
Multi-User Display Walls: Lessons Learned

Kelly Dempski

Accenture Technology Labs
161 N. Clark St.
Chicago, IL 60626 USA
Kelly.L.Dempski@accenture.com

Brandon Harvey

Accenture Technology Labs
161 N. Clark St.
Chicago, IL 60626 USA
Brandon.L.Harvey@accenture.com

Copyright is held by the author/owner(s).
CHI 2006, April 22–27, 2006, Montreal, Canada.
ACM 1-xxxxxxxxxxxxxxxxxxxx.

Abstract

As pixels continue to get cheaper, computer displays tend toward larger sizes, and crop up in new environments. In order to explore the possibilities for robust interactive walls, we built a large, touchable display, which we then deployed in public several times.

We concluded from this that for display walls, free-hand touch will be a dominant mode of user input, and that moreover, very large touchscreens naturally entail multi-user input. But the classic WIMP (Window, Icon, Menu, Pointer) design paradigm—deeply oriented around processing a single stream of user input, with all action staged on a relatively small screen—requires reconsideration if large applications are to operate in a coherent fashion.

Here we will present past insights and future research questions, based on our experience at the intersection of three software design aspects: touchability, cooperation, and large physical size.

Keywords

Touchscreen, Multiuser, Multi-touch, Wall, Bare-hand Control, Display Wall, Visualization

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

Introduction

To explore the possibilities for wall-sized software in a realistic, unencumbered context, we created a high-resolution display wall, 11 feet wide, with an accurate, high-speed multi-touch capability [1,2]. We then designed a set of prototype applications with the intent of satisfying immediate, real-world use cases, in both business and public environments.

We have tested the system in a number of public contexts, with thousands of users. We have had both casual users, at events like the WiReD NextFest, and more technical users, at the KDD Conference (Knowledge Discovery and Datamining) or FOSE (Government and Defense Expo).

What follows is a brief exploration of some of the graphical interface considerations which have arisen in the course of our experience creating the system, testing it, and industrializing it for deployment. The issues we identify are ones we have not encountered in the research literature, nor in our work with a number of interface designers, some of whom have created wall-sized content for feature films.

Large Size

It is obvious that walls are larger than computer screens, but not as obvious that traditional WIMP (Window, Icon, Menu, Pointer) GUIs embed some assumptions about size. First, they typically assume “random access” to their whole visual surface: the user is expected to be able to see the whole desktop. A parallel assumption is that the user can reach or operate a control anywhere on the surface with more or less equal ease. For example, desktop software draws many visual elements a single, standard size. An

element placed in the corner is no larger than those, say, in the center.

However, for a very large, directly-interactive screen which the user is close to, the “random access” assumption no longer holds. A person standing close to a large surface has a limited envelope of clear vision over that surface. Beyond this envelope, the user’s ability to see screen elements is reduced in some way—for some angles and for some contents, the reduction is severe.

We can postulate a threshold—some combination of size and distance—beyond which the user cannot be assumed to have instant access to all the visual information on the screen. Since it is designed in ignorance of the threshold, a traditional GUI on a sufficiently large screen will put some needed information outside the useful viewing envelope. Researchers have investigated GUI improvements which can help users deal with marginally larger screen sizes [5], often in the context of using a mouse. But beyond the threshold to which we are referring, where a difference in degree turns into a difference in kind, very little work has been done.

Parallel with the constraint of limited vision goes the constraint of limited physical reach. Wall software users can only access a certain number of software elements without having to physically move their bodies. We have written elsewhere about these kinds of effects [3]. We highlight, for example, new design possibilities for software that might guide a user’s body through physical space, perhaps in the course of working through some process. We also warn against moving the user through physical space needlessly—for

example, by creating static click targets in the corner of a screen 11 feet wide.

Other considerations apply which may not be obvious at the outset—for example, the value of a high resolution screen. For many large interactive displays, the number of pixels in the user's optimal view envelope (at arm's length) is quite low. If we want to enable the user to see or do something interesting, we must bring the pixels to where they will be used: we must achieve a pixel density, at arm's length, closer to that of a desktop screen than a projected slide presentation.

Direct Interaction

In light of the issues of limited vision and limited reach, some researchers have elected to keep all user interactions far enough away from the viewing surface that full visual access is never compromised [4]. We feel that the benefits of tangible interaction, higher resolution, and the natural human urge to touch and point to things all argue in favor of close, direct interaction with screen contents.

But to build a direct-interaction system, certain choices must be made early on. What is the input mechanism? And is that mechanism to be serial, or parallel—that is, one user at a time, or multiple? Is it to be mediated through some sort of device, or bare-handed?

We have written elsewhere [1,2,3] about the choices we made in our system. In brief, we argue that bare-hand touch—or some other parallelizable, bare-hands input mode—is an appropriate mode of interaction with large screens for a variety of real-world tasks, both in technical or business settings and in casual, public settings. (See [6].)

The applications we have designed, though sometimes complex in their abilities, use simple touching and dragging as their only inputs. By forgoing gesture languages or other nonobvious input mechanisms [7], we can achieve zero-training-time interactions, as arcade games tend to do. And the lack of any user encumbrance makes a touch system appropriate in a public setting where users don't have a special tool with them—and in professional settings, where most of the users we speak to don't want one.

Multiple Users

One of the things we learned by observing public interaction with our screens is that members of the general public—users accustomed only to single-user applications—tend to approach wall-sized software *assuming* that the software accepts multiple simultaneous inputs.

The observation is somewhat surprising. What it implies, in our view, is that the first two elements we have discussed here—large screens, and direct interaction—imply the third. Unlike a kiosk, wall displays naturally accommodate, and therefore tend to attract, multiple users. Users cannot, or will not, queue up their interactions with the system to suit the system. Our design must simply catch up with the expectations that users will bring to it. The system must understand parallel input.

As one example of many which could be produced, the window has classically been an element of screen organization. It helps cue the user as to which software element has focus, so they know where their input will be directed. It is also a means of grouping items together in a logical fashion.

In wall-sized software, a window may be necessary less as an organizing structure, and more as a protective structure, defining a sphere of user context and control. If a window is activated—a callout from a map, for instance—it may make sense for that window to be “safe” from movement or modification initiated by other software actions (perhaps another user moves the map).

In addition to the problem of parallel input, the designer must also think about parallel output. That is to say, a very large piece of software will be viewed from different vantages at the same time, and must be designed accordingly. If more than one viewer is standing at different points on the wall, they may see quite different visual information, and they may be able to access different controls [3].

What’s more, if more than one viewer is standing at different ranges from the large surface, their view of the contents will also be quite different. The closer viewer will see finer-grain detail in a particular area, as governed by their view envelope. The farther viewer will see less detail—may even have their view partly blocked—but they have the benefit of seeing an “abstract” of the screen contents. This means that they will see some things better than the viewer close at hand. For example, a map graphic seven feet wide is actually somewhat difficult for a near-distance viewer to see in its totality. It is easy to work with details of the map at this range, but hard to see larger patterns. Yet a more distant viewer can see such the larger patterns quite clearly, making for a potentially valuable collaborative scenario.

Conclusion

From what we have seen, interface designers are not currently trained to think about the sorts of situations which crop up when designing wall-sized software. Many details of the WIMP paradigm do not immediately translate onto larger screens and multiple users. We are attempting to point a path toward a new kind of interface paradigm.

Citations

- [1] Dempski, K. L. and Harvey, B.L. Natural Support for Multi-User High Definition Visualization and Collaboration. Presented at IUI 2005, Workshop on Multi-User and Ubiquitous User Interfaces (MU3I’05)
- [2] Dempski, K. L. Harvey, B.L., and Korytkowski, B. Multi-User Affordances for Rooms with Very Large, Interactive, High-Resolution Screens. Presented at CHI 2005, “The Virtuality Continuum Revisited” Workshop
- [3] Dempski, K. L. and Harvey, B.L. Touchable Interactive Walls: Opportunities and Challenges. Conference on Entertainment Computing (ICEC 2005)
- [4] Balakrishnan, R., Cao, X. “VisionWand: Interaction Techniques for Large Displays using a Passive Wand Tracked in 3D”. Proceedings of UIST 2003 – the ACM Symposium on User Interface Software and Technology. p. 173-182.
- [5] P. Baudisch, E. Cutrell, D. Robbins, M. Czerwinski, P. Tandler, B. Bederson, and A. Zierlinger. Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems. In Proc. Interact 2003, pages 57–64, 2003.
- [6] Hardenberg, C., B’erard, F., “Bare-hand Human Computer Interaction”. Proceedings of PUI 2001.
- [7] Terrenghi, L. “Design of Affordances for Direct Manipulation of Digital Information in Ubiquitous Computing Scenarios”. Proceedings of Smartgraphics 2005.