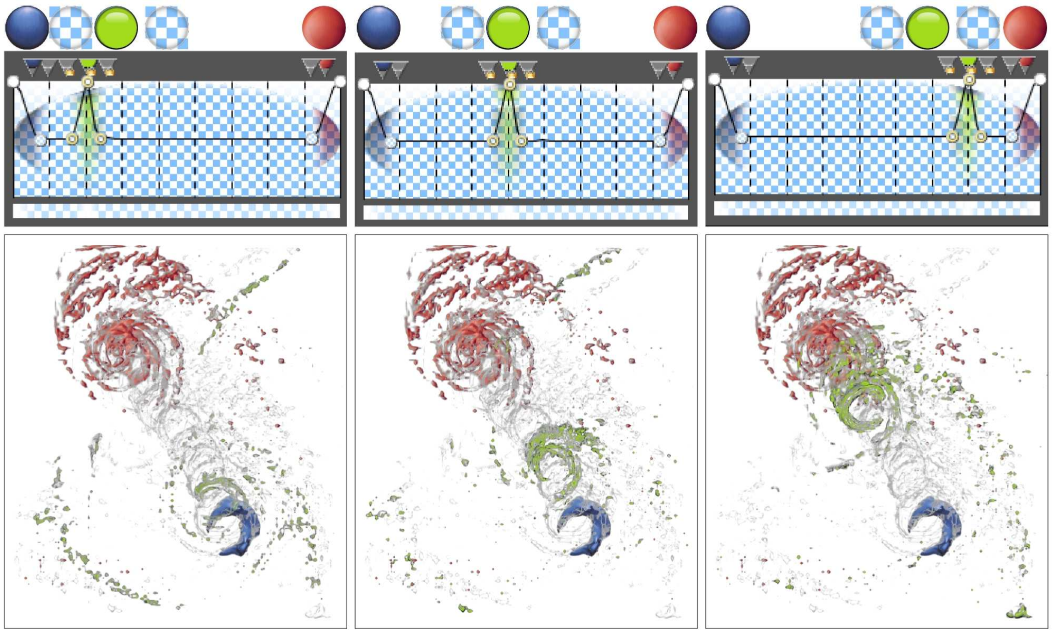

Figure 5. Time-step index TSTF on the hurricane Isabel data set during the interaction. Left: time-step 3 highlighted with a green style, Center: highlight moved to time-step 5, Right: highlight moved to time-step 8.

Temporal Compositor	Super-Ellipsoid	Isabel	Sonar
Temporal Maximum Intensity Projection	0.20	0.25	2.5
Temporal Average	0.25	0.30	2.1
Temporal Maximum Density Change	0.25	0.25	2.2

Table 1. Rendering times in seconds for a viewport of dimensions 512×512**.**

units on the graphics card. Our limitation is the amount of memory available on the graphics card and the maximum resolution of 3D textures (2048^3 on nVidia 8800 GTS).

7. Summary and Conclusions

We have developed a framework for interactive specification of visual parameters for visualization of time-varying data sets sampled on regular and curvilinear volumetric grids with partial overlap. The visualization is carried-out through condensing several time-steps into a single image. To generate this type of visual depiction we used temporal compositors that created a temporal characteristic for the spatially overlapping voxels. We proposed temporal styles to define the visual representation of the temporal characteristics. The resulting images help the user identifying regions of interest in addition to simplifying the interaction by dividing temporal analysis into two components. First a temporal operator, TC, describes a temporal characteristic. Second a temporal style transfer function, TSTF, is designed that highlights only the regions that interest the user.

It is our experience that designing useful temporal styles is just as important and challenging as designing traditional transfer functions. Therefore it is to be expected that a good visual quality of the resulting images will be achieved only after careful design of both, the traditional transfer function as well as the temporal style transfer function. We experienced that the concept of styles significantly simplifies the specification of a useful visual depiction as compared to the traditional transfer functions concept.

The lit spheres concept [13] and consequently style transfer functions have known limitations for shading. Using several differently shaded style spheres in one scene may easily result in inconsistent illumination if the style spheres do not share the same implicitly defined light position. This obviously holds when applying the styles to the temporal domain, however, we have not experienced any undesired visual artifacts related to this limitation. This indicates the known fact that the human observer is not very sensitive to an inconsistent light setup as long as the shape of structures is clearly discernible.

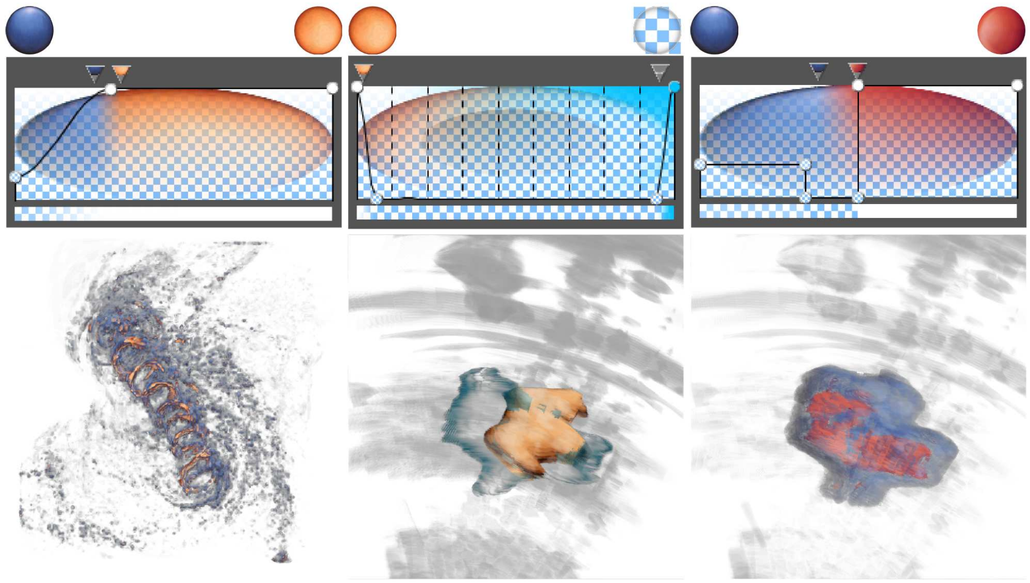

Figure 6. Left: Isabel data set with a maximum density change TSTF, Center: Time-step index TSTF on sonar data, Right: Maximum density change TSTF on sonar data.

Acknowledgments

We thank Daniel Patel for insightful comments and Stefan Bruckner for discussions on style transfer functions. The work has been partially carried out within the exvisation project (FWF grant no. P18322).

References

[1] H. Akiba and K.-L. Ma. A tri-space visualization interface for analyzing time-varying multivariate volume data. In *Proceedings of EuroVis 2007*, pages 115–122, 2007.

[2] L. Andersen, S. Berg, O. Gammelsæter, and E. Lunde. New scientific multibeam systems (me70 and ms70) for fishery research applications. *Journal of the Acoustical Society of America*, 120(5):3017, 2006.

[3] J.-P. Balabanian, I. Viola, E. Ona, R. Patel, and M. E. Gröller. Sonar explorer: A new tool for visualization of fish schools from 3d sonar data. In *Proceedings of EuroVis 2007*, pages 155–162. IEEE, 2007.

[4] S. Bruckner and M. E. Gröller. Style transfer functions for illustrative volume rendering. *Computer Graphics Forum*, 26:715–724, 2007.

[5] E. Coto, S. Grimm, S. Bruckner, M. E. Gröller, A. Kanitsar, and O. Rodriguez. Mammoexplorer: An advanced CAD application for breast DCE-MRI. In *Proceedings of Vision, Modeling, and Visualization'05*, pages 91–98, 2005.

[6] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation. In *Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization VisSym*, pages 239–248, 2003.

[7] Z. Fang, T. Möller, G. Hamarneh, and A. Celler. Visualization and exploration of spatio-temporal medical image data sets. In *Graphics Interface 2007*, pages 281–288, 2007.

[8] S. Grimm, S. Bruckner, A. Kanitsar, and E. Gröller. Flexible direct multi-volume rendering in interactive scenes. In *Proceedings of Vision, Modeling, and Visualization'04*, pages 379–386, 2004.

[9] T. Jankun-Kelly and K.-L. Ma. A study of transfer function generation for time-varying volume data. In *Proceedings of Volume Graphics '01*, pages 51–68, 2001.

[10] A. Joshi and P. Rheingans. Illustration-inspired techniques for visualizing time-varying data. In *Proceedings of IEEE Visualization '05*, pages 86–93, 2005.

[11] J. Krüger and R. Westermann. Acceleration techniques for GPU-based volume rendering. In *Proceedings of IEEE Visualization'03*, pages 287–292, 2003.

[12] National center for atmospheric research. http://www.ucar.edu/, 2007.

[13] P.-P. J. Sloan, W. Martin, A. Gooch, and B. Gooch. The lit sphere: a model for capturing NPR shading from art. In *Graphics interface*, pages 143–150, 2001.

[14] J. Woodring and H.-W. Shen. Chronovolumes: a direct rendering technique for visualizing time-varying data. In *Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics*, pages 27–34, 2003.

[15] J. Woodring, C. Wang, and H.-W. Shen. High dimensional direct rendering of time-varying volumetric data. In *Proceedings of IEEE Visualization'03*, pages 417–424, 2003.

[16] H. Younesy, T. Möller, and H. Carr. Visualization of time-varying volumetric data using differential time-histogram table. In *Workshop on Volume Graphics 2005*, pages 21–29, 2005.