Multi-modal Interface in Multi-Display Environment for Multi-users

Yoshifumi Kitamura, Satoshi Sakurai, Tokuo Yamaguchi, Ryo Fukazawa,
Yuichi Itoh, and Fumio Kishino

Graduate School of Information Science and Technology
Osaka University
kitamura@ist.osaka-u.ac.jp

Abstract. Multi-display environments (MDEs) are becoming more and more common. By introducing multi-modal interaction techniques such as gaze, body/hand and gestures, we established a sophisticated and intuitive interface for MDEs where the displays are stitched seamlessly and dynamically according to the users' viewpoints. Each user can interact with the multiple displays as if she is in front of an ordinary desktop GUI environment.

Keywords: 3D user interfaces, CSCW, graphical user interfaces, perspective correction.

1 Introduction

A variety of new display combinations are currently being incorporated to offices and meeting rooms. Examples of such displays are projection screens, wall-sized PDPs or LCDs, digital tables, desktop and notebook PCs. We often use these multiple displays simultaneously during work. Thus, MDEs are becoming more and more common. We expect to work effectively by using multiple displays in such environments; however, there are important issues that prevent users from effectively taking advantage of all the available displays. MDEs include displays that can be at different locations from and different angles to the user; as a result, it can become very difficult to manage windows, read text, and manipulate objects. Therefore, it is necessary to establish a sophisticated interface for MDEs where the displays are stitched seamlessly and dynamically according to the users' viewpoints, and a user can interact with the multiple displays as if she is in front of an ordinary desktop GUI environment.

Therefore, we propose a system that includes multi-modal interaction techniques utilizing multiple displays. The multi-modal interactions such as gaze inputs, finger gestures, make tasks more comfortable and intuitive. Moreover, they can be used for detecting context of the environment, so that it provides a perspective-correct GUI environment for viewing, reading, and manipulating information for each MDE user.

J.A. Jacko (Ed.): Human-Computer Interaction, Part II, HCII 2009, LNCS 5611, pp. 66–74, 2009.

2 Previous Work

2.1 System Utilizing Multi-Display Environments

Research on computer systems that use several displays has been active for more than 30 years. Early systems focused on the optimization of collaborative processes, and provided personal displays for individual users and large displays for shared use [1]. Newer systems provide more flexible interfaces that integrate combinations of displays as tabletops, personal displays, and large vertical displays [2, 3, 4, 5, 6]. These MDEs typically provide graphical interfaces based on standard WIMP paradigms, perhaps with extended features such as new techniques for cross-display operation [2, 4, 7], novel techniques to manipulate objects [6] and replication techniques to access proxies of distant content from more accessible locations [7, 9]. The geometry that these systems use is mostly inherited from the desktop monitor interface - that is, rectangular elements that assume the user is perpendicular to all displays.

2.2 Multi-modal Interfaces

People communicate with others using multi-modalities in every-day life. Especially we effectively use non-verbal communication channels such as body/hand gestures, eye contacts, and so on. Therefore, many efforts have focused on multi-modal interfaces and non-verbal interfaces.

Voice input has been one of the hot topics since voice commands are effective when combining with other modalities like the gestures [10]. Schmandt proposed a system which adapts the user's complex demands by analyzing his voice [11]. In another example, voices are used as sound signals including information of loudness, pitch and so on [12].

Non-verbal inputs are also used from early years. Bolts et al. proposed a system allowing the users to create and edit figures by using both the gestures and the voice commands [10]. Similarly, the Finger-pointer proposed by Fukumoto et al. supports gestural commands of hands [13].

There are many examples using gaze inputs as another representative type of the non-verbal interactions. MAGIC pointing [14] supports the selections of icons by using gaze information with the cursor. Gaze-Orchestrated Dynamic Windows [15] and EyeWindows [16] allow users to select a window by staring for a while. Although the gaze inputs include some errors, some experimental report that the users can select objects more quickly with the gaze inputs than the mouse.

As described above, the multi-modal interactions have favorable possibility as human-computer interfaces. However, there is no research focusing on the multi-modal interfaces in MDEs.

3 Multi-Display Environment

3.1 Seamless Use of Multiple Displays

We propose a multi-display environment that combines several displays as if they were part of one large virtual GUI environment. The proposing environment defines a virtual plane which is perpendicular to the user as a virtual large display. GUI objects (e.g. windows and cursors) on the virtual plane are projected onto the real displays. As a result, wherever the user's viewpoint is, the user observes GUI objects (cursors and windows) without distortion; just as if they were perpendicular to the user. Even if a GUI object extends to several displays, the user observes it continuously beyond the boundaries of the displays.

Fig. 1. An example of multi-display environment

Figure 1 shows an example of MDE. When the user's viewpoint or some of the displays move, the environment detects these movements with 3D motion sensors and updates the display immediately to maintain the relationship. In the environment, the user controls the cursor on a virtual sphere around the user, so that the cursor can move seamlessly between displays. Also, the user can interact with the multiple displays not only from a certain specific computer, but also from all computers in the environment.

3.2 Middleware Architecture

MDEs can be composed of displays with different characteristics (e.g. resolution, size) located in any position and at different angles. These heterogeneous arrangements present specific interface problems: it is difficult to provide meaningful transitions of cursors between displays; it is difficult for users to visualize information that is presented on oblique surfaces; and it is difficult to spread visual information over multiple displays. Therefore, we proposed middleware architecture designed to support a new kind of perspective-aware GUI that solves the aforementioned problems. Our interaction architecture combines distributed input and position tracking data to generate perspective-corrected output in each of the displays, allowing groups of users to manipulate variety of applications from current operating systems across a large number of displays.

A *3D server* (a dedicated 3D server machine with specific 3D server software) keeps track and processes three-dimensional information of positions and orientations of the users' viewpoints and mobile displays measured through 3D motion sensors. The positions and orientations of user viewpoints and displays are measured by 3D motion sensors that are processed in the 3D server software to calculate the positions and orientations of the GUI objects on the virtual plane. This information is subsequently sent to the client software that runs in each of the client machines.

The system uses an independent *application server machine* that runs actual applications and sends the graphical data to the client machines. The software that carries out the functions of broadcasting the graphical data and receiving input from the client software instances is called the *application server software*.

In addition to presenting the graphical output of applications the system needs to be able to feed user input to these same applications. Users manipulate regular mice and keyboards that are connected to the client machines in order to interact with the applications shown in any display. The client software sends all inputs to the 3D server software, and then the 3D server software relays the inputs to the corresponding windows according to the positions and orientations of the GUI objects in the environment. When the cursor is on top of a window, the 3D server software transfers the cursor inputs to the application server software. For the consistency of the input/output flow, the keyboard inputs on the client machines are sent to the application server software through the 3D server software. In this way, the inputs on all client machines are appropriately processed through the network. Details of each type of software are described in [17].

3.3 Perspective-Aware Features

In MDEs, there are a variety of displays' and users' arrangements so that the users observe the displays at diverse relative positions and postures. If a display is oblique to a user the visibility of information is severely reduced. Moreover, if information extends to multiple displays, part of the information might not be visible, and will consequently be very difficult to interpret. To solve these problems, the system uses the Perspective Window and the Perspective Cursor.

3.3.1 Perspective Window

The Perspective Window provides an optimal view of information in MDEs [18]. They display the same kind of contents as traditional 2D windows (e.g., a web browser or a text processor) but offer extra features derived from the perspective-aware capabilities of the system. Main difference between regular windows and the Perspective Window is that the latter provide optimal visibility to the user regardless of the angle of the display. The windows are rendered using a virtual plane that is perpendicular to the user in the center of the window, and then projected onto the display as shown in Figure 2. If a window is displayed across more than one surface

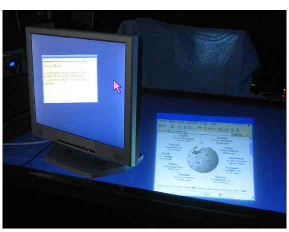


Fig. 2. Perspective corrected windows

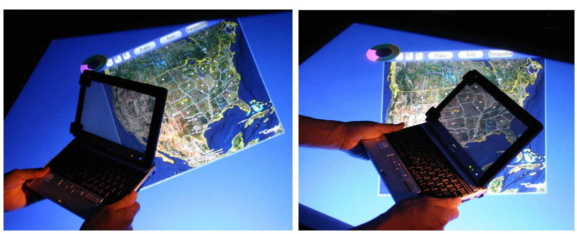


Fig. 3. Window across displays

simultaneously, perspective can help reduce fracture as shown in Figure 3. The perspective windows also help reduce representational disparity since the perspective distortion that affects windows located in different displays is eliminated, making comparisons among their contents easier.

When the viewpoint move measured by a 3D motion sensor. At this time, windows and cursors stay attached to a pixel in a display through an anchor situated in the top left corner of these GUI objects. Their shapes and orientations vary if the display or the user moves, but the objects will remain attached to the same physical point in the display.

When multiple displays are placed at different locations and angles, it can be difficult for people to work with the displays. We investigated the benefit of perspective in multi-display environments [18]. In a controlled experiment that compared perspective windows to flat windows on five basic interaction tasks we found that when using perspective windows, performance improved between 8% and 60%, depending on the task. Our results suggest that where 3D positional information can be obtained, using perspective information in the design of multi-display environments offers clear user benefits.

3.3.2 Perspective Cursor

The system uses the Perspective Cursor as a cross-display user control [19]. Each user manipulates a cursor that is displayed in the user's color. Perspective halos of the same color of the cursor are used to indicate the position of the cursor when it is between displays. Perspective cursor works as follows. The 3D position coordinates of the head of the user is measured and at the same time, a 3D model of the whole environment is maintained, with the actual position of all the screens. The model, together with the point-of-view coordinate of the user's head, lets us determine which displays are contiguous in the field of view, something very different to displays actually being contiguous in 3D space. The position and movement of the pointer are calculated from the viewpoint of the user, so that the user perceives the movement of the pointer across displays as continuous, even when the actual movement of the pointer considered in three dimensional space is not. The pointer travels through the empty space to get from one display into the next. Actually, the cursor can be in any position around the user, even if there is no display there to show graphical representation of the cursor. Figure 4 shows an example movement of the perspective cursor.

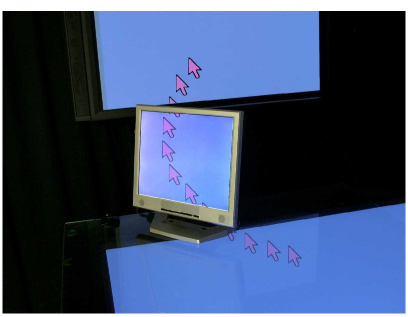


Fig. 4. Perspective cursor

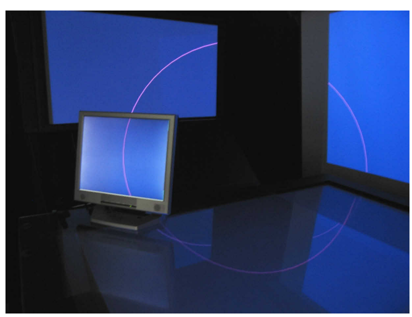


Fig. 5. Perspective halo

There are not many environments in which the users are completely surrounded by displays, meaning that users might lose the pointer in non-displayable space. The solution that we implemented is a perspective variant of halos [20]. Figure 5 shows the perspective halo from the user's viewpoint. Halos are circles centered on the cursor that are big enough in radius to appear, at least partially, in at least one of the screens. By looking at the displayed part of the circle, its position and its curvature, the users can tell how far and in which direction the perspective cursor is located. When the cursor is barely out of one display, the displayed arc section of the halo is highly curved, showing most of the circle. If the cursor is very far away, the arc seen will resemble a straight line.

4 Multi-modal Interface

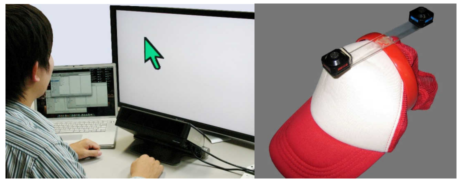
In MDEs, users' workspace is much larger than traditional ones. As a result, the users' tasks often become more difficult and time-consuming. For example, a user needs to move mouse for long distance in order to click a desired target object when it is located far from the current position of the cursor. Moreover, users might lose the cursor when it goes to a gap between displays. In these situations, interactions using gaze inputs show much effectiveness. For example, Figure 6 shows a cursor-motion and a window-jumping to the position where the user stares. It prevents the user from losing the cursor and the window. These interactions reduce the user's load of manipulating the pointing devices for long distance. Besides, the gaze inputs can be effectively used for several other purposes. For example, a particular gaze movement can be registered as often-used actions.

In the system, we use the head direction of the user in addition to the gaze input. Figure 7(a) shows a device for detecting gaze, and (b) shows a device for detecting head direction of a user which consists of two 3D positional trackers. In addition, it supports gestural interactions. Figure 8 shows an example in which one of the users moves a window to the position pointed by his finger. This contributes efficiency in collaborative tasks; for example, a user can take another's notice to the position pointed by the finger with moving an object. Other interactions like voice commands are also effective. We can easily expect that the combinations of these multi-modal interactions will improve the effectiveness and intuitiveness of the tasks. For example, the user can give a command with the voice inputs while indicating the object or the position of the command by the finger gestures. Users can select these options according to their preferences, tasks, attitudes, and so on.

In the system a user can also use ordinary interactions devices such as mice according to her preference. These are still useful for tasks which require accurate paintings.

The perspective windows can be used by multiple users. In this case, the system needs to know which point-of-view to optimize for, i.e., who the owner of the window is. Therefore, the system implements an implicit mechanism for assigning windows to users through the user's attention detected from the user's head directions and the gestures. When the user's head directs to a window for a while, the system interprets that he is working or interested in the window. At this time, the system corrects perspective distortion of the window for him automatically. Figure 9 shows an example in which the system is correcting the window gazed by the user. Similarly,

Fig. 6. Cursor-motion and window jumping caused by gaze input



(a) gaze (b) head

Fig. 7. Gaze and head tracking devices

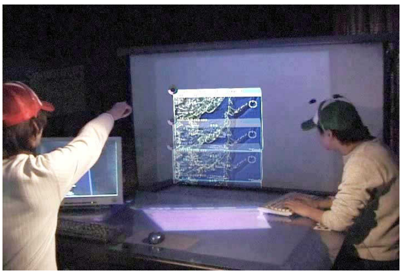


Fig. 8. Window-motion by finger pointing

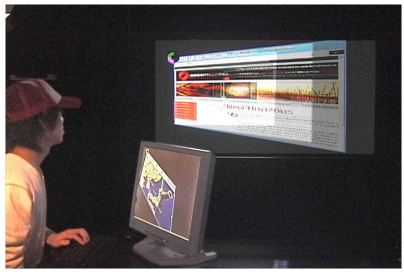


Fig. 9. Perspective correction using gaze input

when the system detects the finger pointing, it adapts the perspective correction. The owner can be changed explicitly by clicking the window with the cursor too.

5 The Prototype

The prototype system was demonstrated at SIGGRAPH Asia Emerging Technologies [21]. As shown in Figure 10, many participants easily understand future possibilities of MDEs through our interactive demonstration. Multi-modal interactions were

Fig. 10. Snapshots of people enjoying our demo at SIGGRAPH Asia

presented using their head positions, directions, and finger gestures. They could also use seamless interaction across multiple displays using the Perspective Window and the Perspective Cursor. Besides, they could experience the automatic and dynamic perspective adaption according to the positions and movements of their viewpoints, their heads' directions, and finger gestures.

6 Summary and Future Work

In this paper we described a system that includes multi-modal interaction techniques utilizing multiple displays. The multi-modal interactions such as gaze inputs, finger gestures, make tasks more comfortable and intuitive. Moreover, they can be used for detecting context of the environment, so that it provides a perspective-correct GUI environment for viewing, reading, and manipulating information for each MDE user.

The MDE technology with multi-modal interface is one of the key technologies to establish future ambient information environment. In an ambient information environment, such explicit instructions as commands are not required. The environment is expected to identify the situation and give necessary information. In such an environment, a multitude of sensors will be installed. The environment is required to comprehensively evaluate the information obtained from these sensors and accurately identify the situation in real time. Various types of information presentation devices will also be installed. The environment will give appropriate people appropriate information by adequately combining these devices depending on the situation.

Acknowledgements

This research was supported in part by and Microsoft Institute for Japanese Academic Research Collaboration and "Global COE (Centers of Excellence) Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. Kraemer, K.L., King, J.L.: Computer-based systems for cooperative work and group decision making. ACM Computing Surveys 20, 115–146 (1988)
2. Biehl, J.T., Bailey, B.P.: ARIS: an interface for application relocation in an interactive space. In: Proceedings of Graphics Interface (GI), pp. 107–116 (2004)
3. Myers, B.A., Stiel, H., Gargiulo, R.: Collaboration using multiple PDAs connected to a PC. In: Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW), pp. 285–294 (1998)
4. Prante, T., Streitz, N.A., Tandler, P.: Roomware: computers disappear and interaction evolves. Computer 37, 47–54 (2004)
5. Rekimoto, J.: A multiple device approach for supporting whiteboard-based interactions. In: Proceedings of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 344–351 (1998)
6. Rekimoto, J., Saitoh, M.: Augmented surfaces: a spatially continuous work space for hybrid computing environments. In: Proceedings of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 378–385 (1999)

7. Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., Smith, M.: Stitching: pen gestures that span multiple displays. In: Proceedings of the Working Conference on Advanced Visual Interfaces (AVI), pp. 23–31 (2004)
8. Khan, G., Fitzmaurice, D., Almeida, N., Burtnyk, N., Kurtenbach, G.: A remote control interface for large displays. In: Proceedings of ACM Symposium on User Interface Software and Technology (UIST), pp. 127–136 (2004)
9. Tan, D.S., Meyers, B., Czerwinski, M.: WinCuts: manipulating arbitrary window regions for more effective use of screen space. In: Extended Abstracts on Human Factors in Computing Systems (ACM SIGCHI), pp. 1525–1528 (2004)
10. Bolt, R.A.: Put-that-there: voice and gesture at the graphics interface. In: Proceedings of Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH), pp. 262–270 (1980)
11. Schmandt, C., Hulteen, E.A.: The intelligent voice-interactive interface. In: Proceedings of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 363–366 (1982)
12. Igarashi, T., Hughes, J.F.: Voice as sound: using non-verbal voice input for interactive control. In: Proceedings of ACM Symposium on User Interface Software and Technology (UIST), pp. 155–156 (2001)
13. Fukumoto, M., Mase, K., Suenaga, Y.: Finger-pointer: a glove free interface. In: Posters and Short Talks of Conference on Human Factors in Computing Systems (ACM SIGCHI), p. 62 (1992)
14. Zhai, S., Morimoto, C., Ihde, S.: Manual and gaze input cascaded (MAGIC) pointing. In: Proceedings of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 246–253 (1999)
15. Bolt, R.A.: Gaze-orchestrated dynamic windows. In: Proceedings of Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH), pp. 109–119 (1981)
16. Fono, D., Vertegaal, R.: EyeWindows: evaluation of eye-controlled zooming windows for focus selection. In: Proceedings of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 151–160 (2005)
17. Sakurai, S., Itoh, Y., Kitamura, Y., Nacenta, M., Yamaguchi, T., Subramanian, S., Kishino, F.: A middleware for seamless use of multiple displays. In: Graham, T.C.N., Palanque, P. (eds.) DSV-IS 2008. LNCS, vol. 5136, pp. 252–266. Springer, Heidelberg (2008)
18. Nacenta, M., Sakurai, S., Yamaguchi, T., Miki, Y., Itoh, Y., Kitamura, Y., Subramanian, S., Gutwin, C.: E-conic: a perspective-aware interface for multi-display environments. In: Proceedings of ACM Symposium on User Interface Software and Technology, pp. 279–288 (2007)
19. Nacenta, M., Sallam, S., Champoux, B., Subramanian, S., Gutwin, C.: Perspective cursor: perspective-based interaction for multi-display environments. In: Proceedings of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 289–298 (2006)
20. Baudisch, P., Rosenholtz, R.: Halo: a technique for visualizing off-screen object. In: Proceedings of Conference on Human Factors in Computing Systems (ACM SIGCHI), pp. 481–488 (2003)
21. Sakurai, S., Yamaguchi, T., Kitamura, Y., Nacenta, M., Itoh, Y., Fukazawa, R., Subramanian, S., Kishino, F.: M3: Multi-modal interface in multi-display environment for multi-users. In: ACM SIGGRAPH Asia Emerging Technologies (2008)