International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.9 (2015), pp.263-270
http://dx.doi.org/10.14257/ijmue.2015.10.9.27

An Exploration of Cursor Positioning Styles of One-Handed Mobile Camera Based Cursor Manipulation

Liang Chen and Dongyi Chen

School of Automation Engineering,
University of Electronic Science and Technology of China,
Chengdu, 611731, P.R China
liangchen0513@gmail.com, dychen@uestc.edu.cn

Abstract

Mobile camera based interaction, which makes use of video input from the inbuilt camera, has been investigated to serve as a supplementary or alternative technique for direct touch manipulation on mobile devices to address usability issues such as the occlusion problem and fat finger problem. Despite some prototype systems have been designed and developed, we still do not possess a good understanding of many fundamental issues closely related to mobile camera based manipulation. In order to enrich this body of knowledge, we conducted a user study using Fitts' reciprocal pointing tasks to explore different cursor positioning styles' influence on users' pointing performance in one-handed mobile camera based cursor manipulation. The results showed that the style of moving the cursor to the opposite direction to which the device moved outperformed the style of moving the cursor to the same direction in terms of both selection time and error rate, although the latter was not statistically significant. In addition, the former gained great popularity among participants.

Keywords: *Mobile Devices, Mobile Camera Based Interaction, Cursor Manipulation, One-handed Interaction, Cursor Positioning Styles, Pointing Tasks*

1. Introduction

Nowadays, mobile smart devices, especially smart phones, have gained widespread popularity among consumers. Comparing to previous mobile devices, *e.g*. feature phones, the functions of current mobile smart devices have no longer been restricted to merely making phone calls or sending text messages, but already been extended to many other aspects of daily routines, assisting people to study, work, entertain and even travel.

With the rapid development of mobile smart devices, techniques for manipulating them have also been continuously developing to better meet people's increasingly growing demands as well as further promote user experience. At present the technology of direct touch is the most popular manipulation technology for off-the-shelf mobile smart devices due to the intuitive and convenient operations which it enables users with. However, its drawbacks, *e.g*. the occlusion problem and fat finger problem [1], lower users' satisfaction when they manipulate mobile touch devices.

To address these usability issues of direct touch manipulation, HCI (human computer interaction) researchers have explored and implemented various techniques over the last decades. For example, Potter *et al*. proposed the Offset Cursor [2] to enable users with precise selection by utilizing a cursor above the point of the finger contact on the touch screen. Also by only making use of the front touch sensitive surface, Vogel and Baudisch developed Shift [3] which supported precise selection by displaying the virtual contents under the fingertip in a pop-up callout at an occluded place on the screen. Other HCI scientists investigated to address these problems by utilizing the rear of or the space around a mobile device. For example, NanoTouch [4] and RearType [5] both made used

ISSN: 1975-0080 IJMUE

of back-of-device input while SideSight [6] and HoverFlow [7] utilized around-of-device interaction through infra-red sensor input.

Mobile camera based interaction, which utilizes the inbuilt camera of a smart device as the input unit, can also be used to address the occlusion problem and fat finger problem. In comparison to the Offset Cursor and Shift, mobile camera based interaction utilizing rear-facing camera input can completely eliminate hand occlusion. On the other hand, comparing with back-of- and around-of-device interaction, camera based interaction is in fact more likely to be deployed on a mobile device since cameras have already been recognized as standard sensors embedded in a mobile device and their performance have also been keeping improving.

Existing mobile camera based techniques can be roughly classified into two categories by the interaction styles which users utilize to manipulate the device – moving the mobile device itself or moving a tracked object in the camera's field of view. Techniques belonging to the first category can enable users with one-handed mobile device manipulation, which frees the other hand for extra tasks. The research conducted by Karlson *et al*. [8] indicated that one-handed interaction gained more popularity among mobile device users in many usage scenarios. Therefore, it is worth exploring users' performance in one-handed mobile camera based manipulation which can contribute to devising more useful and usable mobile interfaces.

In this paper, we present our investigation on one-handed mobile camera based cursor manipulation, mainly focusing on cursor positioning styles. We conducted a user study to compare users' pointing performance by utilizing two cursor positioning styles - moving the cursor to the opposite direction which the device moved to and moving the cursor to the same direction which the device moved to. The quantitative results of our study indicated that the first cursor positioning style outperformed its competitor in terms of selection time. Furthermore, the subjective results demonstrated that the first cursor positioning style also gained more popularity among the participants.

2. Related Work

It is worth noting that although tons of mobile augmented reality studies and systems also take advantage of camera input, most of them mainly focus on enhancing surrounding environment or displaying 3D models rather than manipulate daily used apps running on mobile devices, thus not relevant to our work.

As mentioned above, mobile camera based interfaces can be roughly classified into two categories by the interaction styles users utilize to generate input – moving the mobile device itself or moving an object in front of the camera. Here we introduce several representative examples of both categories.

2.1. Moving the Mobile Device to Generate Input

Rohs [9] presented a mobile camera-based interface utilizing printed or projected 2-dimensional barcodes for detecting phone movement. In his implementation, rotation or tilting of the phone could also be used for manipulation, *e.g*. displaying different information or triggering various functions related to the selected item. Hansen *et al*. [10] proposed a technique to detect the movement of a mobile device by tracking its consecutive positions relative to a stationary circle in the user's vicinity. In contrast to the foresaid two techniques which tracked specific objects for calculating device motion, the work presented by Wang *et al*. [11] and the research conducted by Haro and colleagues [12] proposed another type of solution which estimated device movement by analyzing image changes in successive captured frames. The technique proposed by Sohn and Lee [13] and research reported by Hansen *et al*. [14] both utilized face tracking technology for estimating the movement of a mobile device. Unlike previous four studies, the interfaces in the last two projects employed front-facing cameras rather than rear-facing ones.

2.2. Moving an Object in Front of the Camera to Generate Input

Hachet and colleagues [15] presented an interface which enabled a user to manipulate 2-dimensional and 3-dimensional graphics on a mobile device through camera input. Instead of tracking the movement of the holding device, their technique tracked the movement of a rectangular object holding in the user's hand for generating commands. Kick-Up Menus was an interesting technique developed by Paelke *et al*. [16] which allowed users to interact with contents on the display by kicking actions. Gallo *et al*. [17] and Baldauf *et al*. [18] both employed fingertip detection techniques for mobile device manipulation. Note that all these interfaces exploited rear-facing cameras for video capture.

3. User Study

Cursor input have been widely used for decades for manipulating various electronic devices. Both interaction styles mentioned in related work can be employed to provide users with cursor input for mobile device manipulation. Unlike cursor manipulation realized by the second category which the cursor just follows the tracked object, there are two types of cursor positioning styles for the first category - moving the cursor to the same or opposite direction which the device moves to. So, which cursor positioning style is more efficient and accurate for users to utilize? Which one is more intuitive and easy-to-use in users' point of view? In order to figure out the two questions posed above, we conducted a user study to compare the pointing performance of two cursor positioning styles in one-handed mobile camera based cursor manipulation. The findings could provide user interface designers with serviceable parameters for future mobile interaction design.

3.1. Apparatus

Our entire user study was conducted on a HTC One S smart phone running Android OS. The smart phone had a 1.5 GHz dual-core CPU and 1 GB RAM. It also possessed a 4.3-inch touch screen with a resolution of 960 x 540 pixels and inbuilt 8-megapixel rear-facing camera which we utilized for obtaining video input.

The experimental software, including the color marker tracker was implemented using Java and Android Software Development Kit. The onscreen cursor was controlled by the estimated device movement relative to a color marker which was placed stationary in front of the participant, in absolute mode. Two cursor positioning styles – the style of moving the cursor to the same direction as the device moved to (hereinafter referred to as *the same direction*) and the style of moving the cursor to the opposite direction which the device moved to (hereinafter referred to as *the opposite direction*), were both implemented. Button-based selection was enabled as the only selection mechanism.

3.2. Participants

Ten participants, 5 male and 5 female, were recruited from the university where the first author worked during the study. The ages of the participants ranged from 23 to 31 with an average age of 27 (SD = 3.13). They were all graduate students and researchers. All participants were right-handed and experienced mobile device users. Half of the participants used their mobile devices more than four hours daily. They were all volunteers, thereby receiving no pay from the user study.

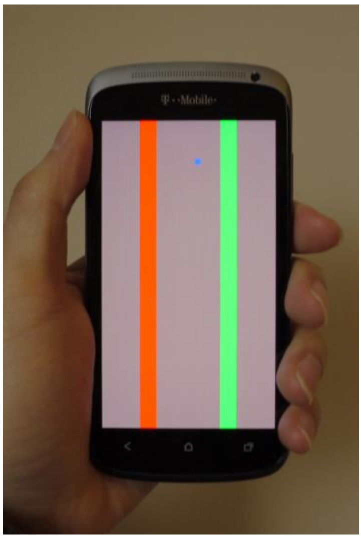

Figure 1. The Appearance of the Fitts' Reciprocal Pointing Task

3.3. Task and Procedure

In our study, we utilized the Fitts' reciprocal pointing task as the experimental task since it had been widely used in previous HCI studies. The appearance of the pointing task is shown in Figure 1. During the study, participants were required to acquire paired vertical rectangular targets with various target widths and different distances from each other. Of the two targets on the display, the active one which should be acquired immediately was rendered in green and the inactive one was displayed in red. Participants were asked to acquire the active target as fast and accurately as possible. Once successfully acquired the active target, it turned into an inactive target right away and the previous inactive target became active simultaneously. If an error acquisition occurred, *i.e.* a participant depressed the button when the cursor was outside the active target, an error sound would be played to remind the participant. In our implementation, cursor positioning was realized by utilizing video input and selection was triggered by pressing down a button on the right side of the device.

Prior to the study, a pre-study questionnaire, which gathered basic personal information of each participant, *e.g.* gender and handedness, was asked to be filled up. After that, a training session was given by the experimenter to introduce the purpose of the user study, the camera based interfaces, and how to perform tasks using the interfaces. Then, the participant started to practice to conduct the tasks. When they felt experienced enough, a total of three blocks of trials were given to them to accomplish. Short breaks were allowed during the study when the experimental software was not timing, *e.g.* between blocks. After completing all trials, a post-study questionnaire was requested to be completed to offer feedback as well as suggestions.

3.4. Experiment Design

$$ID = \log_2\left(\frac{A}{W} + 1\right) \quad (1)$$

$$MT = a + b\log_2\left(\frac{A}{W} + 1\right) \quad (2)$$

In our study, we made use of four target amplitudes (abbreviated as A; 200, 250, 300, 350 pixels respectively) and two target widths (abbreviated as W; 50 and 100 pixels). The widths were based on the widths of UI elements on smart phones. The ID (index of difficulty) values of each A-W condition, calculated by Equation 1 which was the logarithm part of the Shannon formulation (Equation 2) of Fitts' law [19,20], were shown in Table 1.

Table 1. The ID Values (bits)

W \ A	200	250	300	350
50	2.32	2.59	2.81	3
100	1.59	1.81	2	2.17

Participants performed pointing tasks for all eight A-W conditions using both cursor positioning styles. The ordering of cursor positioning styles was counterbalanced and the trials in each A-W condition of each block were appeared randomly. The design of whole experiment could be summed up as follows:

10 participant x
2 cursor positioning styles (*the same direction* and *the opposite direction*) x
3 blocks x
8 A-W conditions (ID values from 1.59 to 3 bits) x
10 trials per A-W condition
= 4800 trials in total.

3.5. Results

3.5.1. Selection Time Analysis

Before conducting the analysis, we first removed records marked as error ones and recognized as outliers. After that, we analyzed the adjusted data using a repeated measures ANOVA. A significant difference for cursor positioning style on selection time was found ($F_{1, 9} = 14.957$, $p < 0.01$), with mean selection times of 2.298 seconds and 2.035 seconds for *the same direction* and *the opposite direction* respectively. As we had expected, there was a significant effect for ID on selection time ($F_{7, 63} = 36.867$, $p < 0.0001$). No significant interaction was found between cursor positioning styles and A-W conditions. Figure 2 demonstrates the mean selection times of both cursor positioning styles for each ID. As shown in Figure 2, we can see that for each ID the mean selection time of *the same direction* is higher than that of *the opposite direction*.

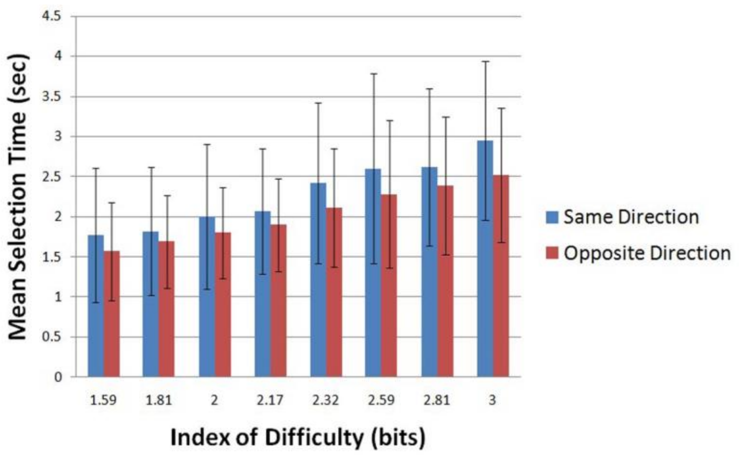

Figure 2. Mean Selection Time for Each ID

3.5.2. Selection Error Analysis

We also conducted a repeated measures ANOVA to analyze the error rates. The results indicated that there was a marginal trend toward significance in selection error between the two cursor positioning styles ($F_{1, 9} = 4.190$, $p = 0.071$), with a mean error rate of 4.6% for *the opposite direction* and 6.20% for *the same direction*. There was a significant main effect for ID on selection error ($F_{7, 63} = 6.669$, $p < 0.001$). No significant interaction was found between cursor positioning styles and A-W conditions. Figure 3 showed the error rates of both selection mechanisms for each ID.

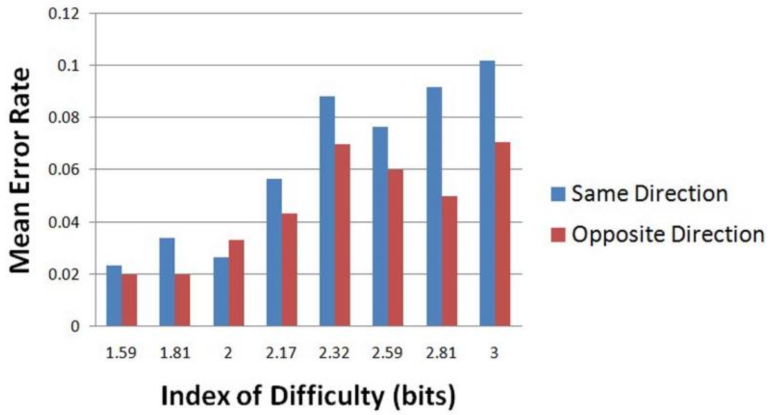

Figure 3. Mean Error Rate for Each ID

3.5.3. User Feedback

The answers we collected from the post-study questionnaires were illustrated in Figure 4. The corresponding questions which were asked in the questionnaire were listed as follows:

(1) Which cursor positioning style do you prefer?

(2) Which cursor positioning style can provide better performance in terms of selection time in your opinion?

(3) Which cursor positioning style is easier to use?

(4) Which cursor positioning style is more comfortable to use?

(5) Which cursor positioning style can get a higher error rate in selecting targets?

From the collected feedback, the cursor positioning style of *the opposite direction* gained a very significant popularity from our participants, for eight out of ten chose this style as the preferred cursor positioning style. Many participants commented that it was more intuitive to control the cursor by this style. The result was also in accordance with our observation that when users first used the style of *the same direction* for controlling the cursor, some of them often moved the cursor to the wrong direction and then corrected it back towards the right direction.

In terms of selection time, four people thought the style of *the opposite direction* did a better performance and five participants believed that there was no significant difference on selection time between the two styles, indicating that almost all participants felt that the style of *the opposite direction* performed at least as well as *the same direction* style did. The quantitative result supported the participants' subjective feelings that *the opposite direction* performed better in terms of selection time.

When asked which cursor positioning style was easier to use, five participants voted for *the opposite direction* while only two participants chose the same direction. Similarly, six people thought *the opposite direction* was more comfortable to use while only two participants selected the same direction. When asked which cursor positioning style might get a higher error rates, seven people chose there was no significant difference between the two styles which conformed to the quantitative results that there was no significant difference for cursor positioning styles in error rate.

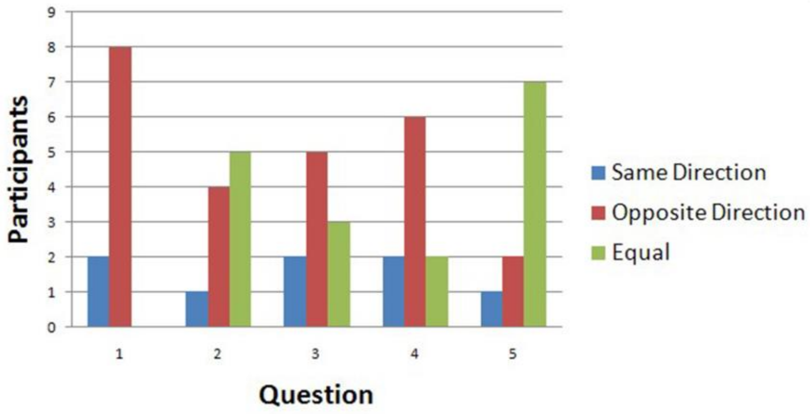

Figure 4. Answers to Questions in Post-Study Questionnaires

4. Conclusion

In this paper we presented the results of a study designed to explore one-handed mobile camera based cursor manipulation. Our findings indicated that the cursor positioning style of *the opposite direction* outperformed in terms of selection time. As for error rate, the style of *the opposite direction* also performed better although this ascendancy was not statistically significant. In addition, the feedback from our participants indicated that the style of *the opposite direction* gained more preference. Taken together, these results could provide mobile user interface designers with more empirical knowledge for developing eligible interactive techniques in the future.

Acknowledgements

The authors would like to thank the ten participants who volunteered to participate in our user studies. The first author was financially supported by China Scholarship Council to conduct his research at University of Toronto in Canada. This project is supported by Sino-German research project (GZ817).

References

[1] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell and C. Shen, “Lucid-Touch: A See-Through Mobile Device”, Proceedings of the 20th annual ACM symposium on User interface software and technology, **(2007)**, pp. 269–278.

[2] R. Potter, L. Weldon and B. Shneiderman, “Improving the accuracy of touch screens: an experimental evaluation of three strategies”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, **(1988)**, pp. 27-32.

[3] D. Vogel and P. Baudisch, “Shift: a technique for operating pen-based interfaces using touch”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, **(2007)**, pp. 657-666.

[4] P. Baudisch and G. Chu, “Back-of-device interaction allows creating very small touch devices”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, **(2009)**, pp. 1923-1932.

[5] J. Scott, S. Izadi, L. S. Rezai, D. Ruszkowski, X. Bi and R. Balakrishnan, “Reartype: text entry using keys on the back of a device”, Proceedings of the 12th international conference on Human computer interaction with mobile devices and services, **(2010)**, pp.171–180.

[6] A. Butler, S. Izadi and S. Hodges, “SideSight: multi-"touch" interaction around small devices”, Proceedings of the 21th annual ACM symposium on User interface software and technology, **(2008)**, pp. 201-204.

[7] S. Kratz and M. Rohs, “HoverFlow: Expanding the design space of around-device interaction”, Proceedings of the 11th international conference on Human computer interaction with mobile devices and services, **(2009)**.

[8] A. Karlson, B. Benderson and J. SanGiovanni, “AppLens and launchTile: two designs for one-handed thumb use on small devices”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, **(2005)**, pp.201-210.
[9] M. Rohs, “Real-World Interaction with Camera-Phones”, Proceedings of the 2nd International Symposium on Ubiquitous Computing Systems, **(2004)**, pp.74-89.
[10] T. R. Hansen, E. Eriksson and A. Lykke-Olesen, “Mixed interaction space: designing for camera based interaction with mobile devices”, CHI’05 extended abstracts on Human factors in computing systems, **(2005)**, pp.1933–1936.
[11] J. Wang, S. Zhai and J. Canny, “Camera phone based motion sensing: Interaction techniques, applications and performance study”, Proceedings of the 19th annual ACM symposium on User interface software and technology, **(2006)**, pp.101-110.
[12] A. Haro, K. Mori, T. Capin and S. Wilkinson, “Mobile camera-based user interaction”, Proceedings of the International Conference on Computer Vision Workshop on Human Computer Interaction, **(2005)**, pp.79-89.
[13] M. Sohn and G. Lee, “ISeeU: camera-based user interface for a handheld computer”, Proceedings of the 7th international conference on Human computer interaction with mobile devices and services, **(2005)**, pp. 299-302.
[14] T. R. Hansen, E. Eriksson and A. Lykke-Olesen, “Use your head: exploring face tracking for mobile interaction”, CHI '06 extended abstracts on Human factors in computing systems, **(2006)**, pp. 845-850.
[15] M. Hachet, J. Pouderoux and P. Guitton, “A camera-based interface for interaction with mobile handheld computers”, Proceedings of the 2005 symposium on Interactive 3D graphics and games, **(2005)**, pp. 65-72.
[16] V. Paelke, C. Reimann and D. Stichling, “Kick-up menus”, CHI '04 extended abstracts on Human factors in computing systems, **(2004)**, pp. 1552-1552
[17] O. Gallo, S. M. Arteaga and J. E. Davis, “Camera-based pointing interface for mobile devices”, Proceedings of the 15th IEEE International Conference on Image Processing, **(2008)**, pp. 1420 - 1423.
[18] M. Baldauf, S. Zambanini, P. Fröhlich and P. Reich, “Markerless visual fingertip detection for natural mobile device interaction”, Proceedings of the 13th international conference on Human computer interaction with mobile devices and services, **(2011)**, pp.539-544.
[19] I. S. Mackenzie, “A note on the information-theoretic basis for Fitts’ law”, Journal of Motor Behavior, vol. 21, no. 3, **(1989)**, pp. 323-330.
[20] I. S. Mackenzie, “Fitts’ law as a research and design tool in human-computer interaction”, Human Computer Interaction, vol. 7, no. 1, **(1992)**, pp. 91-139.

Authors

Liang Chen, is now a doctoral student, majored in Computer Science, in School of Automation Engineering at University of Electronic Science and Technology of China (UESTC). His current research interest focuses on human-computer interaction, especially on prototyping and evaluating mobile or wearable devices and interfaces. He used to work at Department of Computer Science at University of Toronto as a visiting researcher from 2012 to 2014.

Dongyi Chen, is a professor in School of Automation Engineering at UESTC where he directs Mobile Computing Center. He used to work at University of Toronto, Georgia Tech, and some other foreign universities. His current research interest includes ubiquitous computing, mobile computing, wearable computing, wireless sensor network, augmented reality, and so on.

