Designing Interaction for a Multi-touch Wall

Chen Wang[1, 2] Hyowon Lee[2] Alan F. Smeaton[2]

[1]School of Information and Management
National University of Defense Technology
Changsha, Hunan, P.R. China

[2]CLARITY: Centre for Sensor Web Technologies
Dublin City University
Glasnevin, Dublin 9, Ireland

{cwang, hlee}@computing.dcu.ie

alan.smeaton@dcu.ie

ABSTRACT
As large-scale display and multi-touch technologies become more affordable, the market has seen the development of multi-touch walls. This new medium offers a unique mix of information density, direct interactivity and collaboration support, and the new features have radical effects on interaction design. Here we explore some research issues together with proposed solutions and some design suggestions, based on our own approach to three areas of interaction design: multi-touch input, user interface and co-located collaboration.

Categories and Subject Descriptors
H.5.m [**Information Interfaces & Presentation**]: Miscellaneous.

General Terms
Design, Human Factors.

Keywords
Interaction design, multi-touch wall, user interface.

1. INTRODUCTION
In recent years, the market has seen a proliferation of Multi-Touch Wall (hereinafter referred to as MTW) systems, where application settings range from collaborative workspaces in offices to completely public settings such as urban environments. We believe that different application types demand different design guidelines, and there are still many communal issues for interacting with MTWs which need to be investigated. However, so far what is largely missing is a holistic approach that examines the design problems of MTW interaction.

In order to explore the interaction design issues of MTWs, we use a high-resolution multi-touch wall with five MultiTouch Cells [5] (Figure 1). Our wall features a screen 3025mm x 1052mm with a total resolution of 9600 x 1080 pixels, with the capability of detecting multiple points of contact simultaneously in changing light. We designed a set of prototype applications introduced later with the intent of verifying our interaction design thoughts.

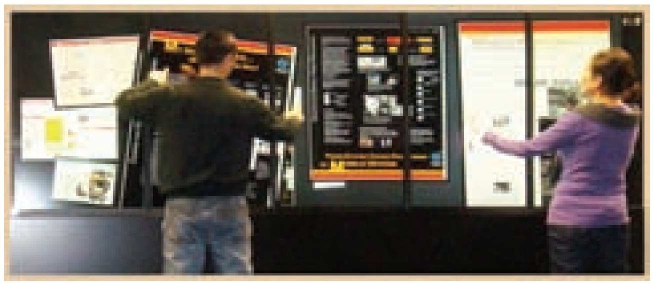

Figure 1. Our multi-touch wall is 3025mm×1052mm with a resolution of 9600×1080 pixels, and capable of detecting multiple points of contact simultaneously.

The following is a brief exploration of some of our interaction design considerations. The issues we identify are ones we have encountered in the course of our prototype application development and which we believe to be representative concerns for MTW interaction design overall.

2. MULTI-TOUCH INPUT
While multi-touch interaction has been extensively studied on iPhone, MS Surface and other similar devices, very little has been studied with respect to the use of wall surfaces for multi-touch input. Practical design experience has gained for us a clearer understanding of some of the questions of using touch input on a very large and stand-up surface.

2.1 Singlehanded or Bimanual
Although many approaches have shown that for many tasks bimanual techniques outperform unimanual techniques due to the parallelism achieved from two hands [1], in terms of some specific tasks utilizing multi-touch input, the situation becomes much more complicated. For example, [4] showed that direct touch with a single finger provides a primary performance benefit over using a mouse while bimanual interaction provides a smaller additional benefit for multi-target selection.

In addition, using a single hand rather than two hands can decrease occlusion due to the fact that we are aware of our own hands and thereby makes it easier to see the targets on the wall display, particularly in a multi-user setting. Furthermore, one important observation of previous studies [8] is that users initially preferred simple gestures, which are familiar from systems with mouse input using the WIMP desktop metaphor.

Therefore, we recommend using simple touch inputs with one finger or multiple fingers from the same hand for general manipulations. Even if special bimanual gestures sometimes have potential benefit for the efficiency of some specialized operations, they should be simple to perform.

2.2 Absolute or Relative Mapping

Most existing multi-touch interaction techniques use an absolute device-pointer mapping in which the system pointer is positioned directly under a finger. Users are well prepared to make this choice since the interaction is based on the analogy of manipulating physical objects. However, there are drawbacks when using absolute mapping in MTW applications. First, the size of the finger tip makes precise selection difficult. Second, hands, arms and fingers all occlude portions of the display, an especially troubling issue for multi-user systems. Third, on large wall displays many objects are simply out of reach. Using an absolute mapping presumes that a user must stretch their arms or physically walk to *distant* parts of the wall at times. In extreme cases, it may become impossible to reach the extents of a very tall wall display. Moreover, when working in a large, multi-user environment distant objects may require the user to interact across other users and even across gaps between displays.

To overcome these limitations, introducing relative mappings, which increases both the range and precision of the cursor by allowing for clutching and cursor acceleration into MTW applications is recommended. The mapping between the mouse and the cursor may be the most successful example of such a mapping. However, in MTWs we rarely use such indirect input devices because they conflict with one of the strong advantages of MTWs namely allowing direct interaction with no intermediary input device. We need to develop new mechanisms to implement relative mappings on multi-touch surfaces.

Researchers have proposed a variety of interaction techniques that enhance pointing precision [7] and increase the user's reach [2] on large touch-screens. But another emerging problem has been rarely studied, that is how to smoothly switch between absolute and relative mappings, sequentially enabling users to benefit from the best of both patterns. HybridPointing [3] provides a good reference for our design. A new "close at hand" widget facilitates free switching between absolute and relative mappings and visual feedback aids the user's understanding of input state.

2.3 Direct or Abstract Gesture

Gestural interaction is another important focal point in understanding the interaction design of MTWs.

By their nature, direct gestures can be easier for users to discover and adopt. They generally correlate to movement semantics (e.g., length of the gesture determines the extent of zooming or distance of dragging), consequently reducing the user's learning curve and making it quite intuitive. However, regarding some complex tasks, merely using one-to-one gestures may be too repetitive and inefficient. Instead, many productivity applications use indirect or "abstract" gestures to initiate, manipulate, and complete activities. These abstract gestures are not associated with direct one-to-one visual representations, such as a "palm swiping" allowing users to quickly close all windows. So substantial practice is usually required for the user to become comfortable with using abstract gestures and the learning curve is somewhat high. Gesture designers have to consider the trade-off between usability and functionality according to application type and end-user genre.

Wendy Yee [9] summed up the criteria for effective gestures and potential limitations of using abstract ones. Here, we want to underline two points: first, allowing users to gesture with minimal effort seems more important for MTW applications. Wall users are more prone to fatigue than tabletop users, as they have to raise arms to perform every gesture, and they are standing, not sitting. For the same reason, we suggest representation of large-scale actions using equivalent gestures with finger movements. Second, the gestures that can be accurately recognized by software is somewhat finite. Unless an application requires a multitude of frequently used complex operations, direct or simple gestures should be the primary operation schema. Designing in this way can not only make gestures obvious and intuitive in the context of relevant tasks but also minimize false-positives or cross-positives in system recognition of subtle gestures.

3. USER INTERFACE

Most of the fundamental principles for GUI design are still applicable to MTW applications. Nonetheless, the new qualities of MTWs give some design rules new meaning and require expansion. Based on some exploration of previously published studies and our own experience of developing MTW applications, here we describe some of our current thinking about how to design more usable GUI interfaces for MTWs.

3.1 High Degree of Interaction Context

Initially, the MTW user interface should make it clear to users that the wall can be touched and likewise it can handle multiple touches and gestural interaction. Another important thing to keep in the designer's mind is to prompt users to use proper gestures within a given context. Users can effortlessly find out and adopt obvious one-to-one gestures, but there are more or less relatively "abstract" gestures which may not be discovered just through intuition. Therefore, a good UI should help users identify applicable gestures for relevant tasks and consistently cue the most efficient gesture for preventing the user from making needless repetition.

An ideal suggestive interface should offer operational cues for users in a timely and forward thinking manner. However the implementation of such *intelligent* interfaces is complicated to some extent as the suggestion system has to track and identify all the user's input patterns in real time. We adopted a substitute passive mechanism: a three-fingered touch on an onscreen element will bring out an array of small thumbnails representing applicable gestures on the current object, and any subsequent gesture performed on the object or clicking on the "Close" button will fade the hint instrument out.

3.2 Rational Interface Elements Layout

One of the noticeable differentiations between comparing MTW UI design with the classic WIMP design paradigm is the size of interface elements. The increased visual component not only results from enlarged screen scale and the fact that the user can get really close to the MTW but also is bound to the input mode via finger-touch. Prior work has shown that targets must be larger than the size of a fingertip to obtain good performance with multi-touch devices. Furthermore, in multi-user settings, greater interaction effects make other simultaneous users capable of sharing the experience and perceiving the collaborative state.

Unlike horizontal multi-touch tabletops, which pay much attention to the orientation, rotation, and position of shared information for the collaboration, the time and physical cost of interactions need to be carefully thought out on a MTW interface. A person closely touching a MTW may be unable see the entire surface. Parallel with the constraint of limited vision, users can only access a certain number of software elements without physically moving their bodies. Even though maybe some wall applications benefit from user's walking and moving which is considered by someone as an opportunity for a better interaction pattern, undue physical activities commonly make users fatigued and tired. Hence, to prevent users from having to frequently walk around or step back from the wall, constraining as much visual information and functionality as possible in the user see-and-touch range is an essential principle of MTW UI design. Fortunately, the high resolution of MTWs makes such design possible.

In addition, most standard UI mechanisms are not appropriate for a collaborative process with multiple users. Compared with dividing the wall space into separate but interoperating workspaces, a single shared space may be preferable for many MTW tasks. This strategy allows multiple users to simultaneously manipulate separate objects within a common view, but concurrency and conflict problems consequently emerge. Although many questions about concurrency and conflict are very complicated and currently unanswered, we suggest that it is important that the representation of interface elements should be carefully considered to facilitate sharing functions and prevent interference among users. In some cases, multiple copies of a common element might be needed to accommodate the existence of multiple users.

3.3 Re-design Interface Widgets

Classic interface widgets are the standard components of a modern desktop GUI. In a MTW environment, to some degree the use of an assistant widget is inevitable in order to provide elemental functionality such as accessing outlying interface objects, switching between absolute and relative mappings, and providing an overview of the screen. Along with considering the usability of these widgets, it's important for designers to take the context, use, and practicality of the MTW into account.

Traditional desktop GUI logic is probably no longer suitable for MTW applications. An outstanding example is that most GUI design has assumed that the whole space is randomly accessible to the user, so most widgets are statically placed at an edge of the screen or in the center of screen while in MTWs, interface widgets should be designed to be close at hand or hovering near the area of the user's interactions. The "close at hand" widgets not only make handy access possible but also possibly diminish interference with other users.

Another interesting UI issue on MTWs is the use of a menu. Although many gestural interaction systems are intended to avoid using any type of menu, we can't help but notice that it has already been a critical constituent of user acceptance and an entire user culture is now built on the WIMP paradigm. When a desktop software user is disoriented, in most instances s/he will intuitively explore the menu bar or click the right button of mouse trying to call out a popup menu to see if an implicit action can be performed. Operating on a MTW, the user similarly prefers an instructional menu to having to remember various implicit gestures. However, the schemes for invoking and dismissing menus have to be re-designed. Special gesture performance can invoke a popup menu while the active zone in which this action can be triggered should be carefully considered. As well, clicking elsewhere on screen, the principal means of dismissing a menu in desktop software is probably unsuitable because the state of the user's interface widgets will potentially be affected by other user's touch in a multi-user setting. Therefore, there probably needs be a "Close" menu item in the bottom of all multi-touch menus. Or, alternatively, multi-touch menus should slowly disappear after a certain period of time if no menu item is clicked.

3.4 Control of Misoperation

Previous studies showed that the touch screen interface was among the fastest interfaces to use but also the least accurate unless special design strategies were conceived [7]. The fat finger problem has been designated as the largest cause of this inaccuracy. From our own observations, we believe there are also two key reasons which lead to this mis-operation on MTWs: one is the deficiency of differentiation among gestures and the other is accidental touching or incorrect contact area.

For the first cause, the software capability of recognizing a variety of gestures should be taken into account, therefore interaction designers can develop sufficiently distinct gestures. For the second cause, limiting gestures to defined active zones helps to minimize the number of mistaken operations. Active zones could be highly context-dependent and appear in close proximity to active or selected objects. Certain gestures might only trigger actions occurring in active zones. In addition, MTW UIs should similarly provide the user with handy instruments to undo operations and require confirmation for unrecoverable processes as the desktop UI does. One predefined gesture (generally we use a four-fingered leftwards swiping) is a simple way for users to master to perform the undo operation. The confirmation strategy widely using in desktop UIs, in which generally a model dialog box blocks all operations while awaiting the user's confirmation, is inappropriate in multi-user settings. Instead, we use an attention-grabbing "sparkling" message box popping up near the last touch point of the unrecoverable operation to encourage the user to confirm his/her action. This scheme can effectively direct the operator's attention without interrupting other users.

4. CO-LOCATED COLLABORATION

A MTW naturally accommodates multiple users, though the technology is still novel and commercial implementations are scarce. We are experimenting with several aspects of designing for multiple users' synergic interaction with a large MTW. Based on previous work in this area and our own feedback, our own design thoughts and several design issues rather than exhaustive considerations, are presented here.

First, users teaming up before using the MTW are often engaged in simultaneous conversation with each other. So they do not have much additional cognitive capacity to cope with complex software state changes, or have a high tolerance for having their attention distracted to the interaction with the wall instead of with other people. Therefore, a simple interface that allows users to manipulate onscreen objects directly will help with fluent interaction, alternating both social and computer communications.

Second, quite distinctly from users seated around a tabletop who are generally egalitarian, collaborators in the region of a MTW are most likely to have diverse stations or roles. There may be secondary users standing behind the direct users who provide suggestions, or more important users standing farther back to administer processes and make decisions. So the interface and visualization should be serviceable for different perspectives. If designers choose to employ the same interface view for dissimilar users in various distances, they have to find out a compromise scale of interaction effects which are not only perceptible for distant users but also steerable for close operators. Alternatively, with the large size of the MTW this offers the possibility of providing another synchronous but larger interface view for distant users. In addition, an easy-to-use and fault-tolerant interface will make those direct users happy about performing their interactions in public.

Finally, the visibility of actions is a fundamental aspect of group awareness which provides a context for users' activities. Researchers have found that higher levels of awareness for the multi-touch condition accompanied by significantly more actions that interfere with each other lead to interactions in this condition which were more fluid and the interference was quickly resolved [6]. So designers of MTW interfaces have to reconsider the strategy of interference disposal in a collaborative situation. The relative rank of the users can be helpful in conflict resolution, compatible with their awareness and their actual interaction. In the present form where there is a lack of awareness of user identity without the aid of specific devices, the only feasible way is to provide resources for collaborators to negotiate such interference.

In our work we have been putting the above design guidelines into practice in the development of MTW applications. Specifically, we are developing a collaborative tool for network management which visualizes a large communications network and allows (multiple) users to query node or link characteristics, a collaborative mindmap creation tool, and a tool for visualizing large (relational) DBMS schemas. Our wall is described earlier in Figure 1.

5. CONCLUSION

Multi-touch walls offer a unique mix of information density, direct interactivity, and collaboration support. Designers and developers of the interaction on these walls encounter several conceptual, methodological, and technical difficulties. We are exploring these issues, and we have put forward some suggestions to contribute to the effectiveness of MTW interactions. We are still near the beginning in understanding this novel medium and improving our design, and the usability of some design notions needs yet to be verified by user studies. We hope that these observations and considerations will help us with further exploration about how to best design interaction for MTWs.

ACKNOWLEDGEMENTS

This work is supported by China Scholarship Council (CSC) under State-Sponsored Scholarship Program for Visiting Scholars and Science Foundation Ireland under grant 07/CE/I1147.

6. REFERENCES

[1] Balakrishnan, R. and Hinckley, K. 2000. Symmetric bimanual interaction. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (The Hague, The Netherlands, April 01 - 06, 2000). CHI '00. ACM, New York, NY, 33-40. DOI= http://doi.acm.org/10.1145/332040.332404

[2] Collomb, M., Hascoët, M., Baudisch, P., and Lee, B. 2005. Improving drag-and-drop on wall-size displays. In Proceedings of Graphics interface 2005 (Victoria, British Columbia, May 09 - 11, 2005). GI, vol. 112. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, 25-32. DOI= http://doi.acm.org/10.1145/1089508.1089514

[3] Forlines, C., Vogel, D., and Balakrishnan, R. 2006. HybridPointing: fluid switching between absolute and relative pointing with a direct input device. In Proceedings of the 19th Annual ACM Symposium on User interface Software and Technology (Montreux, Switzerland, October 15 - 18, 2006). UIST '06. ACM, New York, NY, 211-220. DOI= http://doi.acm.org/10.1145/1166253.1166286

[4] Kin, K., Agrawala, M., and DeRose, T. 2009. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In Proceedings of Graphics interface 2009 (Kelowna, British Columbia, Canada, May 25 - 27, 2009). ACM International Conference Proceeding Series, vol. 324. Canadian Information Processing Society, Toronto, Ont., Canada, 119-124. DOI= http://doi.acm.org/10.1145/1555880.1555910

[5] MultiTouch Cell. http://multitouch.fi/products/cell/

[6] Müller-Tomfelde, C., Schremmer, C., and Wessels, A. 2007. Exploratory study on concurrent interaction in co-located collaboration. In Proceedings of the 19th Australasian Conference on Computer-Human interaction: Entertaining User interfaces (Adelaide, Australia, November 28 - 30, 2007). OZCHI '07, vol. 251. ACM, New York, NY, 175-178. DOI= http://doi.acm.org/10.1145/1324892.1324925

[7] Olwal, A., Feiner, S., and Heyman, S. 2008. Rubbing and tapping for precise and rapid selection on touch-screen displays. In Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 295-304. DOI= http://doi.acm.org/10.1145/1357054.1357105

[8] Schöning, J., Hecht, B., Raubal, M., Krüger, A., Marsh, M., and Rohs, M. 2008. Improving interaction with virtual globes through spatial thinking: helping users ask "why?". In Proceedings of the 13th international Conference on intelligent User interfaces (Gran Canaria, Spain, January 13 - 16, 2008). IUI '08. ACM, New York, NY, 129-138. DOI= http://doi.acm.org/10.1145/1378773.1378790

[9] Yee, W. 2009. Potential Limitations of Multi-touch Gesture Vocabulary: Differentiation, Adoption, Fatigue. In Proceedings of the 13th international Conference on Human-Computer interaction. Part Ii: Novel interaction Methods and Techniques (San Diego, CA, July 19 - 24, 2009). J. A. Jacko, Ed. Lecture Notes In Computer Science, vol. 5611. Springer-Verlag, Berlin, Heidelberg, 291-300. DOI= http://dx.doi.org/10.1007/978-3-642-02577-8_32