

# A Tabletop-Based Real-World-Oriented Interface

Hiroshi Takeda<sup>1</sup>, Hidetoshi Miyao<sup>2</sup>, Minoru Maruyama<sup>2</sup>, and David Asano<sup>2</sup>

<sup>1</sup> Graduate School of Science of Technology, Shinshu University  
4-17-1 Wakasato, Nagano, 380-8553, Japan  
<sup>2</sup> Faculty of Engineering, Shinshu University  
4-17-1 Wakasato, Nagano, 380-8553, Japan  
t09a532@shinshu-u.ac.jp,  
{miyao,maruyama,david}@cs.shinshu-u.ac.jp

**Abstract.** In this paper, we propose a Tangible User Interface which enables users to use applications on a PC desktop in the same way as a paper and pen on a desk in the real world. Also, the proposed system is cheaper to implement and can be easily setup anywhere. By using the proposed system, we found that it was easier to use than normal application user interfaces.

**Keywords:** Tangible user interface, DigitalDesk.

## 1 Introduction

Real-world-oriented interfaces allow more intuitive operation of computers compared to keyboards and mice because the objects are touched directly. We aim to implement a real-world-oriented interface that allows users to intuitively operate computer applications on a tabletop.

Similar research includes the “DigitalDesk” by Wellner [1], “The Everywhere Displays Projector” developed by IBM [2], “PaperWindows” by Human Media Lab [3] and “UbiTable” developed by MERL [4].

In DigitalDesk a way to operate a computer on a desktop without a mouse or keyboard is proposed. The user operates the computer by touching images of computer applications projected onto his table by a projector attached to the ceiling over the table. The shape of the user's hand is determined by processing the image taken by a camera placed next to the projector. Our system is similar in that it also uses a projector to project images of the applications onto a table so that a user can intuitively operate the applications. However, while the images on the table are touched in the DigitalDesk, our system uses real paper and a pen to operate the applications. In this way, we feel that movement of the applications can be done more intuitively. Moreover, our system is cheaper and easier to implement.

The Everywhere Displays Projector is a system that allows multiple users to efficiently work by projecting GUIs to various places in a room. Distortion in the projected image results, but this can be compensated for by using a hardware accelerated projection transform. The purpose of this system is completely different than ours since we are only interested in operating applications on a table, but we also use a similar approach to speed up rendering of the projected images.

UbiTable is a system to share applications among multiple users on a table. In particular, applications in mobile appliances, e.g., notebook PCs and PDAs, can be shared on the table. Our system is similar except for the ability to handle multiple users. However, UbiTable requires the use of a special device called “DiamondTouch”[5]. Our system does not require any special equipment and can therefore be implemented with easy to acquire components.

PaperWindows is a system that allows multiple users to share applications displayed on a projection board that can be moved freely around a room. This system is similar to ours in that a pen can be used to draw and click on the projection board to perform various operations. However, the projection board is flexible and can be moved around a room, so equipment to track the projection board is necessary. This makes the system more expensive and difficult to implement than ours.

## 2 System Overview and Proposed Features

As shown in Fig.1, a projector, infra-red camera and web camera are placed next to the desk. A projection board with infra-red LEDs is placed on the desk. A pointing device is also required. The projector is used to project the application windows on the desk, while the user operates the applications on the projection board using the pointing device.

Our system has the following features:

- As shown in Fig.2, the user can hold the projection board and move it freely. When the board is moved, the application window projected from the projector follows the board. This gives the user the impression that the window is a piece of paper and can be operated as such.
- To operate the application, we use a pointing device. When the button on the side of the pointing device is pressed, infra-red light is emitted from the LED on the tip. When the button is pressed while the pointing device is on the tabletop, it is interpreted as a mouse click, and when the button is held down and the pointing device is moved, it is interpreted as a mouse drag operation. Using the pointing device in this way on the projection board allows the user to operate the applications.
- In order to handle several applications, we implemented a function called the “detach” operation that allows the user to fix a window to the tabletop (Fig.3). To detach a window, the user clicks the “Detach” button on the menu displayed on the side of the projection board. When a window is detached, the image that was displayed on the projection board is fixed in place on the tabletop. To move the window from the tabletop back to the projection board, we provide the “attach” operation (Fig.3). To attach a window, the user moves the projection board to the window fixed on the tabletop and then clicks the “Attach” button.
- The projection board image is always displayed on top of the other windows.
- Sometimes, users may want to move a window fixed to the tabletop to another place. In order to do this, the user just has to drag the window using the pointing device (Fig.4).

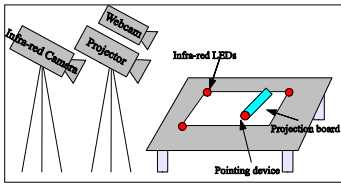


Fig. 1. System overview

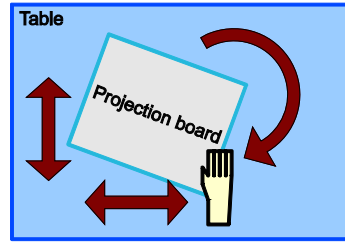


Fig. 2. Movement operation of projection board

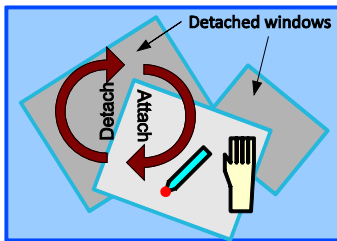


Fig. 3. Attach and detach operation

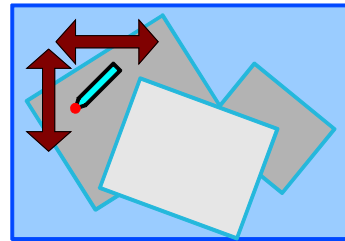


Fig. 4. Operation of detached windows

## 3 System Implementation and Experiments

### 3.1 Implementation

We implemented the above mentioned features in the following way. The necessary equipment is shown in Fig.1. The infra-red camera is an inexpensive Wiimote device. The Wiimote is limited in that it can only receive a maximum of four infra-red spots, so we use three spots to detect the projection board and one spot to detect the pointing device. By detecting these infra-red spots with the infra-red camera, we can construct the image to be projected and extract the operations to be done on the application windows.

**System Calibration.** We aim to create a flexible system that can be used by simply putting the equipment next to any type of desk. In order to achieve this goal, the coordinate systems of the infra-red camera, projector and tabletop must be calibrated. This calibration is done using the following procedure:

1. The web camera and tabletop are calibrated by placing a checker board on the tabletop and then taking an image with the web camera.
2. The projector and tabletop are calibrated by projecting a gray code pattern [6] on the tabletop and then taking an image with the web camera.
3. The projector and infra-red camera are calibrated by projecting several points onto the tabletop and having the user place infra-red LEDs on these points.

**Estimation for the position of projection board.** During the operation, real-time estimate of position and orientation of the projection board on the desk is necessary.

The estimation is carried out by detecting three infra-red spots at vertices of the projection board. Correspondence is taken between the vertices of the rectangular model and the detected spots in the image, using the information on lengths of edges of the rectangle.

Even if one spot is not observed due to occlusion, motion estimation is achieved. For the visible spots, correspondence is inferred based on simple nearest neighbor search from the positions of vertices in the previous frame. Let  $\mathbf{x}_i$  and  $\mathbf{m}_i$  ( $i=1,2$ ) be the observed spots and the corresponding model vertices, where

$$\mathbf{x}_i = R(\theta)\mathbf{m}_i + \mathbf{t}, R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$$

Then, the following equation is derived.

$$\Delta \mathbf{x} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{x}_2 - \mathbf{x}_1 = R(\theta)(\mathbf{m}_2 - \mathbf{m}_1) = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} \Delta m_x \\ \Delta m_y \end{bmatrix}$$

We can estimate  $\theta$  by solving

$$\begin{bmatrix} \Delta m_x & -\Delta m_y \\ \Delta m_y & \Delta m_x \end{bmatrix} \begin{bmatrix} c \\ s \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \theta = \tan^{-1} \left( \frac{s}{c} \right)$$

### 3.2 Experiments

We did experiments with our system. Fig.5 shows an example of an application window displayed on a tabletop. We confirmed that the features described above functioned properly by trying various operations. We also showed that windows could be moved in the same way as moving a piece of paper, which results in an interface that is more intuitive to use than normal computer interfaces.

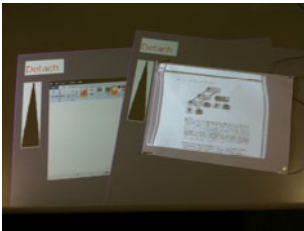


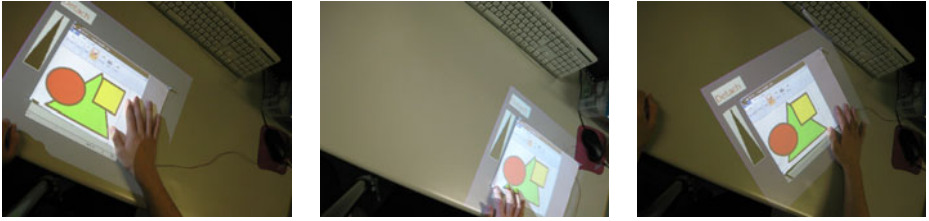
Fig. 5. Example of application windows displayed on a tabletop



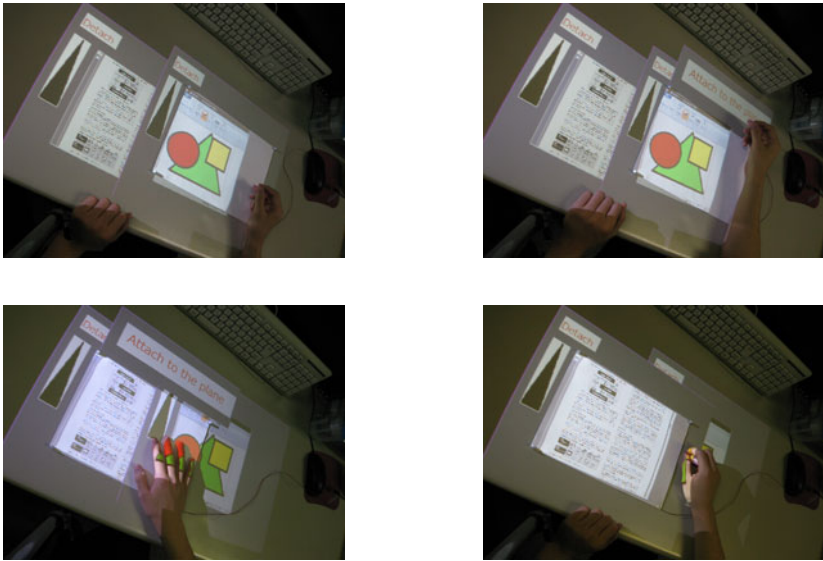
Fig. 6. Environment overview

Experiments were done as shown in Fig. 6. The computer used was a Windows PC, X200(Lenovo), the camera was an easily available web camera, Live! Cam (Creative), the infrared camera was a Nintendo Wiimote and the projector was a low cost BenQ MP620C.

**Moving the projection board on the tabletop.** In Fig. 7 we show how to move an application attached to the projection board (Fig. 2). The projection board, on which has an application displayed, is positioned on the tabletop. The user can move or



**Fig. 7.** Translation and rotation of the projection board

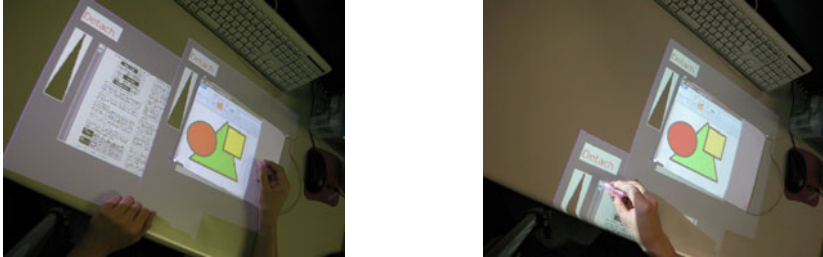


**Fig. 8.** Attach and detach

rotate the application by touching the projection board. If the movement is too quick, a slight delay in displaying the image results. Also, the brightness of the image depends on the distance from the projector.

**Attaching and Detaching Applications.** In Fig. 8 we show how to attach and detach an application and also how to change from one application to another (Fig. 3). First, the user clicks the detach button positioned at the top left of the application with the pointer. This results in no applications being attached to the projection board. In this state, an attach button is created at the top of the projection board. Second, the user moves the application he wants to attach to the projection board. Third, the user clicks the attach button to attach the desired application to the projection board. The attached application is displayed on top of any other applications and becomes the focus of further operations. Experiments with two applications showed no problems switching between them.

**Moving a detached application.** In Fig. 9 we show how to move an application that is not attached and is displayed below other applications (Fig. 4). The position of an application that is not attached cannot be changed by moving the projection board. To accomplish this, we use the pointer. By dragging the pointer on the tabletop, the closest application moves toward the pointer. Rotation of the application is not possible in this case. In experiments with two applications, we were able to achieve the desired results.



**Fig. 9.** Moving a detached application

## 4 Future Work

In future work, instead of a projection board, we plan to use a 3D type of projection object. In Fig. 10, an image being displayed on a cube is shown. In this experiment, we determined the orientation of the cube in a different way in order to render the textures. We plan to use various shaped objects in combination with the projection board.



**Fig. 10.** Projection onto a 3D object

## 5 Conclusions

In our research, we proposed and implemented the basic features for a tabletop interface using a projection board and pointing device and confirmed its operation. Our system has the advantages of being inexpensive (the necessary equipment can be purchased for about \$500) and easy to setup.

Problems that need to be examined are as follows:

- Images far from the projector are darker and have lower resolution. Therefore, text can be hard to read.
- If the projection board is moved too quickly, then the projected image may not be able to keep up.

In future work, we plan to work on methods to solve the above problems and to examine interfaces other than the projection board, such as a cube. Also, we are planning research on using the movement of a user's fingers as an input device.

## References

1. Wellner, P.: Interacting with paper on the Digital Desk. *Communications of the ACM* 36(7), 87–96 (1993)
2. Pinhanez, C.: The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) *UbiComp 2001*. LNCS, vol. 2201, pp. 315–331. Springer, Heidelberg (2001)
3. Holman, D., Vertegaal, R., Altosaar, M.: Paper Windows: Interaction Techniques for Digital Paper. In: *CHI 2005*, pp. 591–599 (2005)
4. Shen, C., Everitt, K., Ryll, K.: UbiTable: Impromptu face-to-face collaboration on horizontal interactive surfaces. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) *UbiComp 2003*. LNCS, vol. 2864, pp. 281–288. Springer, Heidelberg (2003)
5. Dietz, P., Leigh, D.: Diamond Touch: A Multi-User Touch Technology. In: *UIST 2001*, pp. 219–226 (2001)
6. Lee, J., Dietz, P., Aminzade, D., Hudson, S.: Automatic Projector Calibration using Embedded Light Sensors. In: *UIST 2004*, pp. 123–126 (2004)