공학석사 학위논문

TrackLens: Using Rear-Camera as TrackBall-like Input Unit for Single-handed Interaction on Handheld Devices

내장 후면 카메라를 터치 입력 장치로 활용한

휴대용 기기의 한 손 인터랙션 디자인

2013년 8월

서울대학교 대학원

전기•컴퓨터 공학부

고 재 희

TrackLens: Using Rear-Camera as TrackBall-like Input Unit for Single-handed Interaction on Handheld Devices

Advisor Jinwook Seo

Submitting a Master's Thesis of Engineering

August 2013

Graduate School of Seoul National University

School of Computer Science and Engineering

Seoul National University

Jaehee Ko

Confirming the master's thesis written by Jaehee Ko

July 2013

Chair ______Myung-Soo Kim______(Seal)

Vice Chair ______Jinwook Seo______(Seal)

Examiner ______Jehee Lee______(Seal)

Abstract

TrackLens: Using Rear-Camera as TrackBall-like Input Unit for Single-handed Interaction on Handheld Devices

Jaehee Ko

School of Computer Science and Engineering

The Graduate School

Seoul National University

Touchscreens on mobile devices usually require using both hands to efficiently interact with it. However, there are many occasions in daily life that only one hand is available to control the devices. In this paper, we present a single-handed interaction technique (TrackLens) on handheld devices using embedded rear-camera. TrackLens enables users to perform on-the-lens gestures using index finger and precise target selection using both thumb and index finger. It not only resolves the problem of single-handed interaction on mobile devices, but also deals with ingrained problems of touchscreens including no tactile feedback and occlusion. A controlled user study shows that, although the gesture classifier needs some improvement, TrackLens interaction technique can help users control their handheld devices more delicately and comfortably with only one hand.

Keyword : Single-handed interaction, touchscreen, mobile device, camera interaction, gesture, target selection

Student Number : 2011-20783

Content

List of Figures

1. INTRODUCTION

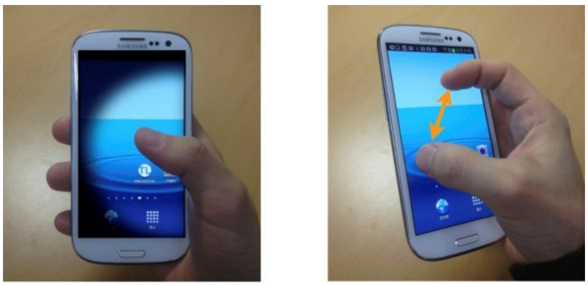
Touchscreen mobile devices are becoming increasingly popular these days. Touchscreen enables mobile devices to be more compact and lighter by integrating input space and output space into one. However, touchscreens on mobile devices have some drawbacks, especially on handheld devices when used with only one hand. In daily lives, there are times when only one hand is available to control them, e.g. when holding a bag in one hand on a street or when holding a strap in one hand on a bus.

First, there are shadow areas on screen (Figure 1a) that a thumb cannot reach due to the limitation of thumb's operational range when used with only one hand [11]. The shadow areas also include the bottom part of a device where hardware function buttons are usually located. Second, it is difficult and uncomfortable to use multi-touch interaction like pinch-to-zoom and two-finger-tapping with one hand (Figure 1b).

These problems of touchscreens on mobile devices when used with one hand are getting worse since the screen size of mobile devices is getting larger and larger. Most of smartphones released these days are equipped with even larger than 4-inch screen. Sometimes it is even impossible to perform multi-touch interaction because the size of device is too big to use with one hand.

Moreover, there are intrinsic problems of touchscreens. Touchscreens do not provide tactile feedback, and thus users do not know where exactly they are touching. This problem often makes users lose orientation of contact point, and thus target selection, especially for small targets, becomes more error-prone [14]. Occlusion by finger and hand when selecting a target on screen also a well-known problem of touchscreens.

In this paper, we present a single-handed interaction technique on handheld devices called TrackLens. TrackLens utilizes rear-camera embedded in mobile devices as a sort of touch sensor that detects finger contact on camera lens and recognize finger movement on it. TrackLens enables users to interact with their mobile devices with only one hand, supporting more precise target selection, and more secure grip of device while reducing fatigue of hand and wrist [3]. TrackLens technique can be enabled in any mobile devices equipped with rear-camera because it does not require any additional devices or accessories.



a) shadow area b) pinch gesture

Figure 1. Using a mobile device in one hand

2. RELATED WORK

TrackLens is a single-handed gesture and pointing interaction technique that uses rear-camera embedded on backside of mobile devices. It is related to previous researches on single-handed touch interaction, interaction on backside of devices, and interaction using embedded cameras.

2.1 Single-handed Interaction using Touch

Many researches have been actively done on methods to increase utilization of touchscreens with one hand since the adoption of touchscreens in mobile devices. Direct Touch is the basic input method for touchscreens where a direct tap on an object selects it. It is an intuitive and fast way for target selection, but targets on unreachable area are hard to select with Direct Touch. Moreover, its selection precision is low when selecting small targets due to occlusion and the fat finger problem [1, 20].

Offset cursor is a technique that shows a cursor at a small constant distance away from finger contact point. It removes occlusion of target object with finger and increases precision for selecting small targets. But selection time with the offset cursor is higher than that of direct touch and there are areas that the offset cursor cannot cover [15, 19]. Adaptive Offset cursor can cover the whole screen by gradually increasing the offset from the center of screen [8]. It has a similar effect of having a small touchpad, which is mapped to the whole screen, to move the cursor across the screen.

In ThumbSpace, a user defines his/her own thumb's reachable area on screen. It presents a radar view for the whole screen in the per-user defined ThumbSpace so that targets on the edge or corner of screen can also be quickly and easily selectable by thumb [9, 10]. In TapTap, users first select an area of interest by a coarse touch, and then a pop-up view showing a magnified version of selected area of interest appears. With a second touch in the pop-up view, users can delicately select the desired target [16]. Fit Your Hand rearranges UI objects on screen to the best easy-to-touch position automatically and dynamically based on users' handedness, finger length, usage habits, and etc [12].

There are researches that try to add more dimensions to touch input by using finger posture, angle, contact area as parameters. MicroRolls detects 6-way rolling of thumb on screen and uses them to build a new touch input vocabulary [17]. FatThumb uses the size of thumb's contact area as a new dimension so that it can distinguish between panning and zooming modes when thumb is dragging on screen without any explicit mode switching.

Studies on using an additional sensory input along with touch input are Sensor Synaesthesia [6] and TapSense [5]. Sensor Synaesthesia provides two types of interaction on mobile devices integrating touch and motion. One is touch-enhanced motion gestures that make use of embedded accelerometers to enable zooming by tilting back and forth the device while touching screen with finger. The other is motion-enhanced touch that detects vibration signals to distinguish soft tap and hard tap. TapSense recognizes unique sound patterns which are generated when different objects make contact with screen. On mobile setting, it can classify finger's tip, pad, nail, knuckle taps and can map different actions to each tap.

2.2 Using Backside of Mobile Devices

Backside interaction on mobile devices expands input space or input dimension. GraspZoom attaches Force Sensitive Resistor (FSR) on the backside of a mobile device to sense touch pressure on the front touchscreen so that multi-state touch input is possible [13]. In RearType, a split QWERTY keyboard is attached on the backside of a tablet-sized mobile device. It provides tactile multi-finger input and removes hand occlusion caused when using soft keyboard on touchscreen [18]. Baudisch et al. developed the NanoTouch device which has a small screen on frontside and has a touch sensor on backside [2]. Enhanced with the Shift technique [21], NanoTouch makes it possible to precisely select a very small target in a very small screen without any occlusion. In addition to these researches, commercial products, e.g. smartphone accessories and gaming devices, are also released recently that supplement additional D-pad (Directional-Pad with up, down, left, right buttons), function buttons, or touchpad on the backside of the devices [24, 25].

2.3 Interaction using Embedded Cemera

As computing power of mobile devices improves and cameras are embedded by default on mobile devices, researches on interaction techniques using camera are also actively being carried out. TinyMotion can detect motion of device without using any accelerometer or gyro sensor by observing movement of background from incoming video [22]. iRotate uses a built-in front camera and detects the face of user. Augmented with a gyro sensor, it automatically rotate screen according to the orientation of device [4].

3. TRACKLENS INTERACTION TECHNIQUE

TrackLens uses an embedded rear-camera as a sort of touch sensor. Therefore, users can use their index finger as well as their thumb to control their mobile device while holding it with only one hand. Also, since it does not require any additional sensor or hardware, TrackLens interaction technique can be applied to any general smartphones with moderate computing power.

3.1 Gesture Recognition and Direction Control

TrackLens can recognize 4-direction swipe gestures – up, down, left, right (Figure 2a to 2d). Instead of horizontal swipe gestures, clockwise or counter-clockwise rotation gestures are also possible (Figure 2e and 2f). These gestures are performed with direct contact on a camera lens not on the air. Thus users can feel tactile feedback from the bump around the lens to adjust their gesture movement.

Users can control direction as well as gestures with TrackLens interaction. Finger position on the camera lens controls the direction (Figure 3). Both 4-way discrete direction control like D-pad and 360° continuous direction control are also possible.

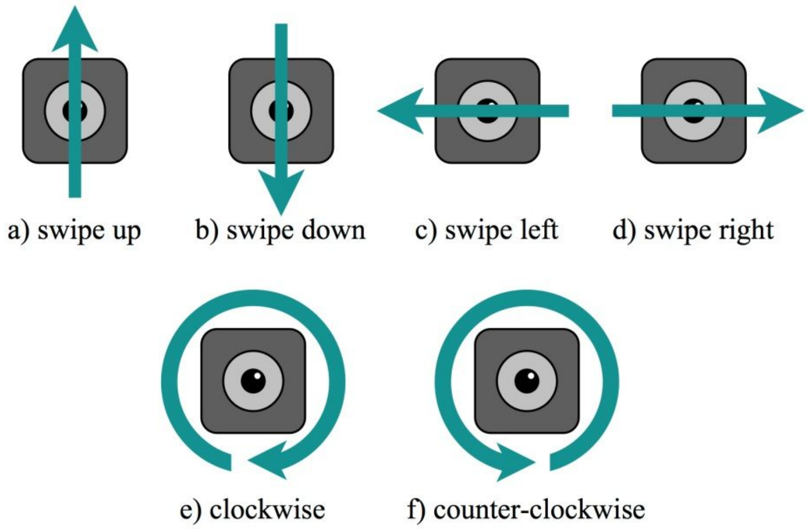


Figure 2. TrackLens gestures

Figure 3. Direction control via touch area on lens

3.2 Mode Switching and Target Selection

TrackLens can distinguish whether users' finger is on the lens or off the lens. Using this feature, mode switching without any external hard button or on-screen soft button is possible. Specifically, users can change modes by tapping the rear-camera with their index finger or a certain mode can be activated while users' finger blocks the camera. This reduces the user effort to find and click a mode switching button [7]. In addition, it can avoid the problem of the mode switching button occluding contents on screen.

With this quick mode switching, TrackLens can be integrated with ThumbSpace or Adaptive Offset cursor techniques to select a target on screen. The radar view of ThumbSpace can appear while users' index finger is on the camera and targets in the radar view can be selected with users' thumb. In the Adaptive Offset cursor mode, a thumb-reachable-sized rectangular touchpad appears on screen replacing the radar view in the ThumbSpace mode. Thumb's movement on the touchpad is mapped to the cursor's movement in the whole screen, e.g. if thumb is on the top-left corner of touchpad then the cursor is on the top-left corner of screen. In this work, we integrate TrackLens with Adaptive Offset cursor and with ThumbSpace as explained above; and compare their target selection performance with Direct Touch.

Target selection could also be done by only using index finger with the TrackLens technique without having to switch mode and use thumb together. However, our pilot test showed that, as for the target selection task, using only index finger does not have comparative advantages over other methods using both thumb and index finger. Target selection time and accuracy fall short of satisfactory performance and thumb is free anyway during selecting a target with index finger. Therefore, we dropped the idea of selecting target only with index finger.

4. IMPLEMENTATION

TrackLens uses simple image processing and machine learning algorithms to recognize users' gestures and direction control. The overall flow of operation is shown in Figure 4. The system is implemented on an Android 4.0 device.

First of all, TrackLens receives the lowest resolution (320x480) video feed from camera. We adjust camera settings to posterize the video feed to increase contrast of image and to remove ambiguous background figures by crushing color tones. This posterized image is stored in the YUV format. We extract only the luminance values to convert the image to grayscale and then downsample it to16x16 size.

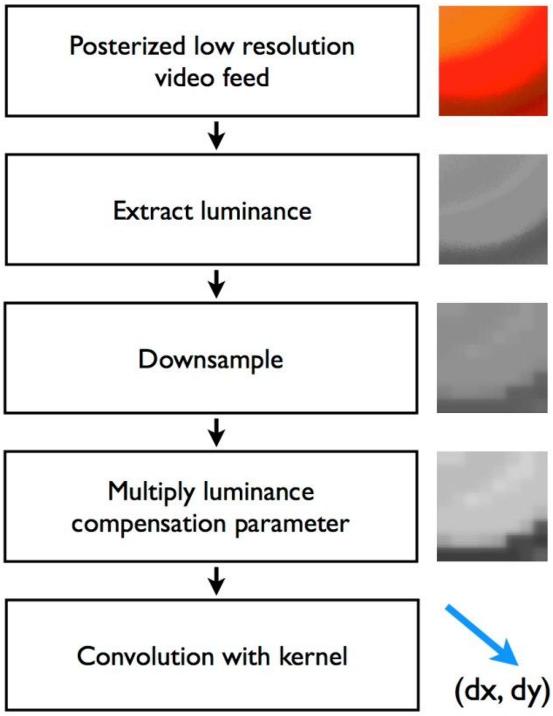


Figure 4. Process to produce direction vector

Each pixel in the downsampled image is multiplied by the luminance compensation parameter L and values above 255 are cut off. The luminance compensation parameter is for making the brightness of each image frame close to 255 in the STANDBY state so that the ACTIVE state when users' finger is on the lens is easily distinguishable. It can be set differently depending on the lighting condition around the camera. In this paper, we empirically set it to a constant value 4, which yields a reasonable outcome in common circumstances.

To represent the image with numeric values, we designed a kernel matrix with the same dimensions as the image frame. Convolution with this kernel produces a direction vector of overall gradient direction in the given image (Figure 5). Before the convolution, the image frame is inverted so that the produced direction vector indicates the location of finger from the center of camera lens. Direction control is done by using this direction vector.

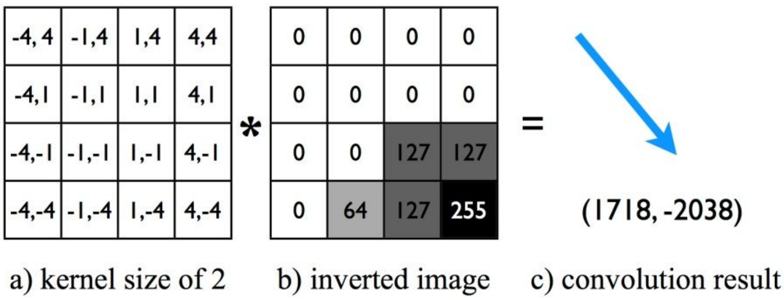


Figure 5. Calculation of direction vector

While in the ACTIVE state, direction vectors of each image frame are stored. To differentiate the same direction but with different size of lens area blocked by finger, we divide the direction vector by the number of black pixels. When the ACTIVE state ends, the gesture created by the transition between the first and the last direction vectors is classified via a logistic regression. The logistic model is built with the Weka machine learning toolkit [25]. The generated logistic model is light-weighted enough to be embedded on the device, so that the classification of gestures can be done in real-time.

By making most of operations done with integer calculation and reusing memory buffers extensively, we minimize processing load and reduce battery consumption.

5. EVALUATION

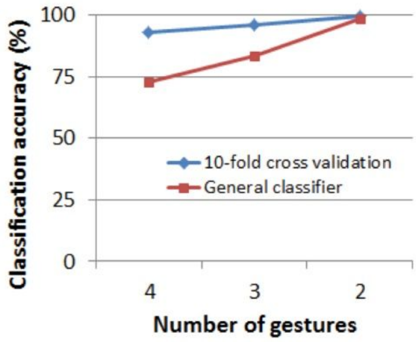
We evaluated our TrackLens interaction technique in a controlled user study. After evaluating TrackLens' gesture recognition performance first, we compared three interaction techniques – Direct Touch, TrackLens with ThumbSpace, and TrackLens with Adaptive Offset cursor – in terms of target selection performance. The study was conducted as a within-subjects design where each participant performed tasks using all three interaction techniques.

First, we evaluated TrackLens' recognition performance using three gesture sets – 4 gestures (↑, ↓, →, ←), 3 gestures (↓, →, ←), and 2 gestures (↓, →). The gesture sets were chosen with a confusion matrix (Figure 6a) which is a result of 10-fold cross validation using 2000 data points created by one user. We dropped the worst-performing gestures one by one starting from the gesture set with 4 gestures. We built general classifiers for the three gesture sets. Participants were asked to follow the gestures shown on screen with arrow marks one at a time and the recognition result (i.e. whether it was successfully recognized or not) was recorded.

Classified as

Intended to	↓	→	↑	←
↓	0.96	0.01	0.03	0.01
→	0.02	0.95	0.03	0.00
↑	0.02	0.08	0.88	0.02
←	0.04	0.01	0.01	0.93

a) confusion matrix



b) classification result

Figure 5. Classifier performance

We then compared Direct Touch, ThumbSpace and Adaptive Offset techniques to evaluate target selection performance. Direct Touch, a baseline condition for comparison, is the basic way of selecting a target on touchscreen. In this study, participants were required to use only one hand for the Direct Touch condition. For this study, we integrated our TrackLens technique with ThumbSpace and Adaptive Offset techniques so that users have to block the camera with their index finger to activate the two interaction techniques and target selection is done with their thumb. Targets of three different sizes are prepared – 24dp (density-independent pixels), 48dp and 72dp. 48dp is corresponded to a physical size about 9mm on screen and it is the recommended size for touchscreen objects on Android devices [23]. We divided screen by an equal-sized 3x3 grid and targets of each size appear on each cell. The order of target size and location was randomized. Participants had to select the targets by using each of three techniques. Target selection time and the number of successful selections were recorded.

15 participants (7 females, aged between 21 and 31) were participated in the user study. All of them are right-handed and familiar with using touchscreen smartphones. The study took less than 30 minutes per participant and each participant got rewarded with $5. We gave 3 minutes practice time before starting each session. User study was done with Samsung Galaxy S3 smartphone which has a 4.8 inch touchscreen and a rear-camera located at the top-center of backside.

6. RESULT & DISCUSSION

In this section, we present TrackLens' gesture recognition performance analysis result and then discuss the results of the comparative controlled user study. Based on the results, we present more possible applications of the TrackLens interaction technique.

6.1 Gesture Recognition Performance

Figure 6b shows that classification accuracy decreases as the number of gestures in a gesture set increases (98.3% for 2 gestures, 83.1% for 3 gestures, and 72.8% for 4 gestures). We found that users have trouble with getting used to swipe left and swipe up gestures. For right-handed person, moving the index finger right-to-left across the camera lens is not as easy as moving it left-to-right because index finger usually sits on the right side of the rear-camera. Also, the swipe up gesture cannot be distinguished clearly unless the finger is completely detached from the lens after the swipe, i.e. lens should not be blocked by finger after the gesture. But, when swiping up, finger still blocks the lens unless the user lifts up his/her index finger deliberately, which is unnatural. So the swipe up gesture is misclassified more often than other gestures (Figure 6a).

One more reason for the sharp drop of the classification accuracy is inherent limitation of general classifiers. In this study, we used general classifiers that do not take into account individual user's unique gesture patterns caused by different hand sizes, finger lengths, grab postures, and etc. Although per-user classifiers would introduce an overhead for users to train the classifier, it will produce a better recognition result, close to that of 10-fold cross validation, once it is fully trained.

In an after-study interview, 2 participants suggested using diagonal direction gestures rather than horizontal direction swipe because it was uncomfortable to move index finger horizontally across the camera lens. One of them further suggested that user-defined gesture sets rather than pre-defined gesture sets by the system would provide better usability.

6.2 Target Selection

Figure 7a shows that Adaptive Offset has the highest target selection accuracy both in whole screen area (86.6%) and in shadow area (89.4%). It was significantly more accurate than Direct Touch in both area ($p < 0.001$ for both area). Since it uses indirect approach for selecting targets using a cursor on screen and the cursor is controlled by movement of thumb within comfortable area, target selection accuracy is not affected by the location of target. But target selection time of Adaptive Offset

(1.6s for whole area, 2s for shadow area) was over twice slower ($p < 0.001$ for both area) than that of Direct Touch due to time taken to fine-tune the pointer position.

When the target size is big enough, Direct Touch has reasonably good selection accuracy (92.1%) while having the fastest selection time (0.7s) at the same time. Adaptive Offset has the highest accuracy for big targets (98.2%) but it does not have significant difference with Direct Touch ($p=0.057$). However, as the target size decreases, the accuracy of Direct Touch drops below 42%, which makes it unusable. For small targets, Adaptive Offset has significantly higher accuracy ($p < 0.001$) but significantly longer selection time ($p < 0.001$) than Direct Touch (Figure 7b). Users can take advantage of both Direct Touch and Adaptive Offset by choosing a better one depending on the selection condition. For example, users can use Direct Touch when selecting large targets near the center of screen and activate Adaptive Offset by blocking the rear-camera with finger when selecting small targets on the corner of screen.

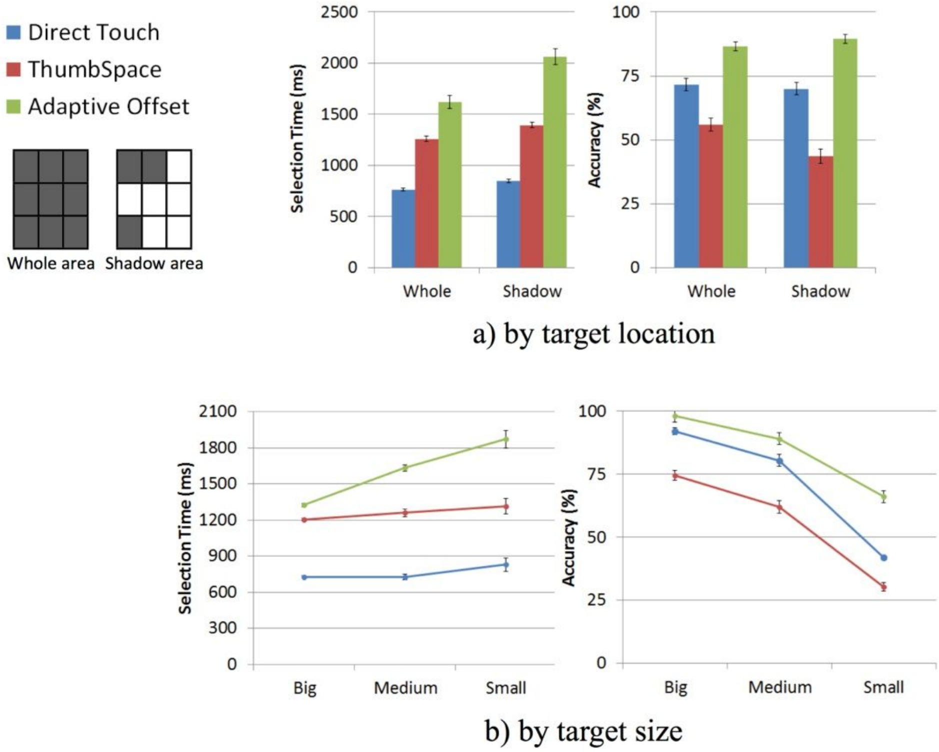


Figure 7. Target selection result

ThumbSpace's low performance in terms of both selection time and accuracy, compared with Direct Touch, is due to the fact that thumb's operational range becomes more limited with index finger on the rear-camera lens. Since the area that thumb can reach becomes smaller while index finger is on the rear-camera, the radar view of ThumbSpace also becomes small. Thus targets on the radar view become too small to tap with thumb. Moreover, as the resolution of touchscreen display is getting higher and higher these days, objects on screen are getting smaller. However, the radar view cannot become larger because a too large radar view makes areas unreachable by thumb on it. This makes the ThumbSpace technique less suitable for current mobile devices.

6.2 Applications

We implemented a web browser supporting the gesture set and the TrackLens with Adaptive Offset cursor technique as an example application of TrackLens. In the common web browser applications, function buttons, like back and bookmark buttons, are located on the top or bottom of screen where the user's thumb is hard to reach. Some web browsers support swipe gestures on screen that activate suitable functions according to the direction of swipe. But the swipe gestures on screen often coincide with scrolling of contents. In the web browser with the TrackLens technique, users can activate those functions by swiping across the rear-camera not on the screen. For example, users can swipe left or right with their index finger across the rear-camera to navigate back and forth between pages and swipe down to bring the bookmark menu.

If a target of interest is too small or too far to select, users can activate the Adaptive Offset Cursor mode by simply blocking the camera with index finger. With the cursor, the target can be selected precisely with thumb on screen without having to change the grip of device.

TrackLens can also be used as a controller for game applications. Many games on smartphone rely on soft D-pad to control movement of game characters. However, controlling direction with soft D-pad causes not only mistakes due to lack of tactile feedback but also content occlusion by finger touch interaction, which could hurt game-playing experience. We expect that our TrackLens interaction technique could enable users to control the direction of movement with more fun and with fewer mistakes thanks to the feeling of bump around the lens.

7. CONCLUSION AND FUTURE WORK

In this paper we presented TrackLens, a single-handed interaction technique on handheld devices using embedded rear-camera. TrackLens enables users to use both thumb and index finger in one hand to do gesture actions and precisely select targets on screen with a comfort and secure grip of device.

Through a controlled user study, we found that the gesture classifier should be improved for more usability. Personalization of gesture classifier, i.e. training per-user classifier and providing user-defined gesture sets, can be done as a further work. The touchpad in Adaptive Offset Cursor mode can also be personalized for maximum utilization of thumb's operational range, i.e. adjusting the size and position of the touchpad according to the user's hand size, thumb length, and grip posture.

The luminance compensation parameter value L was fixed in this paper. By observing average brightness changes from video feed or by using embedded luminance sensor, it can be dynamically adjusted according to the user's environment to archieve better performance. Furthermore, input dimension of TrackLens can be expanded by enabling image-based finger pressure or depth recognition (Figure 8).

a) near the lens b) on the lens c) pressing the lens

Figure 8. Video frame according to distance of finger from lens

REFERENCES

[1] Albinsson, P. and Zhai, S. High Precision Touch Screen Interaction. In Proc. CHI 2003, 105-112.

[2] Baudisch, P. and Chu, G. Back-of-Device Interaction Allows Creating Very Small Touch Devices. In Proc. CHI 2009, 1923-1932.

[3] Boring, S., Ledo, D., Chen, X.A., Marquardt, N., Tang, A. and Greenberg, S. The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction. In Proc. MobileHCI 2012, 39-46.

[4] Cheng, L., Hsiao, F., Liu, Y. and Chen, M. iRotate: Automatic Screen Rotation based on Face Orientation. In Proc. CHI 2012, 2203-2210.

[5] Harrison, C., Schwarz, J. and Hudson, S.E. TapSense: Enhancing Finger Interaction on Touch Surfaces, In Proc. UIST 2011, 627-634.

[6] Hinckley, K. and Song, H. Sensor Synaesthesia: Touch in Motion, and Motion in Touch. In Proc. CHI 2011, 801- 810.

[7] Hinckley, K., Pierce, J., Sinclair, M. and Horvitz, E. Sensing Techniques for Mobile Interaction. In Proc. UIST 2000, 91-100.

[8] Huot, S. and Lecolinet, E. Focus+Context Visualization Techniques for Displaying Large Lists with Multiple Points of Interest on Small Tactile Screens. In Proc. INTERACT 2007, 219-233.

[9] Karlson, A.K. and Bederson, B.B. One-Handed Touchscreen Input for Legacy Applications. In Proc. CHI 2008, 1399-1408.

[10] Karlson, A.K. and Bederson, B.B. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. In Proc. INTERACT 2007, 324-338.

[11] Karlson, A.K., Bederson, B.B. and Contreras-Vidal, J. L. Studies in One-Handed Mobile Design: Habit, Desire and Agility. HCIL Technical Report 2006-02, University of Maryland, College Park, Maryland

[12] Lee, H. and Choi, Y.S. Fit Your Hand: Personalized User Interface Considering Physical Attributes of Mobile Device Users, In Proc. UIST 2011, 59-60.

[13] Miyaki, T. and Rekimoto, J. GraspZoom: Zooming and Scrolling Control Model for Single-Handed Mobile Interaction. In Proc. MobileHCI 2009.

[14] Parhi, P. Karlson, A.K. and Bederson, B.B. Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices. In Proc. MobileHCI 2006, 203-210.

[15] Potter, R.L., Weldon, L.J. and Shneiderman, B. Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies. In Proc. CHI ’88, 27-32.

[16] Roudaut, A. Hout, S. and Lecolinet, E. TapTap and MagStick: Improving One-Handed Target Acquisition on Small Touch-screens. In Proc. AVI 2008, 146-153.

[17] Roudaut, A., Lecolinet, E. and Guiard, Y. MicroRolls: Expanding Touch-Screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. In Proc. CHI 2009, 927-936.

[18] Scott, J., Izadi, S., Rezai, L.S., Ruszkowski, D., Bi, X. and Balakrishnan, R. RearType: Text Entry Using Keys on the Back of a Device. In Proc. MobileHCI 2010.

[19] Sears, A. and Shneiderman, B. High Precision Touchscreens: Design Strategies and Comparisons with a Mouse. International Journal of Man-Machine Studies 34, 4 (1991), 593-613.

[20] Siek, K.A., Rogers, Y. and Connelly, K.H. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In Proc. INTERACT 2005, 267-280.

[21] Vogel, D. and Baudisch, P. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In Proc. CHI 2007, 657-666.

[22] Wang, J., Zhai, S. and Canny, J. Camera Phone Based Motion Sensing: Interaction Techniques, Applications and Performance Study. In Proc. UIST 2006, 101-110.

[23] Android Metrics and Grids Guidelines. http://developer.android.com/design/style/metrics-grids.html

[24] PlayStation Vita. http://us.playstation.com/psvita/features/

[25] SixthSense makes one hand operation.
http://diosystems.com/prod_eng.html

[26] Weka 3: Data Mining Software in Java.
http://www.cs.waikato.ac.nz/ml/weka

국문 초록

내장 후면 카메라를 터치 입력 장치로 활용한 휴대용 기기의 한 손 인터랙션 디자인

서울대학교 대학원

전기컴퓨터공학부

고재희

터치스크린은 입력 공간과 출력 공간을 하나의 하드웨어로 통합함으로써 기기를 보다 작고 가볍게 만들 수 있다는 장점이 있어 최근 스마트폰, 태블릿 등의 휴대용 기기에 널리 사용되고 있다. 그러나 엄지손가락이 닿지 않는 그림자 영역 및 여러 개의 손가락이 필요한 멀티 터치 제스처 등으로 인하여 휴대용 기기의 터치스크린은 한 손만으로 사용하기에 큰 불편이 따른다.

본 연구에서는 내장된 후면 카메라를 활용하여 휴대용 기기를 한 손으로 편하고 정교하게 조작할 수 있도록하는 상호작용 기법을 제시한다. 카메라를 일종의 터치 센서로 활용하는 이 기술은 기기 뒷면에서 이루어지는 상하좌우 네 방향 검지 손가락 제스처를 인식할 수 있고 또한 검지 손가락과 엄지 손가락을 동시에 사용하여 화면 상의 목표물을 보다 정교하게 선택할 수 있게 함으로써 휴대용 기기를 한 손으로 사용했을 때의 불편함을 해소하였다. 특히 본 연구는 별도의 하드웨어 장치 없이 카메라가 내장된 모든 휴대용 기기에서 바로 적용가능하다는데 큰 의의가 있다.

15 명의 사용자를 대상으로한 사용성 실험 결과, 제스처 인식 알고리즘은 다소 개선이 필요하나 본 연구에서 제시한 상호작용 기법은 사용자들이 터치스크린 휴대용 기기를 한 손만으로 보다 편하고 정확하게 조작하는데 도움이 되는 것으로 나타났다. 사용자의 손의 크기, 엄지 손가락의 길이, 기기를 잡는 자세 등을 고려한 사용자별 맞춤 제스처 집합 및 인식 알고리즘을 제공한다면 보다 나은 성능과 다양한 응용 가능성을 가져다줄 것으로 예상된다.

키워드 : 터치스크린, 휴대용 기기, 카메라 상호작용, 한 손 상호작용, 제스처 인식

학번 : 2011-20783