Enabling Interactive Scientific Data Visualization and Analysis with See-Through HMDs and a Large Tiled Display

Ken Nagao* Keio University

Yucong Ye† University of California, Davis

Chuan Wang‡ University of California, Davis

Issei Fujishiro§ Keio University

Kwan-Liu Ma¶ University of California, Davis

Abstract

Validation and exploration of the data generated by large-scale scientific simulations rely on sophisticated visualization and analysis tasks. With the advancement of supercomputing, the growing scale and complexity of the data make some of these tasks challenging, which demands new hardware and software solutions. We believe it is possible to address some of the challenges by utilizing the increasingly affordable see-through head mounted display (HMD) devices together with a low-cost tiled HDTV display. With the tiled display to provide a high-resolution overview of the data, the user can freely choose a small area to explore and analyze using a see-through HMD in stereoscopic 3D with gesture input. During such local exploration and detail data analysis, the user can apply a newly derived visualization parameter setting to the large tiled display for a new overview. In this way, computational costs become more manageable because realtime rendering and response are only required to cover a small screen space and a subset of the data. In our current study, we focus on supporting immersive isosurface and streamline visualization and analysis of 3D flow field data. In this workshop paper, we present our preliminary design and results, and we also discuss our further development and evaluation plan.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, Augmented, and Virtual Realities; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input Devices and Strategies, Interaction Styles

1 Introduction

Numerical simulation is an important tool for conducting scientific research. Large scale volumetric data is generated from the simulations of 3D physical phenomena and complex chemical processes. In order to validate and understand the simulation results, scientists rely on visualization and analysis tools. While many advanced flow visualization and analysis methods have been introduced over the years, the growing scale and complexity of data out of the state-of-the-art scientific simulations continuously demand new, enhanced ways for scientists to depict and dissect data. In particular, when high resolution displays and VR devices become more affordable, it is timely to realize and exploit immersive visualization and analysis of complex data. New opportunities are for both visualization researchers and users.

The value of VR to 3D flow visualization has been demonstrated by many researchers since the 90's. One of the early systems is CAVEvis system [14], which utilizes CAVE system as an immersive environment for interactive visualization of three-dimensional data and can display isosurface and cutplane of the data in stereoscopic images. The other system to be mentioned is the Virtual Wind Tunnel [9], which is a virtual reality system displaying the three-dimensional information of complex fluid flows on a stereoscopic head-tracked display with various visualization techniques including isosurface and streamlines.

VR has not been extensively used in scientific visualization and analysis tasks because it requires high speed, high resolution, high quality rendering solutions and easy input methods to be truly useful. Large-scale data visualization tasks simply make the problem more challenging. However, rendering hardware requirements have become more manageable when we use HMDs together with a tiled display instead of CAVE like environments.

Intuitive and easy-to-use input methods for interacting with the data and specifying intent for analysis tasks are not generally available. Wireless input devices and touch interfaces have been used, but full immersion would require a more natural interface based on simple gesture input. Recent motion capture devices such as Microsoft Kinect enable body gesture inputs, but effective way to utilize body gesture inputs for particular scientific analysis tasks remained to be developed.

We believe it is possible to address some of the challenges by utilizing the increasingly affordable see-through HMD devices together with a low-cost tiled HDTV display to provide a usable immersive analytics solution for studying large-scale complex flow field data. With the tiled display to provide a high-resolution overview of the data, the user can freely choose a small area to explore and analyze using a see-through HMD in 3D stereoscopically with gesture input. During such local exploration and detail analysis, the user can apply a newly derived visualization parameter setting to the large tiled display for a new overview. In this way, computational costs become more manageable because realtime rendering and response are only required to cover a small screen space and a subset of the data. In our current investigation, without losing generality, we aim to support flow field data analysis tasks based on isosurface and streamline visualization methods for scalar and vector fields, respectively. In this paper, we present our preliminary design and results, and we also discuss our further development and evaluation plan. Our prototype implementation can be seen in Figure 1.

2 Related Work

Chandler et al. in their position paper address general design and usability issues of immersive analytics [10]. Our study focuses on supporting interactive scientific data visualization and analysis tasks using a large tiled display and a HMD. In this section, we mainly consider what have been done by others using similar display and interaction devices in the context of visualization and analysis.

*e-mail: nagao@fj.ics.keio.ac.jp
†e-mail: chrisyeshi@gmail.com
‡e-mail: nauhcy@gmail.com
§e-mail: fuji@ics.keio.ac.jp
¶e-mail: ma@cs.ucdavis.edu

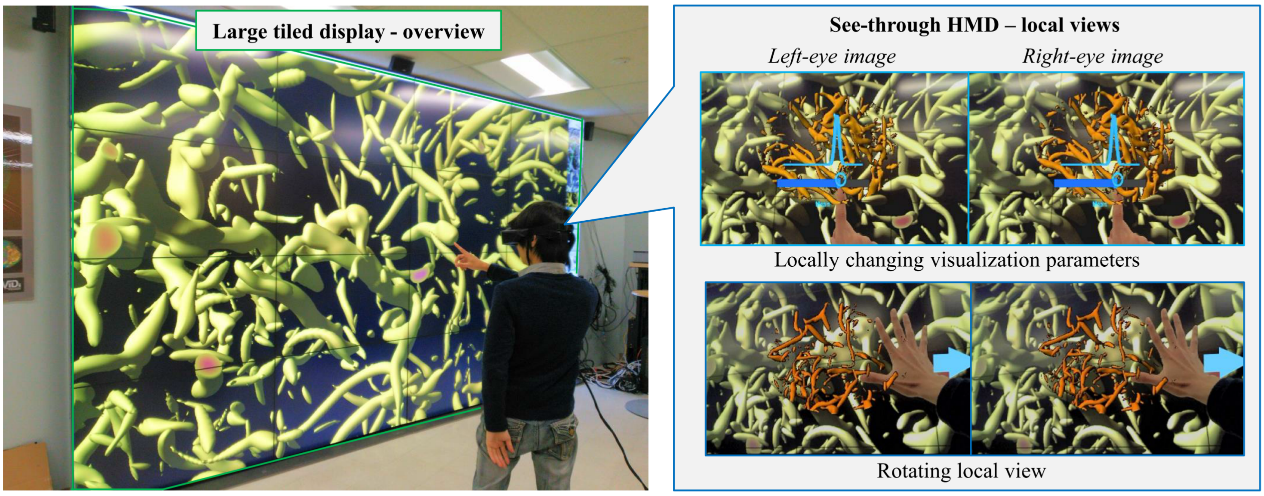

Figure 1: Our prototype implementation and test setting.

2.1 Data Visualization on a Large Display

Large tiled displays are becoming more and more common, and used for a variety of purposes [25]. With its large size display area and high resolution, tiled displays allow users to gain both an overview and the detail of the displayed content [4]; in particular, it is pointed out [5] [24] that search and comparison tasks can be effectively done with large displays to support visualization and analysis tasks. One of the well-known large display systems for data visualization is SAGE2 [22], a complete redesign and reimplementation of SAGE [15].

Using a large high-resolution tiled display for interactive visualization of large-scale data demands high-performance software and hardware solutions [23] [18]. To support stereoscopic viewing the computing requirements and cost could become many times higher. As demonstrated by Reda et al. for the immersive visualization of large-scale atomistic simulation [26], a scalable solution is possible but at the cost of custom software development and dedicated hardware setting.

2.2 Interaction with a Large Display

A variety of technologies for interacting with a large display have been introduced in the past. These technologies can be divided into two categories: with hand-held devices and without. Hand-held devices such as mobile phones have been used to interact with a large display [6] [28]. In addition, the possibility of attaching sensors to hand-held devices, such as pencils, has been discussed [7]. On the other hand, some research results indicate that hands-free devices can encourage more frequent interactions of users than hand-held devices [17]. A touch display is one such hands-free device, but the cost can grow quickly with its display size. Putting costs into perspective, inexpensive motion capture sensors, such as Microsoft Kinect, is quite usable [12], and have been utilized for interacting with a large display. In the system by Johnson et al. [16], together with other devices such as joysticks and mobile phones, Microsoft Kinect is used so that users can intuitively interact with the visualization by gestures. However, effective way to utilize such body gestures for analysis tasks remains to be developed.

2.3 Head Mounted Displays and a Large Display

There are data visualization systems using a large display together with portable display devices for supporting analysis tasks. For example, in [11], mobile phones are used for displaying details of data visualization as well as for interaction. With their increasingly affordable prices, HMDs also become attractive. HMDs enable stereoscopic viewing for data visualization when the screen of the large display is not three-dimensional capable [8]. In the work by Rodrigue et al. [27], the potential of coupling a large display with mobile devices is discussed from the aspect of its high versatility for mixed reality environment.

3 Design and Technologies

Our design goal is to capitalize the strengths of each display device while keeping the computing requirements to a minimum. The overall system configuration is depicted in Figure 2. To make these devices seamlessly work together, we need to integrate several technologies including developing new ones to address existing hardware and software limitations.

3.1 Design

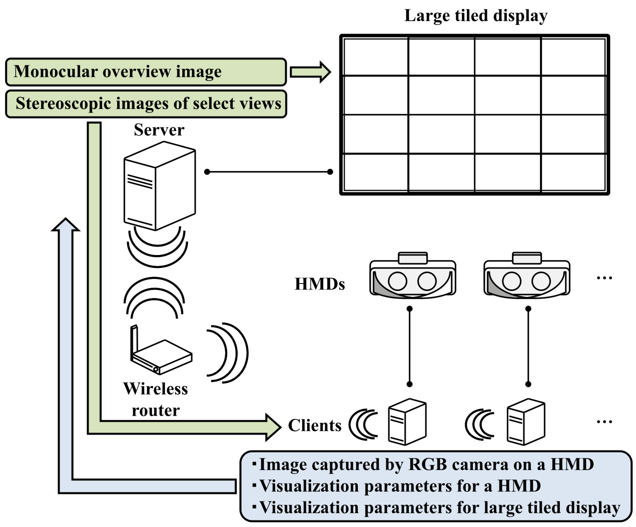

Figure 2: System configuration.

Our system consists of a client-side system and a server-side system, as shown in Figure 2. A large tiled display is connected to the server, and HMDs are connected to clients. The server and clients can communicate with each other through a wireless router. For example, a client sends new visualization parameters to the server, and the server responds to it by updating the visualization. The server delivers stereoscopic images of select views to the client, and monocular overview image to the large tiled display if requested. We note that a client does not render any visualization. Rendering is completely done by the server. In the following subsections, we describe each component of our system.

3.2 Clients

Visualization parameters for both local exploration with see-through HMDs and overview on a large display can be controlled through interfaces of see-through HMDs. Here, we describe the client-side system with a focus on its interactive technology.

3.2.1 Configuration

We use the see-through HMD named Meta1 [2], which has 3D see-through display (960x540 resolution, 35-degree field of view), RGB camera (1280x720 resolution) and depth sensor (320x240 resolution). We connect each see-through HMD to a separate client computer, and it is not necessary that the client computers have high–performance CPUs or GPUs.

3.2.2 Interactions

Each user can interact with the data visualization using his hand. The algorithm is as follows. Firstly, the client gets raw depth data from its depth sensor, and then from the data the client excludes every pixel whose distance from the see-through HMD is greater than the pre-defined threshold so that the client can get the depth data of the user's hand. Next, the client applies edge detection to the depth data. Specifically, the client uses the Sobel operator for the edge detection algorithm. Among the detected edges the system regards the topmost one as the position of the user's index finger. Figure 3 shows how the finger tracking algorithm works. At the same time, the client determines whether the hand is open or not by calculating the number of edges next to the position of the user's index finger.

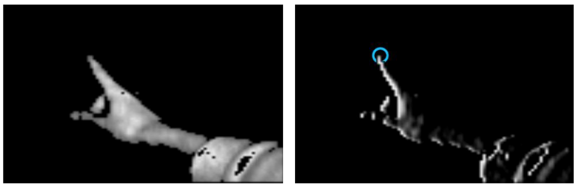

Figure 3: How the finger tracking algorithm works. The left image shows the extracted depth data of a user's hand, and the right image shows the edge detection and estimation of the position of the user's index finger.

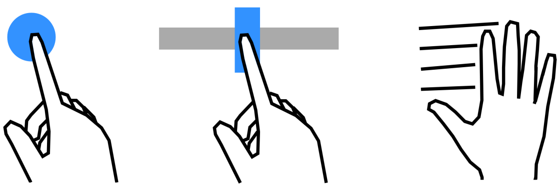

Figure 4: A user can use his hand to interact with the data by selecting buttons, changing slider values, and moving his open hand for rotating local view.

The interface of the see-through HMD consists of buttons and sliders which the user can select or change values with his index finger. Visualization parameters for the HMD are sent to the server every time the user changes its values via the sliders, and visualization parameters for a large tiled display are sent only when the corresponding button is selected. At the same time, by moving the open hand, the user can request the server to render and send rotated local images. Figure 4 shows in what way the user can interact with the virtual interface by his hand.

Each user has his focused region, which is the region of visualization the user selects to perform local data exploration. This is also set through the HMD interface. For setting the focused region, the client sends an image captured by the RGB camera to the server. The focused region detection algorithm is described in Section 3.3.3.

3.3 The Server

Many tasks are performed by the server, including volume rendering to the tiled display, handling network communication with the clients, volume rendering to the clients, and identifying focused regions of the clients.

3.3.1 Configuration

The tiled display consists of sixteen monitors, which are aligned in a four by four grid. The resolution of each monitor is 1920x1080, giving a total resolution of 7680x4320 (8k) for the entire display. The server is a single computer with three graphics cards: one Titan, and two Quadros. The entire tiled display is configured to be one large virtual display by utilizing Xinerama [3]. Xinerama comes with the graphics card driver, which distributes the rendering task to the multiple graphics cards we have in the computer. Xinerama takes care of parallel rendering for us so we don't need to develop a parallel volume renderer. The graphics card driver determines how Xinerama distributes the rendering which we have no knowledge about. That is, without further configuration, any single monitor application can be run across the entire tiled display by simply stretching the application window across multiple monitors. However, parallel rendering enabled by Xinerama is not optimized for our current system. More sophisticated parallel rendering algorithms can be implemented to improve the performance. For the purpose of this study, the performance provided by Xinerama is sufficient.

The architecture of the server follows the model-view-controller design pattern. The model is the volumetric data of interest. Note that the volumetric data is shared among all the views in order to conserve memory. A view-controller duo is used per viewing device. A viewing device can be the tiled display or a client. Each controller maintains its own copy of the rendering parameters, which instructs the volume rendering process of the corresponding view. For the tiled display, users can take full control of the rendering parameters through mouse interactions, while limited change to the rendering parameters is allowed with the clients. The controller of each client maintains a TCP connection, which is used to communicate with the server for receiving rendering results and sending rendering parameters.

3.3.2 Volume Visualization

Rendering for the tiled display and the clients are similar. The few differences are due to the hardware capability and the interaction model. The tiled display is monocular with 8k resolution featuring the traditional mouse keyboard interaction, while the clients have stereoscopic displays with resolution 960x540 featuring hand gesture interaction. Due to the fact that our hardware setup cannot render 8k images at the desired interactive rate, which is at least 10fps, users explore the focused region of the volume through the clients. When the users finish exploring, they can push the rendering parameters to the tiled display to obtain a global image with extreme quality. In the case where multiple users are pushing the rendering parameters to the tiled display, the tiled display uses the last received rendering parameters or that of the lead user. We provide the following visualization for our users to explore the volume: 3D streamline tracing and direct volume rendering, which provides isosurface visualization.

The direct volume rendering algorithm we use is raycasting, as shown in Figure 5. The basic algorithm is to cast a ray through the volume for each pixel on the viewport. As each ray steps through the volume, the resampled scalar values along the ray are mapped into colors and opacities using a transfer function, then the colors are composited according to a simple compositing equation to construct the final color value of the ray. Note that isosurface rendering can be easily done with raycasting using a step function as the opacity transfer function and preintegration to ensure quality [21].

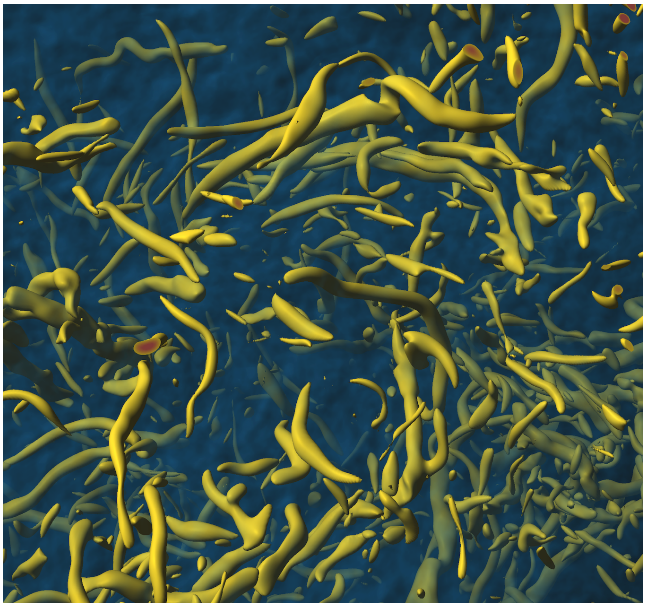

Figure 5: Direct volume rendering and isosurface rendering. We used the forced isotropic turbulence dataset from the John Hopkins Turbulence databases [1].

Tracing streamlines is a common step for analyzing steady flow vector field data. Streamlines are curves tangent to the velocity field, giving a sense of the flow structure to the viewer. To trace streamlines, we first require a seed point to be selected. Each seed point becomes a massless particle injected into the velocity field. Using the Runge–Kutta 4th order algorithm, we move the injected particle following the velocity field, and the trajectory of the particle becomes a streamline. After a streamline is defined, we render it into a tube for better visualization with lighting. The tubes are superimposed in the direct volume rendering so users can better analyze the relationship between the scalar field and the vector field, as shown in Figure 8.

Each client maintains its own virtual camera for exploring the volume. Monocular rendering is done directly using the virtual camera. For stereoscopic rendering, left and right cameras are generated by shifting the virtual camera in order to simulate the left and right eyes. In our setting, we want the focused region of the volume to be approximately 1 meter away from the user in the virtual reality. Due to the fact that the average human interpupillary distance (IPD) is 0.063 meter, we can formulate the shift distance of the left and right cameras as such: $SD = VD * 0.5 * IPD$, where SD is *shift distance* and VD is *view distance* of the virtual camera. The generated left and right eye images are then sent to the corresponding client.

3.3.3 Focused Region Identification

To identify the focused region of a client, we need a client to send an image captured by its RGB camera. This image indicates which portion of the tiled display the user is interested in exploring and analyzing. Since the image captured by the RGB camera is the original image with a combination of translation, rotation, scaling and distortion, we have employed a vision-based technique to find the corresponding region of the captured image I_{cap} in the fully displayed image I_{render}. First, scale invariant local image features such as SIFT [20] are computed for both of the images. The descriptor of each feature in I_{cap} is then used to compare with the features in I_{render} by calculating the Euclidean distances of each pair. Those feature pairs with small distance are selected as matched image features. Since more than hundreds of such matching can be found in our scenario, we adopt RANSAC [13] to calculate the homography matrix H that transforms I_{cap} to its corresponding region R in I_{render}, as shown in Figure 6. Finally, R is displayed in the HMD for the user to explore.

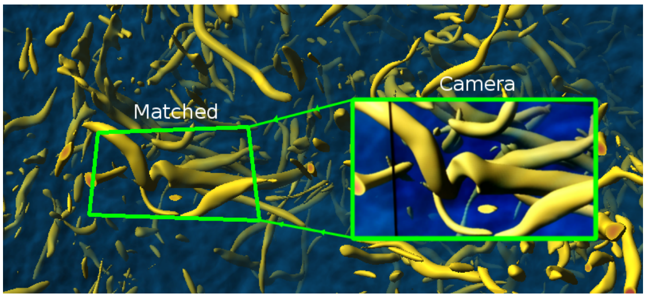

Figure 6: Identifying the focused region of a client. The green box on the right indicates the image taken by the client's camera, and the green box on the left shows the output of our computer vision algorithm, which correctly indicates the focused region.

4 Tests and Discussion

Based on the design described in the last section, we have created a prototype system. A visual analysis session with isosurfaces can be seen in Figure 1. We used the forced isotropic turbulence flow data from the John Hopkins Turbulence databases, which contain many small features. For our purpose, we only used a single time step and the central portion 512^3 of the volumetric data. The velocity field was first processed by a Gaussian filter with $\sigma = 10$ and then Q-criterion was calculated from the velocity field. The visualization results were generated using the Q-criterion variable. As shown, a lot of small details can be easily identified with the 8k resolution display wall, and local data exploration with HMDs in stereoscopic view is achieved in acceptable rendering rates (10-20 frames per second), allowing users to apply a newly derived visualization parameter setting to the large tiled display for a new overview. While hand gesture inputs helped users to intuitively interact with the data visualization, due to random noises within a depth sensor of the HMD device we used, sometimes the tracking did not work accurately. We are planning to utilize another depth sensor for the HMD in order to reduce the noise.

As shown before, our system enables direct volume rendering. However, analytics are sometimes better off with 2D graphs for quantitative analysis. To better support analytics, we are providing a 2D contour plot on an arbitrary slice for scalar field analysis, and a 2D line chart of selected streamlines for vector field analysis.

A contour plot is generated on an arbitrary slice of the volume, as shown in Figure 7. Our system already has the capability of generating such a contour plot. The remaining task is to implement an intuitive user interface. The simple approach is to also use a view aligned slice, and then the user can utilize a slider to determine the depth of the slice. However, it is hard to generate axis aligned contour plot with this approach. With a stereoscopic display and a depth sensor, we hope to develop more natural user interface for specifying the slice and isovalues of the contour plot.

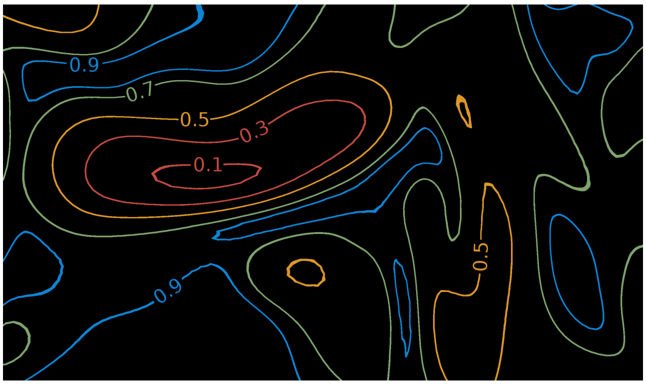

Figure 7: Contour plot of a slice of the volume.

For analyzing the streamlines, we plan to provide a 2D line chart to visualize a select scalar value changes along the streamline. A concept chart is shown in Figure 8. We can show this chart when the user places a seed point, or when the user is selecting an already generated streamline. The generated chart should be an overlay on top of the volume rendered image so that the users can analyze the selected streamline in both the physical space and the streamline space. Again, an intuitive user interface is required. Before we have a capable device which allows high precision 3D picking with bare hands, we can use gaze as input. That is, we always select the streamline that is at the center of the user's viewport.

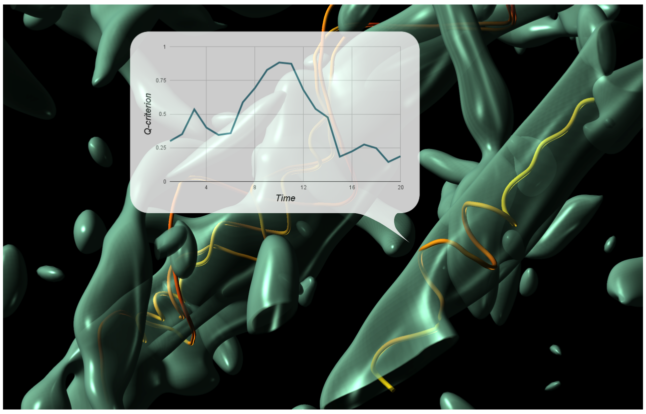

Figure 8: Streamline visualization by tracing particles in the velocity field. The 2D line chart shows how Q-criterion changes along the selected streamline. The horizontal axis is the number of steps taken on tracing the particle, while the vertical axis is the Q-criterion value. Note that we can replace the vertical axis with any other scale variable value.

In order to evaluate and consider the pros and cons of our system, we will design a controlled experiment and perform a quantitative study. One of the common ways to evaluate the effectiveness of visualizations quantitatively is to measure how long it takes for each participant to complete some tasks such as finding specific features in the visualization. In the work by Laha et al. [19], other tasks such as judging the position of some features within the visualization and estimating the values of the visualization may be used for evaluating isosurface visualizations. We will compare conducting tasks in the environment with HMDs and a large tiled display to the environment only with a large display. Through such a comparison study, we hope to show using the see-through HMDs with the large display can help enhance the interactive data visualization for analysis tasks.

5 Conclusion

In this workshop paper, we present our preliminary design and results for interactive data visualization and analysis with isosurface and streamlines. We believe that the utilization of the affordable see-through HMD devices together with a low-cost tiled HDTV display can play an important role for analysis tasks. As we mentioned, the future development and application of the system design can be considered from a variety of aspects. We will keep seeking in what way we can make the most use of the immersive environment and pave the way for immersive analytics.

Acknowledgements

The work is supported in part by U.S. National Science Foundation through grant DRL-1323214, U.S. Department of Energy through grant DE-SC0007443, Japan Society for the Promotion of Science under Grant-in-Aid for Challenging Exploratory Research No. 15K12034, and a Grant-in-Aid for the Leading Graduate School Program for "Science for Development of Super Mature Society" from the Ministry of Education, Culture, Sports, Science and Technology in Japan.

References

[1] Johns Hopkins Turbulence Database. http://turbulence.pha.jhu.edu/, accessed January 2016.

[2] Meta1, Meta Company. https://www.metavision.com/, accessed February 2016.

[3] Xinerama. https://sourceforge.net/projects/xinerama/, accessed January 2016.

[4] C. Andrews, A. Endert, B. Yost, and C. North. Information visualization on large, high-resolution displays: Issues, challenges, and opportunities. *Information Visualization*, 10(4):341–355, 2011.

[5] R. Ball and C. North. Effects of tiled high-resolution display on basic visualization and navigation tasks. In *CHI Extended Abstracts on Human Factors in Computing Systems*, pages 1196–1199, 2005.

[6] R. Ballagas, M. Rohs, and J. G. Sheridan. Sweep and point and shoot: Phonecam-based interactions for large public displays. In *CHI Extended Abstracts on Human Factors in Computing Systems*, pages 1200–1203, 2005.

[7] T. Ballendat, N. Marquardt, and S. Greenberg. Proxemic interaction: Designing for a proximity and orientation-aware environment. In *Proceedings of ACM International Conference on Interactive Tabletops and Surfaces*, pages 121–130, 2010.

[8] H. Benko, E. W. Ishak, and S. Feiner. Collaborative mixed reality visualization of an archaeological excavation. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality*, pages 132–140, 2004.

[9] S. Bryson and C. Levit. The virtual wind tunnel. *IEEE Computer Graphics and Applications*, 12(4):25–34, 1992.

[10] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, K. Marriott, F. Schreiber, et al. Immersive analytics. In *Proceedings of IEEE Big Data Visual Analytics*, pages 1–8, 2015.

[11] K. Cheng, J. Li, and C. Müller-Tomfelde. Supporting interaction and collaboration on large displays using tablet devices. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 774–775, 2012.

[12] R. A. Clark, Y.-H. Pua, K. Fortin, C. Ritchie, K. E. Webster, L. Denehy, and A. L. Bryant. Validity of the Microsoft Kinect for assessment of postural control. *Gait & Posture*, 36(3):372–377, 2012.

[13] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[14] V. Jaswal. CAVEvis: Distributed real-time visualization of time-varying scalar and vector fields using the CAVE virtual reality theater. In *Proceedings of the 8th Conference on Visualization*, pages 301–308, 1997.

[15] B. Jeong, L. Renambot, R. Jagodic, R. Singh, J. Aguilera, A. Johnson, and J. Leigh. High-performance dynamic graphics streaming for scalable adaptive graphics environment. In *Proceedings of ACM/IEEE Conference on Supercomputing*, 2006.

[16] G. P. Johnson, G. D. Abram, B. Westing, P. Navr'til, and K. Gaither. Displaycluster: An interactive visualization environment for tiled displays. In *Proceedings of IEEE International Conference on Cluster Computing*, pages 239–247, 2012.

[17] S. Johnson, M. Gibson, and B. Mutlu. Handheld or handsfree?: Remote collaboration via lightweight head-mounted displays and handheld devices. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1825–1836, 2015.

[18] N. K. Krishnaprasad, V. Vishwanath, S. Venkataraman, A. G. Rao, L. Renambot, J. Leigh, A. E. Johnson, and B. Davis. Juxtaview - A tool for interactive visualization of large imagery on scalable tiled displays. In *Proceedings of IEEE International Conference on Cluster Computing*, pages 411–420, 2004.

[19] B. Laha, D. Bowman, and J. Socha. Effects of VR system fidelity on analyzing isosurface visualization of volume datasets. *IEEE Transactions on Visualization and Computer Graphics*, 20(4):513–522, 2014.

[20] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[21] E. B. Lum, B. Wilson, and K.-L. Ma. High-quality lighting and efficient pre-integration for volume rendering. In *Proceedings of VisSym*, pages 25–34, 2004.

[22] T. Marrinan, J. Aurisano, A. Nishimoto, K. Bharadwaj, V. Mateevitsi, L. Renambot, L. Long, A. Johnson, and J. Leigh. SAGE2: A new approach for data intensive collaboration using scalable resolution shared displays. In *Proceedings of International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 177–186, 2014.

[23] K. Moreland, B. Wylie, and C. Pavlakos. Sort-last parallel rendering for viewing extremely large data sets on tile displays. In *Proceedings of the IEEE Symposium on Parallel and Large-Data Visualization and Graphics*, pages 85–92, 2001.

[24] T. Ni, D. A. Bowman, and J. Chen. Increased display size and resolution improve task performance in information-rich virtual environments. In *Proceedings of Graphics Interface*, pages 139–146, 2006.

[25] T. Ni, G. S. Schmidt, O. G. Staadt, M. Livingston, R. Ball, and R. May. A survey of large high-resolution display technologies, techniques, and applications. In *Proceedings of Virtual Reality Conference*, pages 223–236, 2006.

[26] K. Reda, A. Knoll, K. Nomura, M. E. Papka, A. E. Johnson, and J. Leigh. Visualizing large-scale atomistic simulations in ultra-resolution immersive environments. In *Proceedings of IEEE Symposium on Large-Scale Data Analysis and Visualization*, pages 59–65, 2013.

[27] M. Rodrigue, A. Waranis, T. Wood, and T. Hollerer. Mixed reality simulation with physical mobile display devices. *IEEE Virtual Reality*, (1):105–110, 2015.

[28] J. Sanneblad and L. E. Holmquist. Ubiquitous graphics: Combining hand-held and wall-size displays to interact with large images. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 373–377, 2006.