以單台彩色深度攝影機實現基於視角之遠距直接選取技術

黃志強* 梁容豪* 詹力韋† 陳炳宇†
國立臺灣大學
*{denny, howieliang}@cmlab.csie.ntu.edu.tw †{robin, liwei.chan}@ntu.edu.tw

ABSTRACT
目前市面上智慧電視的體感操作多是以移動游標為主，使用者均是藉由移動手掌來操作螢幕上的游標，並透過手勢執行點選指令。過去有許多研究探討如何徒手直接選取螢幕上的目標，而非透過移動游標的方式，以增加體感操作的效率。然而，前人所提出的遠距直接選取技術多半需要特別的硬體設置，以準確的偵測使用者動作，因此要將該技術佈建於一般的環境中是有困難的。因此，在本論文中，我們提出FingerShot系統，一種基於視角之遠距直接選取技術，且只需要使用單台彩色深度攝影機，即可讓使用者遠距直接選取螢幕上之目標。我們提出之即時偵測演算法會追蹤使用者的上半身、眼睛及指尖，並計算眼睛到指尖之射線與螢幕的交點，即使用者瞄準的位置。我們的系統也支援雙手操作及多人使用。本系統之使用者測試結果顯示，在距離螢幕至少1.6公尺遠的五種不同位置，當使用者閉起非慣用眼僅用單眼瞄準時，可點選到螢幕上11公分寬的目標；當雙眼皆張開時，可點選到12.3公分寬的目標。

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation (e.g. HCI)**]: User Interfaces

General Terms
Algorithms

1. INTRODUCTION
Hand tracking technologies allow users to control a remote display by freehand pointing. The most prominent freehand pointing method is by controlling a body-centric cursor, e.g. Kinect. Using that method, a user can first place the cursor to a rough position on the remote display, move the cursor to the exact position, then commit the selection by a gesture. Although controlling the body-centric cursor is intuitive just like using a PC, it is not efficient for novel users. Inaccurate cursor placement results in long dragging movement, and therefore causes consequent arm fatigue problems.

Perspective-based pointing [4] is another freehand remote pointing method that allows users to select a target on a remote display by directly pointing at where they see. When a user points at an on-screen target, the remote pointing position is defined by the ray casting from the user's dominant eye to his/her fingertip. Perspective-based pointing is efficient because users are allowed to aim the target accurately with their eyes. Nonetheless, since it requires reliable face and fingertip tracking, it usually needs extra cameras and/or motion trackers in real-world deployment [1] that is too heavyweight for general usage.

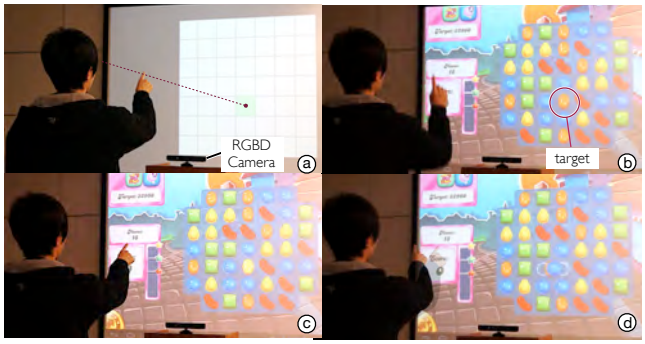

Figure 1: (a) FingerShot is a lightweight system that enable perspective-based remote direct-pointing using just one RGBD camera. In the matching game, a user can (b) lift his arm and (c) rapidly select a target without a cursor, and then (d) swipe to make a match.

1.1 FingerShot: Remote Direct-Pointing
In this paper, we present *FingerShot*, a lightweight system that uses only one RGBD camera to enable perspective-based remote direct-pointing on any deployed remote display. Our method allows users to use the Kinect sensor they already have to control the remote screen, like a SmartTV setting, and acquire the targets on it precisely and comfortably.

Figure 1 shows a matching game, *Candy Crush Saga*[1], as an application to demonstrate the usefulness and possible generalization of our technique. In this game, a user first selects a desired target by simply lifting his arm (Figure1(a)) and then pointing at it (Figure1(b)), as if throwing his finger touch to the display. Once a target is selected, the user can swipe the selected target toward its adjacent one to make the match (Figure1(c)). During playing this game, the user can freely move his position or sit down for more comfortable control. The user also can further alternatively use his another hand or even bi-manually use both hands for better performance.

[1]http://about.king.com/games/candy-crush-saga/

Figure 2: While a user is pointing to an on-remote-screen target, the pointing hand may occlude his face, affecting the tracking reliability.

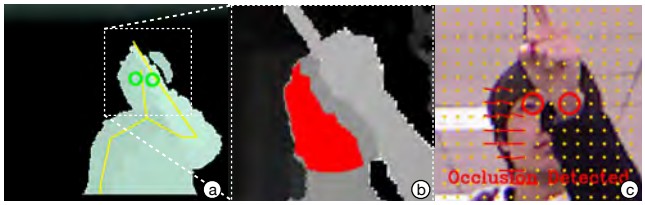

Figure 3: Occlusion-free eye-tracking. (a) The depth image and skeleton information are used to detect occlusion. (b) Selected pixels for calculating optical flow. (c) Result.

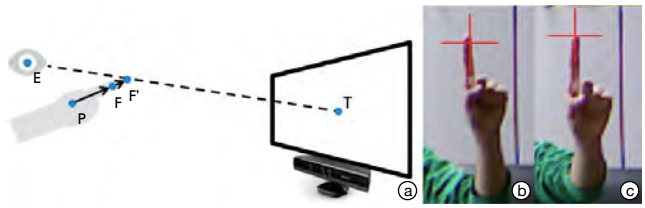

Figure 4: Real-time fingertip-tracking rectifying. (a) Overview of the model. Fingertip position (b) before and (c) after the rectifying.

The above example highlighted several promising features of the proposed FingerShot technique. First of all, FingerShot allows users to acquire remote screen objects rapidly, without wearing or holding any tracker or controller. Moreover, users can perform the selection bi-manually, which opens up new opportunities of remote bi-manual interactions. Furthermore, enabling this technique only requires one RGBD camera deploying in front of the remote display, making it more practical in real environments.

The results of a formal user study to reveal the accuracy of FingerShot are significantly (>6x) more accurate than Kinect PHIZ cursor on land-on pointing tasks in dominant-eye condition, and also significantly (>6x) more accurate in two-eyes condition. The user experiences reported are also generally positive.

In the rest of this paper, we first explain the design challenges of FingerShot, which have been examined by a pilot study. Then, we explain the design and implementation details of the proposed technique and report its usability by a formal user study. Finally, we discuss possible generalization, review the related literatures, and conclude with future research directions.

2. DESIGN CHALLENGES AND PILOT STUDY

2.1 Design Challenges

Robust eye and finger tracking are essential for realizing the perspective-based remote pointing. However, for eye tracking, the user's hand pointing at the remote display may occlude his/her eyes making the eye-tracking mechanism invalid. For fingertip tracking, since the user's fingertip is quite thin according to the distance between the user and the remote display, the exact fingertip position may not be able to be extracted precisely. These two issues thus affect the tracking reliability.

2.2 Pilot Study

To better understand these challenges, we conducted an in-lab pilot study for deeper observation and investigation on how users perform remote direct-pointing. Five participants (2 females) were recruited to use their both hands to perform remote direct-pointing tasks using a Kinect sensor, where no visual feedback is provided. The RGBD images were recorded for further analysis.

2.3 Results and Discussion

The eye tracking is performed by simply using Kinect SDK, and the occlusion problem is observed to be occured shortly before the selection is made. When the occlusion occured, the system fails to trace the users' eyes, making the results of the tracked eye positions become invalid. Nonetheless, we also observed that the duration between the occlusion occurred and the selection made is usually short, because users always move their hands quickly in order to acquire targets rapidly. Moreover, because the duration is short, the users' heads seldom substantially move during this period. These findings indicate that methods which are able to compensate the head movement during the short occlusion duration may be useful to alleviate the occlusion problem. For fingertip tracking, we performed it by superimposing the fingertip position extracted from the RGBD image streams. As a result, though the RGBD camera was calibrated well, there is still an offset between the estimated and actual fingertip positions.

3. DESIGN AND IMPLEMENTATION

Based on the above findings, we propose our solutions to the design challenges.

3.1 Occlusion-Free Eye-Tracking

Based on the findings from the pilot study, we design an optical-flow-based occlusion-free eye-tracking method to mitigate the aforementioned hand occlusion issue. Once the occlusion occurred (Figure3(a)), our system calculates the optical flow of the user's head using the nonoccluded pixels (Figure3(b)). Based on the extracted skeleton information, our algorithm first excludes the background and user's occluding hand from the depth image, and then calculates the remaining pixels' optical flows, which are mainly caused by the movement of the user's head. Based on the results, we can correct the previous valid eyes' positions (Figure3(c)) until the eyes are nonoccluded anymore.

3.2 Real-Time Fingertip-Tracking Rectifying

To rectify the error of the fingertip tracking, we adopt a machine learning approach by applying an MLP (Multi-Layer Perceptron)-based method to correct the fingertip positions. As shown in Figure4(a), F is the initial extracted 3D fingertip position which assumed to be inbetween the eye position E and the center of the pointing target T. Hence, the actual 3D fingertip position F' can be derived by calculating the intersection of the vectors $\vec{u} = \vec{PF}$ and $\vec{v} = \vec{ET}$, where P is the center of the user's palm obtained from the depth image. In the data collection phase, we then can use the

desired compensation vectors $\vec{w} = F\vec{F}'$ for training. After the traning model has been established, we can obtain an MLP regression model $M(F, P) = w$, which allows the system to rectify the fingertip position by $F' = F + M(F, P)$.

To conduct the data collection for training the MLP regression model, four male participants with a mean age of 24.0 years old were recruited. Before the training, these participants were trained with a 10-min practice section to ensure that they are able to perform the perspective-based remote direct-pointing correctly and accurately. Then, they were asked to use their right (dominant) index finger to point at an assigned cross-hair target on a remote display, and click the button on their left (non-dominant) hand to made the selection. After each selection, they need to remove the pointing hand from the remote display and press the button on the right-hand side (i.e., the pointing hand) to reset the task. The participants were asked to perform the tasks in five different positions as shown in Figure5(a), including sitting on three different positions 1.6m-away from the display (P_{center}, P_{right}, and P_{left}), and sitting (P_{rear}) and standing (P_{stand}) on the central position 2m-away from the display. For each position, each participant needs to point to fifteen predefined positions on the remote display in six times of each. For each successful trail, we record the participants' fingertip, palm, and eye positions for further analysis. We hence collected 4(participants) x 5(positions) x 15(targets) x 6(trials) = 1,800 successful trials, and used them to train an MLP regression model.

By comparing the experiment results shown in Figure 7(a) and Figure 7(b) before and after the rectifying, the accuracy can be improved significantly.

3.3 Implementing Perspective-Based Direct-Pointing

To integrate the aforementioned methods into our system, we need to extract the users' 3D eye positions, fingertips, as well as the skeleton information. While a user is standing in front of the display, his/her upper-body skeleton is first tracked. When the user raises his/her forearm above the elbow, the system starts to track the user's eyes and fingertips. If occlusion detected, the optical flows are calculated to compensate the eyes' positions. When the user moves his/her forearm toward the display, the system starts to calculate the pointing position by casting a ray from the user's eye to the rectified fingertip position. If the user steadily points at a remote screen position and dwells for a 100-ms moment, which is obtained from the observation of the pilot study, a selection can be confirmed with audiovisual feedback.

4. EVALUATION

A formal quantitative measurement is conducted to understand the accuracy on performing the remote direct-pointing using FingerShot.

4.0.1 Apparatus

A 42-inch 16:10 SmartTV (Figure 5(a)) is used as a remote display for showing graphical information in 1680 × 1050 resolution, and a Kinect sensor settled at the center bottom of the SmartTV is used as the RGBD camera for tracking the users. The processing is performed on a desktop PC with an Intel i7-3770 3.4GHz CPU and 8GB RAM for achieving the computational performance consistently at 30fps.

4.1 Experiment Design

4.1.1 Participants

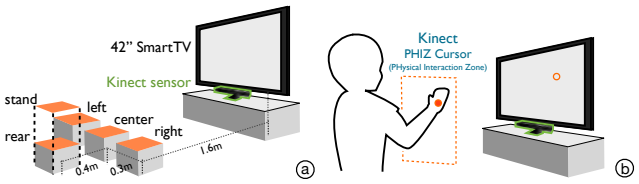

Figure 5: (a) Experimental apparatus. (b) Kinect PHIZ cursor as a body-centric cursor technique is used for comparison.

Twenty-four participants (17 males, 7 females) with a mean age of 22.8 years old were recruited for the experiment. They were evenly divided into two groups: Group A (8 males, 4 females) and Group B (9 males, 3 females). All participants were right-handed.

4.1.2 Tasks and Stimuli

This study use a mixed (within- and between- subjects) design. All participants' dominant eye was first examined in advance. Then, all participants were asked complete a series of target selection tasks by three techniques: FingerShot with the dominant (One-)Eye, FingerShot with Two-Eyes, and Kinect PHIZ cursor (Figure 5(b)). For Group A participants, they need to keep their both eyes open while performing FingerShot (with Two-Eyes). On the other hand, Group B participants need to keep their non-dominant eye closed (FingerShot with One-Eye).

In each task, participants were asked to use their right (dominant) index finger to point at an assigned cross-hair target on the remote display, and click the button on their left (non-dominant) hand to made the selection. After making the selection, the system will give the participant a feedback to indicate a successful selection. After each selection, they need to remove the pointing hand from the remote display and press the button on the right-hand side (i.e., the pointing hand) to reset the task. For further evaluating FingerShot, the participants were asked to perform the tasks in five different positions as defined for training the MLP regression model (Figure 5(a)). For each position, each participant needs to point to fifteen predefined positions on the remote display in six times of each. The sequence of targets is randomized. To remove learning effects, we prepare practice sessions to make sure that every participant can perform each technique well on the smallest targets, and start the testing session only if they felt ready and proceeded to continue. To remove fatigue effects, the participants can pause the tasks anytime. Finally, we collected 5(positions) x 15(targets) x 6(trials) = 450 successful trials on FingerShot, and 1(position) x 15(targets) x 6(trials) = 90 successful trials on Kinect PHIZ cursor from each participant.

In the end of the study, participants were asked to rank FingerShot and Kinect PHIZ cursor from very agree (5) to very disagree (1) according to three different criteria: ease of learn, comfort of manipulation, and independence of visual feedback. A short interview were also performed to provide early user feedback.

4.1.3 Measures

The dependent variable of our designed tasks is the distance between the pointing position and the center of the assigned cross-hair target, and the independent variables are the three techniques and five different positions that were presented to each participant in counterbalanced order. In practice, the target on a remote display cannot be just one pixel, but it also cannot be too large to occupy the whole screen. Hence, if the technique is more accurate, the target

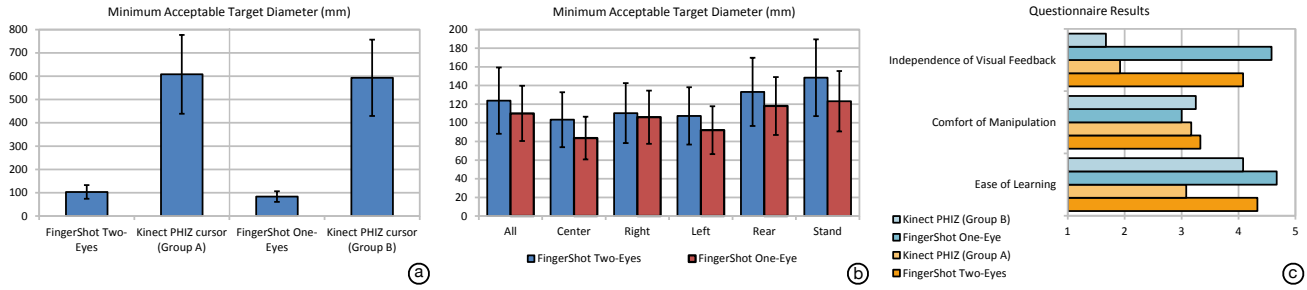

Figure 6: User study results. Minimum target diameters that can encompass 95% of the selections on (a) different techniques and (b) different positions. (c) Questionnaire results.

can be designed smaller. Consequently, to compare the accuracy of the three techniques, we can compare their minimum acceptable target size, which is defined as the minimum target diameter that can encompass 95% of the selections.

4.1.4 Hypothesis

Since FingerShot (both the Two-Eyes and One-Eye versions) allow users to aim at the targets using their eyes, which would be more precise than aiming at the targets just using their kinesthetic memory only. Performing FingerShot with Two-Eyes may also experience the binocular parallax problem. Besides, since FingerShot determines the touch position by the ray casting from the user's eye(s) to the fingertip, the error would be scaled with the distance between the user and the remote display. Hence, our hypothesis are:

H1. Using FingerShot (both the Two-Eyes and One-Eye versions) can have significantly smaller target size than using Kinect PHIZ cursor.
H2. Using FingerShot with One-Eye can have significantly smaller target size than using that with Two-Eyes.
H3. Using FingerShot can have significantly smaller target size when the user's position is closer to the display.

4.2 Results

We first examine the minimum acceptable target sizes (diameters) on P_{center} of all techniques. The result is shown in Figure 6(a). Pairwise *t*-test shows that the minimum acceptable target diameters of both FingerShot Two-Eyes and FingerShot One-Eye are both significantly ($p < 0.01$) smaller than that of Kinect PHIZ cursor. The minimum acceptable target diameters of FingerShot Two-Eyes and FingerShot One-Eye are 103.3mm and 83.6mm, respectively, which can support 9×5 and 11×6 gridded buttons on a 42-inch remote display, respectively. The experiment results of FingerShot (with Two-Eyes) and Kinect PHIZ cursor are shown in Figure 7(b) and Figure 7(c), respectively. Obviously, FingerShot (with Two-Eyes) outperforms Kinect PHIZ cursor a lot. Hence, *H1* is strongly supported.

We then examine all results on all five positions of both FingerShot Two-Eyes and FingerShot One-Eye. The result is shown in Figure 6(b). Welch's *t*-test shows that the minimum acceptable target diameters of all FingerShot One-Eye are significantly ($p < 0.01$) smaller than those of FingerShot Two-Eyes. Hence, *H2* is supported.

We further analyze the results shown in Figure 6(b). One-way repeated-measure ANOVA found a significant effect of different positions in both FingerShot with Two-Eyes ($F(4,44)=15.25$, $p < 0.01$) and FingerShot with One-Eye ($F(4,44)=35.44$, $p < 0.01$). Pairwise *t*-test with Bonferroni correction shows that the minimum acceptable target diameters of both FingerShot Two-Eyes and FingerShot One-Eye are both significantly less accurate on P_{rear} and P_{stand} than on P_{center}, P_{right}, and P_{left} (all $p < 0.01$). Hence, *H3* is supported.

4.3 Questionnaire and User Feedback

The questionnaire results (Figure 6(c)) are discussed with the gathered user feedback.

On *ease of learning*: Wilcoxon signed rank test shows that learning FingerShot is significantly easier than learning Kinect PHIZ cursor in both Group A ($p < 0.01$) and Group B ($p = 0.0418 < 0.05$). Kruskal-Wallis test shows that there is no significant difference ($p = 0.1882$) between FingerShot Two-Eyes and FingerShot One-Eye. Participants generally reported that it is hard to aim the remote on-screen targets using Kinect PHIZ cursor without cursor. One participant further pointed out that the reason could be the different aspect ratio between the display and the physical interaction zone. Nonetheless, one participant who rated FingerShot Two-Eyes in lower score reported that his dominant eye tends to change during selecting targets using FingerShot, because the pointing finger occluded the remote targets.

On *comfort of manipulation*: Both Wilcoxon signed rank and Kruskal-Wallis tests show that there is no significant difference between FingerShot Two-Eyes, FingerShot One-Eye, and Kinect PHIZ cursor (all $p > 0.05$). Some participants rated FingerShot Two-Eyes in lower score because of the fatigue due to visually aiming the remote targets. Some participants rated FingerShot One-Eye in lower score because of the fatigue due to closing one eye. However, participants generally agree that FingerShot really alleviate the fatigue problems of their arm according to Kinect PHIZ cursor.

On *independence of visual feedback*: Pairwise *t*-test shows that using both FingerShot Two-Eyes and One-Eye need significantly less ($p < 0.01$) visual feedback than using Kinect PHIZ cursor. Participants generally agree that, without a cursor shown on the screen, they still can acquire the remote targets on the display. Without visually checking the cursor's position, selection becomes faster as well. Some participants further reported that providing some auditory feedback while selection would be appreciated. A participant reported that FingerShot provides a sense of "throwing" his finger touch to the position where he is looking at on the remote display.

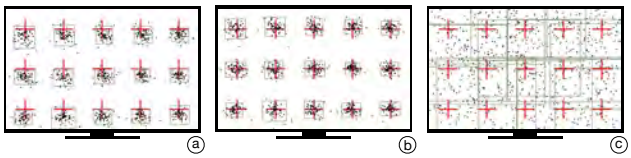

Figure 7: Spread of experiment results of different techniques on P_{center}. Data are plotted with the mean value and the covered area of two standard deviation. (a) FingerShot with Two-Eyes without fingertip rectifying. (b) FingerShot with Two-Eyes with fingertip rectifying. (c) Kinect PHIZ Cursor.

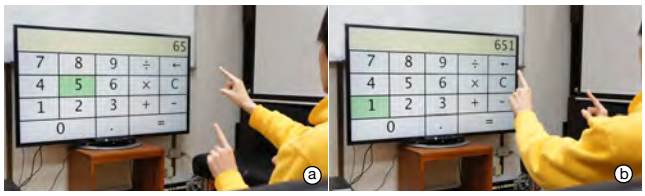

Figure 8: In the calculator example, a user can rapidly enter the numbers and operators without an on-screen cursor, and even bi-manually.

5. POSSIBLE APPLICATIONS AND GENERALIZATION

In this section, we discuss the possible applications and generalization of FingerShot.

Rapid land-on selection. As a remote direct-pointing technique with sufficient accuracy, FingerShot would be potentially suitable for typing. As the calculator example shown in Figure 8. A user can directly point to the remote display to type without using a cursor. He/she can also use his/her both hands to enter the numbers efficiently.

Combing with sliding widgets. As the color matching game example shown in Figure 1, a user can directly perform a swiping gesture after pointing to a target. This makes FingerShot highly suitable for incorporating with sliding widgets [3], and can enrich the input vocabulary of FingerShot.

Precise pointing with absolute+relative cursor. In some applications that require high-precision control, such as the map application shown in Figure 9, a cursor may be needed. Given the current accuracy of FingerShot, a user can first place an absolute+relative cursor very close to the target in the beginning (Figure 9(a)). Then, the user can manipulate an low control-display ratio cursor toward a target precisely, and dwell for a short while to make the selection (Figure 9(b)). In sparse-target condition as shown in Figure 9(c), a user can even activate a bubble cursor mode [2] by a gesture, thereafter the user can move toward another target with shorter correction movement.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented *FingerShot*, a practical perspective-based remote direct-pointing technique using only one RGBD camera. Occlusion-free eyes-tracking and real-time fingertip-tracking rectifying techniques are proposed to resolve the challenges. Results of the evaluation shows that FingerShot significantly outperformed Kinect PHIZ cursors (represented as a body-centric cursor method).

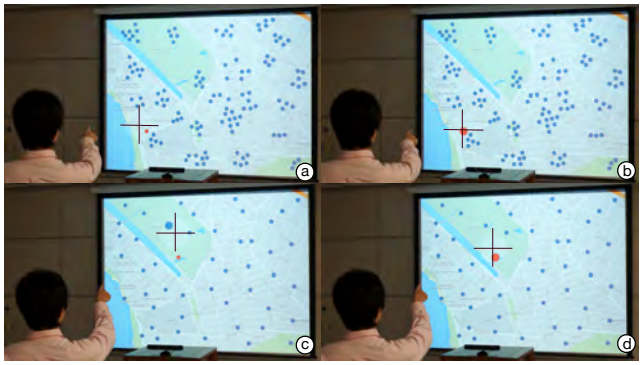

Figure 9: Precise selection in a map application. A user (a) first points to the position he wants to magnify, and then (b) moves the low C/D cursor to make the selection. (c) In sparse-target condition, a user can activate a bubble cursor to ease selection by (d) reducing the correction movement.

Moreover, since FingerShot One-Eye outperformed FingerShot Two-Eyes on accuracy, users can close one eye for precise remote direct-pointing if needed. The limitation on pointing distances and possible generalization are also reported and discussed.

Future research can consider bringing auditory or haptic feedback to help users get better senses of touching an invisible panel for effective interactions. In addition, given its independency of visual feedback, future work can also consider applying FingerShot technique in a display-less condition, such as controlling a smart furniture or electronic appiliances by remote direct-pointing, which would be a promising way of interacting with computers in the coming era – Internet of Things.

7. 致謝

本論文感謝科技部經費補助，計畫編號：NSC101-2219-E-002-026。

8. REFERENCES

[1] A. Banerjee, J. Burstyn, A. Girouard, and R. Vertegaal. Multipoint: Comparing laser and manual pointing as remote input in large display interactions. *IJHCS*, 70(10):690–702, 2012.

[2] T. Grossman and R. Balakrishnan. The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proc. CHI '05*, pages 281–290, 2005.

[3] T. Moscovich. Contact area interaction with sliding widgets. In *Proc. UIST '09*, pages 13–22, 2009.

[4] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, and M. R. Mine. Image plane interaction techniques in 3D immersive environments. In *Proc. I3D '97*, pages 39–43., 1997.