透過魚眼影像偵測手部及與環境互動的戒指型裝置

詹力韋* 謝琦皓† 陳奕麟‡ 梁容豪‡ 陳炳宇‡
*慶應義塾大學 ‡‡國立臺灣大學
*‡{liweichan,yilingchenntu,rhliang,robin}@ntu.edu.tw
†p121225@cmlab.csie.ntu.edu.tw

ABSTRACT
我們提出了基於魚眼相機的戒指型裝置，此裝置穿戴於手指的邊緣且可偵測手以及與影像內容有關的互動，從手指邊緣得到的以手為中心的魚眼影像可以用來偵測手勢還能使得手指和手掌的區域變成觸碰介面，同時因魚眼影像的關係，使用者可以透過手勢與環境中的物體做互動，除此之外由於此裝置是戒指型的穿戴裝置讓使用者保有手部皮膚的回饋。在此論文中我們提出一個概念型證明的裝置和使用隨機決策森林(randomized decision forests)做手勢偵測的辨識率以及包含滑桿(slider)輸入和在手掌寫字(palm-writing)輸入等互動的技術與使用這些技術和環境互動，我們的實驗包含7個手勢並找了15個受測者，手勢的辨識率為84.75

Categories and Subject Descriptors
H.5.m [**Information Interfaces and Presentation (e.g. HCI)**]: Miscellaneous;

General Terms
Design; Research.

1. INTRODUCTION
Human hands are quite powerful as natural user interface for many tasks. To achieve this, camera sensors are placed on different body locations [1, 4, 10, 11] to enable rich and continuous hand-based interactions at everywhere. Among them, Digits [4] using a wrist-worn camera is able to reconstruct the whole 3D hand structure in realtime. However, it has the limitation that the camera has to be elevated in order to observe the user's hand from the wrist position, causing it difficult to be made into miniature form.

By contrast, recent researches on wearable interface dedicated to wrist-worn wearables used various low-level sensing techniques, such as measuring changes in muscles [8], capacitances [7] [9], wrist contour and surface pressures around the wrist. However, it is difficult for low-level sensing techniques to support continuous interactions.

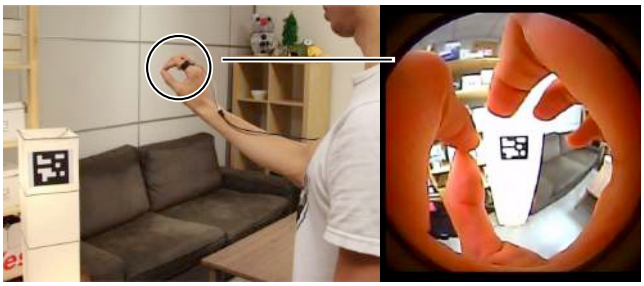

Figure 1: HandSight supports whole-hand and context-aware interactions, allowing users to interact with real-world objects such as the smart lamp by pinch-and-motion input.

Among various wearable forms, ring is widely perceived attractive and challenging owing to its minimal form. Unfortunately, previous researches on ring wearables could merely augment partial input functions, such as tapping on the ring [6], motion input through the wearing finger, and finger mouse at any surface [3]. Other ring wearables allow the finger to interact with real-world elements, such as reading text [5] and textures [2] under the users' finger touches. Until now, there is no wearable device in the form of a ring capable of supporting whole-hand touch and gesture interactions.

1.1 HandSight
We present HandSight, a ring-style fisheye imaging device that is worn at a user's hand webbing. By observing from a central position of the hand through the hand-centric fisheye field-of-view, HandSight is able to observe the whole frontal skin region of the hand, thus allowing to turn the skin region into an interactive surface. Benefiting from the fisheye field-of-view, HandSight further allows to incorporate real-world elements into hand-based interactions. Finally, owing to the form factor as a ring, HandSight preserves skin haptic feedback for the enabled hand-based interactions.

Figure 1 illustrates a scenario that a user wearing HandSight is able to interact with a smart lamp by pinch-and-motion input. We demonstrate a proof-of-concept prototype and a RDF-based algorithm for classifying pixels in the hand-centric fisheye images into different finger labels (*e.g.*, thumb, index finger etc.). Our first experiment demonstrates the RDF-based pixel classification achieves 84.75% recognition rate of hand gesture input from a database of 7 hand gestures collected from 15 participants, suggesting the effectiveness of the fisheye images for rich hand-based interaction.

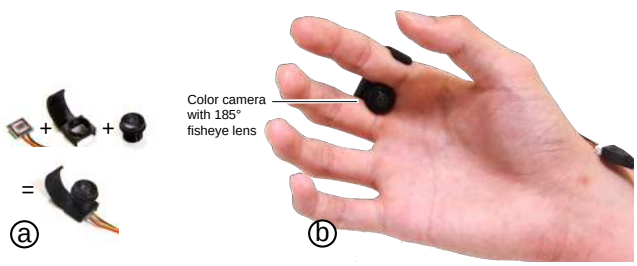

Figure 2: (a) HandSight comprises a miniature 185-degree fisheye camera that is hold by the edge of a 3D printed ring. (b) Wearing it on fingers can position the device at certain hand webbings.

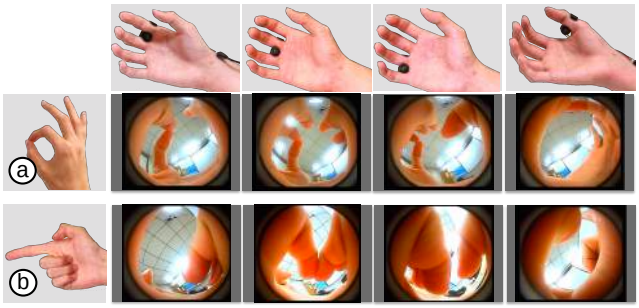

Figure 3: Placement of the ring in different hand webbings.

1.2 Contribution

The main contribution of this paper is the concept of enabling whole-hand and context-aware interactions using a fisheye hand-centric view of a user's hand. To demonstrate the idea, this work (1) presents a proof-of-concept prototype, (2) demonstrates the potential of HandSight for hand gesture input using random decision forest (RDF) method, and (3) presents a set of whole-hand and context-aware interactions.

2. HARDWARE PROTOTYPE

The main component of HandSight is a miniature fisheye camera worn as a ring that sees the frontal skin region of the user's hand. In the following, we demonstrate the components of our hardware prototype, and present the variation of wearing the HandSight at different hand webbings and the benefits of our choice to wear it in index finger.

2.1 Fisheye Ring Devices

Figure 2 shows the hardware prototype. The prototype consists of a miniature camera with a mini-fisheye lens, which has a focal length of 1.2 mm and an aperture of F1.8, allowing for 185-degree field-of-view and its physical size is measured 14 mm in diameter and 15 mm in height. To wear the fisheye camera as a ring, we design and 3D print a ring that holds the fisheye camera by the edge of the ring such that when users put on the ring would position the fisheye camera at a hand webbing as shown in Figure 2b.

This unusual design of the ring wearable is to compensate for the relatively large of the fisheye lens (e.g., 14 mm in diameter) in our prototype while allowing to position the lens as close as possible to a central location of the hand, thus maximizing the ability to see the skin regions of the fingers and palm. It is predictable that the fisheye lens will be greatly shrunk with professional camera optical engineering to let off the compromise in the form design. As a reference, the NanEye camera from AWAIBA[1], despite still far from our requirement in viewing angles, allows 120 degree field-of-view and measures 1.0 mm x 1.0 mm x 1.7 mm in three dimensions.

2.2 Placement of the ring device

Figure 3 displays the observed images of user performing two gestures from different hand webbings by wearing it at the thumb, index, middle, and ring fingers, respectively. The device at different webbings would see considerably different views of the same hand gesture.

In the example of 'Pinch' gesture (Figure 3a), the views from the index, middle, and ring fingers can observe the circle formed by the index and thumb at various distances, but they can barely see the straight fingers. In comparison, the device at the thumb cannot see the circle but is able to clearly observe the straight fingers as if it could count the fingers for knowing the gesture types. In the 'GUN' gesture (Figure 3b), the device at the middle and ring fingers are mostly occluded by the curling fingers. The views from wearing at the index and thumb fingers are also greatly different.

Our current implementation still requires substantial space (1 cm by square) to accommodate the device on users' fingers. We choose to put the device on the webbing of the index and middle fingers by wearing the device on the index finger (Figure 2b), because the index finger is more flexible to yield a space in the webbing for the device without affecting users to perform gestures.

3. INTERACTION TECHNIQUES

Based on HandSight's capability in hand gesture recognition, we further implement interaction techniques including on-finger, on-palm touch interactions, and hand gesture interaction with real-world objects. Each of the interaction techniques is realized with some heuristics by taking advantages of hand-centric fisheye images.

To allow users to switch among various interaction techniques, we adopted the gesture lock-in approach. By detecting the pre-defined hand gestures, HandSight locks in an individual interaction technique and starts to apply the corresponding heuristics, or unlocks from the current interaction. We defined the open hand as the unlock gesture because it is naturally performed when users' hands finish an input and relax. To avoid accidental inputs, HandSight only switches to a new interaction technique from the unlock state.

3.1 On-Finger Pinch-and-Slide Input

Here, users are allowed to perform finger pinches on each of the fingertips, and further adopt the fingers as function sliders. This interaction involves eight lock-in gestures, where each of the four fingers contains two lock-in gestures: thumb-to-finger pinch at the tip and the middle point of the finger. The unlock gesture is defined as open hand.

Figure 4 illustrates the mechanism of gesture locks during the interaction. HandSight determines at which finger the pinch gesture is performed according to the recognized lock-in gesture. Once it

[1]http://www.awaiba.com/product/naneye/

Figure 4: The process of gesture lock for on-finger pinch-and-slide input. From left, the user performs the interaction on the index finger (from unlock to lock state), and releases the pinch (return to unlock state).

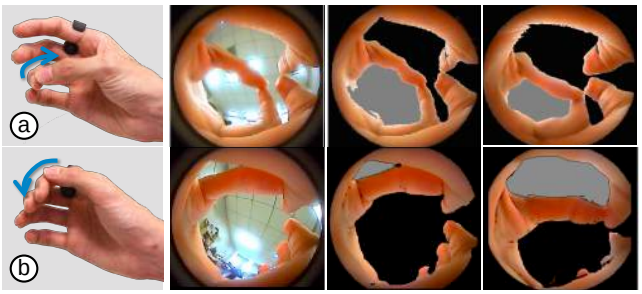

Figure 5: The interaction of multiple on-finger sliders. (a) Decreasing the value at the ring finger slider by moving from the tip, and (b) increasing the value at index finger slider by moving to the tip from a middle point.

detects an lock-in gesture from an unlock state, HandSight is locked in the pinch interaction of that finger. As displayed in Figure 4ab, the interaction is locked in the index-finger-pinch gesture. Then, under the interaction state, we enable finger-slider input by looking for an relatively large enclosed component formed by the pinch gesture in the fisheye images, as shown in Figure 4c. The changes in the component's area size is applied to the changes of slider values. Whenever users release a pinch gesture, the unlock gesture (e.g., open hand) is detected and start to detect next lock-in gesture (Figure 4d).

Figure 5 further demonstrates the image processing techniques and heuristic while users move the thumb along the ring and index finger sliders. When there are two large components appeared, which often happens at non-index-finger pinch gestures where the fake component is accidentally formed by the index finger in fisheye's perspective as shown in Figure 5a, we simply pick up the one whose centroid is on the left. To increase the slider value, the middle-point pinch gesture allows users to pinch at the middle finger segments and moving toward the tip. Figure 5b shows an example on the index finger.

3.2 Palm-Writing Input

Owing to the fisheye view, the skin region in users' palm is partially observable with flat hand posture and is fully observable with half-curved hand posture. This allows us to turn the palm region into a touchpad where users can write with their fingers or pens.

This interaction involves one lock-in gesture: thumb-bent gesture. Figure 6 illustrates the process of gesture lock for palm-writing input. Once HandSight locks in the interaction, we take the skin region in first image frame as the background skin region and enable palm-writing input as follows.

Figure 6: The process of gesture lock for palm-writing input. From left, the user performs palm-writing gesture (from unlock to lock state) to activate the interaction, and straighten the thumb (return to the unlock state) to finish interaction.

Users can write on the palm with their fingers or color-capped pens, as shown in Figure 7. By default, a finger-pen mode is assumed. For each input fisheye image, we determine whether the user is writing with a color pen by searching for a non-skin color blob appearing in the area defined by the background skin region. If a non-skin blob is detected, a color-pen mode is activated and the blob color is recorded.

For finger-pen mode, we detect fingernails using Adaboost method. All possible fingernails are firstly identified and filtered with the area defined by background skin region (Figure 7a). Then, the foreground skin region is identified by subtracting the current skin region with the background skin region (Figure 7d). Note that the foreground skin usually contains the user's writing hand in the lower part, and the lowest line as highlighted is good indicators to where the real fingernail might locate. Therefore, we identify the fingernail below and closest to the line (Figure 7c). We only take the x coordinate of the fingernail position in the fisheye image, and discard its y value because dynamic range in y axis is greatly condensed in the fisheye perspective. To obtain a better substitution for y coordinate, we adopted the size of the foreground hand which is considered the lower foreground component in the foreground skin. The initial foreground size is by default set as the baseline (e.g., the zero value) for the y coordinate.

For color-pen mode, the non-skin color blob in the background skin region indicates where the pen tip locates. Similarly, we only use the x coordinate of the tip position. The size of the color pen body, which is further extracted by region filling from the pen tip in the fisheye image, allow for an greater dynamic range for the y coordinate.

Note that the unit in X and Y coordinates defined above for each of the finger-pen and color-pen mode are different. To compensate the mismatch, we rescale the Y coordinate emperically by 1.2 in finger-write mode and by 0.9 in pen-write model in our implementation.

3.3 In-Air Pinch-and-Motion Input

Following the finger pinch recognition, HandSight also supports in-air pinch-and-motion input. Figure 8 illustrates the process of gesture lock. This interaction involves four lock-in gestures: thumb-to-finger pinches at each of the fingertips except the thumb. Pinching at different fingers allows to increase the input modality.

As shown in Figure 9, once HandSight locks in the interaction, we enable motion input by calculating the moving direction from paired SURF features in the consecutive undistortion fisheye images. Each pair of the features in two consecutive images contributes a candidate displacement. To obtain reliable moving direction, we remove the displacements which are outside standard deviation in

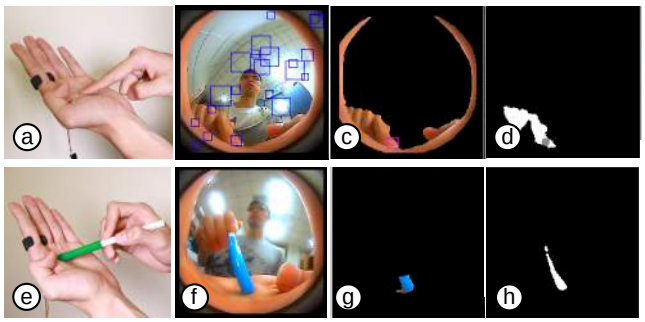

Figure 7: The interaction of palm-writing input. (a) The user writes on the palm with the index finger. (b) Bending the four fingers indicates that the written messages were displayed in protected mode, such as entering password on a public screen. (c) The user can write with different colors using color pens.

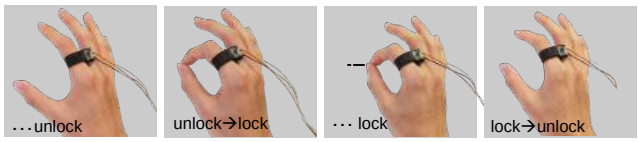

Figure 8: The process of gesture lock for in-air pinch-and-motion input. From left, the user performs pinch gesture (from unlock to lock state), moves to write a stroke in air, and releases the pinch (return to unlock state) to complete the stroke.

magnitude or direction. Then, the average displacement from the remaining candidates is adopted. The motion stroke is formed by aggregating all the displacements until users release the pinch.

3.4 Context-aware interaction techniques

In addition to hand gestures and motions, HandSight allows the input to interact with real world objects such as smart appliances visible to the fisheye images during interaction. The interaction with real world objects can be realized by common computer vision techniques such as object recognition using AR tags or feature matching, face detection, and natural feature tracking.

3.4.1 Pinch-and-motion input with object recognition

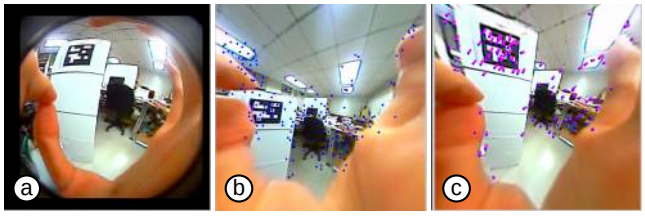

Figure 9: The interaction of in-air pinch-and-motion input. (a) The user performs the pinch gesture on the index finger. (b) The SURF features is computed in the undistortion image. (c) The displacements of paired features with the last image frame.

Figure 10 demonstrates three example interactions with the environment. In Figure 10a, the user performs pinch gestures toward a smart lamp. From the fisheye view, the HandSight recognizes which appliance appears under the user's pinch by detecting the AR tag on the lamp. The user then write a "tick" in air to turn on the lamp by moving the pinch gesture, and releases the pinch to deliver the operation. Note that the AR tag can be replaced with object recognition techniques using natural features such as SURF.

3.4.2 Pinch gestures with face detection

With face recognition, users interact with people nearby using pinch gestures. In Figure 10b, the user sends digital files to a friend in front of him. From the glass display, he sees the undistortion version of the HandSight's view and the faces therein were highlighted. The cross displayed at the center of the image allows the user to aim a face by moving the pinch gesture. Releasing the pinch, the interaction is delivered to the person after face recognition.

3.4.3 Finger copy functions

In Figure 11, the user copies an image on a paper by dragging the finger across the image. To realize the interaction, we first identify the position of the index fingertip that seemed in contact with the paper. This is achieved by searching for the pixel with greatest x coordinate in the skin region of the left-upper quarter in the undistort images (Figure 11d). Note that we can define the upper-left quarter as region-of-interest due to the fact that the hand-centric view allowed by HandSight maintains a relatively consistent spatial arrangement of fingers associated with same gestures.

To determine the user's finger stroke on the paper (Figure 11f), we compute homography transformations between every two consecutive undistort images (Figure 11e), by identifying at least four pairs of SURF features therein. We can obtain the homography transformation because the paired SURF features are laying on the same planar surface. The homography transformations allow to re-project all the fingertip positions along the finger stroke onto the first undistortion image where the stroke was initiated.

With the stroke, we are ready to extract the content-of-interest on the paper. While the stroke seems nicely capturing the diagonal line of the content-of-interest, we can not simply take the rectangle defined by the diagonal line unless the HandSight squarely look at the paper. To rectify the image, we apply a homography transformation, H_o, which is obtained in advance to capture how HandSight inclines to a planar surface, such as, for the finger-copy function. For example, H_o can be used to reproject the four corners of a square on paper appeared in the HandSight view of the finger-copy gesture. By applying Ho, the content-of-interest is rectified (Figure 11g) and can be extracted by the rectangle of the diagonal line (Figure 11h). Note that the homography transformation only needs to be computed once as long as users perform the interaction (e.g., approaching a planar surface) in similar ways.

4. EXAMPLE APPLICATIONS

We demonstrate three categories of applications related to whole-hand and context-aware interactions.

4.1 Whole-hand interactions

4.1.1 Gestural interaction for virtual reality

Rich input modalities are desirable for immersive visual environment such as virtual reality (VR). Hand gestures are in particular attractive in VR gaming as players can take advantages of metaphors

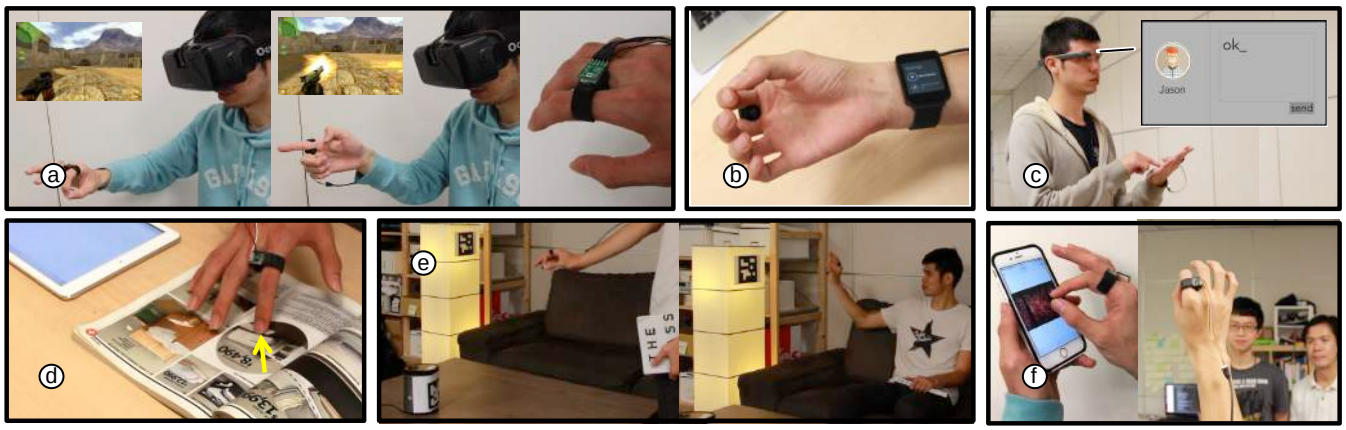

Figure 12: Example applications of the HandSight device for gestural interaction in virtual reality, single-handed interaction,.

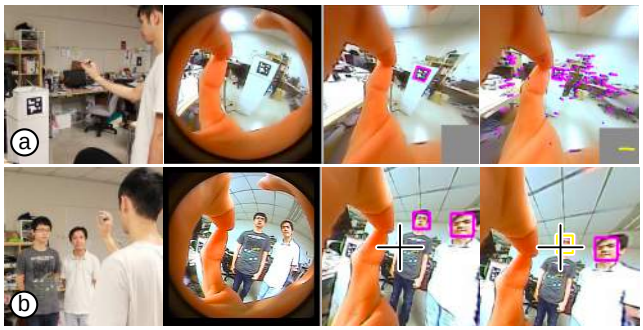

Figure 10: More context related functions can be incorporated with in-air pinch-and-motion input. (a) Direct interaction with an smart lamp through an AR tag. (b) Pick-and-drop information to a person via face detection.

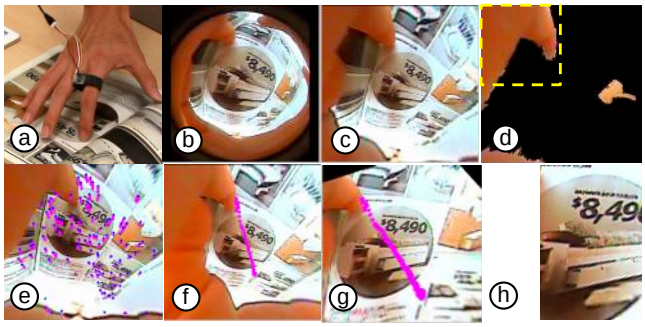

Figure 11: Finger copy of an image on the physical paper. (a) Photo taken from third-person view. (b) The raw fisheye image. (c) Undistort image. (d) Fingertip position. (e) Paired SURF features. (f) The finger stroke and (g) after rectification.

in various hand gestures to recall rich game functions. In a first-person shooting gam demonstrated in Figure 12a, the player form a 'GUN' gesture to recall the pistol barrel. Triggers a gun shoot by bending down the thumb which is defined as different gesture. Pinching with index finger retrieves a grenade. To incorporate more interactivity, we augmented the HandSight with a 6 DOF IMU to allow aiming with pistol, and detect throwing grenades.

4.1.2 Single-handed interaction for smart watches

Current interaction on smart watches is designed around touch input, which requires users' both hands during interaction. In Figure 12b, on-finger pinch-and-slider input enabled by HandSight allows users to perform rich operations such as setting clock alarm on watches single-handedly.

4.1.3 Palm-Touch Interaction for glass displays

We implemented the user's palm into a remote touch pad for glass displays. Instead of reaching to the glass or an external touch pad for input, the user simply writes on the palm to touch control the glass display. As shown in Figure **??**c, the user navigates and replies messages with palm-writing input.

4.2 Context-aware interactions

4.2.1 Real-world Clipboard

HandSight allows to bridge the digital document and its physical printout. In real-world clipboard, the user drags the index finger cross the region of interest, say a photo, on the physical paper (Figure 12d). The content in the specified region is clipped digitally, and directly passed to the pad beside. Again, this interaction is implemented with finger-copy functions.

4.2.2 Pinch Into the Context

HandSight sees what the user is intended to reach to with the pinch gesture, allowing an intuitive way to associate pinch input with the real world objects. In smart home, users reach out an smart device with the pinch gesture and start stroke input in air to control the smart device over air (Figure 12e). In the example of pick-and-drop with persons (Figure 12f), users pick up an digital content in the smart phone with the pinch gesture, reach out the pinch gesture toward a person in face, and drop the content by releasing the pinch gesture such that the content is sent to that person.

5. 致謝

本論文感謝科技部經費補助，計畫編號：MOST103-2218-E-002-024-MY3與MOST103-2218-E-002-014。

6. REFERENCES

[1] G. Bailly, J. Müller, M. Rohs, D. Wigdor, and S. Kratz. ShoeSense: A new perspective on gestural interaction and wearable applications. In *Proc. ACM CHI '12*, pages 1239–1248, 2012.

[2] L. Jing, Z. Cheng, Y. Zhou, J. Wang, and T. Huang. Magic ring: A self-contained gesture input device on finger. In *Proc. ACM MUM '13*, pages 39:1–39:4, 2013.

[3] W. Kienzle and K. Hinckley. Lightring: Always-available 2D input on any surface. In *Proc. ACM UIST '14*, pages 157–160, 2014.

[4] D. Kim, O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis, and P. Olivier. Digits: Freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proc. ACM UIST '12*, pages 167–176, 2012.

[5] S. Nanayakkara, R. Shilkrot, K. P. Yeo, and P. Maes. Eyering: A finger-worn input device for seamless interactions with our surroundings. In *Proc. ACM AH '13*, pages 13–20, 2013.

[6] M. Ogata, Y. Sugiura, H. Osawa, and M. Imai. iRing: Intelligent ring using infrared reflection. In *Proc. ACM UIST '12*, pages 131–136, 2012.

[7] J. Rekimoto. GestureWrist and gesturePad: Unobtrusive wearable interaction devices. In *Proc. ISWC '01*, pages 21–27, 2001.

[8] T. S. Saponas, D. S. Tan, D. Morris, R. Balakrishnan, J. Turner, and J. A. Landay. Enabling always-available input with muscle-computer interfaces. In *Proc. ACM UIST '09*, pages 167–176, 2009.

[9] M. Sato, I. Poupyrev, and C. Harrison. Touché;: Enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proc. ACM CHI '12*, pages 483–492, 2012.

[10] T. Starner, J. Auxier, D. Ashbrook, and M. Gandy. The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In *Proc. ISWC '00*, pages 87–94, 2000.

[11] E. Tamaki, T. Miyaki, and J. Rekimoto. Brainy hand: An ear-worn hand gesture interaction device. In *Proc. ACM CHI EA '09*, pages 4255–4260, 2009.