

Depicting Time Evolving Flow with Illustrative Visualization Techniques

Wei-Hsien Hsu¹, Jianqiang Mei^{1,2}, Carlos D. Correa¹, and Kwan-Liu Ma¹

¹ University of California, Davis,
One Shields Avenue, Davis CA 95616, USA,
whhsu@ucdavis.edu

² Tianjin University, Tianjin
92, Weijin Road, Nankai District, Tianjin 300072, P.R. China

Abstract. Visualization has become an indispensable tool for scientists to extract knowledge from large amounts of data and convey that knowledge to others. Visualization may be exploratory or illustrative. Exploratory visualization generally provides multiple views of the data at different levels of abstraction and should be highly interactive, whereas illustrative visualization is often made offline at high quality with sufficient knowledge about the data and features of interest. Techniques used by professional illustrators may be borrowed to enhance the clarity and aesthetics of the visualization. This paper presents a set of visualization techniques for presenting the evolution of 3D flow. While the spatial features of the data is rendered in 3D space, the temporal behaviors of the flow are depicted using image-based methods. We demonstrate visualization results generated using three data sets obtained from simulations.

Key words: volume visualization, time-varying data visualization, image processing, evolution drawing, non-photorealistic rendering

1 Introduction

Visualization transforms numbers into vivid pictures that capture the essence of the data generated in the study of a complex problem in physical sciences, social science, engineering, biology and medicine. Visualization if properly made can help a scientist to see what she expects to see, possibly see previously unknown phenomena, and communicate with others her findings more effectively. Advances in visualization over the years have been made possible largely due to the incorporation of concepts in art and visual perception into visualization design, rapid development of graphics hardware technology, and coupling of advanced display technology and interaction techniques into an overall visualization solution. A visualization system should support its users to make visualizations according to the purpose of visualization, be it for exploration or illustration.

This paper presents a set of techniques for crafting scientific visualization to be illustrative, expressive and aesthetic. We show how to achieve our goals by combining advanced rendering methods, image processing techniques, together

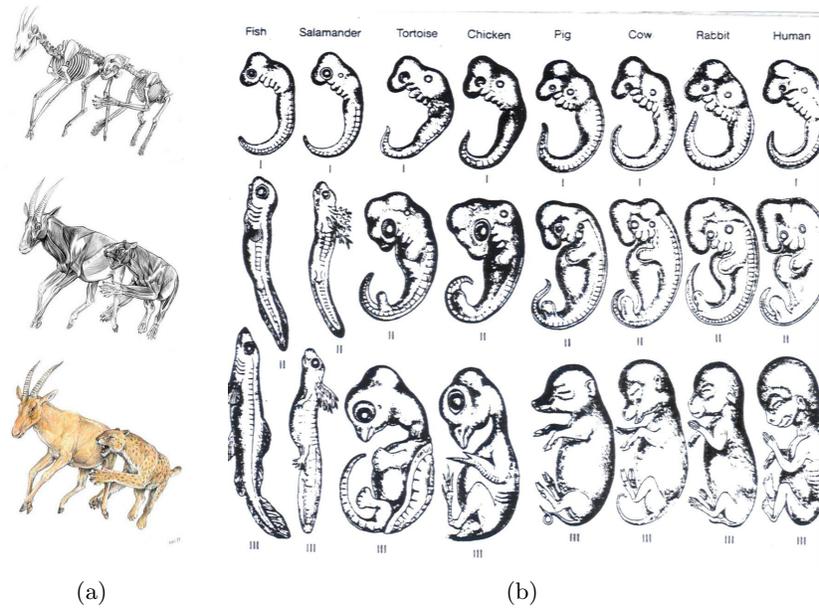


Fig. 1. Examples of professionally made illustrations (a) Depiction of *Metailurus* major based on the proper body proportions. Preying on the Miocene antelope, *Tragoportax*, by Zoologists and artist Velizar Simeonovski. (b) A comparison of vertebrate embryos (from *What Evolution Is* by E. Mayr[11]).

with styles, compositions, and layouts commonly found in art. Skilled artists are able to apply pigments using a brush or pen in a manner that creates meaningful abstractions of reality. On a piece of paper, they can unambiguously convey three-dimensional shape, express spatial relationship, and give a sense of movement. Figure 1(a) uses three pictures of the same pose to depict the anatomy of an extinct genus of false saber-toothed cats called *Metailurus*, illustrated by Zoologists and artist Velizar Simeonovski. Figure 1(b) shows a sequence of vertebrate embryo images for the comparison of different species at different time periods [11]. Like an artist, we can create visualization emphasizing features that might not otherwise be visible, while de-emphasizing those features of less importance. With proper artistic composition and arrangement, the resulting visualization is often more expressive and compelling. In this paper, we demonstrate techniques operating in either 3D volume data space or 2D image space to create desirable visualization of time-varying 3D flow fields.

2 Related Work

Time-varying volume data visualization leads to many unique challenges from data encoding to feature extraction and volume rendering. Ma has given a survey of early work on time-varying volume data visualization [10].

Researchers have been paying attention to the approaches to visually represent the time evolving phenomena in the physical space. Attempts have been made to visualize time evolving data without animation. Inspired by the idea of chrono-photography, Woodring and Shen [19] integrated a time-varying 3D dataset into a single 3D volume, called a chronovolume, using different integration functions. In [20], Woodring et al. considered time-varying dataset as a four-dimensional data field and applied high dimensional projection techniques to create an image hyperplane. Lu and Shen [9] introduced a method by first reducing time steps and only visualizing the essential information in a storyboard layout. Balabanian et al. [1] used temporal style transfer functions to visualize temporal characteristics into a single image.

Depicting time evolving data by focusing on a small number of features in the data is becoming increasingly popular especially when the data size is large. The feature-based approach requires multiple steps, including feature extraction, feature tracking, and the visual representation. Wang and Silver proposed several volume tracking methods for regular grid [15] and unstructured grid [14] by utilizing the spatial overlap among corresponding isosurfaces in consecutive time steps. Muelder and Ma [12] introduced a prediction-correction method to make the best guess of the feature region in the subsequent time step to achieve a better tracking result. Instead of explicitly tracking features, Lee and Shen [7] used Time Activity Curves to describe time-varying features and visualized the vector field of the motion tendency of the feature. To draw time-varying features, Post et al. [13] used icons as a symbolic representation of essential characteristics of features to visualize time-dependent data. Joshi and Rheingans [6] used the techniques inspired by conventional illustration such as speedlines and flow ribbons to depict the motion of the feature.

Non-photorealistic rendering (NPR) techniques as used in [6] have been shown great promise in conveying information. In contrast to the traditional photorealistic rendering, illustrative rendering aims to mimic various styles of human paintings and artistic illustrations to highlight hidden structures where a specific meaning has to be conveyed to viewers. Considerable work has been done in making stylish artistic images from both 2D images [5, 18] and 3D objects [4, 8]. NPR techniques have also been applied to volume visualization. Stoppel and Ma [16] introduced a set of feature enhancement techniques based on NPR to generate more perceptually effective visualization of multivariate and time-varying volume data. Svakhine et al. [17] used a combination of illustrative and photorealistic rendering techniques to visualize flows and volumes.

3 Object-based Approach

Our approach to 3D time-varying data visualization can be performed on either object data space, i.e. the 3D domain of the data, or in image-space, i.e., using the 2D images generated by 3D volume rendering. In this section, we describe our technique for illustrating time-evolving flow in 3D data space. To obtain a good 3D volume rendering of time-varying data, we follow a series of steps. First, we find a suitable transfer function that maps the data values to color and opacity. To follow the evolution of a feature, we employ a feature extraction and tracking method [12]. Finally, we render the extracted features using illustration-inspired techniques.

3.1 Transfer function specification

Transfer functions are used to map data values to rendering characteristics, such as opacity and color. Before assigning data values to color and opacity, we must first normalize the entire collection of time steps, since the data range for each time step might be different. This normalization is necessary to ensure that the regions highlighted with a transfer function are consistent across the different time steps. We then use a 1D transfer function based on the normalized scalar data value. Transfer functions let users select regions of interest and highlight intervals in the data domain, such as regions of high vorticity.

3.2 Feature Extraction and Tracking

Feature Extraction and Tracking(FET) lets us extract meaningful features and follow their evolution through time. The FET technique we used in this paper predicts each feature’s region and then adjusts the surface of this region to extract the correct feature. It utilizes information from previous time steps to predict a feature’s region in the current time step, then correcting the predicted region to extract the feature’s actual boundaries via region growing and shrinking. Users can select an individual feature, and the corresponding features over time would be extracted and tracked. If no special features are selected, the system automatically extracts and tracks all the features.

3.3 Spatio-Temporal Fade-in and Contour Rendering

Once we have extracted different features, we render them in a way to convey their evolution through time. To this purpose, we use fade-in effects. The oldest time step is depicted using only gray levels, while the newest time step is rendered with color. To convey a smooth transition, we use linear blending between these two extremes for intermediate time steps. The idea of this transition is to convey a natural direction of time, often found in art forms. Older instants of time of a moving object appear blurred whereas the current instant is crisp and clear. Since we want to preserve the rich detail of each time step, we choose

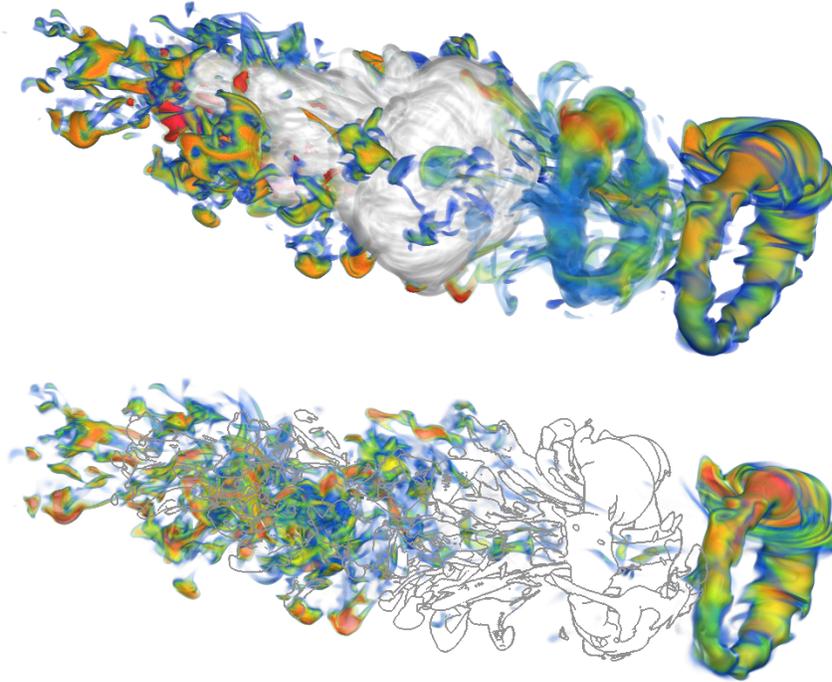


Fig. 2. Top: The separation process of the ring structure and its left-to-right motion is presented. There are three time steps in the image, the first time step is rendered in gray-color, while the last time step is full volume rendering, and the middle one is transparent color. Bottom: The contour line drawing depicts the exact position of the smoke ring in earlier time step more clearly.

to use less saturated colors to achieve the same effect. In addition, we compute contours based on the 3D depth of features. Contours are useful to simulate line drawing, useful when we want to preserve the overall shape of features with a minimal impact in the image. Combined with contour rendering, the motion of interesting features and their evolution can be conveyed and easily obtained by users. Furthermore, interactive tools to modify the color and opacity of each time-varying feature lets users create any desired fading effect to emphasize earlier or more recent stages of evolving flow.

3.4 Results

Our technique has been implemented and tested on several datasets. We show here an argon bubble data set, which is $640 \times 256 \times 256$, and the turbulent vortex data set (used by D. Silver [15]), which is $128 \times 128 \times 128$ voxels. Both datasets are time-varying scalar volumes. For the bubble data set, the ring structure is what we want to capture, while for the turbulent vortex data set, the features

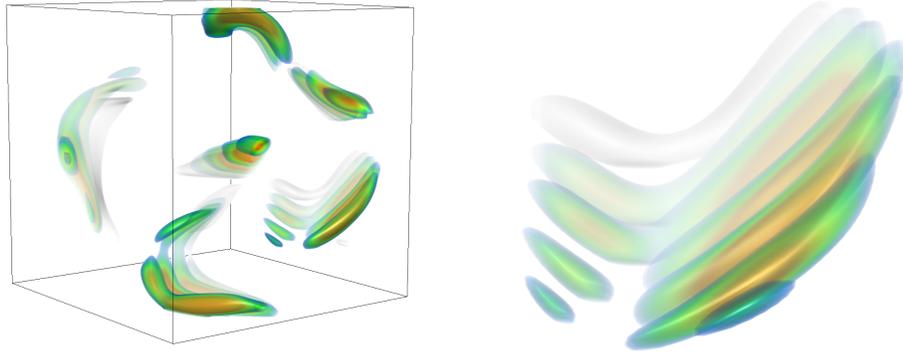


Fig. 3. (a) Evolution of five features in the original volume from vortex data set. Time-varying behavior of the volume could be obtained. (b) From birth to death, one single feature from vortex data set. As time progresses, more transfer function colors have been used to render the feature, and the evolution through its lifetime could be captured at all.

of interest we focus on are regions of high vorticity. All tests were run on a Dual-Core of a Mac Pro with a 2 x 2.66GHz Intel Xeon processor and 6GB RAM.

Figure 2 is an example created with the argon bubble data set. This data set shows the result of the simulation of a shockwave interacting with a bubble of argon gas surrounded by air creating a swirling torus-shaped "smoke ring" along with smaller turbulence structures. By looking at how the bubble deforms, scientists can understand how the air shockwave interacts with the argon bubble and generates vorticity, in an effort to model hyperbolic conservation laws. As we can see in Figure 2(top), the separation process of the ring structure and the left-to-right motion of the interested region is presented. As time progresses, gray-scale color becomes more and more transparent, meanwhile, the chromatic color becomes more apparent. Finally, a full-colored volume rendering is shown to depict the last time step of the data. Because the different time steps overlap in 3D space, obtaining a clear view of the smoke ring may be difficult. Figure 2(bottom) uses a line drawing style for one of the time steps to show the exact position of the smoke ring in relation to the other stages.

Figure 3(a) and figure 3(b) are examples created with a Computational Fluid Dynamics(CFD) simulation with turbulent vortex structures. The vortex structures are line-type features with strongly curved, and tube-like shape. The relationship between different features at different times could be easily delivered through Figure 3(a), which contains six features with five time steps, from frame 1 to frame 17, 4 frames interval. One special feature, from birth to death, has been shown in Figure 3(b), the spatio-temporal changing of the feature through

its life can be obtained from the result, which could also convey a visual impact to the users.

4 Image-based Approach

Some visual effects which are rather difficult to implement in 3D volume rendering can be easily done in image space. An obvious example is the layout of the temporal depiction. It is not always true the time-varying volume data have spatial translation over time. Sometimes its temporal characteristic is the shape or the density. In such a case, rendering several time steps into a single volume becomes too crowded to effectively show the evolution of the data. Instead, we use a combination of image processing techniques to create a time evolving flow depiction entirely in image-space.

The overall process of our image-based approach is shown in Figure 4. The input is two sets of sequences of the volume rendering snapshots and the contour images extracted in volume space. Each snapshot is first sent to an edge detector. We used these images and the edge maps to obtain line-drawing renderings. Both the original frames and the edge maps are then sent to a GPU shader, which applies illustrative rendering such as toon-shading, contour enhancement and sharpening. Finally, the final image is then composed according to a specified 2D layout.

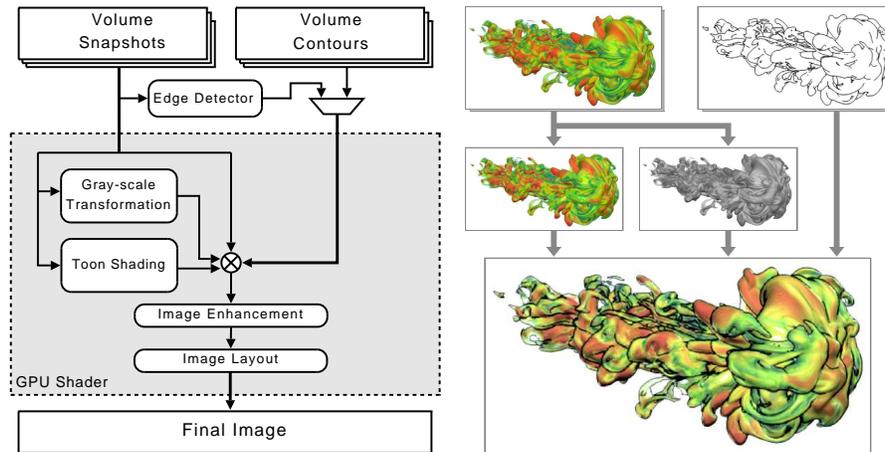


Fig. 4. The data flow of the image-based approach. The snapshots and the contours that are generated from volume rendering are sent to the system. Additional edge maps, gray-scale luminances and toon colors are computed to compose a compact depiction.

4.1 Edge Detection

Edge detection attempts to localize significant variations of the gray level image in the hopes of finding meaningful object boundaries and contours [21]. However, edge detection is not a trivial task due to the similarity of the image content, the contrast of the image, and noise. In this paper, we use the Canny edge detector [2], for its robustness and quality. The output of this edge detector is an image that highlights the most important edges in a flow that can be used to enhance the original images obtained from volume rendering. Since the original images are obtained from volume rendered images, there is not much noise when compared to natural images, which makes this approach appropriate for simulating a line-drawing rendering style.

4.2 Toon-Shading

Toon-shading is a rendering technique which aims to mimic the style of a comic book or cartoon and makes the computer graphics appear to be hand-drawn [4]. To achieve the same effect from a 2D image, we simply quantize the color of images at a number of levels as : $C^* = \text{Round}[C * t]/t$, where C^* is the resulting toon color, C is the original pixel color in the range of $[0, 1]$ (for each red, green and blue components), and t is the quantization level.

4.3 Color Blending and Enhancement

In addition to shading, we perform color blending and enhancement to highlight different stages using rendering properties. In our work, the final color of a pixel is given by:

$$C = (t_1C_0 + t_2T + t_3G)(1 - t_4E) \quad (1)$$

where C_0 is the original color in the input image, T is the tone color computed in the previous step, G is the gray intensity of the color, and E is the edge intensity as provided by the previous step or as an input. The gray level intensity is computed as the luminance $G = 0.3R + 0.59G + 0.11B$, where R, G, B are the red, green, blue components of a pixel color, respectively. t_1, t_2 and $t_3 \in [0, 1]$ are weighting parameters that add up to one and are used to smoothly blend from gray-scale images, through tone shading, to full color images, depending on the position of the snapshots in the final sequence. In our examples, earlier time steps are assigned to gray-scale, while the final time step is rendered at full color. t_4 is a sequential factor that determines the saturation of the edge map in a snapshot. As t_4 becomes closer to 1, the contours are more apparent.

There are several methods for image enhancement [3]. Here we simply use gray-scale manipulation which maps the original gray/color intensity to a new value according to a specific curve. Gray-scale manipulation can effectively increase the image contrast. In our system, the manipulation is done in the graphics hardware on the fly, which also enables interactive adjustment.

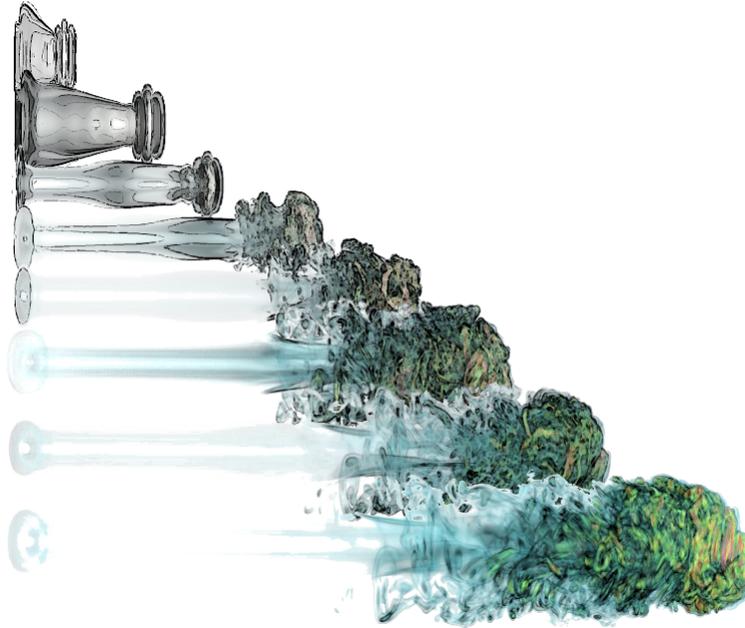


Fig. 5. Plume Data Set. The vertical layout shows the the plume in different time periods and provides a better view of how it changes over time.

4.4 Interactive Composition

Once all the snapshots are processed, the final step is to arrange each snapshot to the right place according to the specific layout. In order to accommodate to different types of images, our system provides interactive image composing. Users can add or remove a snapshot in real-time and the system will automatically update the sequential index of each snapshot. The layout can also be altered by moving, rotating, or scaling up and down the snapshots.

4.5 Results

Figure 5 shows the visualization of the simulation conducted by researchers at the National Center for Atmospheric Research (NCAR) that models 3D thermal starting plumes descending through a fully-compressible adiabatically-stratified fluid. In order to better estimate the penetration of the plume into the solar interior, scientists mostly focus on the stability of the plume as it evolves. The vertical layout of the plume in different time periods in Figure 5 provides a better view of how it changes over time. The line drawing and the luminance highlight in early and intermediate snapshots show the structural evolution more clearly. And at the final step, we use the full-colored snapshot from the volume rendering which highlights the intensity of the plume vorticity.

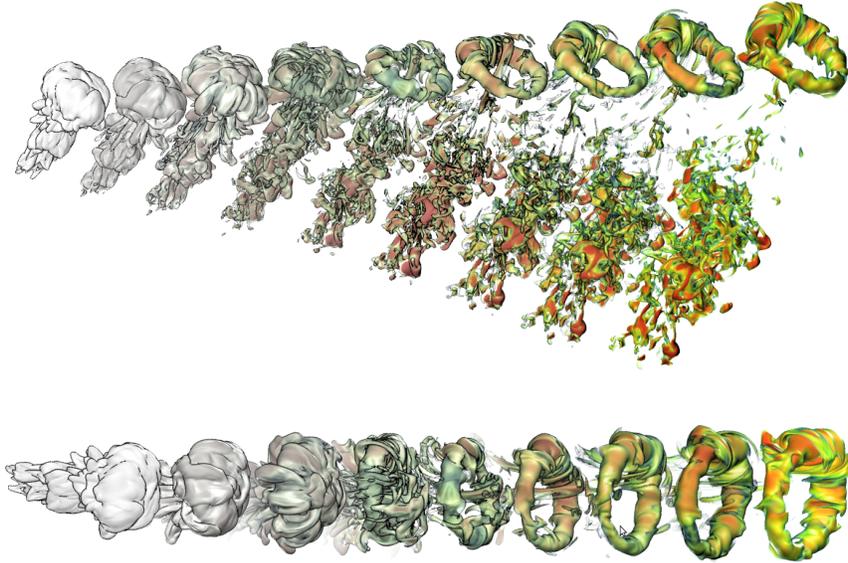


Fig. 6. Argon Bubble Data Set. The top image is a sequential arrangement of the snapshots showing the entire evolving process. By applying a color mask, we can achieve a more compact image showing on the features of interest (the smoke ring), as the bottom image shown.

As for the argon bubble data set, Figure 6(top) shows a similar layout to the plume illustration. However, we also apply the masking technique to the snapshots which de-emphasizes the turbulence structure and only show the feature of interest, the smoke ring. In doing so, we are then able to put the ring structure compactly as shown in Figure 6(bottom). Likewise, the contour drawing and the fading gray-scale coloring depict the evolving structure more clearly.

5 Discussion

The traditional approach to visualizing time varying data relies on animations, which may be lengthy and only convey one instant at a time. As an alternative, we have produced illustrations that capture flow motion in a single image using a combination of both 3D object- and 2D image-based approaches. Object-based approaches have the advantage of showing all steps in the evolving flow in their original 3D position and orientation. This is useful for correlating the different stages in time and the space they occupy. Image-based approaches, on the other hand, decouple the space dimension and lets us experiment with different layouts to depict the different stages of an evolving flow. Although the actual 3D information is not encoded, a different layout lets us see different stages without

the inherent overlapping that may occur otherwise. Compare, for example, the illustrations generated in Figure 6 (image-based) and Figure 2 (object-based). Although the latter clearly shows the actual advance of the argon bubble, it is not possible to show many of these stages in the same image. The former, on the other hand, lets us place many stages in a single image. In general, different illustration needs demand different strategies. For example, we have shown that the use of transparency and contour rendering is important for de-emphasizing certain stages so that others are clearly depicted. In other cases, the use of transparency is used to emphasize entire regions, such as the smoke ring in the argon bubble illustration, Figure 6(bottom). With the use of transparency, we have traded completeness for compactness, and produce an effective illustration.

6 Conclusions and Future Work

We have presented a series of techniques for obtaining effective illustrations of time evolving flow from volume data. At the core of our approach is the premise that flow and motion can be represented effectively in a single 2D image. To convey the idea of flow, we apply different rendering styles to different stages in a flow, such as contours and toon shading. Inspired by hand-crafted illustrations, we have also experimented with different layouts to convey the different stages of flow clearly without the ambiguity that may be raised with naive image overlapping. Obtaining a good illustration that communicates the essence of time evolving flow remains a craft. For this reason, a combination of both 3D object- and 2D image-based techniques is necessary. Coupled with interactive tools to allow the user to explore the different rendering and layout parameters, these techniques highlight important features of the data, such as contours and shading, that help understand the evolution of a complex shape over time. There are a number of directions for future work. We will explore of other metaphors for depicting flow motion extensively used in art forms, such as speed lines and arrows, and motion blur. Directionality is also important when conveying motion. We will experiment with other layouts that preserve the directionality of flow.

Acknowledgement. This work is sponsored in part by the U.S. National Science Foundation’s ITR and PetaApps programs and the U.S. Department of Energy’s SciDAC program. The argon bubble data set was provided by the Center for Computational Sciences and Engineering at the Lawrence Berkeley National Laboratory. The plume data set was provided by the National Center for Atmospheric Research. Chris Muelder of the VIDDI group at UC Davis provided the feature extraction and tracking code.

References

1. Jean-Paul Balabanian, Ivan Viola, Torsten Möller, and Eduard Gröller. Temporal styles for time-varying volume data. In Stephan Gumhold, Jana Kosecka, and

- Oliver Staadt, editors, *Proceedings of 3DPVT'08 - the Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, pages 81–89, June 2008.
2. J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.
 3. Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
 4. Bruce Gooch, Peter-Pike J. Sloan, Amy Gooch, Peter Shirley, and Richard Riesenfeld. Interactive technical illustration. In *ISD '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 31–38, 1999.
 5. Christopher G. Healey, Laura Tateosian, James T. Enns, and Mark Remple. Perceptually based brush strokes for nonphotorealistic visualization. *ACM Trans. Graph.*, 23(1):64–96, 2004.
 6. A. Joshi and P. Rheingans. Illustration-inspired techniques for visualizing time-varying data. pages 679–686, Oct. 2005.
 7. Teng-Yok Lee and Han-Wei Shen. Visualizing time-varying features with tac-based distance fields. pages 1–8, April 2009.
 8. Yunjin Lee, Lee Markosian, Seungyong Lee, and John F. Hughes. Line drawings via abstracted shading. In *ACM SIGGRAPH '07*.
 9. Aidong Lu and Han-Wei Shen. Interactive storyboard for overall time-varying data visualization. pages 143–150, March 2008.
 10. Kwan-Liu Ma. Visualizing time-varying volume data. *Computing in Science and Engg.*, 5(2):34–42, 2003.
 11. Ernst Mayr. *What evolution is*. Basic Books, New York, 2001.
 12. Chris Muelder and Kwan-Liu Ma. Interactive feature extraction and tracking by utilizing region coherency. In *Proceedings of IEEE Pacific Visualization 2009 Symposium*, April 2009.
 13. Frits H. Post, Frank J. Post, Theo Van Walsum, and Deborah Silver. Iconic techniques for feature visualization. In *VIS '95: Proceedings of the 6th conference on Visualization '95*, pages 288–295, 1995.
 14. D. Silver and X. Wang. Tracking scalar features in unstructured data sets. pages 79–86, Oct. 1998.
 15. Deborah Silver and X. Wang. Volume tracking. In *VIS '96: Proceedings of the 7th conference on Visualization '96*, pages 157–164, 1996.
 16. Aleksander Stompel, Eric Lum, and Kwan-Liu Ma. Feature-enhanced visualization of multidimensional, multivariate volume data using non-photorealistic rendering techniques. pages 1–8. *Proceedings of Pacific Graphics 2002*, IEEE, 2002.
 17. N.A. Svakhine, Y. Jang, D. Ebert, and K. Gaither. Illustration and photography inspired visualization of flows and volumes. pages 687–694, Oct. 2005.
 18. Fang Wen, Qing Luan, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Color sketch generation. In *NPAP '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering*, pages 47–54, 2006.
 19. Jonathan Woodring and Han-Wei Shen. Chronovolumes: a direct rendering technique for visualizing time-varying data. In *VG '03: Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume graphics*, pages 27–34, 2003.
 20. Jonathan Woodring, Chaoli Wang, and Han-Wei Shen. High dimensional direct rendering of time-varying volumetric data. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, pages 417–424, 2003.
 21. Djemel Ziou and Salvatore Tabbone. Edge Detection Techniques-An Overview. *Pattern Recognition & Image Analysis*, 8:537–559, 1998.