# BayScope

*Yun Jin*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-148
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-148.html

May 31, 2012

Acknowledgement

This thesis would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.
First and foremost, I wish to thank my advisor, Bjoern Hartmann, professor in Department of Computer Science. He has been supportive since the day I began working on the project. He guided me and led me to understand how to do this project.
Dr. Jitendra Malik, Professor in Dept. of Computer Science, for his kindness to be willing in my thesis committee.
Mr. Viraj Kulkarni, Mr. Hong Wu for their unselfish and unfailing support as

my project colleague; it’s a great journey to work with all of them. Their insights have always enlightened me.

# BayScope

Yun Jin

23271229

Department of Electrical Engineering and Computer Science

*In partial fulfillment of the*

*requirements for the Degree of*

*Master of Engineering*

*University of California at Berkeley*

Very-high-resolution wall-sized displays are becoming affordable and widespread throughout the design, education, business, and entertainment industries. Such displays offer new opportunities not only for technical large data visualization, but also for collaborative workspaces, such as a meeting room, a design studio. In such environment, a combination of multi-user input, new interaction techniques and applications for large displays is pressing. BayScope is aimed at exploring new techniques and applications with multiple user input to enhance interaction and collaboration between groups of people and a large wall-sized display in a collaborative workspace. In this paper, we first focus on four different interaction techniques to share and manipulate content on large displays by using mobile phones. We compare speed and accuracy of these techniques by gathering and analyzing data from a user study with ten participants. From the data, we find touchpad is the most appropriate interaction technique, so we choose touchpad as our primary technique for developing applications. Since the cost of developing new applications on large displays is high, we investigate how to adapt existing web applications without access to the applications' source code. Our approach is to execute multiple copies of several existing applications in parallel and keep them synchronized, and they can be used by multiple users through touchpad interaction. Finally, we conduct a user study with ten participants to interact with our system. The evaluation result shows that it is convenient and efficient to use our system in a collaborative environment.

## Table of Content

# 1. Introduction

High-resolution wall-sized displays are widely used to visualize large data sets in scientific areas. For instance, astrophysicists navigate and annotate very large telescope imagery; biochemists use them to visualize and interact with complex molecules; crisis management centers interact with highly detailed maps of very large areas. In addition to the application scientific areas, large displays are also useful in our daily lives, such as meeting rooms and collaborative workspaces where participants are willing to share their ideas and make comments on the displays at the same time. In a collaborative room, large displays would provide enhanced value if users could interact across multiple interactive computing devices, such as a large display wall, tablet computers and smartphones. Users also prefer to use large display walls in a collaborative room with multi-user inputs.

Interaction with large displays is different from desktop computers which have small displays. For small displays, they can be operated with traditional input devices like keyboards and mice. However, for large wall-sized displays, they pose different sets of trade-offs. Inputs should be location independent and should require neither a hard surface such as desk nor clumsy equipment: users should have the ability to move freely in front of the displays and interact at a distance. This precludes use of conventional input devices such as keyboard and mice which are used for small displays. However, using mobile phones as inputs for large wall-sized displays enables users move freely in front of the displays and interact with them. Since large wall-sized displays with multiple user input devices have become more common, the need for applications that take advantage from such facilities is essential. Existing applications often do not easily scale to wall-sized displays for several reasons. First, wall-sized displays often require computing clusters, where each machine drives a subset of displays, while most existing applications cannot be run on distributed cluster environments. Second, existing applications are frequently restricted to single-user input while large displays are especially appropriate to multiple users' collaboration. Third, existing applications are usually written with single screen resolution in mind. Resizing application interface windows to large resolution would break usability. One approach to address this challenge is to develop new UI framework and rewrite applications using framework. While this approach requires a large upfront investment into re-engineering of common applications, so it takes long time to do. Another approach is to investigate whether and how existing applications can be adapted to run on multi-display walls without modifying applications and source code. While such adaptations cannot provide the full advantages of rewriting applications, they are easier and faster to produce and they can also be used for applications for which source code is not available.

Our goal is to develop a collection of application instances whose contents and views are coordinated and can be used by multiple users through mobile remote control on large wall-sized displays. In this paper, we first present four interaction techniques between mobile phones and wall-sized displays in section 3. Then, we evaluate and compare speed and accuracy of these four interaction techniques by analyzing the data results from user study with ten participants. After careful evaluation, we choose touchpad as our major interaction technique. In section 4, we introduce adapted existing web applications on multi-display walls. And we also propose mobile remote controller as input to navigate the applications. Finally, we design several tasks and open questions for users and thus the result shows that users prefer to use our system in a collaborative environment.

## 2. Related Work

Our work is at the intersection of many HCI research areas, such as large displays, interaction techniques and multi-user input. This section highlights strongly related work to our project.

### *2.1. Large Displays*

Large displays have been the focus of much research and evaluation over ten years. Ni *et al.* [1] survey hardware configurations, rendering techniques as well as interaction techniques for many different types of large displays. Overall, the body of empirical work on large displays suggests that users can generally benefit from their use. It also shows that the design of interaction techniques has to be carefully adapted to the characteristics of these large displays and to their context use. Early studies investigated how users could benefit from larger displays in different settings. Baudisch *et al.* [2] found advantages of using a large focus and context screen over zooming and overviews to information from large documents such as maps and schematics of circuit boards. Ball *et al.* [3] discovered that for pan-zoom tasks, such as navigating to a known location, searching for specific targets, users perform better with larger viewport sizes that requires less virtual navigation, promoting physical navigation instead. Results from other recent studies suggest that large displays are also beneficial for information visualization and analysis tasks thanks to the larger amount of data that can be displayed. [4, 5]

### *2.2.Interaction Techniques*

A lot of recent research effort has gone into developing new interaction techniques that substitute mouse and keyboard input. Khan *et al* [6] proposed a new GUI widget, called the Frisbee, designed to provide comfortable manipulation of remote parts of a large display workspace for arm's-length interaction. There is also some related work by using mobile phones to interact with large displays. Boring *et al.* [7] proposed Touch Projector, a system that enables users to interact with remote screens by taking a live video image on their mobile phones. It also allows users to manipulate content on distant displays that are unreachable. Chang *et al.* [10] created Deep Shot which supports two novel and intuitive interaction techniques, deep shooting and deep posting, for pulling and pushing work states, respectively, using a mobile phone camera. By using these two interaction techniques, users can migrate tasks across devices such as mobile phones and laptops. Ballagas *et al.* [11] proposed two interaction techniques called "sweep" and "point & shoot" to enable users to use mobile phones to interact with large public displays. Users can use mobile phones like optical mouse without pointing the camera at the display and they can also select objects using visual codes to set up an absolute coordinate system on the display by using mobile phones.

Some other research focused on navigation techniques on large displays, such as pan and zoom navigations. Malacria *et al.* [8] proposed clutch-free panning and integrated pan-zoom control on large touch-sensitive surfaces by drawing circles clockwise or counterclockwise. And their experiments suggest that these two techniques outperform flicking and rubbing techniques. From this paper, we found that navigation techniques should also be taken into account in designing interaction techniques for large displays. Some researchers also compared and analyzed different interaction and navigation techniques. Nancel *et al.* [9] studied and evaluated different families of location independent mid-air interaction techniques for pan-zoom navigation on wall-sized displays. We used similar analytical methods to compare our interaction techniques.

### *2.3.Multi-user Input*

Users can gain from multi-user input to wall-sized displays since it can help users collaborate with each other by using different inputs to point and manipulate contents together. Hartmann [12] introduced an interactive tabletop system that enhances creative collaboration across physical and digital artifacts. Wallace [13] proposed a virtually shared model that enables users to use remote displays as extensions of their local displays and to allow multiple users to use multiple cursors and keyboards to input and control shared applications and their windows simultaneously. In general, multi-user input helps multiple users manipulate, discuss, and share contents together on large displays.

## 3. Interaction Techniques

### *3.1.Implementation*

In this paper, we introduce four interaction techniques between mobile devices and large wall-sized displays. All network communication between these devices is handled through the Open Sound Control (OSC) library. Afterwards, we build a test application to evaluate these four interaction techniques. The application is developed using Processing API which runs on top of Java. It consists of a canvas that is spread over the entire screen space of the display. Multiple images of various sizes are shown on the canvas (Figure 1). Users can move the images around the canvas by using any of the four interaction techniques we have implemented.

### *3.2.Hardware Details*

Our system setup includes a display wall built out of 6 Samsung 460UTN-UD monitors in a 3x2 configuration, as shown in Figure 2. Each monitor offers a full HD resolution of 1920x1080 pixels. The entire wall, thus, offers a resolution of 5760x2160 pixels. This wall is driven by a single Mac Pro - 2.4 GHz, Quad-Core Intel Xeon, 8 GB RAM and two ATI Radeon HD 5770 GPUs. A Microsoft Kinect controller connected to a Windows 7 machine provides gesture input. Both systems, the Mac Pro and the Windows machine, are connected to the Internet over the same Wi-Fi network.

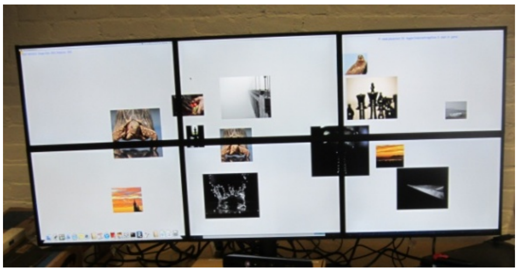

**Figure1.** Test application canvas

**Figure2.** Our setup containing six displays

### *3.3.Four Interaction Techniques*

In this paper, we present and evaluate four interaction techniques, which are 4-way remote control (Figure 3), Omni-directional remote control (Figure 4), touchpad (Figure 5) and mobile phone coupled with Kinect (Figure 6). The first one technique contains four navigating buttons and a click button in the center. The four directional buttons are used for moving cursors, while the click button is responsible for click. The omni-directional remote control is an improvement to the 4-way remote control. Users are not constrained to the orthogonal axes, instead, they can move in any directions in the circle, and the center of the circle is for click. For touchpad technique, we use relative mapping for mapping the touch input to on-screen coordinates. And users could click to confirm by pressing and holding the screen. The last interaction technique is a combination of mobile phones and Kinect. The Kinect is used for quickly moving the cursor and the mobile device is responsible for fine tuning.

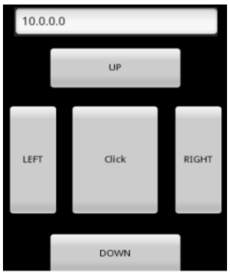


**Figure3.** 4-way remote control


**Figure4.** Omni-directional remote control


**Figure5.** Touchpad

**Figure6.** mobile phone coupled with Kinect

When using the smaller screen of the mobile phone to control the much larger display wall, an important factor that demands due consideration is the control-display ratio. A smaller control-display ratio affords fine granularity in interaction which, in our case, enables the user to finely control the pointing location on the wall. This, however, comes at a trade-off of speed of motion. A large control-display ratio affords faster movement across but comes at the expense of accuracy. We reason that an effective interaction technique should offer the user a choice between fine grained and coarse grained interactions. This is especially important for wall-sized displays as the penalty incurred for the wrong choice of granularity increases with display size. Hence, all of our interaction techniques are designed to use the tilt data from the accelerometer of the mobile phone to manipulate the control-display ratio.

### *3.4.Evaluation*

We conducted an initial study to verify our hypothesis that using phone tilt as a mechanism to let the user change the control-display ratio improves task completion times. In this study, we recruited five participants to move cursors on the large display by using phone tilt comparing with not using phone tilt function. The study results strongly confirmed our hypothesis and we designed the four interactions based on this observation. We then conducted a user study to determine the relative efficiency of each of these four interaction techniques in order to compare them with each other. We present the results and analysis of this study in this section.

*Design*

The primary task the users are asked to perform is a drag-and-drop task on images which are displayed on the large wall sized screen. One image is highlighted and the user is asked to select that image and drag it to a target box. The placement of the images and the selection of the target image and target box are done randomly.

Our test is a [2x2x4] within subjects design with three primary variables: Image size (SIZE_BIG, SIZE_ SMALL), Distance (DIST_BIG, DIST_SMALL) and the interaction techniques (4 techniques). We measure the time required by the user to select an image, the movement time required to drag the image to the target box, and the number of incorrect placements or errors.

*Participants*

We conducted the user study on 10 participants, 8 males and 2 females, in the age group 20-30 years of age. All of them were right handed and were daily computer users.

*Procedure*

In order to aid familiarization with the technique and the system, we developed a game that users play for 3-5 minutes before measurements are taken. The game instruction stated that participants caught with each other by moving pictures on the large display. In order to mitigate the effects of learning, the order of the interaction techniques is chosen at random.

### *3.5.Results*

*Selection time and movement time*

In accordance with the Fitt's law, the size of the image to be selected affects the selection time while distance of the target box from the image affects the movement time required to drag the image. As expected, shorter distances and larger image sizes translate to shorter task completion time. Figure 7 illustrates the selection time of different image sizes by using the four interaction techniques. Touchpad interaction technique outperforms the others but for larger images, the omni-directional technique is faster than the touchpad. Figure 8 shows the movement time for different distances. Surprisingly, for the omni-directional remote control and mobile phone coupled with Kinect, smaller distances took longer time than larger distances! The Kinect implementation is good for coarse-grained interaction but it's not very well suited for tasks that demand accuracy. There is also a noticeable latency in our

Kinect implementation which might have caused this anomaly.

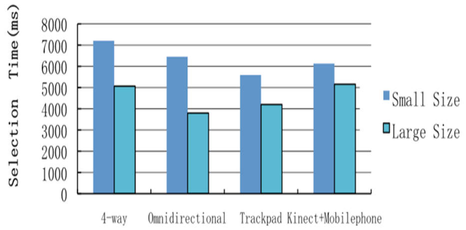


**Figure7.** Selection time

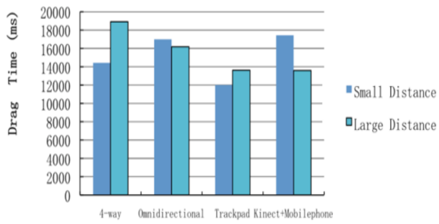


**Figure 8.** Movement time

*Overall time and incorrect placements*

Overall time is the total test completion time and it sums up the selection time and movement time for each technique regardless of the image size and the distance. Figure 9 shows the total test completion time. The touchpad interaction works the fastest while the Mobile phone coupled with Kinect is the slowest. Omni-directional is faster than 4-way remote control as we expected. Figure 10 shows the average number of incorrect placements for each interaction. As far as accuracy is concerned, the 4-way remote control technique performed the best while the Kinect was the worst.

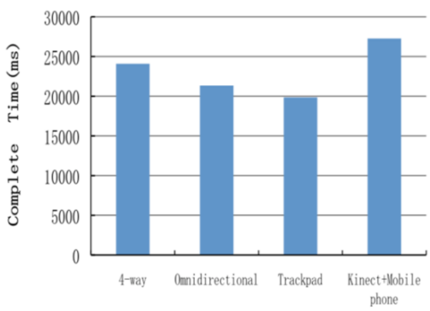


**Figure 9.** Task complete time

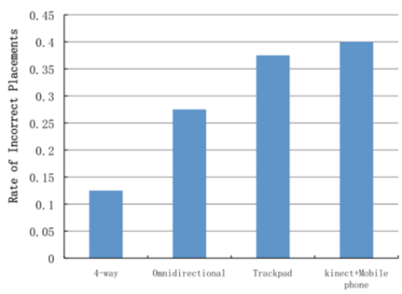


**Figure 10.** Number of incorrect placements

*Open questions and qualitative data*

The qualitative data collected from the participants gives insights into some interesting observations. As seen in Figure 11, the touchpad is the most preferred technique since it is easy to use and very tractable. Though omni-directional remote control is faster than the 4-way remote control, users much preferred the 4-way to the omni-directional technique. Users claimed that the omni-directional technique was harder to use. One of the reasons might be that users are more familiar with up/down/left/right keys from using keyboards and joysticks. Another reason is that unconstrained motion in 2D space requires more cognitive efforts than orthogonally constrained motion. To sum it up, the touchpad technique was the most widely preferred and also the most efficient one. However, for applications where a higher degree of accuracy is desired, the 4-way remote control is more preferred. Although the omni-directional remote gives users more freedom and more speed and accuracy of movement, users tend to dislike it because it requires more cognitive effort. Users found the Mobile phone coupled with Kinect the most exciting, and also the easiest to learn. In general, users preferred to use touchpad interaction techniques most, so we decided to choose touchpad as our primary interaction technique for further development.

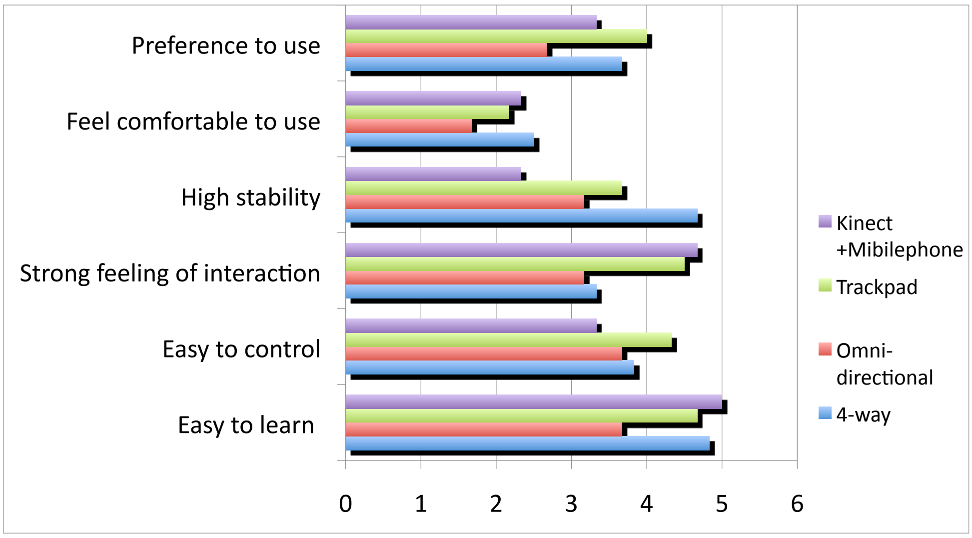


**Figure11.** Qualitative date from users for four interaction techniques (0 - not agree at all, 5 - strongly agree)

## 4. Web Applications

Since the cost of developing new applications on large displays is high, we investigate the techniques to adapt existing web applications without access to the applications' source code. Users continue to operate existing software packages and gain additional value from additional displays. Our approach consists of executing multiple copies of the application in parallel, and keeping the copies synchronized. And these applications are used by multiple users though mobile devices. In this paper, we present five sample web applications, which are search, presentation, finance, maps, and document editing.

### *4.1.System Architecture*

Our system consists of interface instance managers (IIM), a system input manager (SIM), a central synchronization server and interface instance managers (IIM) and mobile remote controls for different applications. IIM runs on each display server to monitor view changes and update views. Concretely, when a user interacts with one of the applications, the IIM senses application view changes in the active application; and sends interface commands to other application interfaces to synchronize their views accordingly. SIM runs on each display server to permit use of multiple cursors across multiple machines. A synchronization server is responsible for transmitting messages between different IIMs and SIMs across display servers and it also conveys input events from mobile controllers. Mobile remote controls enable users to assign applications to displays.

We use a setup with two display servers. A Mac Pro with 2 GPUs drives a 3x2 tiled display of 6 monitors with a total resolution of 5760x2160 (1980x1080 per monitor). A Mac Mini drives a single monitor with a resolution of 1600x1200. This second server is used to demonstrate that our architecture runs across multiple servers. We use Android and iOS devices to run the mobile controllers.

### *4.2.Implementation*

We implemented our browser-based web applications by writing Google Chrome extensions. Each screen runs its own Chrome window. A startup script launches one browser per screen and positions browser windows to fill each screen. The script then initializes each view by loading a shared application URL, and creates the initial view setup (e.g., by moving different screens to different pages/slides). Each page in each browser window is augmented with Javascript code of the interface instance manager (IIM). This client IIM code observes whether users interact with the application. If users change the view, the client IIM code in the window that triggered the change passes a message to a shared server IIM. This server then determines the proper synchronization steps that should be taken, and issues appropriate update commands to all other client IIMs. The server IIM is implemented in JavaScript using Google Chrome and background pages, while client IIM code is implemented in Chrome and content scripts. The central synchronization server is used NodeJS and socket.io.

### *4.3. Five Web Applications*

We developed five web applications: search (Figure 12), presentation (Figure 13), finance (Figure 14), maps (Figure 15) and document edit (Figure 16). In search app, the top left and bottom left monitors show the results of Google search. The rest monitors first show Wikipedia, maps, YouTube and images. If users change views to next, the rest monitors would show the first four pages of results of Google search. In presentation app, each monitor shows one slide. If there are more slides than number of monitors, all monitors will advance through the deck in synchrony. In finance app, each monitor shows stocks of one company. Users can modify the company name and compare stocks between different companies on six monitors. For maps app, each monitor shows a map tile such that a unified view of a single map is created. Users can pan and zoom, with all tiles updating accordingly. Users can also switch from a single tiled view into small multiple views where each screen shows the same map region, but different data for that region (e.g., roads vs. topographic vs. satellite images). Document edit app is used for manipulate contents on the documents by multiple users through mobile devices at the same time.

**Figure12.** Search App

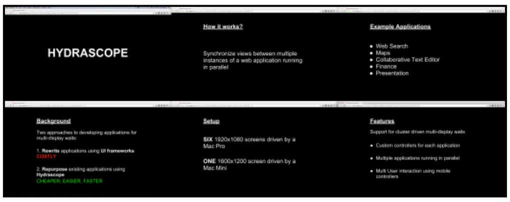


**Figure13.** Presentation App

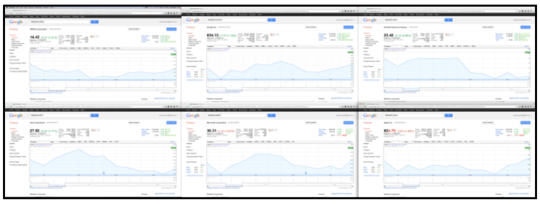

**Figure14.** Finance App

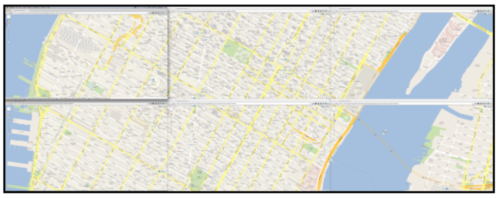

**Figure15.** Maps App

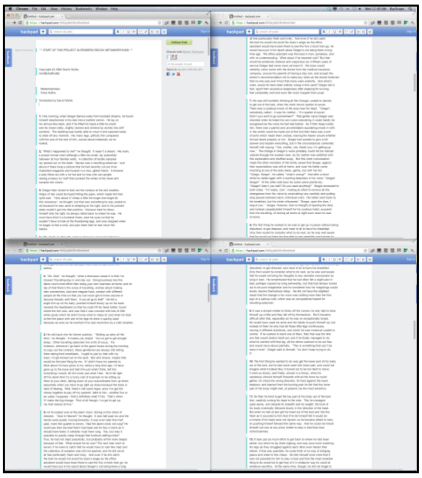

**Figure16.** Document Edit

### *4.4.Mobile Remote Controller*

Wall-sized screens are often used in shared multi-user environments. In order to let multiple people move around and interact with the display, we built a mobile remote controller. In this section, we demonstrate how mobile remote controller can be used to provide input to multi-display web applications. We implemented mobile controller by using HTML5 and jQuery Mobile. It consists of an Application Launchpad for launching and killing applications, the Screenscape screen manager for assigning different parts of the wall to different applications, and a set of controllers for interacting with the multi-display wall. Five web applications are currently available on our mobile app websites, as shown in Figure 17. Users can launch and close each app by clicking on it. Besides, we designed six small icons on the interface of mobile phone representing each monitor on the large wall-sized display. After launching a web app, users should first choose the order of each monitor on large displays by tapping on the six small icons on the mobile phone interface by sequence. Two types of controllers are designed to manipulate contents on these web applications. The first one is the generic keyboard and mouse controller, which uses touchpad interaction technique (Figure 18). The mouse controller is responsible for move and click, whereas keyboard controller is used for input. The second type uses individual controller for search, presentation and finance applications. For search controller (Figure 19), users can click on the small button for preserving the corresponding screen on large displays. They could also input keywords in the search bar by tapping the search bar. The function of previous and next buttons in search, presentation and finance apps is to jump to the previous or next page. Users have the option to use either of the two controllers for interacting with web applications on multiple displays. Moreover, we implemented multi-user input for web applications. Multiple users could use different mobile phones, keyboard and mouse for input on the large display such that they can collaborate with each other.

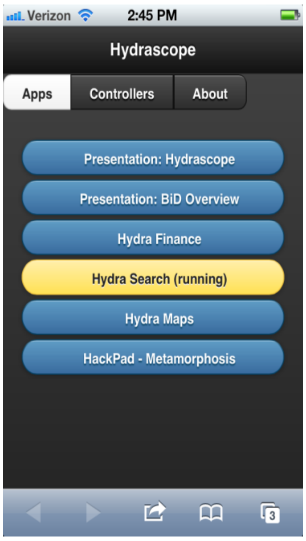


**Figure17.** Homepage of mobile app


**Figure18.** Generic Controller

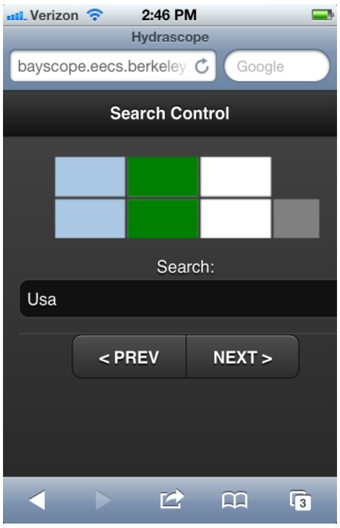


**Figure19.** Search Controller

### *4.5.Evaluation*

In order to evaluate the usability and utility of our web applications by using mobile phones as input, a user study was conducted involving small, co-located groups performing collaborative tasks. The study focused on the usability and potential of supporting such group work by using the system and was designed with the following objectives in mind:

1. understand the advantages of using web applications with multiple views on large displays,

2. test the effectiveness and convenience of using multiple mobile application as controllers for collaboration , and

3. observe real user experiences with mobile application interface: two types of controllers.

*Design*

We designed two sets of tasks for pairs to demonstrate the usability and utility of the system. Each task was carried out three times: once with a single display and mouse and keyboard control, once with six displays and use one mobile phone, one keyboard and one mouse and once with six displays and two mobile devices. The purpose is to help users compare different interfaces and different interaction techniques. In the first task, participants were asked to use search web application to answer questions from “AGoogleADay”. Each of these questions requires participants to form multiple queries and then synthesize the found information into an answer. For example, one question is: “Did your brain’s frontal lobe or temporal lobe have more to do with planning out which movie you'd like to see this evening?”

For the second task, participants used finance web application to find the highest and lowest stock price among a set of technical companies at a given time.

*Participants*

Ten participants from university of California, Berkeley were asked to test our system. There were 2 females and 8 males between the ages of 18-25. All participants were divided into 5 groups of 2. Four groups were familiar with each other; only one group was not familiar with each other.

*Procedure*

Participants were welcomed and explained the purpose of the study. They were then given brief instructions on using our system. They were given 5 minutes of freeform play to become familiar with the system and discover system functionality. Next, participants were given a formal demonstration of the main features of the system with a special focus on web applications with multiple views on large displays and mobile application for controllers. The demonstration took approximately 10 minutes. Afterwards, the tasks were explained to the group. The group then spent 20 minutes averagely performing the task. Finally, participants were asked to complete a post-study questionnaire consisting of 10 questions. The study session took approximately 45 minutes to complete in total.

### *4.6.Results and Discussion*

By using our system, participant could collaborate with each other on our wall-sized display by providing simultaneous input from two mobile phones. For instance, as we observed in the test of using search application, one participant used the generic controller (Figure 18) to scroll search result pages and find useful information, while the other one used search controller (Figure 19) to input keywords and navigate the result pages.

When comparing keyboard and mouse with mobile controllers, users preferred to use mobile phones for input, because they are more convenient and enable them to change distances from the large display. But there is one problem with the interface of mobile phones that participants proposed frequently—the cursor control and specific application controller interface were located on different pages on mobile phones. This forced users to switch between two pages frequently. Such problem can be addressed by adding a controller widget on all mobile application controllers.

Participants also commented repeatedly about the benefits of using web applications on large wall-sized displays. They discovered that it is more convenient and efficient to compare different information six displays rather than switch windows many times.

## 5. Conclusion

We implemented four interaction techniques between mobile phones and large wall-sized displays. After quantitative and qualitative analysis of the results from user study, touchpad has been selected as the major interaction technique for mobile remote control on the large displays. On the other side, we proposed a concept of a collection of application instances whose contents and views are coordinated and they can be used by multiple users through mobile devices. More concretely, we revised existing web applications to show multiple view interfaces on large displays. Finally, we conducted an evaluation to demonstrate the usability and utility of our system.

From the results of the user study, we find several areas of future work. First, we would like to implement and evaluate more interaction techniques. Second, we are interested in developing better tools for developers to facilitate our system. Finally, we also want to build some native applications (opposed to web-based applications) on multi-display walls.

## 6. Reference


[1] T. Ni, G. Schmidt, O. Staadt, M. Livingston, R. Ball, and R. May. A survey of large high-resolution display technologies, techniques, and applications. In Proc. VR ’06, 223–236. IEEE, 2006.

[2] P. Baudisch, N. Good, V. Bellotti, and P. Schraedley. Keeping things in context: a comparative evaluation of focus plus context screens, overviews, and zooming. In Proc. CHI ’02, 259–266. ACM, 2002.

[3] R. Ball, C. North, and D. Bowman. Move to improve: promoting physical navigation to increase user performance with large displays. In Proc. CHI ’07, 191–200. ACM, 2007.

[4] C. Andrews, A. Endert, and C. North. Space to think: large high-resolution displays for sensemaking. In Proc. CHI ’10, 55–64. ACM, 2010.

[5] B. Yost, Y. Haciahmetoglu, and C. North. Beyond visual acuity: the perceptual scalability of information visualizations for large displays. In Proc. CHI ’07, 101–110. ACM, 2007.

[6] Azam Khan, George Fitzmaurice, Don Almeida, Nicolas Burtnyk, and Gordon Kurtenbach. A remote control interface for large displays. In Proc. UIST 2004, pp. 127-136.

[7] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, Patrick Baudisch. Touch Projector: Mobile Interaction Through Video. In Proc. CHI 2010, pp. 2287 – 2296

[8] Sylvain Malacria, Eric Lecolinet and Yves Guiard. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the cyclostar approach, in Proc. CHI 2010.

[9] Mathieu Nance, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. Mid-air Pan-and-Zoom on Wall-sized Displays. In Proc. CHI 2011 pp. 177 – 186.

[10] Tsung-Hsiang Chang, Yang Li. Deep Shot: A Framework for Migrating Tasks Across Devices Using Mobile Phone Cameras. CHI 2011 ACM Conference on Human Factors in Computing Systems.

[11] Rafael Ballagas, Michael Rohs, and Jennifer G. Sheridan. 2005. Sweep and point and shoot: phonecam-based interactions for large public displays. In CHI '05 extended abstracts on Human factors in computing systems (CHI EA '05).

[12] Hartmann, B., Morris, M. R., Benko, H., and Wilson, A. D. 2010. Pictionaire: supporting collaborative design work by integrating physical and digital artifacts. In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (Savannah, Georgia, USA, February 06 - 10, 2010). CSCW '10. ACM, New York, NY, 421-424.

[13] Grant Wallace and Kai Li. Virtually Shared Displays and User Input Devices. In Proc 2007 USENIX Annual Technical Conference. Pp. 375-380.