Nine Years of the Tromsø Display Wall

Tromsø Large Display Wall

Tromsø Large Display Wall with users studying weather visualizations

Otto J. Anshus
Dept. of Computer Science
University of Tromsø,
Tromsø, Norway
otto@cs.uit.no

John Markus Bjørndalen
Dept. of Computer Science
University of Tromsø,
Tromsø, Norway
jmb@cs.uit.no

Daniel Stødle
Northern Research Institute
Tromsø, Norway
daniel@norut.no

Lars Ailo Bongo
Dept. of Computer Science
University of Tromsø,
Tromsø, Norway
larsab@cs.uit.no

Tor-Magne Stien Hagen
DIPS,
Bodø, Norway
tsh@dips.no

Yong Liu
NSD,
Bergen, Norway
yong.liu@nsd.uib.no

Bård Fjukstad
Dept. of Computer Science
University of Tromsø,
Tromsø, Norway
bard.fjukstad@uit.no

Lars Tiede
Dept. of Computer Science
University of Tromsø,
Tromsø, Norway
lars.tiede@uit.no

Abstract
A display wall is a wall-sized tiled display comprising several smaller displays and computers. Desktops and applications are hardly designed to control and do output on many displays, let alone on display walls. Consequently, performance suffers, and the human-computer interaction makes a users life miserable. The last nine years we have constructed a display wall from commodity hardware, and adapted existing and developed new approaches on how to control and use display walls. We report on some of the systems built and lessons learned. Most systems were done from scratch to better understand the issues involved.

CHI 2013 Extended Abstracts, April 27 – May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Author Keywords
Display wall; programmable displays; device-free user interaction; high-performance visualizations;

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction
Programmable standalone displays are growing in number, size, and resolution, while becoming lighter, thinner and cheaper. Commodity displays are increasing in pixel count from 2Mpixels to 8Mpixels, and display walls at 20-1000Mpixels can realistically be put together. The large size and high-resolution allows users at a distance to view the totality of data, while letting users closer by the display read the fine details.

Software designed for PCs can be adapted and ported to a display wall. However, the performance will suffer when centralized models are used because they may not scale to many enough displays. The user experience will suffer because buttons, menus and notifications are put at awkward locations on the large display, and because user input is done in inflexible and inconvenient ways not taking sufficiently into account the size of the display wall and that it can have multiple users.

Figure 1. Navigating a large collection of comics on the Tromsø Large Display Wall.

Figure 2. Tromsø Portable Display Wall in public use.

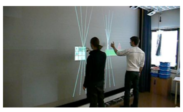

Figure 9. Camera-based triangulation to find objects in front of display wall

Tromsø Display Walls
Construction of the Tromsø Large Display Wall, shown in figure 1, was started in 2004-2005. It consists of 28 projectors arranged in a 7x4 grid creating a 22-megapixel rear-projected display. A 28-computer display cluster running Linux is used to drive the projectors. The network is a switched gigabit Ethernet. Sixteen cameras and eight Mac minis are used to locate objects in front of the display wall, see figure 9. This is used to enable device- and touch-free interaction with the display wall. A set of microphones is used to pick up audio-based commands to the system.

In 2012, we applied lessons learned to build the Tromsø Portable Display Wall, Fig. 2, comprised of a fold-up canvas, four projectors in a 2x2 grid and four Mac minis and a Microsoft Kinect camera with custom software for multi-point and touch-free input.

Architectures and Designs for Display Walls
Using more than one display at a time can be done by connecting several displays to one computer in a multi-monitor setup. By tiling several displays and computers, large display walls [5] can be built.

The use of a display cluster for the Tromsø Display Wall precludes running existing software directly. To relatively easily support existing desktop applications, we use VNC [8]. A single computer runs all applications as well as a VNC server with a virtual framebuffer covering the display wall. Each display node runs a viewer requesting updates from the server. Because of the scanning of a large framebuffer, and the need to compress frame updates before they are sent to the viewers, the server computer sees a heavy CPU load. The frame rate at the display wall is good enough for static documents, but for videos and image panning, the framerate at 1-4 FPS becomes too low for an interactive user experience.

To increase the performance when using VNC, we did several optimizations and redesigns of VNC [9]. Through a combination of optimizing an existing VNC implementation, and letting the server push updates to viewers, we were able to increase the framerate to 10-15 FPS. We also tried using multi-threaded approaches executing on GPU's, but could not gain better performance from doing this.

To relieve the VNC server from some work, we developed a decentralized VNC approach [6]. In this approach, the VNC viewers at the display nodes share data with each other when they can. For videos, this gives no benefit, but for panning large documents, images and maps, reuse of data between viewers results in a significant increase in FPS giving a satisfying interactive response for users.

In some cases, it is acceptable to have lossy visualizations on the display wall. By recording an on-the-fly video of output from an application, and streaming the video to the display wall, we achieved 30 FPS [10].

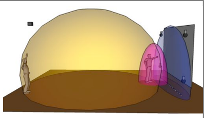

Figure 3. Three different interaction spaces in front of a display wall

Figure 10. Cameras along the floor detect objects along the wall.

However, for best performance, new applications should be written as parallel applications executing on the display nodes. We describe a few of these later in this paper.

We have also developed models suitable both for a set of standalone displays and for display walls. We describe the NAD [12] model later in this paper. A work in progress is the display cloud model [7]. From a mobile device a user can select local and remote programmable displays, and add them to a user specific *cloud display* where the user can arrange them relative to each other as needed. The user can then select remote computers and steer output to his cloud display. The physical displays can simultaneously show output from several user PCs.

Interaction

Interaction with display walls can be accomplished using input devices such as mice, keyboards and tablets, or in a device-free manner using only the user's presence to provide input. With devices, each user needs a device to provide input, limiting the number of simultaneous users to the number of input devices available. Other issues include limited battery life and confusion over where the needed device is physically located. For these reasons, our research in interaction has been focused on developing systems for device-free interaction with display walls. We have taken a systems approach to interaction research. Rather than focusing on particular use-cases or whether users like or dislike a particular gesture, objective measurements of system performance have been made, with a focus on metrics like accuracy, precision and latency.

We define the concept of an Interaction Space [13] as a volume within which user interaction is detected. Different interaction spaces can cover differently sized and overlapping areas, with each interaction space existing orthogonally to the computers they act on and supporting one or several users at a time. The input mechanism is no longer bound to a single computer, but instead becomes a property of the space users occupy. Based on this concept, three systems for device-free interaction have been developed: Camera-sense, Snap-detect and Arm-angle, see Figure 3.

The Camera-sense system uses 16 FireWire cameras to create an interaction space directly in front of the display wall, see figure 10. The system determines the location and 3D extent of multiple objects such as hands or fingers, which is then interpreted by applications as gestures of different kinds, such as panning or zooming. The system requires no markers and enables fully touch-free and multi-point interaction with the display wall.

The Snap-detect system uses four microphones to create an interaction space that encompasses an entire room. The system detects users snapping their fingers or clapping their hands, and determines the approximate 2D location of the user in relation to the display wall. Applications can use this input in different ways, such as zooming in or out at the location where a user is, or to make windows move to a user's location.

The Arm-angle system uses a single ceiling-mounted, pan-tilt-zoom (PTZ) camera to create a small, on-demand interaction space. A user snaps her fingers to make the PTZ-camera bring the user into its field of view. The system then determines the angle at which

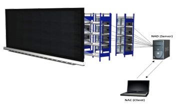

Figure 4. NAD Architecture.

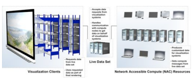

Figure 5. WallScope Architecture.

Figure 6. WallGlobe: Navigating Earth at High Resolution.

Figure 7. WallView: Navigating a gigapixel image on the Tromsø Display Wall.

the user's arm is pointing, without the use of markers. This information is then used by applications to identify far-way targets on the display wall.

The three systems all use a shared event bus we have developed to communicate their state to applications.

NAD - Network Accessible Displays

The Network Accessible Display (NAD) model [12] enables integration of a computer with any display, including a display wall (Fig. 4). By using NAD, a display wall can be used over a network connection as if it was physically connected to a laptop/desktop computer.

A user visits a web page containing a link to initiate a customization of the user's computer. When clicking on the link the computer will be customized by software transparently downloaded from the display wall front-end. This software enables the user to select one or several parts of the local desktop display to be mirrored onto the display wall. The display protocol between the computer and the display wall is pixel-based, ensuring cross-platform compatibility. The software is implemented in Java, and it supports Windows, Linux and Mac OS X.

WallScope, WallGlobe, & WallView

To support interactive visualization of data on the display wall we created WallScope [11]. WallScope, see Fig. 5, comprises a display-side and a compute-side connected through a live data set.

The display-side is comprised of a set of display clients (one per display-node) driven by a state server that broadcasts the state of the visualization a configurable number of times per second.

The display clients request data from the live data set. The data set is live in that it receives requests for processed data from the display-side that it translates into compute requests and forwards to local or remote compute resources that are connected to the system. The results of these computations are transferred to the display-side where it is included in the rendering and displaying.

Currently we have developed two visualization systems using WallScope. WallGlobe, fig. 6, is a visualization system for planets. It is currently used to visualize the Earth using data from several different resources. The back-end computational resources of WallScope are used to do on-demand processing on the data for the display nodes. For example, we do this to manipulate image tiles before they are used as part of the rendering of the Earth.

WallView is a visualization system used to visualize high-resolution images, see Fig. 7. One of the images used by WallView is a 13.3 Gigapixel image showing Tromsø, Norway.

Weather Forecasting

We have ported the Norwegian Meteorological Institute's DIANA system [1] for use on the Display Wall (Fig. 8). The DIANA system is executed on a standalone computer. In this case, the Display Wall is used just as a very large high-resolution display. We have conducted several experiments with operational weather forecasters using the Display Wall for tasks related to regular operational weather forecasting [2].

DIANA has demonstrated that resolution matters. A display wall allows for both a combination of good

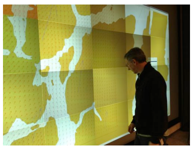

Figure 8. Weather forecasting

Figure 11. Genomic data exploration by our collaborators at Princeton using our reimplementation of [4].

overview of the forecast for a large region, and at the same time access to the finest details from the numerical models. Actual forecasters report to us that the display wall is very useful in reducing the time it takes to look for details in a large geographical region. This is because both the whole region and sufficient details are visible simultaneously. A forecaster no longer has to do frequent panning and zooming.

Porting DIANA was simple because DIANA already existed for Linux. We use DIANAs GUI. This results in having to moving a cursor over most of the wall to select buttons and menus, including close by the ceiling. Touch based cursor positioning without rearranging the GUI elements is not usable.

Genomic Data Exploration

Interactive visual analysis by biology and bioinformatics researchers is critical in extracting biological information from genomic data [3]. Many of the data analysis techniques developed share a common need to identify and examine patterns in large visualizations comprising of thousands of datasets with millions of samples. In addition, such analysis is often collaborative.

High-resolution display walls can be used to simultaneously display multiple datasets (figure 11). This approach has been demonstrated to provide biological insights that are not apparent using techniques designed for low-resolution displays [4]. Display walls have also been identified as an important enabling technology for other genomic data visualizations [3].

To our knowledge, few genomics data visualization approaches utilize high-resolution display walls. We have been approached by several biology and bioinformatics researchers that would like to use our display wall for their analysis. We are therefore currently working on two approaches for genomics data visualization on display walls. First, we are building a middleware system for scalable integrated visualization of thousands of datasets [14]. Second, we are developing special purpose applications for interactive 3D exploration of very large data sets.

There are technical challenges in achieving good interactive performance of very large data visualizations. There are also several HCI challenges including how to ensure that human experts can detect patterns in the data and not be overwhelmed by the amount of data shown, how to utilize the large size of a display wall for collaboration, and how to interactively control the different visualizations.

Conclusions

We have so far primarily focused on researching and developing architectures, designs, and implementations of a set of new applications and systems for display walls. Doing this helps us understand the issues better.

Display walls are complex instruments with many challenges of both a principal and practical nature. Much research remains to be done to understand how to make platforms that can be used by applications and systems to scale in performance and flexibility to the large size and high resolution of a display wall.

Acknowledgements

This work was funded in part by the Norwegian Research Council, projects 187828, 159936/V30, 155550/420, and Tromsø Research Foundation.

References

[1] Martinsen, E.A. and Foss, A. and Bergholt, L. and Christoffersen, A. and Korsmo, H. and Schulze, J. *Diana: a public domain application for weather analysis, diagnostics and products.* 5th Annual Meeting of the European Meteorological Society, 2005

[2] Fjukstad, B. and Anshus, O. *High resolution numerical models on a Display Wall.* 7th EMS Annual Meeting / 8th ECAM. 2007.

[3] S. I. O'Donoghue, A.-C. Gavin, N. Gehlenborg, D. S. Goodsell, J.-K. Hériché, C. B. Nielsen, C. North, A. J. Olson, J. B. Procter, D. W. Shattuck, T. Walter, and B. Wong, "Visualizing biological data-now and in the future.," Nature methods, vol. 7, no. 3 Suppl, pp. S2–4, Mar. 2010.

[4] M. Hibbs, G. Wallace, M. Dunham, K. Li, and O. Troyanskaya, "Viewing the Larger Context of Genomic Data through Horizontal Integration," in 2007 11th International Conference Information Visualization (IV '07), 2007, pp. 326–334.

[5] Kai Li, Han Chen, Yuqun Chen, Douglas W. Clark, Perry Cook, Stefanos Damianakis, Georg Essl, Adam Finkelstein, Thomas Funkhouser, Timothy Housel, Allison Klein, Zhiyan Liu, Emil Praun, Rudrajit Samanta, Ben Shedd, Jaswinder Pal Singh, George Tzanetakis, and Jiannan Zheng. Building and Using A Scalable Display Wall System. Computer Graphics and Applications 20(4):29-37, December 2000.

[6] Stødle, D., Bjørndalen, J.M., Anshus, O. De-Centralizing the VNC Model for Improved Performance on Wall-Sized, High-Resolution Tiled Displays. In Proceedings of the Norwegian Informatics Conference, NIK 2007, pp. 53-64, Oslo, Nov. 2007.

[7] Tiede, L., Bjørndalen, J.M., Anshus, O., Cloud Displays for Mobile Users in a Display Cloud, HOTMOBILE 2013, Jekyll Island, USA, Feb. 2013.

[8] http://ipinfo.info/html/vnc_remote_control.php

[9] Yong Liu, John Markus Bjørndalen, and Otto. J. Anshus. "Using multithreading and server update pushing to improve the performance of VNC for a wall-sized tiled display wall". In InfoScale 2009: The 4th International ICST Conference on Scalable Information Systems, 2009.

[10] Yong Liu, Otto J. Anshus, Phuong H. Ha, Tore Larsen, and John Markus Bjørndalen. "Multistream a cross-platform display sharing system using multiple video streams". Distributed Computing Systems Workshops, ICDCS'08. 28th International Conference, pages 90-95, 2008.

[11] Tor-Magne Stien Hagen, Daniel Stødle, Otto Anshus. On-demand High-performance Visualization of Spatial Data on High-Resolution Tiled Display Walls. In Proceedings of the International Conference on Information Visualization Theory and Applications, pp. 112-119, May 2010.

[12] Tor-Magne Stien Hagen, Espen Skjelnes Johnsen, Daniel Stødle, John Markus Bjørndalen, Otto Anshus. Liberating the Desktop. In Proceedings of the First International Conference on Advances in Computer Human Interaction, ACHI, pp. 89-94, Feb. 2008.

[13] Stødle, D., Troyanskaya, O., Li, K., Anshus, O.J., Device-Free Interaction Spaces. In Proceedings of IEEE International Symposium on 3D User Interfaces 2009, Lafayette, Louisiana, USA, March 14-14, 2009

[14] Bongo, L. A., Bulk Synchronous Visualization. The 2013 International Workshop on Programming Models and Applications for Multicores and Manycores (PMAM'2013), ACM. February 23, 2013, Shenzhen, China.