One-finger Interaction for Ubiquitous Environment

Takashi Nakamura, Shin Takahashi, Jiro Tanaka
Department of Computer Science
University of Tsukuba
Tsukuba, Ibaraki, Japan
Email: {takashi,shin,jiro}@iplab.cs.tsukuba.ac.jp

***Abstract*—We propose new interaction techniques named ”One-finger Interaction” in the ubiquitous environment in a home. One-finger Interaction is an interaction technique for doing various operations by moving the fingertip while pointed by the index finger. It is possible for the user to operate from a distant place. Moreover, because the user uses only a LED device and the recognition of the fingertip is robust, this method can be used easily in different environments. The user can also change the menu according to the preference and applications. Therefore, it is possible to use it in various environments. We developed two prototype interfaces for One-finger Interaction. One is an interface to execute operations by moving the fingertip and crossing icons. The other is that the user selects a rough position of the menu by the direction of movement of fingertip, then selects a detailed position and decides the menu by the gesture which moves it like drawing circle. These interfaces allow the user to perform various operations by moving the fingertip. Moreover, because it is possible to move from the end of one operation to the next action at once, the user can operate menus continuously.**

***Keywords*-Hand gesture; Finger movement**

I. INTRODUCTION

Along with the development of sensors, network, and computer technologies, the usage of the computer in a home will change in the near future. Various consumer electronics (e.g. television, PC, audio, photo viewer, DVD player, etc.) will be embedded in the home, and the user will interact with them on one display. It is also expected that, in a home, many equipments (e.g. light switches, sensors, etc.) are connected by the network, and the user can check and control them on the TV screen. In such situations, it is desirable that the user can control them from the remote position so that the user does not need to move around to control the device at the close position.

One possible interaction technique for that purpose is hand gesture interactions[9], [13]. For example, the user's hand movement, the posture of the hand, and the tilt of the hand can be used as input, and, by those gestures, he/she can execute various operations. The early work of hand gesture interactions started around 1980s[5], and many researchers have designed various interaction techniques[7], [15], [12].

However, there are several problems in hand gesture interations. First, the recognition rate of the hand gesture is usually not so good due to various reasons. One reason is that the position of the hand is blurred because of the user's hand tremor. Another reason is the difficulties of making recognition system for complex gestures.

Another problem of the hand gesture is the lack of versatility. They usually assume a specific environment, and it is hard to be used in various environments and applications. It is also a problem that the user is soon tired because these interactions need to move his/her hand a lot.

To cope with these problems, we focus on some simple movements of fingertip. They are very simple so that it is easy to recognize, and the user can do them without hand tremor. By combining these basic gestures, we can design sufficiently complex interaction system.

We call this interaction technique as ”One-finger Interaction”. In this technique, the user can perform all operations by the combination of the basic movements of his/her fingertip (e.g. index finger). Currently, we define three types of movements as basic operations. The user can select buttons, operate menus, and even input texts with the combination of these three operations.

In this paper, we describe One finger Interaction, and also explain the implementation of two prototype systems that use One-finger Interaction. Our prototype systems only uses a usual web-camera for recognition of the gesture. Only by putting a LED marker on the user's fingertip, the system can robustly recognize One-finger Interaction. We also show the result of preliminary evaluation of the system.

II. ONE-FINGER INTERACTION

When we design One-finger Interaction, it is important how the user interact with a screen using movement of fingertip. In addition, it should be able to correspond to various applications flexibly. In One-finger interaction, the user can interact using menu selections by movement of fingertip. It also can correspond to various applications by changing the contents of the menus freely.

A. Basic operations in One-finger Interaction

In One-finger Interaction, the user can execute all operations using finger movement. However, we thought we should define the operations using the movement of fingertip before designing interaction methods. In this time, we defined three basic operations named ”double-crossing”[14], ”direction of the movement”, and ”rotate operation”.

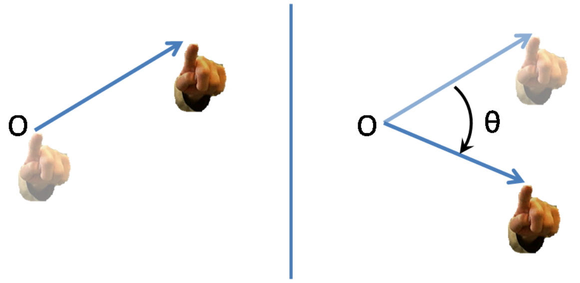

Figure 1. Gestures using movement of fingertip (left: moving to a direction, right: rotation)

1) Double-crossing: We paid attention to the crossing technique[1] used by the pen base interface[3], [6] as a technique. In this technique, the user can do a operation by intersecting with the target. We tried the technique which combined movement of fingertip and crossing technique. However, it is not possible to operate like putting down the pen in One-finger interaction. Therefore, the user sometimes selected other targets which not intended to select, because the pointer was always moved by movement of fingertip. Because he/she needs to make the pointer a detour to avoid the wrong selection, it is not possible to operate efficiently.

We used the technique that the user was able to select a menu if crosses it twice in short time. The pointer can be moved by the beeline, because the operation is not done even if the user crosses other menus one time by using this technique. The influence of the hand tremor is also not so high compared with the crossing of one time, and the user can decrease the non-intended operations.

2) Direction of the movement: "Direction of the movement" uses in which direction the fingertip moved. When the user moves his/her fingertip like the left of Figure 1, the system judges the direction of the movement and executes an operation corresponding to it. Directions are not detail degree of the angle but uses eight azimuths (up, down, right, left, upper right, upper left, lower right, and lower left). The system can recognize easily and the misidentification is also low because of using a rough direction. Moreover, the user can operate quickly because he/she can move his/her fingertip roughly.

3) Rotate operation: "Rotate operation" is an operation technique using the movement of fingertip like drawing a circle from a point (see the right of Figure 1). The user uses turning degree (θ in Figure 1) made in this rotation. The user can execute operations (e.g. scrolling, adjustment of volume etc.) when the turning degree is more than the threshold. The user can repeat a same operation quickly because he/she can keep drawing a circle by his/her fingertip.

B. Menu operation by basic operations

Though the user can interact by using three basic operations, it is difficult for him/her to obtain visual feedback which operation he/she can do now. In One-finger Interaction, we display the menus, and the user can interact through selecting them using three basic operations. The user can understand visually which operation he/she can do now or which menu he/she selected by displaying the menus.

We made two prototypes of One-finger Interaction. One is called "crossing-type interface" that the user can execute operations by selecting the icons and using "double-crossing". The other is called "direction-rotate interface" that the user selects the menus by combining "direction of the movement" and "rotate operation". It is not necessary for the user to memorize the movement of fingertip for operations, because he/she only has to move it according to the displayed menus. We explain the details of them in Section III.

C. Usage of the interfaces

In this method, the user can change the menus according to the applications. For example, when the user watches the TV, he/she can allocate the operations such as "change of the channel" and "adjustment of the volume" to the menus. The user can also do such as "scroll" and "switch of the tabs" to them in case of web browsing.

One-finger interaction can correspond to different applications like this by the change of the content of the menu. However, the menu frequently used is different in each user even if it is the same application. Therefore, the menus should be able to be set freely according to the user. In One-finger Interaction, the menus that suits the user are generated by reading the XML files that the user made beforehand when the system starts. XML files have the information about each menu such as the function, the position, information on the display and so on.

The system arranges many menus in one window. The user can execute them by selecting the arranged menus. However, it is difficult to display all menus in the window because we assume to use them by many applications and there is a lot of numbers of menus. In one-finger Interaction, the user can switch the displayed menus. The user can switch the menus by manual. The user can also do them automatically when the application starts or changes.

D. System architecture overview

We explain about the basic structure of One-finger Interface. The basic structure of our system consists of one PC and one camera. The system takes a image of the movement of user's fingertip by using the camera, then detects the position of the finger and interacts with the target object. In case of using a large display environment, a camera set up in front of the user as shown in Figure 2. A PC recognizes the finger and interacts with a large display.

As for hand detection, the user put a LED device like Figure 2. The system detects the brightness of LED and detect the finger position from the light position. The system can detect the finger at a real time and become robust by using this method. The strain to the user is also reduced because we use only one simple device.

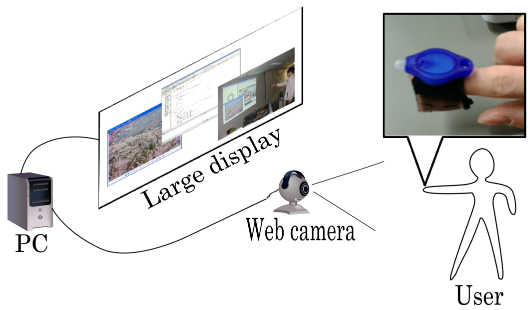

Figure 2. System setting in case of a lerge display

III. Prototypes of One-Finger Interaction

A. Crossing-type interface

Overview of the crossing-type interface is Figure 3 and it displays on the existed GUI in overlay, and the user can easily find the menu for operation. This interface consists of two parts. First is called header (see the upper part of Figure 3). Second is menu window (see the lower part of Figure 3). Menu window consists of the icons called click icon (see Figure 4) and menu icon (see left of Figure 5). The user can create this part individually. The detail of the header is Figure 6. The system creates this part automatically. It is easy for the user to select the menu immediately after having found it and also to continue doing the same menu.

1) Flow of interaction: The user moves the pointer by movement of fingertip in the interaction using this menu interface. He/she also selects the menus using the pointer movement and double-crossing, and then the system does a operation. To select the same menu many times, he/she executes the same operations continuously.

2) Function of the crossing-type interface: The crossing-type interface has three basic functions, ”window movement”, ”switch of the menu window”, and ”switch of the display of the window”.

”Window movement” is a function that moves as the window follows according to the movement of the pointer so that the window is always in surrounding of the pointer. The window moves according to the direction and distance when the pointer parts from the window. The user can switch on/off of the following by selecting the pin icon which is in the header (see Figure 6).

”Switch of the menu window” is a function that change the menu window. The crossing-type interface is composed of some menu windows. The menu window changes by selecting the tab that exists in the header (see Figure 6), and the menus that can be used also change. In the use of the tabs, the user can create menus of various applications

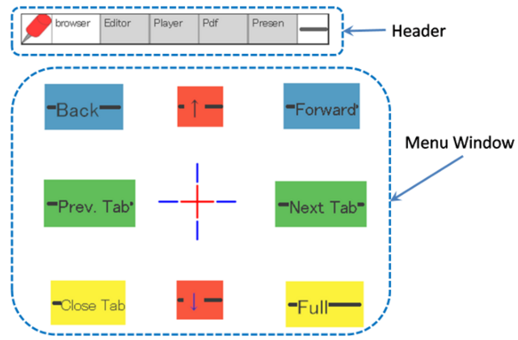

Figure 3. Overview of crossing-type menu interface

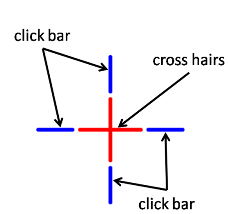

Figure 4. Click icon

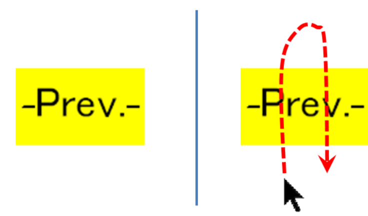

Figure 5. (left) Menu icon, (right) Select menu icon

in the tabs. It is also possible to make it by dividing the menus to operate one application in two or more tabs.

The menu window might sometimes hinder user’s works spaces because the menu window is always displayed in front of the existing GUI. ”Switch of menu window” is a solution of this problem which the user can minimize the menu window if it is necessary or not. When he/she selects the minimize icon (see Figure 6), all menus become hidden, and the system displays only the minimize icon. In such a condition, the menu can be returned to former size by selecting the minimize icon again.

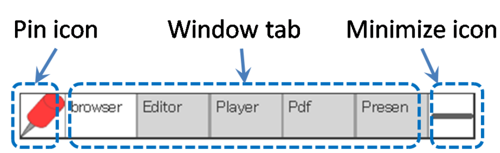

Figure 6. Header

3) Menu icon and Click icon: This interface has two kinds of icon. First is menu icon (see left of Figure 5). Second is click icon (see Figure 4).

The user can do basic menu operations by using the menu icons, and most of the menu window consists in them. The command such as shortcut keys is allocated to each menu icon, and the user can execute by selecting the icon. The user can freely change the size, color, and the commands of these icons by writing XML files.

It is necessary to do the operation that corresponds to

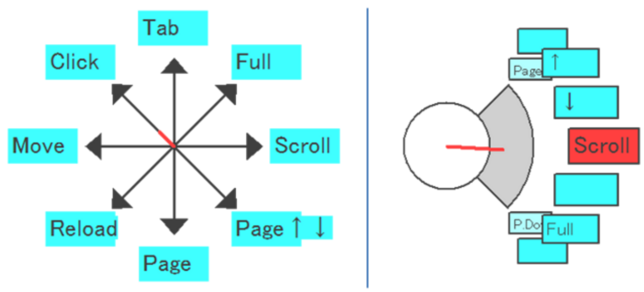

Figure 7. left: Overview of direction-rotate menu (initial state), right: Expansion of submenus based on the right direction

mouse's left-click through the menu because One-finger Interaction does not use any buttons. The click can have to be operated in a place without using the position of the pointer because the pointer is not steady by the influence of hand tremor. To cope with the problems, we made a click icon and the user can execute operation of mouse's left-click through this icon. This icon consists of "click bar" and "cross-hair". The system executed the click action at the center of cross-hair. Cross-hair can also be moved like pushing it by the pointer, and the user can set it at the point where he/she wants to click. He can execute the click operation when the user crosses the click bars twice in a short time.

B. Direction-rotate interface

The system, which differs from the crossing-type interface, detects the direction of hand or rotation in this interface. Then, the system judge actions (expand, scroll and select) and execute them to the menus. Compared with crossing-type interface, this interface needs not move the hand greatly. The operations such as scrolling the window or the increasing / decreasing the value are also easy by using movement of fingertip which draws a circle. User's work is also not so obstructed because the size of the interface is compacter than crossing-type interface.

1) Flow of interaction: In this method, "direction of the movement" executes the expansion of the submenus and decision of the menu, and "rotate operation" does selection of the submenu and scrolling. The user can select a menu by the following steps. 1) the user stays his finger for a while in the vicinity of the center of the chest. 2) he/she selects the direction of the menu which wants to select by using "direction of the movement". 3) he/she makes the menu scroll using "rotate operation". 4) he/she moves his finger to the neutral position and can select the menu.

The system displays the menu like the left of Figure 7 after operation 1). The menu in the left of Figure 7 disappears by operating 2). Then it displays submenus (such as a right of Figure 7) that correspond in that direction. It expands submenus to the vicinity of the label where the user moved his finger. The system decides which directions of submenus expand by the direction of the movement of fingertip (e.g. the inclination, direction and so on).

In addition, when the user wants to do scrolling or change values, he should do following operations after 3). 4) after he/she moves his finger from the beginning point, he/she does "rotate operation". 5) he/she moves his finger to the beginning position, then finishes the scroll actions. The user can increase the value when he/she moves his finger clockwise. The user can also decrease when he/she moves anticlockwise.

The user needs not to move his hand widely in these operation flows. It is also easy for the user to execute menu operations continuously because the end point is near the beginning point. The user can also increase or decrease values by the gesture of drawing a circle.

2) Detail of the interface: We designed the direction-rotate interface as shown in Figure 7 for selecting the menus by movement of fingertip. In this interface, The system displays a part of the corresponding menu of each direction in the initial state. As for the initial state, the system displays the direction and the menu corresponding to it by the arrow and the label as shown in the left of Figure 7.

Each menu of direction has submenus. When the user moves his finger and progresses to the menu, the system expands the submenus based on them, which direction the user moves his finger, as shown in the right of Figure 7. The system expands not only the submenus in the selected direction but also the submenus in the surrounding directions at the same time.

IV. PERFORMANCE EVALUATION

We did a menu selection experiment to consider the problems and the improvement points about the two menu interfaces. In this experiment, we examined the speed and accuracy. We also examined how the user selected the menu by observing the appearance of the experiment. We consider present problems in these interfaces from the measurement data and the appearance of the experiment.

A. Methods

In this experiment, the users select a menu of the same number which displayed on the screen randomly. We measured a time and error rate of one operation, which the users select a menu item. We prepared 16 and 32 menus. Each menu had a number from 0 to 15 (or 0 to 31), and all menus were different number. The users select all menus once in one trial. They tried two times in each case. They were able to practice being accustomed to the each operation before the trials.

We set the menus up 4x4 and 8x4 in the crossing-type menu interface. We united all sizes of the menu by 100x50 pixels. We set the intervals of the icons to all 50 pixel. In direction-and-rotate menu interface, we set the eight directions and each direction had 2 submenus in case of

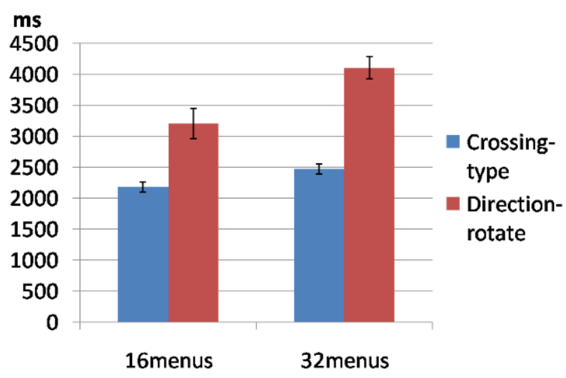

Figure 8. Average of selection time

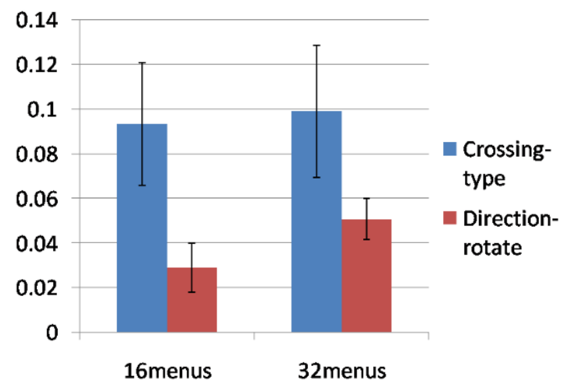

Figure 9. Average of selection error

16 menus and 4 submenus in case of 32 menus. We also set the thresholds for operations below. When user's finger moves 100 pixels from the center of camera, the system extends submenus. The user can select a submenu when the distance between the finger and the center of the camera becomes below 50 pixel. Rotating submenus is done when the user rotate his/her finger more then $pi/4$.

We defined the selection time and error rate as follows. The selection time was not the time till selecting a menu of correct number, but till selecting a menu. Therefore, we counted the selection of a wrong menu and it of the cancellation menu as a menu selection of one time. The equation of the error rate was (wrong selection counts in one trial) / (all selection counts in one trial). Notice that we did not count to either when the user selected a cancel menu in case of calculating the error rate.

We experimented in the large screen which was 6.0 x 1.8 meter (resolution was 3840 x 1024). The users operated from 3 meters far from the display. We put the camera in front of the user. We united the resolutions of the camera in 640x480 to experiment as much as possible on the same condition. We did not change the color of the menu which was the target, but displayed the number as a text on the screen, and the users searched it. The reason of using this method is that the users can find the target menus more easily than direction-rotate menu, when they use crossing-type menu interface. The users were 6 persons who were students in our laboratory and right-handed.

B. Results

The averages of the selection time and error rate of all users were Figure 8 and Figure 9. Crossing-type menu interface was 1 second faster than direction-rotate menu, and the difference between the users was also smaller. On the other hand, the error rate of the direction-rotate menu was less than the other, and the difference of it between users was also smaller. Especially, the difference of individual of the error rate was large in the crossing-type menu. In 32 menus, one user's error rate was 0, the other was more than 0.25.

C. Consideration

1) Crossing-type interface: In this menu, the users searched a target menu with moving the pointer. Then, they did crossing operation immediately after searching. As a result, it seems that the selection time shortened. However, some problems occurred because of this operation. First problem is that the users crossed other target twice by suddenly returning the pointer. We saw this tendency well when they found the target where they moved the pointer before. Second problem is that a user who vastly moved the pointer had many errors while searching a target. Especially, the user whom the error rate was more than 0.25 was often seen the tendency. They selected other menus many times in selecting the target once. When we assumed in a real environment, the malfunctions with such a false selection might hinder the work of the user and give him the stress.

2) Direction-rotate interface: In this menu interface, opposite to crossing-type menu, the users moved their hand after checking the position of a menu. Therefore, the false selection of another menus was not seen, when they were searching for the target . They also scrolled submenus with confirming a menu whether it was the target. However, most of the false selection were caused by scrolling submenus. They moved their finger too much and did an extra scrolling. In this case, there were two patterns. First pattern is that they who did not notice had selected a different menu. Second pattern is that they did scroll in the opposite direction when they noticed. The other case of false selections is that they moved them finger by the inertia when draw a circle using them and went back to the center. They did an extra scroll and caused a false selection. We think that they took many time to select a menu because of these factors. There were some opinions such as wanting to scroll a lot of menus by the scroll once.

V. Related work

Lenman[11] designed the interface that operates the home appliance by using the gesture of the shape of the hand, Marking Menu[10], and FlowMenu[8]. These menus are designed for pen based interface, and the menu selection is done using direction of the trajectory of pen movement. Moreover, continuous direction inputs can execute a lot of commands easily. In the research of Lenman, Marking Menu is operated by the direction of pointing, and a lot of commands can be executed by a little gesture. However, the advantage of continuous inputs using trajectory is lost, because detection of the direction of finger need a many time and it can be always done before executing commands.

A lot of menu operation techniques that use trajectory is seen in the pen based interface. Bailly[4] designed the interface to execute the menu by drawing trajectories, which are shown in the display with the pen. Apitz[2] made the technique for selecting two or more menus at one time by using the crossing with pen stroke. Zhao[16] selected the menus by doing an easy gesture continuously two or many times. These techniques that use trajectory is also effective in One-finger Interaction. However, it is difficult to draw exact trajectory and correspond to operation that raises a pen in movement of fingertip. Therefore, when we use these techniques with hand gestures, it is necessary to improve according to the properties of them.

VI. Conclusions

We proposed One-finger Interaction as a technique of interacting the object from a distant place by movement of fingertip. In this technique, we were able to do various interactions by combining movement of fingertip with menu interface. We also developed two type menu interfaces as a prototype of One-finger Interaction. One is crossing-type interface that the user can do operations when crossing with a menu twice in a short time. The other is direction-rotate interface, which uses the direction and draws a circle by movement of fingertip.

Future works include the improvement of operativeness and the selection technique of a menu. Direction-rotate interface needs more improvements because the user took much time for selecting a menu. The improvements of customizing the interfaces which the user can make are also need. In addition, we will develop other interfaces and discuss which method is suitable for One-finger Interaction.

References

[1] J. Accot and S. Zhai. More than dotting the i's - foundation for crossing-based interfaces. In *CHI' 02*, pages 73–80, 2002.

[2] G. Apitz and F. Guimbretiere. Crossy: A crossing-based drawing application. In *UIST '04*, pages 3–12, 2004.

[3] G. Bailly, E. Lecolinet, and L. Nigay. Wave menus: Improving the novice mode of hierarchical marking menus. In *INTERACT 2007*, pages 475–488, 2007.

[4] G. Bailly, E. Lecolinet, and L. Nigay. Flower menus: a new type of marking menu with large menu breadth, within groups and efficient expert mode memorization. In *AVI '08*, pages 15–22, 2008.

[5] R. A. Bolt. "put-that-there": Voice and gesture at the graphics interface. In *SIGGRAPH '80*, pages 262–270, 1980.

[6] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. An empirical comparsion of pie vs. linear menus. In *CHI '88*, pages 95–100, 1988.

[7] P. Dhawale, M. Masoodian, and B. Rogers. Bare-hand 3d gesture input to interactive systems. In *CHINZ '06*, pages 25–32, 2006.

[8] F. Guimbretiére and T. Winograd. Flowmenu: combining command, text, and data entry. In *UIST '00*, pages 213–216, 2000.

[9] T. Hisamatsu, B. Shizuki, S. Takahashi, and J. Tanaka. A novel click-free interaction technique for large-screen interfaces. In *APCHI2006*, pages (CD–ROM), 2006.

[10] G. Kurtenbach and W. Buxton. The limits of expert performance using hierarchic marking menus. In *CHI '93*, pages 482–487, 1993.

[11] S. Lenman, L. Bretzner, and B. Thuresson. Using marking menus to develop command sets for computer vision based hand gesture interfaces. In *NordiCHI '02*, pages 239–242, 2002.

[12] S. Malik, A. Ranjan, and R. Balakrishnan. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In *UIST'05*, pages 43–52, 2005.

[13] K. Miyaoku, S. Higashino, and Y. Tonomura. C-blink: A hue-difference-based light signal marker for large screen interaction via any mobile terminal. In *UIST '04*, pages 147–156, 2004.

[14] T. Nakamura, S. Takahashi, and J. Tanaka. Double-crossing: a new interaction technique for hand gesture interfaces. In *APCHI2008*, pages 831–838, 2008.

[15] D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *UIST'05*, pages 33–42, 2005.

[16] S. Zhao, M. Agrawala, and K. Hinckley. Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. In *CHI '06*, pages 1077–1086, 2006.