LETTER

Calibrating Coordinates of a Tabletop Display with a Reflex in Eye-Hand Coordination

Makio ISHIHARA† and Yukio ISHIHARA††, Members

SUMMARY This manuscript introduces a pointing interface for a tabletop display with a reflex in eye-hand coordination. The reflex is a natural response to inconsistency between kinetic information of a mouse and visual feedback of the mouse cursor. The reflex yields information on which side the user sees the screen from, so that the screen coordinates are aligned with the user's position.

key words: reflex, eye-hand coordination, screen calibration, pointing interface, tabletop displays

1. Introduction

Human computer interaction or HCI [1] becomes more important in helping people interact with computers more intuitively and implicitly. In the field of HCI, tabletop displays have been studied recently and become one of interesting challenges for application and interaction style. A tabletop display is a horizontal display which users sit down around and use together. A tabletop display is most likely equipped with a touch screen so that the user can click on and drag contents on the screen in an interactive way. The contents could be water surface, a pool table, a satellite map etc.

Nowadays tabletop displays are often used for collaborative work such as brainstorming and meetings where multiple people work together around a single tabletop display. This trend leads to a strong demand for a large screen space.

A pointing interface is a system that locates the spot on a computer screen, which the user wants to focus on. One of key issues around a tabletop display is how to make the pointing interface if the tabletop display has a large screen so that the user cannot reach to the contents by his/her own hands. Traditionally the pointing interface is assembled with sophisticated sensors such as 6 DOF magnetic ones in this case. CAVE systems take this way for pointing. In research, the trend is that the pointing interface functions by capturing the quadrilateral of the screen with a handheld camera and locating the center of the camera as a graphical cursor. Nintendo Wii takes this approach for pointing. Infrared LEDs are attached on a TV screen, which represent the quadrilateral of the screen. The Wii remote captures the LEDs and locates the graphical cursor in real time. This way works well as far as the camera captures the quadrilateral of the screen.

One of the solutions to this is to use Gray-code. Gray-code is a way to encode values 0 up to $2^n - 1$ into $n$ bits binary sequences with the property that only one bit changes between any two consecutive sequences. A gray-code binary pattern is a visual representation of the binary sequences at a specific bit. In our previous work [2], we used black and white gray-code binary patterns. They appear on the screen in order then the temporal pattern captured by a handheld camera at a position of the screen uniquely represents its own position. Thus it is not necessary that the handheld camera captures the entire screen but one position. In an environment of projection, Lee, J. et al. [3] proposed a method for making the visible gray-code binary patterns imperceptible by transmitting the gray-code binary patterns through frequency-shift keying or FSK modulation. They [4] later made a prototype of a projection system that is capable of emitting both infrared and visible rays simultaneously. This way is applicable to tabletop displays if their screens are projection.

Our pointing interface uses a common mouse so that it relies on neither the projection of infrared rays nor the quadrilateral of the screen. One of the major problems with the use of a mouse for pointing on a tabletop display is that the coordinates of the screen (hereafter the screen coordinates) are not always aligned with the user’s position. Widgeor, D. et al. [5] showed that the correlation between display positions and mouse control orientations would have a significant impact on performance. Our pointing interface uses a reflex in eye-hand coordination in order to align the screen coordinates automatically with the user’s position. The reflex in eye-hand coordination is a natural response to inconsistency between kinetic information of a mouse and visual feedback of the mouse cursor. For example, when you see a computer screen from the left side and move the mouse upwards, then you will see the mouse cursor moving leftwards. This conflict between the mouse moving upwards and the mouse cursor moving leftwards makes the mouse cursor move in circle counterclockwise. We call it the reflex in eye-hand coordination.

This manuscript shows that the radius of circle or orbit in which the mouse cursor moves depends on the angle at the user’s position from the right front of the screen. So the reflex can be exploited to align the screen coordinates with the user’s position. Once the screen coordinates are aligned, the user can manipulate the mouse cursor in a usual manner. This manuscript also shows that the reflex is viable to
align the screen coordinates within a practical time. In this manuscript, we discuss the reflex in a laptop environment without loss of generality of the result.

Section 2 describes properties of the reflex in eye-hand coordination. Section 3 conducts an experiment on performance of calibration of the screen coordinates and Sect. 4 gives the concluding remarks.

2. A Reflex in Eye-Hand Coordination

This section shows the behavior of the reflex in eye-hand coordination and its properties.

2.1 Experiment 1

The objective of this experiment is to show a significant correlation between the angle at the user’s position from the right front of the screen, and the radius of circle in which the mouse cursor moves by the reflex.

Figure 1 shows the setup of the experiment on properties of the reflex. The rectangle represents a laptop screen of Dell XPS M1210, which is 261 mm wide and 164 mm high (1280x800 pixels in resolution). The laptop has a processor of 1.73 GHz. It is also equipped with a graphics card of nVidia GeForce Go7400. In the figure, a user sees the laptop screen right in front.

To represent the user’s position around the laptop screen, only the screen coordinates are set in angle as shown in Fig. 1. In the experiment, the screen coordinates are set in each angle of 30, 60, 90, 120 and 150 degrees both clockwise (represented by “cl”) and counterclockwise (represented by “cc”). While the user moves the mouse by the reflex, the radius (mm) of circle in which the mouse cursor moves is measured about 30 Hz in real time. Since the user hit a specified key to tell the system that he/she was moving the mouse by the reflex at a certain radius, the minimum radius having held the same over 1 second is sampled. There were 4 subjects between the ages of 21 and 23. All were right-handed, had an experience working with a mouse. Each subject had 10 trials of each angle, resulting in 100 trials.

4 subjects ×
10 angles of screen coordinates ×
10 trials = 400 total trials

To avoid learning effects, the sequence of trials is in a random order but it is the same between subjects.

2.2 Properties

Figure 2 shows the result of the experiment. The horizontal axis shows the angle of the screen coordinates (deg.). Each angle represents both the clockwise and counterclockwise one. The vertical one shows the average of the observed radii (mm) at each angle for each subject. The line labeled “all” shows the average at each angle across all the subjects.

The details are shown in Table 1. As seen in the figure, it would be possible to separate each angle from the others by the observed radius. Analysis of variance shows that there is a significant difference in the observed radius between angles of the screen coordinates \(F(4,395) = 150.285\) at \(p = 0.1\). A further analysis of t-test between any consecutive angles of the screen coordinates shows that there is a significant difference in the observed radius between them \([T(395) = 13.01, 3.98, 2.41, 2.73\) at \(p = 0.025\).

In addition, it was observed that the subject moved the mouse cursor in circle clockwise (or counterclockwise) when the screen coordinates were set in angle clockwise (or counterclockwise).

3. Performance

This section evaluates performance of calibration of the
screen coordinates in terms of accuracy, completion time and usability.

3.1 Experiment 2

The objective of this experiment is to show that the reflex in eye-hand coordination is applicable to align the screen coordinates within a practical time.

The setup of the experiment is the same with the one shown in Fig. 1. Each average listed in Table 1 is used as a threshold of the radius at the corresponding angle of the screen coordinates. If the observed radius for example is less than the threshold of 20.26, the angle of the screen coordinates is expected to be more than 30 degrees.

In the experiment, a subject sits down right in front of the laptop screen. The screen coordinates are initially set in each angle of 30, 60, 90, 120 and 150 degrees both clockwise and counterclockwise (hereafter initial angles of the screen coordinates). The subject is asked to align the screen coordinates. The procedure is following. First, the subject moves the mouse by the reflex. Next, the subject hit a specified key to tell the system that he/she is moving the mouse cursor in circle by the reflex at a certain radius. Since then the minimum radius having held the same over 1 second is sampled. Finally, the system tries to align the screen coordinates by comparing the sampled radius with each threshold and taking account of the mouse cursor moving in circle clockwise or counterclockwise. The subject needs to do this procedure (hereafter update) again until he/she aligns the screen coordinates completely with his/her position. There were 4 subjects between the ages of 21 and 23. They were different from those in Experiment 1. All were right-handed, had an experience working with a mouse. Each subject had 5 trials of each angle, resulting in 50 trials.

4 subjects ×
10 angles of screen coordinates ×
5 trials = 200 total trials

To avoid learning effects, the sequence of trials is in a random order but it is the same between subjects.

3.2 Accuracy

Figure 3 shows the result of the experiment. The horizontal axis shows the update of the screen coordinates. The vertical one shows the average of the angles of the screen coordinates at each update for each initial angle. Each initial angle represents both the clockwise and counterclockwise one. At the first update, the system aligned the screen coordinates with the subject’s position within approximately 40 degrees error. Then at the second update, the system did it within approximately 10 degrees error. Ideally, calibration should be completed at once. Figure 4 shows the progress of the calibration at each update for each initial angle of the screen coordinates. Figure 5 shows the progress of the calibration at each update across all the initial angles. As shown in Fig. 5, the screen coordinates are aligned with 73.3% accuracy at the first update, and then 92.7% accuracy at the second update.

3.3 Completion Time

Table 2 shows the average of elapsed time for the calibration at each initial angle of the screen coordinates. The elapsed time here refers to the period from the time when the subject hit the specified key till the time when the system finishes doing just a single update to the screen coordinates. In the table, the row “1st” shows the elapsed time for the first update and the row “Final” shows the total elapsed time to
Table 2 Average of elapsed time for calibration at each initial angle of the screen coordinates. The row “1st” shows the elapsed time for the first update and the row “Final” shows the total elapsed time to complete the calibration.

| Initial angle (deg.) | 30   | 60   | 90   | 120  | 150  |
|---------------------|------|------|------|------|------|
| 1st (ms)            | 1817 | 2020 | 2137 | 1633 | 1864 |
| Final (ms)          | 1908 | 3545 | 5095 | 4783 | 4079 |

Table 3 Average of elapsed time for calibration across all the initial angles of the screen coordinates. The row “1st” shows the elapsed time for the first update and the row “Final” shows the total elapsed time to complete the calibration.

| Initial angle (deg.) | all  |
|---------------------|------|
| 1st (ms)            | 1894 |
| Final (ms)          | 3882 |

complete the calibration. The system aligned the screen coordinates with the subject’s position in approximately 2 seconds at the initial angle of 30 degrees. The system did it in 3.5–5 seconds at the other initial angles. Table 3 shows the average of the elapsed time for the calibration across all the initial angles of the screen coordinates. As for the first update, it took less than approximately 2 seconds. So, it is fair to say that the system aligned the screen coordinates with the user’s position with 73.3% accuracy in 1.89 seconds.

3.4 Usability

Our pointing interface is not required to capture the quadrilateral of the screen, so it can work in a large-screen tabletop-environment. It is also not required to use projection of infrared rays, so it is applicable to various settings of a tabletop display such as LCDs, projection, plasma displays and organic LED displays. Our pointing interface exploits the reflex in eye-hand coordination to align the screen coordinates with the user’s position. It leads to having the interface work implicitly, so it is not likely that the user needs to learn how to use. One of major problems is how the pointing interface separates the user’s response to the mouse cursor in the reflex from his/her usual response to it. In our pointing interface, the user needs to start calibration by hitting a specified key.

Judging by the result of accuracy and completion time, our pointing interface is good at aligning the screen coordinates quickly, but roughly, with the user’s position. Manual configuration, which allows the user to rotate the screen coordinates by wheeling the mouse or pressing the keys, is good at aligning the screen coordinates precisely, but slowly, with the user’s position. These two kinds of methods complement each other in terms of accuracy and completion time.

4. Conclusions

This manuscript introduced a pointing interface for a tabletop display with a reflex in eye-hand coordination. The reflex is a natural response to inconsistency between kinetic information of a mouse and visual feedback of the mouse cursor. The reflex yields information on which side the user sees the screen from, so that the screen coordinates are aligned with the user’s position. The pointing interface relies on neither the quadrilateral of the screen nor infrared rays as traditional ways do. It can work with various settings of a large-screen tabletop-environment.

Acknowledgments

This work was partly supported by Computer Science Laboratory of Fukuoka Institute of Technology.

References

[1] B. Shneiderman and C. Plaisant, Design the user interface: strategies for effective human-computer interaction —4th ed., Addison-Wesley Publishing Company, Boston, 2005.
[2] M. Ishihara and Y. Ishihara, “An approach to remote direct pointing using gray-code,” AVI’06: Proc. Working Conference on Advanced Visual Interfaces, pp.75–78, ACM, New York, NY, USA, 2006.
[3] J. Lee, S. Hudson, J. Summer, and P. Dietz, “Moveable interactive projected displays using projector based tracking,” Proc. UIST2005, pp.63–72, ACM, Oct. 2005.
[4] J. Lee, S. Hudson, and P. Dietz, “Hybrid infrared and visible light projection for location tracking,” UIST’07: Proc. 20th Annual ACM Symposium on User Interface Software and Technology, pp.57–60, ACM, New York, NY, USA, 2007.
[5] D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan, “Effects of display position and control space orientation on user preference and performance,” CHI’06: Proc. SIGCHI Conference on Human Factors in Computing Systems, pp.309–318, ACM Press, New York, NY, USA, 2006.