ISBN 0-473-07559-8© 2001 SIGCHI NZ

Development and Application of Large Interactive Display Surfaces

Mark Apperley, Barry Dahlberg, Amos Jeffries, Lance Paine, Malcolm Phillips and Bill Rogers
Department of Computer Science
University of Waikato
Private Bag 3105
Hamilton, New Zealand
coms0108@cs.waikato.ac.nz

ABSTRACT

Recent hardware developments have made possible the construction of relatively inexpensive large interactive display surfaces (LIDS). We have built prototype systems with office whiteboard sized displays using off-the-shelf data projectors and pen digitising equipment. Our group is investigating the use of these prototype systems in a variety of situations, including: programming, teaching, and both local and remote meeting support.

The challenge of this research is to provide a style of user interaction in which the computer is an unobtrusive support agent – offering assistance, recording and replaying data – but not distracting from the interaction between people using the system.

Keywords
large interactive display surfaces, computer supported collaborative work.

1. Introduction

In recent years there has been growing interest amongst HCI researchers in providing computing devices that break the 'desktop' metaphor – to improve the bandwidth of communication between human and computer (or human and human via computer), and to more fully support the rich modes of communication that people routinely use in their day-to-day lives. The area of human computer interaction that we are addressing in this paper is the situation in which a small group of people are interacting and have or are producing shared documents. This includes formal meeting settings in which the shared documents include an agenda, prepared papers and minutes. It also includes design or teaching situations in which people may work to produce notes, sketches, and even prototype computer software.

In these situations all participants need to be able to see shared documents. If the document is not changed or is only changed slightly during the meeting it is sufficient to have a paper copy prepared in advance – an agenda for example. If the document is being written it is a considerable advantage for all members to be able to watch the process. Data input, in a group situation, is normally by one person at a time – although many people may participate during the course of a meeting, especially in design or teaching situations.

A number of technologies are in common use to support group interaction. These include blackboards, whiteboards, overhead projectors, photocopying whiteboards, and projected computer displays. Each has its advantages and disadvantages. Whiteboards and blackboards are probably the most satisfactory for data input – the process is completely transparent. People may take turns at being author by walking up and taking a pen. It is obvious at any time who is doing the input. Everyone can watch the input process as it occurs. The author can stand wherever they like with respect to the document and can gesture directly at it. The audience can see the author and the document together. Everyone, including the author is looking at the same version of the document at the same time. A simple whiteboard however, cannot save the shared document. Someone must take notes or a photocopying whiteboard must be used to automate that process. The main weakness of a whiteboard though, is that it cannot be used to modify or annotate a pre-prepared document.

Overhead projectors allow presentation and annotation of existing documents. The modified documents can be kept and copied. Quality of interaction between author and audience is arguable. Some would say that it allows the author to always face the audience, an advantage in some teaching situations. An alternative view is that the author has been somewhat distanced from the audience, by being behind the projector, and distanced from the shared document by writing and looking in a different place to the audience. Changing authors is not quite as comfortable as with a whiteboard.

Both boards and overhead projector have the advantage that input is by pen. Ideas can be quickly sketched. There are no constraints on the kinds of documents produced. Landay and Myers [1] argue that *sketching is especially valuable for creative tasks*, avoiding concern for inappropriate detail.

The advantage of the computer in a groupwork situation is the flexibility it provides in presenting and recording information. Networked computers also offer the possibility of supporting geographically diverse groups. Individual computers isolate people. Small groups may try to work clustered around one desktop computer. However, a normal desktop display is too small for more than two or at most three people to view at a time. Even then, their interaction is badly constrained. They are forced to sit closer together than would be normal in an office situation. Sharing of input devices – keyboard and mouse – is very awkward. Projecting a computer display onto a screen helps. Many people can then comfortably see shared data. However, a projector doesn't address the input problem. The author usually sits behind the computer and their entry mechanism – mouse or keyboard – is not visible to the audience.

The Large Interactive Display Surface (LIDS) is a wall mounted computer display supporting direct interaction, ie: it is possible to 'draw' on the surface and have the computer respond to and record what is drawn. This is a scaled up version of the pen computing metaphor, however, we have found it more productive to think in terms of a 'whiteboard' metaphor. In our view the large interactive display surface provides a significant step forward from the desktop computer in supporting groups when generating or modifying documents. It combines the transparency offered by a whiteboard, with the presentation and recording power of a computer. Multiple LIDS systems can provide support for people at more than one site.

LIDS equipment is not new. Examples of existing systems are Xerox's Liveboard [2] and DynaWall [3]. However, both systems are expensive. Liveboard systems were sold at approximately $40,000 US until Xerox dropped the product in 1997. What is new is increasingly inexpensive data projection and pen digitisation hardware. As video projectors become more popular in home theatre applications, further price reductions seem likely. We have constructed prototype devices from off the shelf components and are experimenting with these devices and their use. Hurst et al [7] describe a similar system to ours in a lecture capture application.

In section two we describe our prototype systems. The software we are developing for teaching and meeting support is the subject of section three, and finally we present some interim conclusions.

2. Prototype Systems

Our prototypes are modelled on office whiteboards – approximately 900mm high by 1200mm wide, mounted about 1m above ground level. This provides a surface comfortably in reach for most people, and of the same aspect ratio as a desktop computer display (Figure 1).

We use a standard data projector, operating in rear projection mode. In the first prototype the projector operated at VGA (640 by 480 pixels) resolution with a brightness of 300 lumens. Later versions have used SVGA (800 by 600) and XGA (1024 by 768) at 600 lumen brightness. The low brightness projector required subdued light for satisfactory operation. The 600 lumen projectors operated satisfactorily in brightly lit rooms.

Cloth rear projection screens are unsuitable because a firm writing surface is needed. We have experimented with several surfaces. Satisfactory results have been obtained with frosted glass and with plain glass backed by tracing paper, although some problems remain. There is a parallax error between pen and image when the projection surface is on the back of the glass, glass thickness being 4mm to provide reasonable strength. Parallax error can be avoided with glass frosted on the front, but the surface is difficult to keep clean.

To allow pen input we use a Mimio Whiteboard Digitizer [4]. The digitiser takes the form of a bar attached to the top left of the screen (see Figure 1). In normal Mimeo operating mode the user draws with a felt pen held in a Mimio pen holder. The holder and capture bar provide ultrasonic position detection for the pen tip,

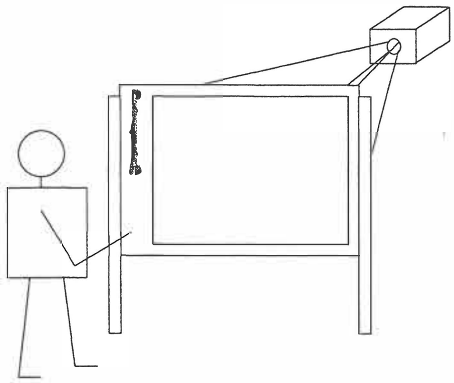

Figure 1. The first prototype

when the pen is pressed to the display surface. In our application we use a 'blank' pen – one with no ink – and Mimio's mouse emulation software. We have experimented with pen tips. A hard plastic tip with glass surface is slippery and noisy to write with. An old felt tip pen with all the ink washed out works well. Simply covering a plastic tip with selotape is also satisfactory.

Prototype one used glass mounted in a picture frame on a stand as shown in Figure 1. It proved difficult to make the stand strong enough to allow writing without some movement in the glass surface. Nevertheless overall operation is satisfactory and the result is a reasonably portable system. The second prototype, built at the Auckland University of Technology was built in a heavy wooden box and used two surface silvered mirrors to shorten the optical path. This system provides a better supported working surface, but is too large and heavy to move. The third prototype is again fixed. This time the projector is mounted in the ceiling, pointing down. A single mirror projects the image onto a screen, build as a 'window' in the wall between two laboratories. Having the screen fitted into a wall again provides good stability. The physical arrangement minimises the space taken in the adjacent room. All three prototypes are operated by standard PC's running Microsoft Windows operating systems. All operate in a satisfactory manner. Experiments with geometry, and pen/screen surfaces are ongoing.

3. Software initiatives

The goal of our software development has to achieve lightweight interaction – to preserve so far as possible for the user their familiar whiteboard style of use, whilst providing the advantages of computer data storage and processing. As noted earlier we are mainly interested in meeting support and teaching, but are keen to explore the use of LIDS systems in as many applications as possible.

Our first experiments were with standard application software. The Mimio mouse emulation software allows use of the Microsoft Windows operating system and much application software. In this mode we have had users play simple games and use the file system explorer to arrange files. Using Paragraph's[5] character recognition system we have also experimented with word processors, and programming environments. The response of users to these experiments was very encouraging. Use of a programming environment to support code inspection was particularly successful.

In this experiment we set a small programming exercise to software engineering students. Each student was asked to write a procedure to lay out some images inside a window. Their procedure formed part of a larger graphic user interfaced program. The code of the procedure was brief enough to display on a screen. Procedure authors were required to explain how their code worked to fellow students in small (4-6) tutorial groups. When the group had formed an opinion about the procedure, interactive testing to verify behaviour was done. The LIDS display was particularly good for testing. Everyone could see clearly. The actions of the person doing the testing were all visible. Use of character recognition software for writing C++ code was not satisfactory. In the few cases where software modification was undertaken, it proved more satisfactory to use a keyboard.

During the summer of 2000/2001 we started development of our own software to use for meeting and teaching support. At present the program is not focussed on one or other objective. It just provides a collection of useful features which we are intend evaluating in each setting. We have developed the features discussed in following paragraphs.

(a) The software uses Microsoft PowerPoint for display and storage. PowerPoint is well suited to lecture presentation. It can also be used in meetings, to present an agenda and summaries of prepared papers. In our system it provides a flexible way of allowing material to be introduced at the start of a session. Using the COM programming interface it was possible for us to write software that controlled and modified a PowerPoint presentation. A particular strength of the approach is that records of a session are simple PowerPoint files. They can be transmitted, replayed, have data extracted, or further edited without our software being involved. This is particularly useful in the teaching situation.

The program implements its control of PowerPoint via a transparent overlay window. PowerPoint runs, but all user input is intercepted by our software. Where user gestures request slide advance, etc, we pass on an appropriate request. The overlay window is also used to display annotations and video.

(b) Annotations: When using an overhead projector, it is common for a teacher to write on slides to make corrections or additions. Often the prepared material is an outline only, and important parts of the lecture content are delivered by annotation. In the extreme case the lecturer begins with blank slides and writes as they go. This freedom to amend a presentation is usually lost when using computer presentation. We consider it an important part of the 'whiteboard' metaphor and strongly support it in our software.

When the user draws with a pen, our system records pen strokes and echoes them onto our screen overlay. Some strokes are interpreted as annotations, and in the final system we intend that some should be recognised as text, but the remainder are taken to be simple drawing. Provision is made for deletion, as described in paragraph (d). When the system reaches the slide advance stage, our system displays the new slide and stores remaining gestures directly into the old PowerPoint slide. If the presenter chooses to review a slide, it will appear fully annotated, as it was when last seen.

Annotations are written into PowerPoint with animation times set to duplicate the timing of presentation (within the time used to show one slide). On replay of a saved presentation, the annotations occur as they were written, synchronised with the Audio track, see paragraph (e).

An original goal was to fully animate the drawing of each annotation, just as the presenter drew it. In the current system, we use the PowerPoint 'swipe' entry for each stroke. Stroke presentation begins at the time (within slide) at which it was actually drawn. Swipe is programmed to occur in the predominant direction of the stroke. Where a simple line is entered the effect is quite good. It appears as it was drawn, although not with exactly the same timing. Where a stroke is circular, both sides of the circle will appear simultaneously and eventually join. For most text entry the illusion of watching it being written is satisfactory. For diagrams the result is not great, but is significantly better than having gestures appear all at once. The 'drawing' process attracts the observers' attention to the new stroke.

(c) Concurrent Use: The first copy of the software to be started functions as a server. Other copies can attach as clients. On startup the user selects the PowerPoint presentation to be used as a basis for the presentation. We assume that each copy begins with the same presentation. Events occurring on each running copy of the software are broadcast to all other copies. The result is that annotations appear live on each display and slide advance is synchronised. The amount of data transmitted is quite low – mostly mouse coordinates while annotating. We also transmit audio and video data as discussed in sections paragraphs (e) and (f).

The software does not manage timing of contributions – or who is presenter at any time. If two people try to draw at the same time, then two annotations appear. Deciding 'whose turn it is' is a social issue. In a single room it is obvious who is holding the pen and reaching for the board. In the concurrent situation presenters must use information from the audio and video channels.

(d) Gesture Recognition: Our goal was to have a lightweight interaction with the system – to impose as little additional overhead as possible in comparison with a standard whiteboard. Nevertheless, there is a need to issue commands to the computer, for example advance slide. The options were to use buttons on the Mimio device; to have display buttons on the screen; or to capture gestures with the pen. We rejected the Mimio buttons because they were too hard to find in a darkened room. Displayed buttons were rejected partly because the drawing surface was too small – we wanted to keep all the available space for user data – and partly because the drawing surface was too big – users wouldn't want to reach over to one side when issuing commands.

Our experience with gestures has been mixed. At first we tried dots (mouse clicks) for advance and later a long left to right line. In both cases the chance of accidental entry when drawing was high. At present the system uses large arrows (Figure 2) and boxes. Left and right retreat or advance slide. Up terminates the session. Drawing a box around a gesture or an element of a presentation deletes it.

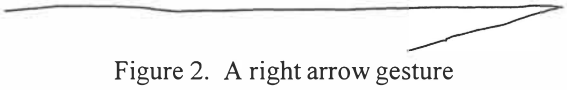

Figure 2. A right arrow gesture

Further development of gesture recognition is an important priority for us. We are experimenting with a partially developed character recognition system based on the work of Rubine [6].

(e) Audio: In the teaching application audio is a central part of the record of a session. After lecture presentation, the PowerPoint file will hold the initial data, the audio track and annotations. In meeting support, the audio data is required for communication between sites.

For replay, audio segmentation is an issue. A person replaying a session may not want to watch/listen to the data associated with an entire slide. We would like them to be able to 'click forward' to a point of interest, rather than wait for sound segments to complete. In the ideal case the presenter would speak on a point and then make an annotation or advance slide. In practice sound and annotation overlap in complex ways. We are experimenting with the use of silence and annotation timing to segment sound. In cases where neither is helpful we fall back on fixed length time periods.

(f) Silhouette Video: A significant difficulty in social interaction for participants in remote meetings, is knowing when it is appropriate to speak or write. In face to face situations we can tell if a person is about to speak, or about to finish speaking from various cues. It is particularly easy to tell if they are about to write on a whiteboard. They have to be standing next to it holding a pen. For our experimental system we are not providing audio input throughout the meeting room. We assume that speakers will stand at the board, and draw on it from time to time.

A full video connection would help with social interaction, but it would be difficult to place the video monitor in the room. If it was on the screen it would take space and distract from the data being discussed. If it were elsewhere it would be distracting or ignored. As a compromise we have implemented a silhouette video, lightly superimposed over the data display. The idea is that it represents the shadow of a person standing behind the display. Being light and monochrome it is not too distracting. It is generated from a low resolution web camera and can be transmitted at 10 frames per second at about 1kbps. The image is not very detailed, but the human eye is very good at finding or inventing detail in moving images. The result provides the social cues required – knowing when a person is about to write on the other LIDS display in a concurrent situation – at acceptable cost in computer performance and data communications load.

4. Conclusions

Only limited usability testing has been attempted so far. Our main objective has been to give as many people as possible an opportunity to try the system. We have used it in some usability testing, on University Open Days with the public, and shown it to any visitors to our department. Our programming team has even tried to use the system for interactive video games. Reactions have been positive.

Our principle goal has been to achieve lightweight interaction. In our view the meeting support software achieves this end. Users seem to catch on to the gestures quite quickly, although we have had problems with gesture annotation collisions.

Acknowledgements

The research described in this paper has been carried out with the support of the Foundation for Research, Science and Technology under the New Economy Research Fund contract number UOWX9910.

References

1. Landay, James A. and Myers Brad, A. (2001) Sketching Interfaces: Toward More Human Interface Design, *IEEE Computer,* March 2001, 56-64.

2. Moran, T.P., van Melle, W. and Chu, P. (1998) Spatial Interpretation of Domain Objects Integrated into a Free Form Electronic Whiteboard, *Proc UIST'98*, ACM, New York, 175-184.

3. Streitz, Norbert, A.; Tandler, Peter; Muller-Tomfelde, Christian; Konomi, Shin'ichi, (2001), Roomware: Towards ..., *HCI in the New Millenium (Ed J.A.Carroll)*, Addison Wesley.

4. Virtual Ink Corporation (1999): http://www.virtual-ink.com/.

5. Paragraph (1999); http://www.paragraph.com/.

6. Rubine, D. (1991) Specifying Gestures by Example, *Computer Graphics,* 25 (4), 329-337.

7. Hürst, W., Maass, G., Müller, R. and Ottmann, T. (2001) The "Authoring on the Fly" System for Automatic Presentation Recording, *Proc ACM SIGCHI 2001, Extended Abstracts*, 5-7.