PAPER *Special Section on Electronic Displays*

A Comparison Study on Camera-Based Pointing Techniques for Handheld Displays

Liang CHEN[†a)], ***Member and*** **Dongyi CHEN**[†b)], ***Nonmember***

SUMMARY Input devices based on direct touch have replaced traditional ones and become the mainstream interactive technology for handheld devices. Although direct touch interaction proves to be easy to use, its problems, e.g. the occlusion problem and the fat finger problem, lower user experience. Camera-based mobile interaction is one of the solutions to overcome the problems. There are two typical interaction styles to generate camera-based pointing interaction for handheld devices: move the device or move an object before the camera. In the first interaction style, there are two approaches to move a cursor's position across the handheld display: move it towards the same direction or the opposite direction which the device moves to. In this paper, the results of a comparison research, which compared the pointing performances of three camera-based pointing techniques, are presented. All pointing techniques utilized input from the rear-facing camera. The results indicate that the interaction style of moving a finger before the camera outperforms the other one in efficiency, accuracy, and throughput. The results also indicate that within the interaction style of moving the device, the cursor positioning style of moving the cursor to the opposite direction is slightly better than the other one in efficiency and throughput. Based on the findings, we suggest giving priority to the interaction style of moving a finger when deploying camera-based pointing techniques on handheld devices. Given that the interaction style of moving the device supports one-handed manipulation, it also worth deploying when one-handed interaction is needed. According to the results, the cursor positioning style of moving the cursor towards the opposite direction which the device moves to may be a better choice.

key words: *handheld devices, handheld displays, camera-based interaction, Fitts' law, pointing techniques*

1. Introduction

Over the past decade, direct touch input has replaced keypads and been widely used on various handheld devices, becoming the mainstream mobile interaction technology. Direct touch provides users with a number of intuitive, easy-to-learn and effective touch gestures to improve the efficiency of completing miscellaneous daily tasks on handheld displays. With the assistance of direct touch input, the functions of handheld devices have been widely extended from the most basic communication-related applications (e.g. to make a phone call or send a message) to supporting many other everyday activities, such as learning, working, gaming, traveling, and even bodybuilding.

While the advantages of direct touch input have improved user experience in mobile operation, its limitations (such as the occlusion problem [1], [2] and the fat finger problem [3]) cause negative effects on the usability of handheld devices. For example, the occlusion problem, caused by the operating hand(s) over the touchscreen, prevents the user from viewing the full digital contents on the handheld display during manipulation, thereby disrupting the visual experience and even possibly causing incorrect operations. On the other hand, the fat finger problem, which is caused by the size and softness of a fingertip [4], decreases the user's ability to make precise selections on a handheld display, especially when the target is much smaller than the fingertip. Therefore, developing novel interactive techniques, which can avoid the above usability problems, for handheld displays is very meaningful and of great importance.

Consequently, HCI (human-computer interaction) researchers have devised and implemented a number of new interactive techniques to alleviate the negative effects of hand occlusion and "fat fingers". Some researchers developed software techniques to solve the problems. Potter et al. presented Offset Cursor [5] which realized precise selection by creating a software cursor above the contact point of the finger on the touchscreen. The user could move the cursor by dragging his/her finger across the display. Vogel et al. presented Shift [6] that enabled precise selection by triggering a callout which rendered the contents currently covered under the fingertip. Other researchers explored making use of the previously unused rear surface or the surrounding space of a handheld device for manipulation. For instance, NanoTouch [4] utilized a rear-mounted touch panel to enable touch manipulation on very small devices; RearType [7] took advantage of a keyboard attached on the device's rear to provide back-of-device text input; Granell and Leiva [8], [9] presented tap-based back-of-device interaction realized by using inbuilt sensors of an off-the-shelf smartphone. On the other hand, SideSight [10] and Hoverflow [11] both used the input from infrared sensors to provide the users with around-of-device interaction; Xiao et al. [12] utilized vibro-acoustic sensors to extend touch interaction to the surfaces near the mobile device. Since the operating hands are not over or on the display during using back-of-device or around-of-device interaction, the problems mentioned above can be avoided.

Camera-based mobile interaction that employs the input of an inbuilt camera (e.g. the rear-facing camera of a smartphone) for manipulation can also be utilized to resolve the above usability problems. Compared to the software techniques such as Offset Cursor and Shift, camera-based

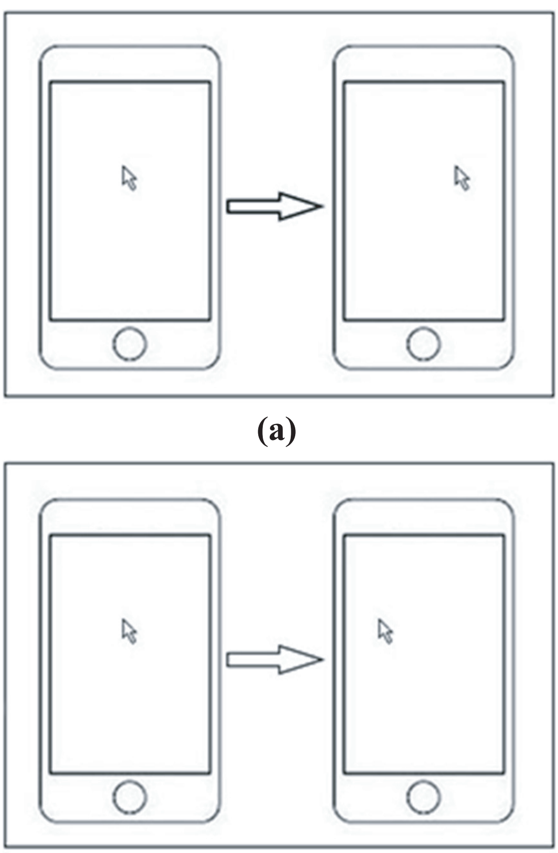
Manuscript received February 29, 2020.
Manuscript revised June 1, 2020.
Manuscript publicized August 4, 2020.

†The authors are with the University of Electronic Science and Technology of China, Chengdu, China.
a) E-mail: liangchen0513@gmail.com
b) E-mail: dychen@uestc.edu.cn

DOI: 10.1587/transele.2020DIP0003

mobile interaction is able to completely avoid hand occlusion. Compared to back-of-device or around-of-device interaction, it is more feasible and easy to implement camera-based interaction on handheld devices because nowadays almost all handheld devices possess at least one inbuilt camera.

There are two types of interaction styles to utilize a camera-based interactive technique on a handheld device [13]. The first interaction style generates interactions by directly moving the handheld device and the second interaction style generates interactions by moving something (e.g. a finger) within the field of view of the camera. The two interaction styles can both be used to realize camera-based pointing interaction. There are two types of cursor positioning styles when utilizing the first interaction style to enable pointing interaction: the cursor moves to the same direction as the device moves to (see Fig. 1(a)) and the cursor moves to the opposite direction as the device moves to (see Fig. 1(b)). The results in [13] showed that the interaction style of moving a finger performed better than the interaction style of moving the device itself (the cursor positioning style is to move the cursor to the same direction which the device moves to) in terms of efficiency when completing pointing tasks.



(a)

(b)

Fig. 1 The two types of cursor positioning styles in the interaction style of moving the handheld device itself to generate pointing interaction. (a) The cursor moves to the same direction as the device moves to; (b) the cursor moves to the opposite direction as the device moves to. The arrow indicates the direction which the device moves to.

In the interaction style of moving the device, does cursor positioning style affect user performance? In addition, which cursor positioning style will achieve better user performance? If the cursor positioning style of moving the cursor towards the opposite direction which the device moves to achieves better performance, will it outperform the interaction style of moving a finger? In order to answer these questions and get a more complete understanding on camera-based pointing techniques that make use of the rear-facing camera, we carried out a user study, which compared the pointing performance of three camera-based pointing techniques. The three pointing techniques are Finger (moving the index finger of the participant's right hand in front of the rear-facing camera to move the cursor), DeviceOPPO (moving the device to move the cursor towards the opposite direction which the device moves to), and DeviceSAME (moving the device to move the cursor towards the same direction as the device moves to). The paper is organized as follows. First, we briefly review the literature which is most relevant to our study. After that, we describe the details of our user study, including the information about apparatus, participants, tasks, etc. At last, the results of the analysis on the collected data are presented.

2. Related Work

Many mobile AR (augmented reality) projects [14]–[16] also utilize the input from the inbuilt camera. However, since our research focuses on camera-based pointing techniques that are used to manipulate the applications that we use daily on handheld devices rather than enhancing the environment around us or viewing 3D objects from various angles, most AR studies are not related to our study.

As described in the previous section, there are two types of interaction styles to enable camera-based mobile interaction: move an object before the camera or move the handheld device itself. Here, we briefly review a number of representative studies on the two interaction styles.

2.1 Moving the Device Itself

A camera-based interactive technique which made use of a 2D barcode tracking system was introduced by Rohs [17]. The tracking system calculated the motion of the handheld device relative to the barcode and then utilized it to generate interaction. Hansen et al. [18] introduced a similar interactive technique which utilized a tracker to detect and track a circle placed in front of the camera. Unlike the above two techniques which employed certain tracking objects, Wang et al. [19] and Haro et al. [20] introduced another kind of approach that estimated the motion of the handheld device through analyzing image differences in successive frames captured by the camera. Researchers also explored making use of the front-facing camera to generate mobile interaction. For instance, Sohn et al. [21] and Hansen et al. [22] employed computer vision techniques to track the user's face. Their interactive techniques utilized the movement of the handheld device relative to the user's head to generate interactions.

2.2 Moving an Object before the Camera

Hachet et al. [23] proposed an interactive technique which utilized a tracker to track a rectangular target with special designed pattern. The user interacted with the mobile user interface through moving the rectangular target in front of the camera. Gallo et al. [24], Chen et al. [25], and Makino [26] developed fingertip trackers and utilized them to realize pointing manipulation on mobile devices. Song et al. [27], Krejov et al. [28], and Fanello et al. [29] developed more complex algorithms which enabled camera-based mobile interaction with multiple fingers.

3. User Study

We conducted a user study to try to find out which interaction style and which cursor positioning style (in the interaction style of moving the device to generate interaction) would provide users with better pointing performance. In the user study, we compared the performance of three camera-based pointing techniques (Finger, DeviceOPPO, and DeviceSAME) by letting the participants complete pointing tasks.

3.1 Apparatus

The experimental software was developed on Android platform. The test device was a smartphone that owned a 1.5 GHz dual-core CPU and 1 GB RAM. The mobile device was also equipped with a 4.3 inches touchscreen. The resolution of the display was 540×960 pixels, approximately one pixel each millimeter. It possessed an 8-megapixel rear camera that was used to provide the camera input to our experimental software. The rear camera was in the center of the top of the smartphone's back.

A color marker in blue was used as the tracking object in our study for the two interaction styles. For Finger, the marker was worn on the participant's index finger of the right hand; for DeviceSAME and DeviceOPPO, the marker was placed nearby in front of the participant. In a pilot study, we found that when participants conducted the pointing tasks by Finger, the average distance between the camera and the finger was about 15cm. This distance could ensure effective control of the cursor movement and reduce fatigue during manipulation. Therefore, in the formal study, we requested the participants to maintain an approximate distance of 15cm between the camera and the finger when conducting the pointing tasks by Finger. In this case, the ratio between the amount of movement of the finger and the cursor was about 2. In order not to affect the results due to the settings of the experiment, when using DeviceOPPO and DeviceSAME to complete the tasks, we also asked the participants to maintain a distance of about 15cm between the camera and the marker. The ratio between the amount of movement of the device and the cursor was around 2. In addition, when using DeviceOPPO and DeviceSAME, we asked the participants to control the cursor by moving the device horizontally. It was prohibited to move the cursor by rotating the device.

In our study, there were mainly two reasons to track a color marker. On one hand, it was easier to develop a tracker to detect and track a color marker, and running such a tracker required less computing resources on a handheld device. On the other hand, it would not introduce other factors that might have impact on the results of our study. For instance, an algorithm for detecting and tracking a finger had more complexity compared with that for detecting and tracking a rectangle in the nearby environment. As a result, the algorithm for finger tracking would require more computing resources and time to process the camera input and this may influence the results of our study.

3.2 Participants

Twelve participants from our university, aged 20 to 26 years (mean=23.08 years, SD=1.73 years), were recruited. All of them were right-handed. They were experienced in using mobile touch devices, and none of them had used any pointing techniques on handheld devices through camera-based input before our user study. All participants volunteered to take part in our study without any payment.

3.3 Task and Procedure

In this research, the Fitts' reciprocal pointing task [30] was leveraged because this task had been widely adopted in prior HCI researches [31], [32]. When completing the pointing tasks, the participants were requested to hold the device with a normal and comfortable posture, where the angle of the device relative to the horizon was about 70 degrees.

After the start button was successfully selected, two rectangular targets (one in red and the other in green) would be rendered on the display. The two targets were at equal distance from the center of the display. The green target was the active target that participants were asked to select it as fast as they could; the other target was inactive. A small blue cursor could be manipulated by each of the three camera-based pointing techniques. The participant was required to select the active target as quickly and accurately as possible. If the active target was successfully selected, it would turn to inactive immediately and the former inactive target would turn to active at the same time. If the participant made an unsuccessful selection (i.e. the selection was made outside the active target), the experiment software would play an error sound to remind the participant to be more careful. In the current implementation, camera input was utilized to position the cursor and the click action was realized by pressing a button on the device's right side.

At the beginning of the study, a pre-questionnaire was

Table 1 The ID values (bits) of each type of pointing trials.

		Width	
		50	100
Amplitude	200	2.32	1.59
	250	2.59	1.81
	300	2.81	2.00
	350	3.00	2.17

required to be filled out to collect the participant's personal information. After that, a training session was given to let the participant learn and practice how to conduct the pointing tasks by the three camera-based pointing techniques. After the participant deemed that he or she was skilled enough, five blocks of trials would be given to accomplish. During the experiment, short breaks were allowed when not timing.

3.4 Experiment Design

A within-subject design was used in this research. During the study, the participants were asked to complete five blocks of experimental tasks using each of the three camera-based pointing techniques. The ordering of the pointing techniques was counterbalanced. In the pointing task, 2 target widths (50 and 100 pixels) and 4 target amplitudes (200, 250, 300 and 350 pixels) were used, thus resulting eight different types of trials. The selected target widths were respectively similar to the width of the soft keyboard's key (50 pixels) and the application icon (100 pixels) of our experiment device. The ID (index of difficulty) values of each type of trials were calculated by Eq. (1) and illustrated in Table 1. In each block, the participant conducted pointing tasks for all the eight Amplitude-Width conditions, and the trials in each Amplitude-Width condition would appear in random order.

To sum up, the design of the experiment is:
12 participants ×
3 Pointing Techniques ×
5 blocks ×
8 Amplitude-Width conditions ×
10 trials
= 14400 in total.

$$ID = log_2(\frac{Amplitude}{Width} + 1) \tag{1}$$

4. Results

4.1 Completion Time Analysis

Before performing statistical analysis on the data collected from the experiment, the data set was first preprocessed. During the preprocessing, the trials marked as error ones and outliers were excluded from the data set. Here, the outliers refer to the trials whose completion time was longer than three standard deviations of the same participant's mean completion time by using the same pointing technique.

A repeated measures ANOVA was applied to the preprocessed data. A significant main effect for Pointing Technique on the completion time ($F(2,22)=15.399$, $p<0.001$) was found, with the mean completion times of 1.77s (SD=0.55s), 1.94s (SD=0.60s), and 1.47s (SD=0.38s) for DeviceOPPO, DeviceSAME, and Finger. Post hoc analysis indicated that the completion time of Finger was significantly shorter than those of DeviceOPPO ($p<0.01$) and DeviceSAME ($p<0.001$), but the difference in the completion time between DeviceOPPO and DeviceSAME was not significant ($p=0.429$).

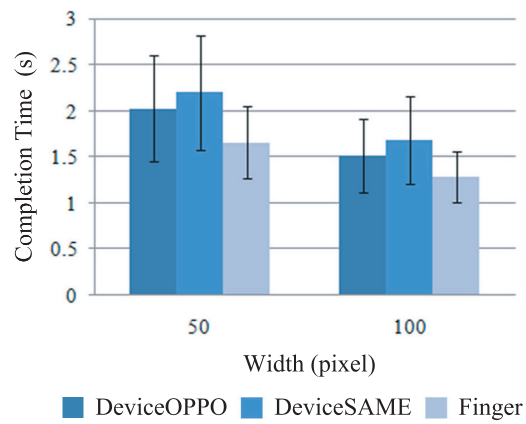


Fig. 2 The mean completion times of each pointing technique under each width condition.

As expected, a significant main effect for Width ($F(1,11) = 235.661$, $p < 0.001$) was found. For the 50-pixel target, the mean completion time was 1.96s (SD=0.58s); and for the 100-pixel target, the mean completion time was 1.50s (SD=0.42s). The mean completion times of each pointing technique under different Width conditions are illustrated in Fig. 2. There was a significant interaction between Pointing Technique and Width ($F(2,22)=7.539$, $p<0.01$). A post hoc test indicated that all three pointing techniques took significant longer times to complete the 50-pixel pointing tasks than the 100-pixel ones (all p values are less than 0.001).

Figure 3 shows the mean completion times for the three pointing techniques under the eight IDs. From Fig. 3, we can see that under each ID, Finger took the shortest time while DeviceSAME took the longest time. We also can see that the mean completion times of all three pointing techniques increase as the ID values increase, which is in line with the expectation of Fitts' law.

4.2 Selection Error Analysis

An error selection happened when the participant made a selection before the cursor was moved into the active target. A repeated measures ANOVA revealed a significant main effect for Pointing Technique on the selection error ($F(2,22)=5.721$, $p<0.05$), with the mean error rates of 0.071, 0.071, and 0.044 for DeviceOPPO, DeviceSAME, and Finger respectively. Post hoc analysis indicated that the error rate of Finger was significantly lower than that of DeviceOPPO ($p<0.05$), but not significantly different to that of

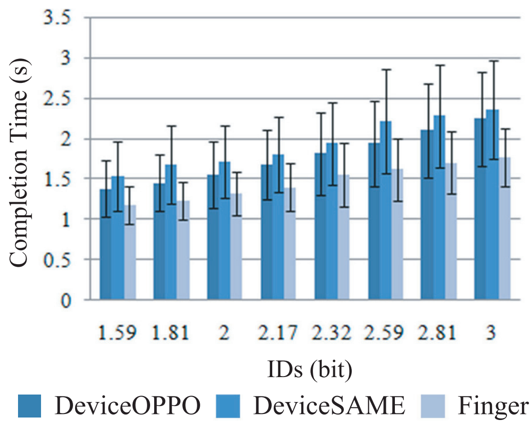


Fig. 3 The mean completion times of each pointing technique under each ID value.

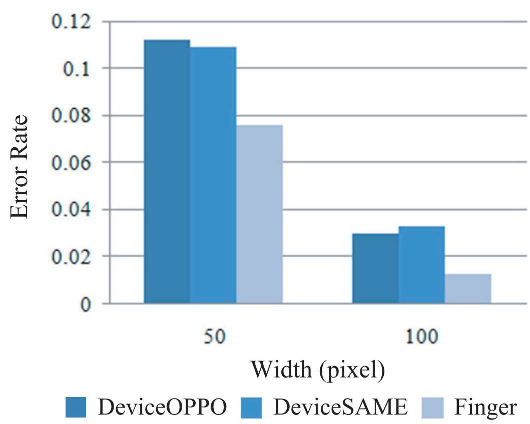


Fig. 4 The mean error rates of each pointing technique under each width condition.

DeviceSAME (p=0.086) which is in line with the result in [13]; and the difference in error rate between DeviceOPPO and DeviceSAME was not significant (p=1.000).

As expected, a significant difference for Width ($F(1,11) = 72.191$, $p < 0.001$) was also found. For the narrow target, the mean error rate was 0.099; and for the wide target, the mean error rate was 0.026. There was no significant interaction between Pointing Technique and Width ($F(2,22)=1.610$, $p=0.222$). The mean error rates of each pointing technique under different Width conditions are illustrated in Fig. 4. From Fig. 4, we can see that all three pointing techniques committed more errors to complete the 50-pixel tasks than the 100-pixel tasks, and Finger committed fewer errors than the other two pointing techniques under both target widths.

Figure 5 shows the mean error rates for the three pointing techniques under each ID. From Fig. 5 we can see that when using the three pointing techniques to complete the pointing tasks with the same amplitude, the mean error rates of the tasks with a narrower width (50 pixels) were much higher than those of the tasks with a wider width (100 pixels). We also can see that Finger committed much fewer errors than the other two pointing techniques under each ID condition.

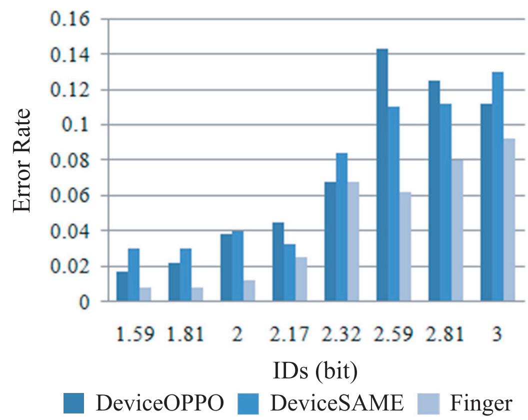


Fig. 5 The mean error rates of each pointing technique under each ID value.

4.3 Fitts' Law Model and Throughput

A Fitts' law model (see Eq. (2)) is obtained by regressing the mean completion times, which are also called the movement times (MT), on the associated Amplitude-Width conditions. Each Amplitude-Width condition is expressed as an ID value which can be calculated by Eq. (1), as mentioned above. Index of performance is an important throughput metric that can be used to compare the performances across studies. A vital consideration when comparing throughput is whether or not the speed-accuracy tradeoff is normalized. In this paper, we utilized Welford's technique [33] to normalizing the variability using participants' error rate. For each Amplitude-Width condition, the target width was first transformed to an effect target width (W_e) by using Eq. (3) [34], and then the effective ID (ID_e) could be calculated by using Eq. (4), see Table 2. After that, MT was regressed on ID_e, see Eq. (5). Table 3 summarizes the models for the three camera-based pointing techniques. We can see that the squared correlations (R^2) of the models are high, which indicates that all three models provide very good fits to the experiment data.

$$MT = a + b \times ID \tag{2}$$

$$W_e = \begin{cases} Width \times (\frac{2.066}{Z(1-Err/2)}) & if\ Err > 0.0049\% \\ Width \times 0.5089 & otherwise. \end{cases} \tag{3}$$

$$ID_e = log_2(\frac{Amplitude}{W_e} + 1) \tag{4}$$

$$MT = a + b \times ID_e \tag{5}$$

For performance comparisons, we turned to throughput, also called index of performance (IP). IP represents

Table 2 The values of W_e and ID_e under different conditions.

Technique	Width	Amplitude	W_e	ID_e
DeviceOPPO	50	200	56.69	2.18
		250	70.58	2.18
		300	67.33	2.45
		350	65.04	2.67
	100	200	86.30	1.73
		250	89.98	1.92
		300	99.75	2.00
		350	103.06	2.14
DeviceSAME	50	200	59.68	2.12
		250	64.60	2.28
		300	64.97	2.49
		350	68.26	2.62
	100	200	95.20	1.63
		250	95.20	1.86
		300	100.63	1.99
		350	97.08	2.20
Finger	50	200	56.67	2.18
		250	55.32	2.46
		300	59.01	2.61
		350	61.27	2.75
	100	200	78.31	1.83
		250	78.31	2.07
		300	81.92	2.22
		350	92.17	2.26

Table 3 Fitts' law models of the three camera-based mobile pointing techniques.

Technique	a	b	R	R^2
DeviceOPPO	-0.417	1.014	0.969	0.939
DeviceSAME	0.034	0.885	0.943	0.889
Finger	-0.090	0.679	0.929	0.863

Table 4 The throughput of each mobile pointing technique.

	DeviceOPPO	DeviceSAME	Finger
IP (bits/s)	1.23	1.11	1.57

the human rate of information processing in conducting the pointing tasks. We utilized Eq. (6) [34] to compute each pointing technique's throughput. The results are listed in Table 4. By comparison, it can be seen that the performance of Finger is higher than those of DeviceOPPO and DeviceSAME respectively by 27.6% and 41.4%; and the performance of DeviceOPPO is 10.8% higher than that of DeviceSAME

$$IP = \frac{ID_e}{MT} \tag{6}$$

4.4 Participant Feedback

At the end of the experiment, each participant filled out a post-study survey. The main purpose of the survey was not to compare the three camera-based mobile pointing techniques but to get a preliminary understanding on how the participants felt about the techniques during the experiment. From the survey, 9 of our participants deemed Finger was their most preferred pointing technique, 2 participants favored DeviceOPPO, and 1 participant preferred DeviceSAME. Likert-scale questions were also utilized in the survey. The collected results are illustrated in Fig. 6. From Fig. 6, we can clearly observe that overall the participants believed that Finger was easier to learn and use, and they also deemed that Finger was more efficient and accurate to use. These results are in line with the results in the completion time and the selection error analysis, in which Finger took shorter time and caused few errors. We also can see that DeviceOPPO was deemed to be slightly easier to learn and use than DeviceSAME.

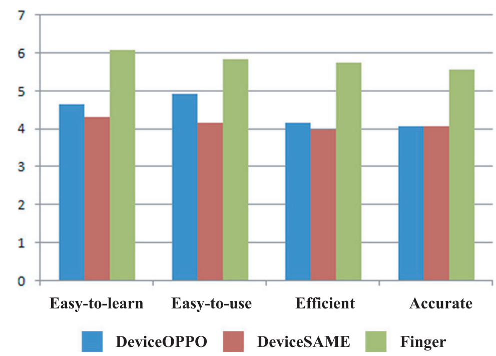


Fig. 6 The subjective feedbacks from the participants.

5. Conclusion

According to the results of the user study, we can clearly see that Finger was better than the other two pointing techniques in terms of efficiency, accuracy, and throughput. Finger was also the most preferred camera-based mobile pointing technique according to the participants' feedback. Therefore, when deploying camera-based pointing technology on handheld devices, priority should be given to the interaction style of moving a finger.

Although the interaction style of moving the device was not as good as the other interaction style, it also possesses its own strengths. For example, it only needs one hand (the holding hand) for mobile manipulation. Among the two pointing techniques in this interaction style, DeviceOPPO performed better (but not significant) than DeviceSAME in efficiency. In addition, the throughput of DeviceOPPO was also slightly higher than DeviceSAME. Hence, DeviceOPPO may be the better choice when applying camera-based one-handed pointing technology on handheld devices.

It is also worth mentioning that the 50-pixel target is apparently not large enough to be accurately acquired by the camera-based pointing techniques. Even with Finger, the mean error rate was close to 8% (see Fig. 4). According to the W_e values (the width adjusted by Crossman's correction to enforce a 4% post hoc error rate) in Table 2, the target width should be greater than 60 pixels for Finger; and it should be at least 65 pixels for DeviceOPPO and DeviceSAME, preferably greater than 70 pixels.

According to the findings of previous literature [2], we can see that even Finger is not as fast as direct touch input. Therefore, camera-based pointing technology should be seen as a complement rather than a substitute for direct touch input. For example, when direct touch input does not work properly, or when screen occlusion should be avoided as much as possible during operation, camera-based pointing technology can be used to manipulate handheld devices.

Acknowledgements

The authors would like to thank the participants who took part in our user study. This work is supported by the National Key R&D Program of China (No. 2016YFB1001401).

References

[1] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen, "Lucid Touch: a see-through mobile device," UIST2007, pp.269–278, Oct. 2007.
[2] L. Chen, D. Chen, and X. Chen, "A Comparison Study on Front- and Back-of-Device Touch Input for Handheld Displays," IEICE Trans. Electron., vol.E101-C, no.11, pp.880–883, Nov. 2018.
[3] K. Siek, Y. Rogers, and K. Connelly, "Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs," INTERACT2005, pp.267–280, 2005.
[4] P. Baudisch and G. Chu, "Back-of-device interaction allows creating very small touch devices," CHI2009, pp.1923–1932, April 2009.
[5] R. Potter, L. Weldon, and B. Shneiderman, "Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies," CHI1988, pp.27–32, May 1988.
[6] D. Vogel and P. Baudisch, "Shift: a technique for operating pen-based interfaces using touch," CHI2007, pp.657–666, April 2007.
[7] J. Scott, S. Izadi, L. Rezai, D. Ruszkowski, X. Bi, and R. Balakrishnan, "Reartype: Text Entry Using Keys on the Back of a Device," MobileHCI2010, pp.171–180, Sept. 2010.
[8] E. Granell and L.A. Leiva, "Less Is More: Efficient Back-of-Device Tap Input Detection Using Built-in Smartphone Sensors," ISS2016, pp.5–11, Nov. 2016.
[9] E. Granell and L.A. Leiva, "βTap: back-of-device tap input with built-in sensors," MobileHCI2017, Article No. 52, Sept. 2017.
[10] A. Butler, S. Izadi, and S. Hodges, "SideSight: Multi-"touch" Interaction Around Small Devices," UIST2008, pp.201–204, Oct. 2008.
[11] S. Kratz and M. Rohs, "HoverFlow: Expanding the Design Space of Around-device Interaction," MobileHCI2009, Article No. 4, Sept. 2009.
[12] R. Xiao, G. Lew, J. Marsanico, D. Hariharan, S. Hudson, and Chris Harrison, "Toffee: enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation," MobileHCI2014, pp.67–76, Sept. 2014.
[13] L. Chen and D. Chen, "An empirical study of interaction styles of mobile camera-based cursor manipulation," J. Comput. Theor. Nanos., vol.13, no.1, pp.989–992, Jan. 2016.
[14] W. Hürst and C. van Wezel, "Gesture-based interaction via finger tracking for mobile augmented reality," Multimed. Tools Appl., vol.62, no.1, pp.233–258, Jan. 2013.
[15] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui, "Mobile augmented reality survey: from where we are to where we go," IEEE Access, vol.5, pp.6917–6950, April 2017.
[16] W. Chun and T. Hollerer, "Real-time hand interaction for augmented reality on mobile phones," IUI2013, pp.307–314, March 2013.
[17] M. Rohs, "Real-world interaction with camera phones," UCS2004, pp.74–89, 2004.
[18] T.R. Hansen, E. Eriksson, and A. Lykke-Olesen, "Mixed interaction space: designing for camera based interaction with mobile devices," CHI2005 extended abstracts, pp.1933–1936, April 2005.
[19] J. Wang, S. Zhai, and J. Canny, "Camera phone based motion sensing: Interaction techniques, applications and performance study," UIST2006, pp.101–110, Oct. 2006.
[20] A. Haro, K. Mori, T. Capin, and S. Wilkinson, "Mobile camera-based user interaction," ICCV2005, pp.79–89, 2005.
[21] M. Sohn and G. Lee, "ISeeU: camera-based user interface for a handheld computer," MobileHCI2005, pp.299–302, Sept. 2005.
[22] T.R. Hansen, E. Eriksson, and A. Lykke-Olesen, "Use your head: exploring face tracking for mobile interaction," CHI2006 extended abstracts, pp.845–850, April 2006.
[23] M. Hachet, J. Pouderoux, and P. Guitton, "A camera-based interface for interaction with mobile handheld computers," I3D2005, pp.65–72, April 2005.
[24] O. Gallo, S.M. Arteaga, and J.E. Davis, "Camera-based pointing interface for mobile devices," ICIP2008, pp.1420–1423, 2008.
[25] L. Chen, D. Chen, and X. Chen, "MoCamMouse: Mobile Camera-based Mouse-like Interaction," ICSAI2017, pp.132–136, 2017.
[26] Y. Makino, "Thumbnail input for head-mounted display," Haptic Interaction, LNEE, vol.277, pp.197–200, June 2015.
[27] J. Song, G. Sörös, F. Pece, S.R. Fanello, S. Izadi, C. Keskin, and O. Hilliges, "In-air gestures around unmodified mobile devices," CHI2014, pp.319–329, Oct. 2014.
[28] P. Krejov and R. Bowden, "Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima," FG2013, pp.1–7, 2013.
[29] S.R. Fanello, C. Keskin, S. Izadi, P. Kohli, D. Kim, D. Sweeney, A. Criminisi, J. Shotton, S.B. Kang, and T. Paek, "Learning to be a depth camera for close-range human capture and interaction," Acm T Graphic, vol.33, no.4, pp.1–11, July 2014.
[30] P.M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," J. Exp. Psychol., vol.47, no.6, pp.381–391, June 1954.
[31] C. Forlines and R. Balakrishnan, "Evaluating Tactile Feedback and Direct vs. Indirect Stylus Input in Pointing and crossing Selection Tasks," CHI2008, pp.1563–1572, April 2008.
[32] J.O. Wobbrock, B.A. Myers, and H.H. Aung, "The performance of hand postures in front- and back-of-device interaction for mobile computing," Int. J. Hum-Comput. St., vol.66, no.12, pp.857–875, Dec. 2008.
[33] A.T. Welford, "The Fundamentals of Skill," London: Methuen, 1968.
[34] R.W. Soukoreff and I.S. MacKenzie, "Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI," Int. J. Hum.-Comput. Stud., vol.61, no.6, pp.751–789, Dec. 2004.

Liang Chen received his master's degree and doctor's degree in computer science respectively from the School of Computer Science and Engineering and the School of Automation Engineering at the University of Electronic Science and Technology of China. He is currently a Postdoctoral Researcher with the Mobile Computing Center at the University of Electronic Science and Technology of China. From 2012 to 2014, he was a Visiting Researcher with the DGP lab at the University of Toronto. His current research interests mainly focus on human-computer interaction and wearable computing.

Dongyi Chen received his doctor's degree from the School of Electronic and Information Engineering of Chongqing University in 1997. From 1992 to 2000, he was an Associate Professor at Chongqing University. From 1997 to 1999, he worked at the University of Toronto as a Visiting Scientist. In 2000, he was promoted to be a Full Professor at Chongqing University. From 2002 to 2005, he was a Visiting Professor with the GVU Center, Georgia Institute of Technology. Since 2004, he has been a Full Professor with the School of Automation Engineering at University of Electronic Science and Technology of China, where he directs the Mobile Computing Center. His main research interests include wearable computing, ubiquitous computing, augmented reality, and human-computer interaction.