A Remote Pointing Technique Using Pull-out

Takuto Yoshikawa, Yuusaku Mita, Takuro Kuribara,
Buntarou Shizuki, and Jiro Tanaka

University of Tsukuba, Japan
{yoshikawa,mita,kuribara,shizuki,jiro}@iplab.cs.tsukuba.ac.jp

Abstract. Reaching objects displayed on the opposite side of a large multi-touch tabletop with hands is difficult. This forces users to move around the tabletop. We present a remote pointing technique we call HandyPointing. This technique uses pull-out, a bimanual multi-touch gesture. The gesture allows users to both translate the cursor position and change control-display (C-D) ratio dynamically. We conducted one experiment to measure the quantitative performance of our technique, and another to study how users selectively use the technique and touch input (i.e., tap and drag).

Keywords: bimanual interaction, multi-touch, gesture, tabletop.

1 Introduction

A large multi-touch tabletop is used for a collocated collaborative work that involved multiple users. These users surround the tabletop and touch the tabletop from their respective positions. However, reaching a distant object displayed on the opposite side of the tabletop is difficult due to the largeness of the touchscreen. Toney et al. reported that more than 90% of users' touch interactions are performed within 34 cm from their respective positions [11]. Users are forced to lean forward from the tabletop or move around the tabletop to reach the object.

To solve this problem, indirect-pointing devices, such as a mouse, are complementary used for touch input. Using these devices enable users to reach the distant objects. However, they require physical space in which to place them around a tabletop for each user. Furthermore, preparing the devices in advance is troublesome because tabletops are used by an unspecified number of users simultaneously.

Therefore, we present a remote pointing technique we call *HandyPointing*. This technique uses pull-out, a bimanual multi-touch gesture [12], to determine a cursor position. A pull-out gesture requires no additional devices because the technique uses touch input only. Furthermore, a pull-out gesture allows users not only to translate the cursor position but also to change the control-display (C-D) ratio dynamically, similar to [3]. This means that they can selectively perform rough pointing with a large C-D ratio and precise pointing with a small C-D ratio. Therefore, users can precisely point at a distant position quickly by combining these rough and precise pointing techniques.

M. Kurosu (Ed.): Human-Computer Interaction, Part IV, HCII 2013, LNCS 8007, pp. 416–426, 2013.

2 Related Work

There are related works about remote pointing techniques on tabletops. We classify these techniques into direct-pointing, and indirect-pointing. Here the former uses the position at which users point as a cursor position, and the latter translates a cursor position according to the movements of devices.

Direct-pointing. Parker et al. used the shadow of the tip of stylus to point at a distant position [9]. In the work of Banerjee et al. [3], users could point a finger at objects on tabletops and dynamically change C-D ratio by one hand while performing a pointing with the other hand. The above techniques required additional devices that obtain the position of users' hands to realize direct-pointing. In HandyPointing, we adopt indirect-pointing in order not to use such devices. Our technique requires only the touch coordinates.

Indirect-pointing. Bartindale et al. [4] and Matejka et al. [8] developed an onscreen mouse for multi-touch tabletops that allows users to point, similar to a conventional physical mouse. These research realized an indirect-pointing technique. However, they required to recognize the shape of hands, while our technique can be applied to tabletops that detect more than three touch points. I-Grabber [1] is an onscreen widget manipulated by multi-touch interactions. Users can select and translate a distant object with the widget. Although our technique also uses multi-touch interactions, the technique allows users to determine a cursor position by a single multi-touch gesture.

Bimanual Interaction. Guiard modeled the asymmetric bimanual behaviors of humans as Kinectic Chain Model [6]. Tokoro et al. presented a pointing technique that utilized two acceleration sensors, and postures of both hands determined a cursor position [10]. Furthermore, Malik et al. developed a bimanual pointing technique by using image processing [7]. In contrast with these techniques, our technique is realized by using touch based gestures. In addition, Bailly et al. utilized the number of touches and their strokes of both hands to execute commands in a distant menu bar [2]. While their technique enables command invocations that utilized discrete input, our technique enables indirect-pointing that utilized continuous input by the stroke of a pull-out gesture.

3 Interaction Techniques

Our pointing technique utilizes both hands to move a cursor. In this section, we describe HandyPointing, a remote pointing technique, and its additional remote manipulation technique.

3.1 Pointing Technique

Figure 1 shows the procedure of HandyPointing. First, users put two fingers of their non-dominant hand (base-fingers) on a tabletop, as shown in Figure 1a.

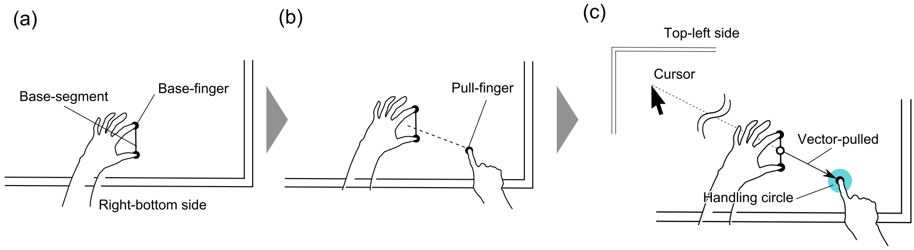

Fig. 1. Pointing procedure using HandyPointing

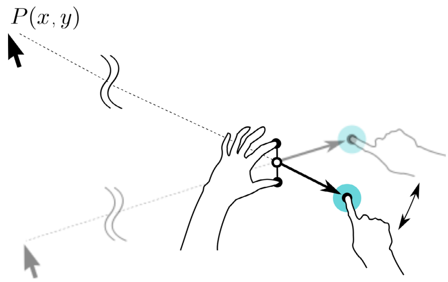

Fig. 2. Cursor translation according to vector-pulled

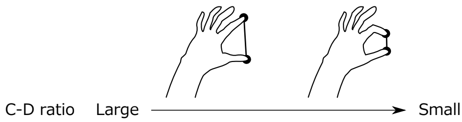

Fig. 3. Dynamic C-D ratio according to length of base-segment

When users drag their finger of their dominant hand (pull-finger) to cross the segment between base-fingers (base-segment) as shown in Figure 1b, a cursor is displayed on an extension of the opposite direction of the vector from the center of the base-segment to pull-finger (vector-pulled) as shown in Figure 1c. Users can quit pointing by taking base-fingers off from the tabletop.

If users arrange the vector-pulled, the cursor position changes in accordance with the vector as shown in Figure 2. C-D ratio also changes depending on the length of the base-segment. This means that users can simultaneously move the cursor by using the dominant hand while controlling C-D ratio dynamically by using the other hand as shown in Figure 3.

An advantage of this bimanual manipulation is that users can selectively perform rough pointing with a large C-D ratio or precise pointing with a small C-D ratio. For example, users can move a cursor precisely with a short base-segment, while they can quickly move the cursor at a distance with a long base-segment as shown in Figure 4.

3.2 Determination of a Cursor Position

This section describes the procedure to determine a cursor position. Suppose that $P_i(x, y)$ is the i-th cursor position after i frames have passed since users placed base-fingers on the tabletop. Then P_i is given by the following expressions:

$$P_i = G_0 - \sum_{j}^{i} k_j \Delta V_j,$$

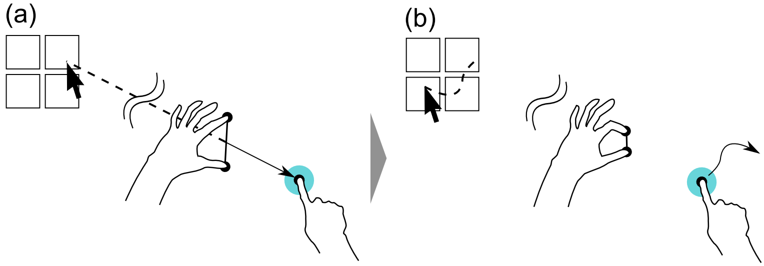

Fig. 4. Usage of dynamic C-D ratio. Users (a) point at far position quickly with large C-D ratio, and then (b) precisely point at object with small C-D ratio.

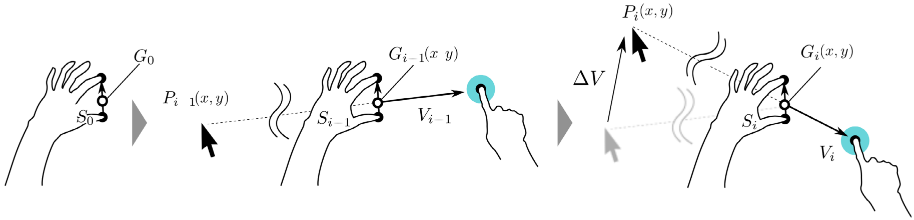

Fig. 5. Determination of cursor position

$$\Delta V_i = V_i - V_{i-1},$$
$$k_i = \alpha \times \frac{|S_0|}{|S_i|}.$$

As shown in Figure 5, S_0 is the base-segment when base-fingers were placed on the tabletop, and G_0 is the gravity point of S_0. Furthermore, S_i and V_i are the i-th base-segment and vector-pulled, respectively. Then, G_i is the gravity point of S_i. α is a constant. That is, our technique determines i-th C-D ratio by k_i, and then the cursor position P_i is moved by k_i and ΔV_i, which is the difference of V_i, frame by frame.

3.3 Remote Manipulation

We implemented a function to manipulate a distant object. As shown in Figure 1c, a circle is shown around the pull-finger (handling-circle) when users begin pull-out. Users can select the object under the cursor by tapping the handling-circle after they have translated a cursor. Moreover, they can translate the cursor again by dragging the handling-circle. They can unselect the object by tapping the circle again.

The selected object moves according to the cursor when users drag the handling-circle. This means that they can select a distant object, and drag it to another distant location. Note that they can dynamically change C-D ratio while they are using not only a normal pointing but also a remote manipulation. Therefore, they can select the object precisely and move it quickly.

4 Evaluation

We conducted two experiments. One was to measure the quantitative performance of HandyPointing, and the other was to study how users selectively use HandyPointing and touch input (i.e., tap and drag). We implemented a prototype of HandyPointing using a 1470mm x 800mm 60-inch tabletop. Ten volunteers participated in this experiment. They were undergraduates or graduate students with ages ranging in age from 21 to 24 years. They were all right-handed and familiar with a mouse.

4.1 Experiment 1

To compare the performance of HandyPointing and another in-direct pointing technique, we used a pointing task similar to those in [3, 5]. We used a mouse as it is the most common in-direct pointing device. We divided participants into two groups. One group first performed the task by HandyPointing; the other group first performed the task by mouse. We asked participants to sit in a chair that placed in the middle of the short side of the screen during the task. We explained how to use each technique before the task, and asked them to practice the techniques sufficiently. To measure the base-line of our technique, we assigned the C-D ratio of the mouse to achieve the best performance such that the mouse never needed to be clutched.

Task. We asked participants to point at and select a target object. When participants selected a start point, a target object was displayed. We radially arranged the positions of target objects as shown in Figure 6. Once participants selected the target object, the object was removed.

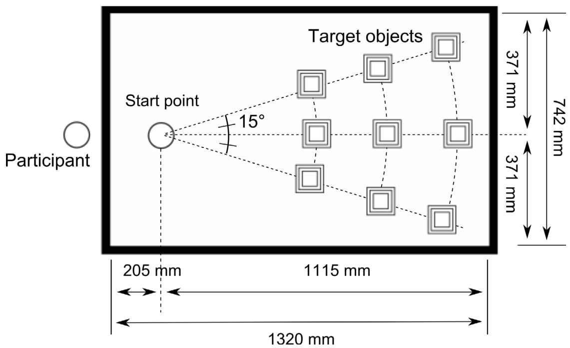

Fig. 6. Positions of start point and target objects

In this experiment, independent variables were: target distance (500, 700, and 900 pixels, i.e. approximately 515mm, 715mm, and 915mm, respectively), target angle (-15, 0, and 15 degree), target size (40, 80, and 120 pixels, i.e. approximately 41mm, 82mm, and 123mm, respectively), and technique (HandyPointing and mouse). Each participant performed 2 trials for each combination of factors, thus they performed 108 (3x3x3x2x2) trials in total. All trials were presented in a randomized order.

Result. We measured the time to complete a trial (trial-time) and the number of errors. In HandyPointing condition, trial-time begins when participants started HandyPointing on a start point and ends when they selected a target object by tapping a handling-circle. In mouse conditions, trial-time begins when participants clicked a start point and ends when they selected a target object. When participants failed to select a target object, we treated it as an error.

Figure 7 to 10 show the result of the experiment. Figure 7 and 8 show the average trial-time and the number of errors in HandyPointing condition. Figure 9 and 10 show those in mouse condition. In these figures, the blue graph illustrates the result for each target distance (500, 700, and 900 pixels), green one illustrates the result for each target angle (-15, 0, and 15 degree) and red one illustrates the result for each target size (40, 80, and 120 pixels). The average trial-times were 3812ms in HandyPointing condition and was 1266ms in mouse condition. The average numbers of errors were 1.72 in HandyPointing condition and was 0.053 in mouse condition. Figure 11 and 12 show average trial-time of HandyPointing trials and mouse trials for each participant.

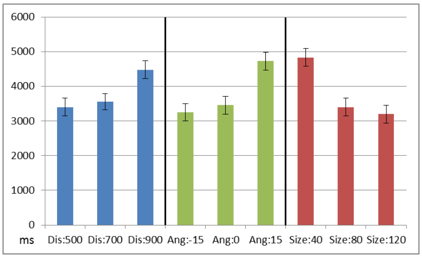

Fig. 7. Mean and variance of trial-time for each target distance (blue), target angle (green), and target size (red) in HandyPointing condition

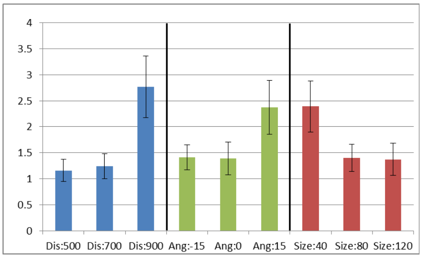

Fig. 8. Mean and variance of number of errors for each target distance (blue), target angle (green), and target size (red) in HandyPointing condition

Discussion. The trial-time of HandyPointing was larger than that of a mouse, as shown in Figure 7 and 9. However, trial-time gradient and the number of errors were similar for each condition. That is, trial-time was in accordance with the increase in distance and the decrease in angle. This result indicates that HandyPointing seems to follow Fitts' law.

The number of errors of HandyPointing increased when target distance was 900 pixels, as shown in Figure 7. This means participants failed to tap a handling-circle when they selected the most distant targets. This is because they could not tap the circle while focusing on a distant target object. On the other hand, participants

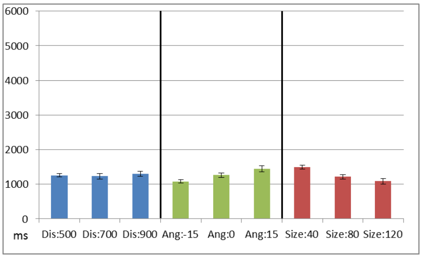

Fig. 9. Mean and variance of trial-time for each target distance (blue), target angle (green), and target size (red) in mouse condition

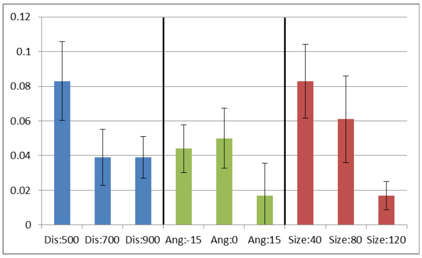

Fig. 10. Mean and variance of number of errors for each target distance (blue), target angle (green), and target size (red) in mouse condition. Note that maximum of Y-axis is different from Figure 8.

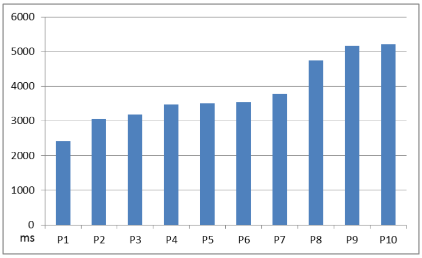

Fig. 11. Mean of trial-time for each participant in HandyPointing condition

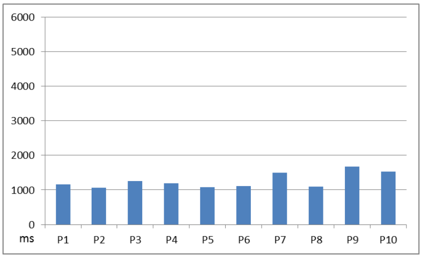

Fig. 12. Mean of trial-time for each participant in mouse condition

successfully tapped the circle more when they selected a near target object. This is because they could focus on the circle and the object simultaneously.

Furthermore, the trial-time and the number of errors also increased in HandyPointing condition as the angle changed from -15 to 15 degree, as shown in Figure 7 and 8. This increase was more significant than for the mouse. A cursor is displayed on the non-dominant hand side first, when they begin to perform HandyPointing. For example, the cursor is displayed on the left side for right-handed participants. Thus, they can easily select the non-dominant hand side objects. However, selecting the dominant hand side objects forces participants to move arms outside of the natural range of motion of arms. Hence, they had to move their right arms to the left side and left arms to the right side. This makes it difficult to select the dominant hand side objects and shows that users can easily select non-dominant hand side objects by using HandyPointing.

Next, we discuss average trial-time for each participant. These trial-times are shown in Figure 11 and 12. In HandyPointing condition, the largest average trial-time (5211ms) was 2.15 times larger than the smallest average trial-time (2420ms). In mouse condition, the largest average trial-time (1524ms) was 1.57 times larger than the smallest average trial-time (1165ms). Moreover, the smallest average trial-time of HandyPointing (2420ms) was 2.08 times larger than that of a mouse (1165ms). Trial-time greatly differed among participants. In this experiment, we asked participants to practice HandyPointing and a mouse, until they had become familiar with these techniques. However, some participants proceeded the experiment as soon as they had learned only how to use HandyPointing but without becoming familiar with the technique. This meant the degree of proficiency in HandyPointing for each participant was significantly different, resulting in the large differences in trial-time among participants. On the other hand, they were already familiar with the mouse. Thus, the differences in trial-time in mouse condition was small. This result means that the degree of proficiency significantly affects the performance of HandyPointing among participants. However, with a little training, pointing with HandyPointing takes at least only twice time as long as pointing with a mouse. In addition, more training may lead to better performance.

4.2 Experiment 2

To study how users selectively use HandyPointing and ordinary touch input, we used select & docking task, similar to those in [3, 5]. This experiment was sequentially conducted after Experiment 1. This means that the same participants joined this experiment, and the same apparatuses were used.

Task. The same as Experiment 1, we asked participants to point at an object, select it, and move it into a dock. In this task, objects appeared on the top, middle, or bottom of the screen as illustrated in Figure 13, and docks appeared near or far from participants. Participants were allowed to use touch input and HandyPointing simultaneously during the experiment.

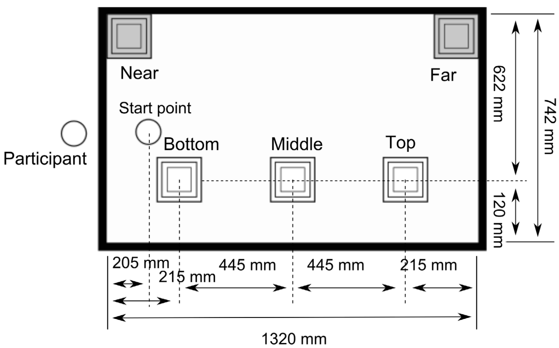

Fig. 13. Positions of target objects and docks

In this experiment, independent variables were: target size (40, 80, and 120 pixels, approximately 40mm, 80mm, and 120mm respectively), target position (top, middle, and bottom), and dock position (near and far). The dock was the same size as the target in each task. Each participants performed 3 trials for

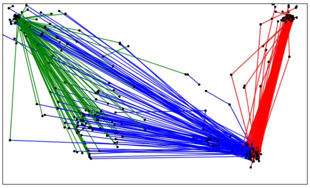

Fig. 14. Movements of top objects

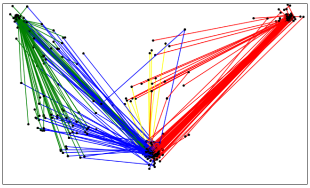

Fig. 15. Movements of middle objects

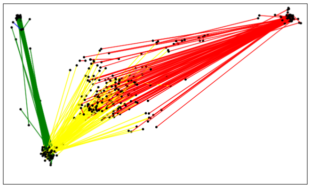

Fig. 16. Movements of bottom objects

each combination of factors, thus they performed 54 (3x3x2x3) trials in total. All trials were presented in a randomized order.

Result. Figure 14 to 16 shows the movements of objects, where blue and green segments represent the movements of objects toward a near dock that are manipulated by HandyPointing and touch input, respectively. Red and yellow ones represent those of objects toward a near dock that are manipulated by HandyPointing and touch input, respectively. The segments connect the positions where objects began and the stopped moving.

Figure 14 shows the movements of top objects. All top objects toward a near dock were manipulated by HandyPointing only, and no manipulations were performed by touch input. In contrast, those toward a near dock were manipulated by the combination of HandyPointing and touch input. This means participants used HandyPointing to move a distant object to the near position and touch input to move it into the near dock. Figure 15 shows the movements of middle objects. 89% of the objects toward a near dock were manipulated by HandyPointing only, and the remaining 11% were done by the combination of HandyPointing and touch input. This is because some participants first moved the objects to the near position by touch input and then into the far dock by HandyPointing. 39% of the middle objects toward a near dock were manipulated by touch input only, and 3% were performed by HandyPointing only. The remaining 58% were performed by the combination of HandyPointing and touch input. Figure 16 shows the movements of bottom objects. All objects toward a near dock were manipulated by the combination of HandyPointing and touch input, and 99% of the objects toward a near dock were manipulated by touch input.

Discussion. Participants' behavior changed depending on the distance to a target object. Participants used HandyPointing to drag top and middle objects into a near dock. They used the combination of HandyPointing and touch input to drag them into a near dock (Figure 14 and 15). In contrast, they used the combination of HandyPointing and touch input to drag bottom objects into a near dock and used touch input to drag them into a near dock (Figure 16). This difference is owing to whether they can reach the objects with hands. That is, participants used HandyPointing for a distant object and touch input for a near object. The result shows that they selectively used one of these techniques in accordance with the distance to a target object.

When they combined the techniques, participants first put an object into the near position by using one technique, and then they dragged it into a dock by using the other technique. In our observation, they seemed to use the first technique to move the object into the position where they could select easily with the second technique. This indicates that they preferred to use HandyPointing for a distant object and touch input for a reachable object.

5 Conclusions and Future Work

We designed and implemented a remote pointing technique, HandyPointing. The technique allows users to point at a distant position that their hands cannot reach. Furthermore, users can simultaneously change C-D ratio dynamically by using their non-dominant hand while determining a cursor position by using the dominant hand. Therefore, they can selectively use rough pointing or precise pointing. We conducted two experiments. Their results showed that users can selectively use HandyPointing and ordinary touch input, and pointing with HandyPointing at least takes only about twice as long as pointing with a mouse with a little training.

We will continue to measure the performance of HandyPointing because the C-D ratio of a mouse in the experiments was the ratio that maximizes the performance of the mouse to measure the base-line of our technique. Therefore, we will conduct a similar experiment to measure the maximum performance of our technique by seeking the ideal C-D ration in the future and compare it with the results in this paper.

References

1. Abednego, M., Lee, J.H., Moon, W., Park, J.H.: I-Grabber: expanding physical reach in a large-display tabletop environment through the use of a virtual grabber. In: Proc. of ITS 2009, pp. 61–64 (2009)
2. Bailly, G., Lecolinet, E., Guiard, Y.: Finger-count & radial-stroke shortcuts: 2 techniques for augmenting linear menus on multi-touch surfaces. In: Proc. of CHI 2010, pp. 591–594 (2010)
3. Banerjee, A., Burstyn, J., Girouard, A., Vertegaal, R.: Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops. In: Proc. of ITS 2011, pp. 11–20 (2011)
4. Bartindale, T., Harrison, C., Olivier, P., Hudson, S.E.: SurfaceMouse: supplementing multi-touch interaction with a virtual mouse. In: Proc. of TEI 2011, pp. 293–296 (2011)
5. Forlines, C., Wigdor, D., Shen, C., Balakrishnan, R.: Direct-touch vs. mouse input for tabletop displays. In: Proc. of CHI 2007, pp. 647–656 (2007)
6. Guiard, Y.: Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. Journal of Motor Behavior 19, 486–517 (1987)

7. Malik, S., Ranjan, A., Balakrishnan, R.: Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In: Proc. of UIST 2005, pp. 43–52 (2005)
8. Matejka, J., Grossman, T., Lo, J., Fitzmaurice, G.: The design and evaluation of multi-finger mouse emulation techniques. In: Proc. of CHI 2009, pp. 1073–1082 (2009)
9. Parker, J.K., Mandryk, R.L., Inkpen, K.M.: TractorBeam: seamless integration of local and remote pointing for tabletop displays. In: Proc. of GI 2005, pp. 33–40 (2005)
10. Tokoro, Y., Terada, T., Tsukamoto, M.: A pointing method using two accelerometers for wearable computing. In: Proc. of SAC 2009, 136–141 (2009)
11. Toney, A., Thomas, B.H.: Applying reach in direct manipulation user interfaces. In: Proc. of OzCHI 2006, pp. 393–396 (2006)
12. Yoshikawa, T., Shizuki, B., Tanaka, J.: HandyWidgets: local widgets pulled-out from hands. In: Proc. of ITS 2012, pp. 197–200 (2012)