EXHIBIT 1

MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

DiamondTouch SDK: Support for Multi-User, Multi-Touch Applications

Alan Esenther, Cliff Forlines, Kathy Ryall, Sam Shipman

TR2002-48 November 2002

Abstract

DiamondTouch is a multi-touch input technology that supports multiple, simultaneous users; it can distinguish who is touching where. We present the DiamondTouch SDK; it provides support for the development of applications that utilize DiamondTouch s capabilities to implement computer-supported collaboration and rich input modalities (such as gestures). Our first demo illustrates the basic utilities and functionality of our system. Our second demo, a multi-user map application, highlights DiamondTouch s ability to sup-port input from multiple, simultaneous users and exploits Di-amondTouch s ability to identify the owner of each touch. Our third demo illustrates DiamondTouch s ability to run with existing applications by providing a mouse emulation mode. DiamondTouch is well-suited to shared-display applications. It is suitable for front-projected video of the computer display, which facilitates direct manipulation of user interface elements and provides a shared focus of attention for collab-orating users. Possible applications include command-and-control command posts, control rooms, business or technical meetings, and a variety of casual applications in the home, at schools, and in retail settings.

CSCW 2002 Demo

DiamondTouch SDK: Support for Multi-User, Multi-Touch Applications

Alan Esenther, Cliff Forlines, Kathy Ryall, Sam Shipman
Mitsubishi Electric Research Laboratories
201 Broadway
Cambridge, MA 02139
{esenther, forlines, ryall, shipman}@merl.com

ABSTRACT

DiamondTouch [1] is a multi-touch input technology that supports multiple, simultaneous users; it can distinguish *who* is touching *where*. We present the DiamondTouch SDK; it provides support for the development of applications that utilize DiamondTouch's capabilities to implement computer-supported collaboration and rich input modalities (such as gestures). Our first demo illustrates the basic utilities and functionality of our system. Our second demo, a multi-user map application, highlights DiamondTouch's ability to support input from multiple, simultaneous users and exploits DiamondTouch's ability to identify the owner of each touch. Our third demo illustrates DiamondTouch's ability to run with existing applications by providing a mouse emulation mode. DiamondTouch is well-suited to shared-display applications. It is suitable for front-projected video of the computer display, which facilitates direct manipulation of user interface elements and provides a shared focus of attention for collaborating users. Possible applications include command-and-control command posts, control rooms, business or technical meetings, and a variety of casual applications in the home, at schools, and in retail settings.

INTRODUCTION

DiamondTouch is a multi-user, multi-touch technology first presented at UIST 2001 [1]. It allows multiple people to simultaneously interact with a surface; it can distinguish *who* is touching *where*, and also if a person is touching in multiple locations. DiamondTouch is currently being used on a tabletop display with front projection, although it can be layered onto other surfaces.

This type of input device is particularly relevant for single-display groupware applications [4].[1] DiamondTouch enables a group of people to interact with a surface without interfering with each other. Furthermore, objects placed on the surface do not interfere with DiamondTouch operation.

Figure 1: **Prototypical DiamondTouch setup: front-projection onto a tabletop surface.**

DiamondTouch uses an array of antennas embedded into a surface, with each antenna transmitting a unique signal. Each user has their own receiver, generally attached to their chair. When a person touches the surface, energy from nearby antennas is coupled through the user to their receiver. Using this mechanism, the system determines who is touching where. For a more detailed discussion of DiamondTouch and comparison to other input devices see Dietz and Leigh [1]. Figure 1 illustrates a typical DiamondTouch setup.

DIAMONDTOUCH SDK

The DiamondTouch surface consists of overlapping vertical and horizontal arrays of antennas. The hardware periodically produces frames of data containing scalar values that measure the proximity of the user's finger(s) to each antenna.

The DiamondTouch Library (dtlib) reads these data frames

[1] We and others prefer the term *shared-display groupware* to indicate that multiple shared displays may be used, but to the authors' knowledge this term has not yet been popularized.

from the DiamondTouch device and affords access to the raw data and to various abstractions and interpretations of that data, such as the location of the maximum proximity (the touch point) and the bounding box of the area touched. Other abstractions are possible and are the subject of ongoing research. A weighted interpolation algorithm increases the effective resolution to 2500 x 1500. Median filtering, hysteresis, and adaptive touch thresholding are used to improve robustness in the face of RF interference and other environmental variables.

The SDK consists of dtlib (ANSI C), jdt (a Java interface layer), merldt (a Windows application providing mouse emulation, projector calibration, and various diagnostic displays), and a simple multi-user application example.

RELEVANCE TO CSCW

Computer-assisted collaborative environments allow multiple users to work together to solve difficult problems. However, simply scaling the techniques that work well for a single user creates awkward systems. In a collaborative environment, sharing a single mouse is very awkward. Providing multiple mice only makes matters worse - it is extremely difficult to keep track of what each user in the group is doing. Touch screen technologies appeared promising since a glance reveals each users activities. However, touch technologies are usually restricted to detecting a single touch point. They also tend to be somewhat fragile.

DiamondTouch is unique hardware that provides multi-touch input; it is a touch technology specifically designed for a multi-user environment. Enabling input from multiple participants is important for computer-supported collaboration and access to this hardware will facilitate exploration in these areas. Furthermore, the DiamondTouch technology is still a research prototype; MERL relies on outside collaborators (in this case academic institutions) for feedback on and development of the technology and its applications. DiamondTouch has been successfully used by the University of Maryland in conjunction with MID [3] in applications such as KidPad [5] and SearchKids [6]. In the near future MERL will be distributing DiamondTouch to a number of other universities.

DEMO APPLICATIONS

Our first demonstration illustrates the basic utilities and functionality of DiamondTouch and its SDK. The basic display, shown in Figure 2, provides low-level information about each user's interactions with the DiamondTouch surface. A different color is used for each user. In this example, each user is touching the table with two fingers; the bar graph shows the signal strength for each user's touch. The outlined rectangles indicate the bounding box of the area touched by each user. Note that the boxes overlap — the users' touches are tracked independently by DiamondTouch, and do not interfere with each other. Other diagnostic information (touch thresholds, calibration, touch points, etc.) is included in the display.

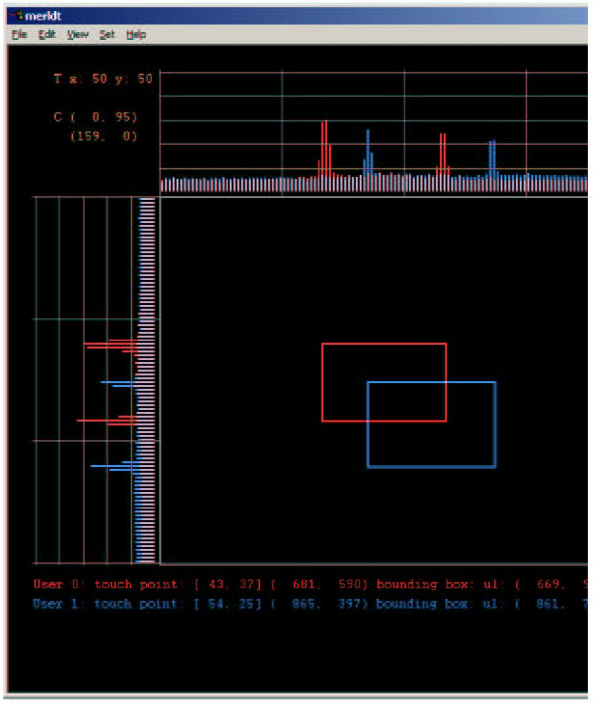

Figure 2: **The display from `merldt` provides low-level information about DiamondTouch and users' interactions with it.**

In our second demo we present a prototype multi-user map application. In this application the display, which is projected onto a DiamondTouch surface, contains a satellite map image. Different views can be overlayed onto the map. One lens, for example, provides a street map view of the area in question. Another lens provides an annotated view highlighting areas of interest (e.g., ice cream parlors, museums and other attractions). A third view displays traffic information. Each user selects his or her own view, which may be the same as or different from other users' views. As each user touches the DiamondTouch display his or her lens appears, revealing the appropriate customized information. Figure 3 shows two users interacting with the application.

More generally, any layered information can be displayed in this application. In Figure 4 we see a sample session with our application in which three people are interacting with a nautical data set. In this sample demonstration application, for example, four 'layers' of information are provided: satellite imagery, depth charts, topographic data, and street maps. The users can interact with the application in parallel, and each has his own customized view of the data.

Figure 5 provides the details of the application by showing a sample screen shot (contents of the display only). In the bottom right corner of the display we see a thumbnail view of the entire data (map) set. Users navigate the larger map

Figure 3: **Two users interacting with a sample DiamondTouch application.**

Figure 4: **An overview shot of DiamondTouch running DTMap, a multi-user, layered information, map application.**

display by touching the image directly, or using the compass buttons to move the currently selected region, which is shown by a red rectangle. That region is then shown in detail on the work surface. Users may select which 'view' will serve as the background image from any of the available views; in this case the satellite view is used as the background image for the work surface. The toggle button shows the currently selected background. The region of interest and the background image are globally shared resources — any user may interact with them, and all users have the same 'state'. This aspect of the application is similar to single-user applications.

A novel aspect of our application is that each user interacts with his or her own view of the data. The screen shot in Figure 5 is taken from the interaction in Figure 4. The oval on the left reveals topographic information. The rounded rectan-

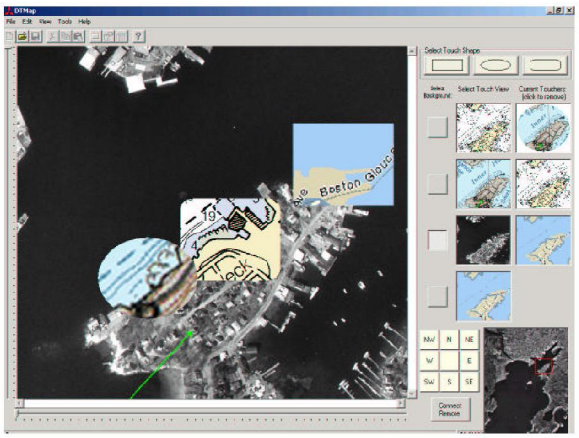

Figure 5: **DTMap: A screen shot to illustrate the application details.**

gle in the center provides depth information about the water near this island. The rectangle in the upper right provides a street map view.

Further, because it knows who is touching where, DiamondTouch can provide functionality and privileges to individual users. For example, each user can control not only the contents of his or her lens, but also its shape and size. The goal of this demo is to highlight DiamondTouch's capabilities; we hope other users will contribute to developing content for DiamondTouch.

Our third demo illustrates the mouse emulation capabilities of DiamondTouch. There are two aspects to this demo. First, this capability allows traditional software to be used with DiamondTouch. We currently have several mouse modes (one-touch, two-touch, etc) and are experimenting to determine how best to implement a fully-functioning mouse with DiamondTouch. Our mouse emulator works with traditional software. Second, under Windows the DiamondTouch SDK can generate 'augmented mouse events' that contain the user ID info in an extra field. In this manner C++ Windows code using a Win32 API call can get the ID information for each mouse event as it is processed by the application. The application then needs to store the state for the different users, and also some mouse information. We have a sample drawing program (a modified MFC demo) to illustrate the use of 'augmented mouse events' for writing applications for DiamondTouch.

ACKNOWLEDGMENTS

The authors would like to thank other members of the DiamondTouch team at MERL: Shane Booth, Paul Dietz, Darren Leigh, Joe Marks, Kent Wittenburg and Bill Yerazunis.

REFERENCES

1. Dietz, P. and Leigh, D. DiamondTouch: A Multi-User Touch Technology. In *Proc. of UIST '01*, ACM, NY, 2001, pp. 219–226.

2. Bier, et al. Magic Lenses. Toolglass and Magic Lenses: The See-Through Interface. In *Proc. ACM SIGGRAPH Conference 1993*, ACM, NY, 1993, pp 73–8.

3. Hourcade, J. P. and Bederson, B. Architecture and Implementation of a Java Package for Multiple Input Devices (MID). *Tech Report HCIL–99–08*, Computer Science Dept., Univ. of Maryland, College Park, MD, 1999.

4. Stuart, J. and Bederson, B.B. and Druin, A. Single Display Groupware: A Model for Co-present Collaboration. In *Proc. of CHI '99*, ACM, NY, 1999, pp pp. 286–293.

5. Juan-Pablo Hourcade et. al. KidPad: Collaborative Storytelling Tool for Children. In *Extended Abstracts CHI 2002*, ACM, NY, 2002.

6. Juan-Pablo Hourcade et. al. SearchKids: A Digital Library Interface for Young Children. In *Extended Abstracts CHI 2002*, ACM, NY, 2002.