Layout of Multiple Views for Volume Visualization: A User Study

Daniel Lewis, Steve Haroz, and Kwan-Liu Ma

University of California at Davis
{dglewis, sharoz, klma}@ucdavis.edu

Abstract. Volume visualizations can have drastically different appearances when viewed using a variety of transfer functions. A problem then occurs in trying to organize many different views on one screen. We conducted a user study of four layout techniques for these multiple views. We timed participants as they separated different aspects of volume data for both time-invariant and time-variant data using one of four different layout schemes. The layout technique had no impact on performance when used with time-invariant data. With time-variant data, however, the multiple view layouts all resulted in better times than did a single view interface. Surprisingly, different layout techniques for multiple views resulted in no noticeable difference in user performance. In this paper, we describe our study and present the results, which could be used in the design of future volume visualization software to improve the productivity of the scientists who use it.

1 Introduction

Volume visualization has become an increasingly important tool in many areas of scientific research and industry applications. It aids in the evaluation of engineering designs, the understanding of large-scale simulations, the viewing of multi-modal medical data, the mining of enormous web databases, and the analysis of homeland security information. Historically, the primary challenges in 3D visualization have been the limitations of hardware and algorithms. With the increasing power of graphics hardware and the maturity of rendering algorithms, the ability to generate useful visualizations is now more often limited by human interaction with the system. The development of appropriate user interfaces for performing volume visualization tasks, however, has not received sufficient attention. This need with respect to time-variant data is particularly great, as the amount of data can quickly overwhelm both the user and the computer. Innovative interface designs and layouts have been introduced, but most have not been adopted by commercial visualization tools. The effectiveness of these designs with respect to user performance needs to be evaluated against that of simpler conventional interfaces.

While the helpfulness of multiple views to time-varying data is obvious, we want to find the extent to which that is true. Furthermore, we want to determine if the arrangement of views influences user performance. Expecting the increased effort required by the more flexible layout schemes to result in longer task completion times, we were surprised to find that hypothesis incorrect. We compared the

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 215 – 226, 2006.

conventional single-view interface with variations of multiple-view layouts by measuring how well users performed a set of tasks requiring fine tuning visualization parameters and combining visualization data. We conducted the study on both time-variant and time-invariant volumes to determine the utility of multiple-views.

2 Visualization Interfaces

In data visualization, initially raw data is filtered, mapped, rendered, and finally viewed. The filtering and mapping steps select which data the user wants to see and convert that data into an intermediate representation. The rendering step then generates images based on view position, lighting conditions, and any other parameters that help better present the data visually. A typical user performs this process iteratively, converging on a good representation that shows information of interest in the data. A good visualization interface should abstract the internal steps and complex parameter selection of visualization algorithms leaving the user with simple, high-level decisions made through an intuitive visual interface [16]. Prior research efforts on optimizing this process have primarily focused on improving the individual steps such as accelerating the feature extraction or rendering calculations. The user interface aspects of the problem, as well as the workflow for the entire process, have often been neglected.

The typical volume visualization interface has a single rendering window and a control panel that displays all the visualization parameters. Finding optimal settings for the myriad of parameters by manipulating many individual controls can be overly complicated and inefficient, and it can unnecessarily muddle the simplicity of a user's mental model of how an application works [4]. Having too many individual controls can reduce context and is unintuitive to users without expertise in graphics or visualization. One solution to this problem, the data-flow interface, represents the data exploration process by a directed graph of connected components. These components act upon the datasets or can be chained together in a pipeline. A few commercial visualization systems employ such an interface [1][18][19]. Data-flow interfaces display a graph showing the flow of information, which can be shared amongst collaborators to communicate the process needed to generate a requisite result. One shortcoming of the data-flow approach is that it does not indicate the history of the visualization process [9][10]. As is true in most interfaces, changes to the settings of one of the flow nodes cause the previous image to be lost.

Other approaches treat data exploration as an exploration of dimensions [15] or a search in the parameter space [9][7][10][11] resulting in simpler and more intuitive interface designs. The user can see the consequence of changes rather than the means by which to do so. Multiple views generally arranged in an evenly spaced layout show the same data with different parameters. The user, however, is required to browse through a large number of images, which could be quite time-consuming, and the user may have difficulty finding where to focus attention.

While these approaches to visualization interface design show promise, incorporating them into existing commercial visualization software, if possible at all, could require major restructuring of the systems. Commercial software tools, however, have begun to add simple displaying and browsing options such as multiple linked views and focus plus context. Much research has been done on the benefits of multiple linked views [2] and their results are promising. A common theme repeatedly surfacing in volume visualization interfaces is the use of multiple views [3][13]. Nevertheless, none has studied or measured the extent of the helpfulness of giving volume visualization users multiple views, nor has anyone examined the effectiveness of the layout techniques for those views. In turn, we performed this study to learn exactly that.

3 Layout

Layout is an important and fundamental aspect of graphical user interface design. Appropriate arrangement of controls in a form is vital to usability, as it simplifies the flow of attention and localizes the proximity of widgets for related or sequential tasks. Control layout has even been used as a metric for evaluating user interfaces [14].

Most modern graphical interfaces have a variety of panels with each having its own unique set of controls and data displays. Statistical examinations of the usage of particular panels can reveal the appropriateness of the size and location these panels. The layout of these panels has been found to vary widely based on the specific type of application used [8]. Nevertheless, many of the same layouts are used for a variety of different applications without sufficient tests of the effectiveness of the layout in that particular type of use.

A more general approach to the examination of layout appropriateness regards the conformity of panels to a “regular pattern” [2]. Panels or windows with irregular shapes or inconsistent sizes have been found to benefit user controlled layout rather than automatic arrangement, yet when a simple and automatic layout is feasible, the extra burden on the user of manual arrangement is unnecessarily time consuming. The appropriateness of an automated layout should therefore be dependant on the regularity of data between multiple views.

The data sizes of different time steps in volume visualizations are usually consistent, so time variant volume visualization should benefit most from an interface with automatic layout. Unique layouts have already been found to be effective for video editing, which involves the manipulation of 2D time-variant data [5][17]. In contrast, we examined the effect of layout on time-variant and time-invariant 3D data.

4 Layout Variations

To test the helpfulness of multiple-view layouts, we created a volume visualization application that can utilize any one of four layout techniques. The application has a common control panel that allows the user to manipulate the colors and opacity levels of the volume’s transfer function. The visualization is in a separate panel. The camera for each view can be independently controlled by dragging the mouse. The arrangement of views is the only difference between the four layout interfaces (see figure 1).

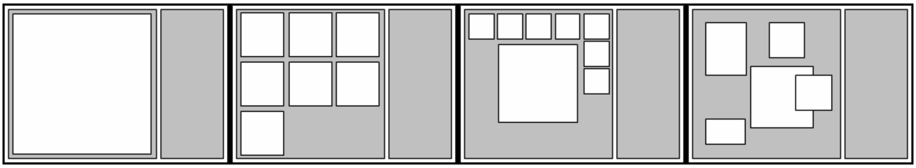

Fig. 1. Layout designs from left to right: simple (single view), split (evenly divided), radial (large primary view), and freeform (user controlled)

The simple layout has a single interactive view of the visualization. At any given time, the user can only see a single perspective of the data using the current transform and will not be able to see any previous actions or results. This basic interface is common in many applications and is therefore used as a control for our experiment. The split interface allows the user to evenly split the screen into different views of the data. All of the views are independent, interactive, and the same size. If the number of views is not a square number, blank space fills in the remaining area to make the boxes square. A user can create a new view with a 'New Window' button, which will copy the current view into a newly spawned view box. Each view also has a 'Load Data' button that loads a new set of data. All of the views have standard 'Close' and 'Maximize' buttons that remove or maximize the view respectively. If a square number of views remain, the views rearrange themselvess into a smaller square layout.

The radial layout consists of a large view in the center of the visualization panel with additional views surrounding the perimeter of the primary view. The outer views are smaller than the primary view, but they are all equally sized. They also have the same aspect ratio as the primary view. If not enough views can fill the perimeter, blank space is used. If the primary view is closed, one of the perimeter views will move to the center. Each view has a 'New Window' and 'Load Data' button that behaves similarly to that of the split interface. In this layout scheme, the maximize button moves the window into the central focus if the view is in the perimeter or maximizes the window to full screen if it is already in the center.

The freeform interface provides users with complete control over the layout of the views with no constraints. When a new view is opened, it appears in the left corner of the viewing pane with half of the width and height of the available area. The users can then freely move and resize the view within the viewing pane. Other than this distinction, it behaves similarly to the radial and split interfaces.

The common control panel to all four interface designs allows the user to modify the basics of the transfer function: color, curve shape, position, and opacity. The user can add predefined curve shapes to the transfer function or choose to hand draw it. The predefined shapes can be easily modified with the mouse to change the opacity, position, and size. Multiple transfer functions can be added or loaded from a previously saved transfer function to easily crop parts of the volume that are of interest (see figure 2).

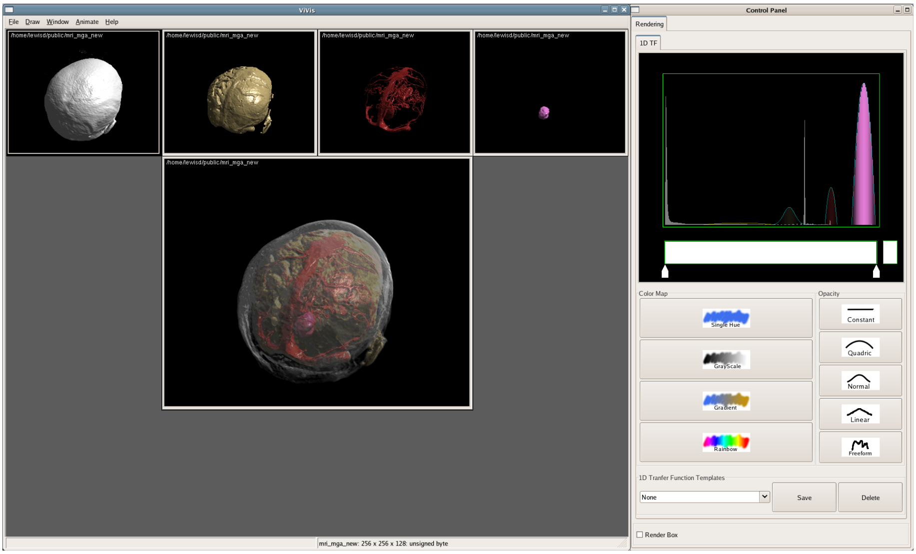

Fig. 2. This shows the brain dataset in the radial interface. Each of the brain's volume components is separated in a radial view, while the center shows the components with different levels of transparency. The control panel with the center view's transfer function is on the left.

5 Experiment Design

Twenty-four participants with a mixed range of technical backgrounds were involved in our study. They consisted of undergraduate and graduate students in Computer Science, Electrical Engineering, Civil Engineering, Mechanical Engineering, Mathematical, and Biological fields. We began each session by having our participants fill out a questionnaire to find their experience with computers and visualization. Their computer experience varied from having little experience (just email and typing documents) to being advanced developers. We randomly assigned users to interfaces and we manually reviewed the experience levels to assure even distribution of experience for each layout.

As another compensation for the range of experience, the users were given a tutorial to introduce them to the application. We explained the purpose of the transfer function and walked each person through the basics of configuring it to selectively crop parts of the volume. We also explained the benefits of differentiating parts of the volume by adjusting the color and opacity of ranges in the transfer function. We then had each user demonstrate competence in the program by creating a transfer function to show the bone structure, muscle, and skin layer of a dataset consisting of two scanned feet. We made sure that the users could achieve this task without any help from the experimenter to prevent any questions arising during the timed tasks. The tutorial given used the first layout that the user would encounter. After the first segment of the experiment was complete, a second tutorial was given to acquaint the user with the second layout interface. We trained each user for approximately ten minutes or until the user could easily complete simple tasks without our help.

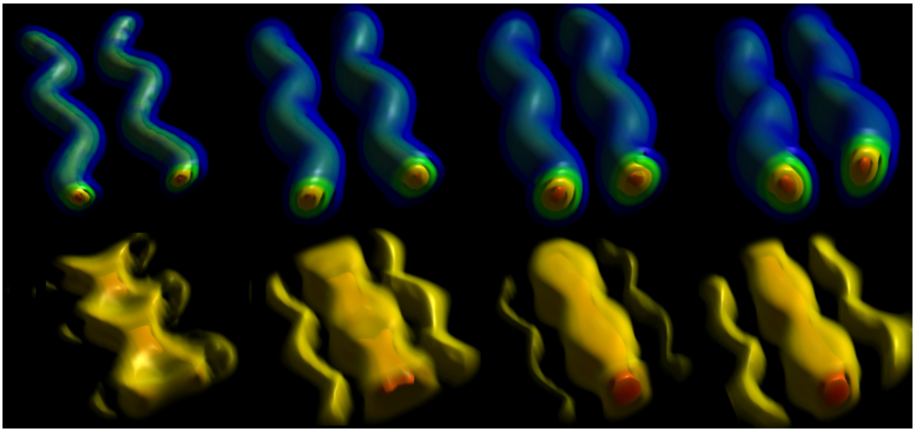

Fig. 3. These images show time steps 1 through 4 of the velocity data (top) and the vorticity data (bottom). Two spinning vortices produce high velocity winds in a tube-like shape (top), and they produce high vorticity, or local rotation, in the area in between the two tubes. Their progressive increase in strength and order can be seen in the successive time steps.

Each user performed two tasks, and each task was done using a different layout system. The tasks were individually timed to track the effect of the different interface designs on user performance. The first dataset is an MRI scan of a human head (see figure 2), and the second dataset is the velocity and vorticity produced by two spinning vortices over five time steps (see figure 3).

For the data containing the brain, the user was asked to adjust the transfer function to differentiate the skin, brain, blood vessels, and tumor inside the brain by varying their respective colors and opacities. We instructed the users to make all other features of the dataset transparent.

After the first test, the experimenter would explain the second user interface and allow the user to become comfortable with it. The other task was performed using two time-varying datasets with five time steps each. One dataset showed the magnitude of the velocity of two spinning vortices, while the other showed the vorticity, or local rotation, resulting from those vortices. The users were given the task of roughly sketching each of the five time steps for both the velocity and vorticity. After the users drew each time step, we asked them to explain the changes that occurred to velocity and vorticity throughout the time steps. The final time step for both velocity and vorticity was pre-drawn to ease the task of finding the best transfer function that best displayed the important features. After the test, the users were asked to explain what changed in the velocity and vorticity of the object in order to gauge their understanding of the volumes' transforms.

Each user performed the task on a different ordered pairing of the two datasets and two of the interfaces. Each permutation of the ${}_4P_2 = 12$ possible ordered couplings of the interfaces were given to two users. Each of those users was given a different ordering of datasets for a total of 24 combinations of two datasets and two out of four interfaces. The specific ordered pair was randomly assigned to each user to test all combinations and counterbalance any ordering or learning effects. We also had a different set of users examine time-variant brain tumor datasets on all four interfaces (a within-user study) and found experience inconsistently affected performance.

The multiple views should allow the user to have multiple workspaces that allows for separating the three different volumes individually. Using multiple views should help modularize the tasks and combine all of the features in a final step. Using the single view, the users need to save the transfer function after each feature is found and then reload the previous transfer functions to add them together. The users however, may use a different strategy, such as trying to do everything from one view regardless of the interface used. We observed their use and asked them to describe their strategy when finished. We also asked which interface they preferred and why.

6 Results

Performance with time-invariant dataset (figure 4) was unaffected by the interfaceinterface used. All the interfaces had nearly identical average use times. The time-varying dataset showed a difference, but that difference only existed between the single-view layout and the multiple-view layout. The task completion times for the multiple-view layouts depended only on the actual dataset used.

6.1 Time-Invariant Data

Six people used each of the layouts for the fixed brain dataset. The users of the single-view layout, the experiment's control, for the time-invariant dataset required an average of 5:47 to complete the task (standard deviation of 1:50). The average times for the split, radial, and freeform layouts were 5:47, 5:38, and 5:30 respectively. The statistically insignificant differences show that both the availability and layout of multiple views was irrelevant to users' performance (see figure 4 and table 1).

6.2 Time-Varying Data

As with the time-invariant dataset, six participants used each of the layouts. With this dataset, however, a significant difference between the single-view times and the multiple-view times is apparent (see figure 5 and table 2). The average task-completion time for the single view is around five minutes longer than the average times of the multiple views. The single-view also had a significantly higher standard deviation. A clear difficulty arises from comparing images when an intermittent task (loading another volume) must be performed. This small pause appears sufficient to break users' attention to the point of being unable to identify small differences [12]. While the difference in quantity of views resulted in a strong difference, little performance difference resulted from changing the type of multiple-view layout.

6.3 User Preference

To evaluate user preference of the layouts each participant was asked to comparatively rate their preference of the two interfaces used on a scale from one to ten. The majority of those who had no preference between the first and second

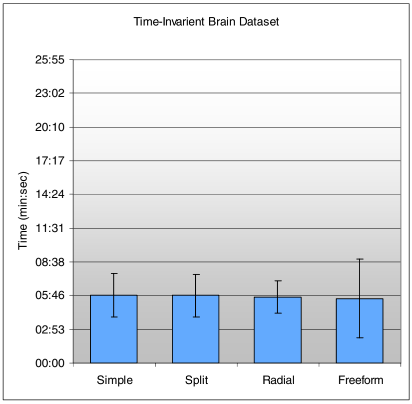

Fig. 4. Time-Invariant task execution times

Table 1. Time-invariant data results (note that average times are nearly identical)

Time-Invariant Brain Times

	Simple	Split	Radial	Free	All
Avg	05:47	05:47	05:38	05:30	05:41
Std Dev	01:50	01:49	01:24	03:21	02:05

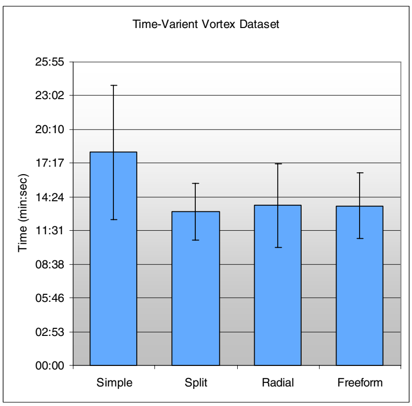

Fig. 5. Between-subject task execution times

Table 2. Time-varying data results for the vortex. The P-values were calculated between each layout's times and those of simple layout.

Time-Variant Vortex Times

	Simple	Split	Radial	Free	All
Avg	18:11	13:08	13:40	13:39	13:29
Std Dev	05:44	02:26	03:36	02:49	02:49
P-value	-	0.074	0.133	0.112	0.013

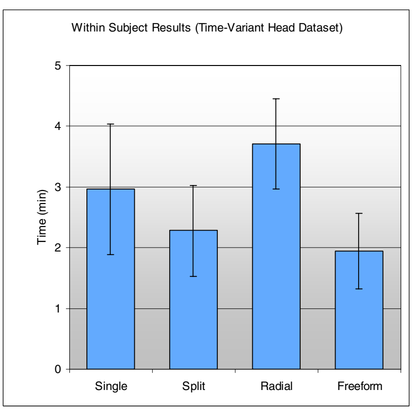

Fig. 6. Within-subject task execution times

Table 3. Time-varying data results. The P-values were calculated between each layout's times and those of simple layout.

Time-Variant Brain Times

	Simple	Split	Radial	Free	All
Avg	2:58	2:17	3:43	1:57	2:33
Std Dev	1:04	0:45	0:45	0:37	1:06
P-value	-	0.17	0.18	0.05	0.003

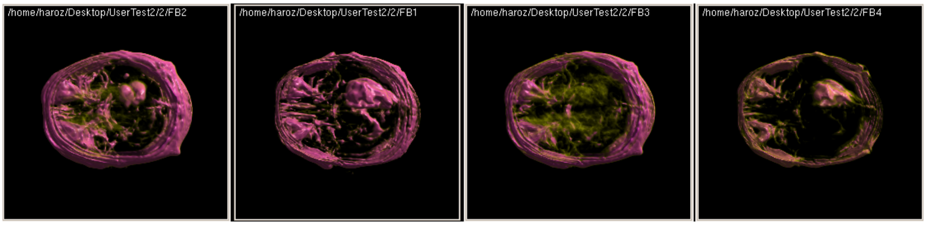

Fig. 7. These images show a time-variant dataset in which a tumor can be seen growing and shrinking inside a brain. This is one of the datasets used in the with-user study.

interfaces used the simple single view for the brain dataset and a multiple-view for the vortex dataset. The lack of necessity of the multiple views for the brain dataset resulted in few even realizing that a difference existed. The preferences were as follows: one participant preferred the simple interface; six preferred the split interface; five preferred the radial interface; and seven preferred the freeform interface. As with the task completion times, the user preferences between the different multiple-view layouts were equivalent.

6.4 Within-Subject Study

To examine within-subject time differences, we performed a separate study with twelve participants. Though different, these users were selected from a similar population to that of the first study. Each person tracked the size of a brain tumor as it grew and shrank between five time steps (see figure 7). After undergoing a training session using a sample dataset, the users were timed as they performed the task using each interface with a different dataset. Their only feedback was to state whether the tumor grew, shrank, or remained unchanged between each time step. The orders of the datasets and interfaces were selected randomly to give a nearly even distribution of each.

After examining the data, we realized that the times for all of the users' first trials were highly inconsistent. Due to the unusually high variation in that set of numbers, we chose to drop everyone's first trial. Therefore, only three of each user's four timed trials were included in the analysis. Even so, all dataset and interface combinations were presented and used in the analysis at least twice. Training the participants using the interface of their first trial was a likely culprit for the anomaly.

The results (see figure 6 and table 3) show long times and high standard deviations for the simple and radial layouts. In contrast, the split and free layouts have lower times and standard deviations. With the exception of the poor performance of the radial layout, these results are similar to those of the between-user study. The split and free layouts ultimately performed better than did the simple layout, but the performance difference between the split and free layouts is statistically insignificant.

7 Discussion and Future Work

While the between-user study was conducted on twenty-four people, each unique combination of two layouts and two datasets was given to only two people.

Conducting the study on more people could have helped reduce the high variance in the time measurements, which would lower the P-values to more acceptable levels. Had we had the opportunity, we would have also preferred to conduct an additional study on specialists in a particular field that makes use of time-variant volumes. If all of our users had the knowledge to describe complex datasets using precise and consistent language, we could have conducted a far more detailed study that required precise, more falsifiable responses. Though the vortex and tumor datasets had easily identifiable features, the volumes' minimal complexity made for somewhat simple responses from the participants such as "it got skinnier". In spite of this simplicity, the users of the single view still had a very difficult time identifying the changes in the volumes through time.

Although, we only looked at four view layouts, their simplicity and generality covered a significant number of the arrangements used in the few existing multiple-view volume visualization systems. The number of possible layouts is infinite, so we tested simple views that we believe are most practical. Nevertheless, this limited testing could have resulted in a missed opportunity to find a better, albeit more complex, layout.

An interesting observation was that some subjects using the simple view with the time-varying data expressed the desire to use multiple views. The numerous switching between views directly affected their overall time. This lack of direct comparison clearly limited their ability to perceive changes in time-varying data without knowing precisely where to attend.

Another notable find was the users' tendency to arrange the free layout views into a tiled layout similar to that of the split design. According to [2], this wasted expenditure of time with window management should have resulted in longer times for the free layout. A possible explanation for this discrepancy is the difficulty in making our application rearrange windows for the constrained layouts. The lack of a high affordance means of achieving this simple goal resulted in a couple of users closing the views and then reopening them in the desired order. The split view times and the radial view times may have been shorter if rearrangement were simpler.

8 Conclusions

This study identifies the differences in effectiveness of a variety of simple layouts for multiple-views of volume visualizations. We have presented four interfaces for volume visualizations with varying layout constraints and arrangements. The results support the obvious advantage of multiple views and their ability to aid in achieving faster and more accurate performance of feature identification for time-varying data. The experiment's control, the single view, resulted in slower task completion times as well as significantly decreased accuracy in describing changes between datasets. As was expected, for a time-invariant dataset, the number of views had no noticeable effect on the accuracy or speed of task accomplishment.

The more surprising observation is the irrelevance of the type of multiple-view layout as well as the level of constraint for that layout. We hypothesized that the constrained layout system would outperform the others, yet our results show that the layout of multiple volume visualization views has little, if any, effect on users' performance. While we only tested three multiple-view layout designs, a wide range of designs exist. The lack of a significant and consistent performance difference between the free layout and the automated layouts may be implicative that time spent on window management is counteracted by the constraints of the automated layouts. The regularity in the size of the time variant dataset should have resulted in the automated layouts eliminating effort spent on window management [2]. Perhaps a similar user study carried out with a volume that significantly changes size, shape, and aspect ratio over time could yield different results, but obtaining such a dataset may be impractical.

Acknowledgements

We greatly appreciate all those who participated in our user study. This work is supported in part by NSF under grants IIS-0552334 and ACI-0222991.

References

1. Abram, G., Treinish, L. An extended data-flow architecture for data analysis and visualization. In *Proceedings of the IEEE Visualization '95 Conference*., pp. 263–270. October 1995.
2. Bly, S. A. and Rosenberg, J. K. A comparison of tiled and overlapping windows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. M. Mantei and P. Orbeton, Eds. CHI '86., New York, NY 1986 ACM Press.
3. Calhoun, P.S., Kuszyk, B.S., et al. Three-dimensional volume rendering of spiral CT data: theory and method. In *Radiographics*. 19(3). pages 745-64. 1999.
4. Cooper, A., Reimann, R. *About Face 2.0: The Essentials of Interaction Design*. Wiley, New York, 2003.
5. Girgensohn, A., Boreczky, J., Chiu, P., Doherty, J., Foote, J., Golovchinsky, G., Uchihashi, S., and Wilcox, L. A semi-automatic approach to home video editing. In *UIST 00: Proceedings of the 13th Annual ACM Symposium on User interface Software and Technology*. pages 81-89. New York, NY, 2000. ACM Press.
6. Gresh, D. L., Rogowitz, B. E., Winslow, R. L., Scollan, D. F., Yung, C. K. 2000. WEAVE: a system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *Proceedings of the Conference on Visualization*. pages 489-492. IEEE Visualization. IEEE Computer Society Press, Los Alamitos, CA.
7. He, T., Hong, L., Kaufman, A., Pfister, H. Generation of transfer functions, with stochastic search techniques. In *Proceedings Of IEEE Visualization '96 Conference* (October 1996), pp. 227–234.
8. Henderson, D. A., Card, S. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. In *ACM Trans. Graph.* 5(3) pages 211-243. 1986.
9. Jankun-Kelly, T. J., Ma, K.-L. A spreadsheet interface for volume visualization. In *Proceedings of IEEE Visualization 2000 Conference* (2000).
10. Ma, K.-L. Image graph: A novel approach to visual data exploration. In Proceedings of *IEEE Visualizaton '99 Conference* (1999), pp. 81–88.

11. Marks, J., Andalman, B., Beardsley, P., Freeman, W., Gibson, S., Hodings, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., And Shieber, S. Design Galleries: A general approach to setting parameters for computer graphics and animation. In *Proceedings of SIGGRAPH '97 Conference* (August 1997), pp. 389–400.
12. O'Regan, J. K., Rensink, R. A., & Clark, J. J. Change-blindness as a result of 'mudsplashes'. In *Nature*, 398(6722), pages 34-34. 1999.
13. Roberts, J. C., On Encouraging Multiple Views for Visualisation. In *Proceedings of the international Conference on information Visualisation* IEEE Computer Society, Washington, DC, 8. 1998.
14. Sears, A., Layout appropriateness: a metric for evaluating user interface widget layout, In *IEEE Transactions on Software Engineering*. 19(7), July 1993, pp. 707-719.
15. Tory, M., Potts, S., And Möller, T. A Parallel Coordinates Style Interface for Exploratory Volume Visualization. In *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 1. pages 71-80. Jan/Feb 2005.
16. Tzeng, F.-Y., Ma, K.-L., Lum, E. An intelligent system approach to higher-dimensional classification of volume data. In *IEEE Transactions on Visualization and Computer Graphics* 11, 3 (2005), 273–284.
17. Ueda, H., Miyatake, T., Sumino, S., and Nagasaka, A. Automatic structure visualization for video editing. In *CHI 93: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. pages 137-141. New York, NY, USA, 1993. ACM Press.
18. Upson, C., Faulhaber, T., Kamins, D., Schlegel, D., Laidlaw, D., Vroom, J., Gurwitz, R., And Van Dam, A. The application visualization system: A computational environment for scientific visualization. In *IEEE Computer Graphics and Applications* 9, 4 (1989), 30–42.
19. Young, M., Argiro, D., And Kubica, S. Cantata: Visual programming environment for the Khoros system. In *ACM SIGGRAPH Computer Graphics* 29, 2 (1995), 22–24.