A New Eye Tracking Method as a Smartphone Interface

Eui Chul Lee[1] **and Min Woo Park**[2]
[1] Department of Computer Science, College of Software, Sangmyung University
110-743, 7 Hongji-Dong, Jongno-Gu, Seoul, Republic of Korea
[e-mail: eclee@smu.ac.kr]
[2] Department of Computer Science, Graduate School, Sangmyung University
110-743, 7 Hongji-Dong, Jongno-Gu, Seoul, Republic of Korea
[e-mail: nogood79dle@gmail.com]
*Corresponding author: Eui Chul Lee

Received December 31, 2012; revised March 14, 2013; accepted April 8, 2013; published April 30, 2013

Abstract

To effectively use these functions many kinds of human-phone interface are used such as touch, voice, and gesture. However, the most important touch interface cannot be used in case of hand disabled person or busy both hands. Although eye tracking is a superb human-computer interface method, it has not been applied to smartphones because of the small screen size, the frequently changing geometric position between the user's face and phone screen, and the low resolution of the frontal cameras. In this paper, a new eye tracking method is proposed to act as a smartphone user interface. To maximize eye image resolution, a zoom lens and three infrared LEDs are adopted. Our proposed method has following novelties. Firstly, appropriate camera specification and image resolution are analyzed in order to smartphone based gaze tracking method. Secondly, facial movement is allowable in case of one eye region is included in image. Thirdly, the proposed method can be operated in case of both landscape and portrait screen modes. Fourthly, only two LED reflective positions are used in order to calculate gaze position on the basis of 2D geometric relation between reflective rectangle and screen. Fifthly, a prototype mock-up design module is made in order to confirm feasibility for applying to actual smart-phone. Experimental results showed that the gaze estimation error was about 31 pixels at a screen resolution of 480×800 and the average hit ratio of a 5×4 icon grid was 94.6%.

Keywords: Eye tracking, Gaze tracking, Smartphone interface

This research was supported by a 2013 Research Grant from Sangmyung University.

http://dx.doi.org/10.3837/tiis.2013.04.013

1. Introduction

In recent years touch screen based smartphones have become increasingly popular. Tablet PCs, PDAs (Personal Digital Assistants) and UMPCs (Ultra Mobile Personal Computers) are also widely used. Following the appearance of various applications and their increasing complexity, the hardware performance of these devices has continued to improve. In contrast, their input devices are similar to those of desktop computers, because there is no alternative input method. The buttons or touch regions on the keyboards of these devices are much smaller than conventional keyboards for a desktop PC. Therefore, incorrectly touched errors frequently occur. Also, there is dirt problem caused by fingerprint and finger sweat in case of using capacitive touch screen interface. In case of hand disabled person or busy both hands, touch interface cannot be used as a smartphone interface. In such mobile devices, a built-in camera is included in the default configuration. Thus, this input problem can be solved by using gaze tracking technology to determine exactly where users are looking on any given screen by analyzing the user's eye and facial movements. The gaze tracking method in mobile device has following advantages. Firstly, intuative and rapid interaction can be available because watching equals to pointing. In case of using touch interface, pointing by touching is perfomed after gazing at interesting position. Second, there is no dirt problem caused by finger contact. Thirdly, eye image based additional functions can be easily appended without placing any camera equipment. For example, iris recognition for security lock function and emotion estimation for user context awareness by analyzing pupil size and eye blink can be used because the kinds of their used image are equal to the eye image for gaze tracking.

Gaze tracking methods for desktop screens have been extensively researched [1][2] and gaze tracking has been applied to many kinds of application such as virtual reality [3], pilot training [4], games [5], and usability evaluation [6] for over twenty years. Several commercial products have already been released [7].

There are many kinds of gaze tracking methods. In terms of the approach, they can be categorized as either 2D image based methods [8-10] or 3D eye structure based methods [2][11][12]. In each method, the number of cameras, and the number of illuminators vary as needed, as can additional components such as location tracking sensors [13], HMDs (Head Mounted Displays) [2], and eyeglasses frames [14].

In the desktop screen based environment, the various kinds of gaze tracking methods can be easily adopted because most desktop screens are larger than ones of mobile devices. In other words, the gaze estimation error of about 1~2 degrees in conventional systems may be acceptable because the FOV (Field Of View) of desktop monitors is large. For example, 1~2 degrees error can be negligible in some applications since a 23 inch monitor has a FOV of 52° when the distance between the monitor and user's eye is 60cm. Due to the much smaller margin for error there has been comparatively little research into gaze tracking for mobile environments

Several studies have reported on eye tracking for mobile devices as shown in Table 1. The research by Kristian tried to apply a gaze tracking method for hand-held devices [15]. This system was a prototype on the bases of headgear and a PDA (Personal Digital Assistant). To obtain the 3D position of a user, they analyzed the relations among the user's head, eyes and the mobile device by using a magnetic location tracking sensor. It is not suitable when working with mobile devices because this method required great computational complexity and users often experienced inconvenience due to the heavy headgear. Next, methods of tracking gaze directions were proposed under the assumption that users gazed at eight directions on mobile phones [16]. Through the use of eye-gaze gestures, users were able to simply control mobile phones by using a few enrolled commands. However, this system was only a prototype which used a conventional eye-tracking camera designed for desktop screens and did not allow for facial movements. Furthermore, since the mobile phone was attached to the desktop monitor, this method would also not work with actual mobile environments. In the study of Gustav et al., they evaluated the readability of text presentation formats on a mobile screen by analyzing data acquired from the gaze tracking system [17]. They used a commercial gaze tracking system. This system consisted of a pair of goggles including IR-LEDs (Infra-Red Light Emitted Diodes) and a processing module which estimated the user's gaze position. This method did not allow user's facial movements, as also mentioned in the above research [16]. Hansen *et al.*, studied a new input scheme which worked by tracking user's face on mobile devices [18]. This method used a scheme which worked with the translation and rotation of the mobile device instead of tracking a user's facial movements. In order to calculate gaze points, researchers simply analyzed the face position of a given image on the basis of a probabilistic color histogram of facial regions. Therefore, it was difficult to detect the face when other similar objects were included in the input image. Also, performance was affected by environmental lighting. In the other previous work, a facial gaze tracking method based on an active appearance model was proposed for ultra-mobile PCs [19]. Although features of eye region were included, the method is not intuitive because of their dragging navigation scheme. Also, the active appearance model used cannot be operated in real-time in a low powered processing environment such as a smartphone.

Table 1. Previous methods of gaze tracking for mobile device and their Pros and Cons

Methods	Pros	Cons
Kristian's method [15]: Applying gaze tracking method to hand-held devices as prototype by using a headgear, a PDA, head tracking sensor.	- Calculating the 3D geometric relation between the screen and user's head.	- High computational complexity. - Inconvenience caused by wearing headgear.
Heiko's method [16]: Estimating 8 gaze directions for performing several pre-enrolled simple commands.	- Low computational complexity.	- Not validated for applying into mobile device because of just using a prototype by attaching mobile phone onto the desktop screen. - Not allowing facial movement.
Gustav's method [17]: Evaluating the text readability of mobile screen using commercial gaze tracking system	- Light-weight goggle based eye tracking system. - Evaluating usability of mobile screen by using eye tracking method.	- Not gaze tracking algorithm but just application. - Not allowing facial movement.
Thomas's method [18]: Facial gaze tracking of mobile screen using geometric relation between face and screen by moving mobile device instead of facial movement.	- Low computational complexity because of only using facial position in color image.	- Environmental lighting condition problem because of using color image. - Not intuitive because of using dragging navigation scheme.
Lee's method [19]: A facial gaze tracking method based on an active appearance model for UMPC.	- Accurate gaze estimation because of using many features of facial and eye regions.	- Not intuitive because of using dragging navigation scheme. - High computational complexity caused by operating active appearance model.

The above-mentioned previous methods also used some additional equipment such as a location tracking sensor or head-mounted devices to overcome the accuracy problems caused by gazing at the small sized screens of mobile devices. Thus, users often experienced inconvenience, system costs were increased and the modification of conventional mobile devices was required. Whereas in Hansen's research they did not use additional devices (Thomas et al., 2006), the accuracy of gaze tracking was not acceptable because they only used facial positions that were based on face color.

To solve these problems, we propose a new eye tracking method for mobile devices. We use an Adaboost method for eye detection and analyze the relation between the pupil center and three specular reflections generated by three IR-LEDs for extracting facial features. To confirm the feasibility of the proposed method, we designed a smartphone prototype which includes three IR-LEDs and a fixed focus zoom lens camera. The proposed method can be adaptively applied to various applications since two specular reflections are selectively analyzed according to the device direction such as landscape and portrait modes. Since the specular reflection can be regarded as the area of the screen and the gaze position is relatively calculated based on this region, additional error compensation methods required due to facial movement are not necessary in our method.

In section 2, the designed device and eye tracking algorithm are respectively given. In section 3, the experimental setup, results and their analyses are discussed. Conclusions and future research plans are given in section 4.

2. Material and Methods

2.1 Device

To implement an eye tracking system into a mobile device, we designed a smartphone based prototype as shown in **Fig. 1**. Since conventional mobile phones do not meet the hardware requirements to apply this eye tracking method, we supplemented the device with infrared LEDs and a USB webcam including a zoom lens as shown in **Fig. 1**.

First of all, the camera is the most important device in order to implement a vision based eye tracking system. Even though a conventional smartphone has two cameras, front and rear facing, the front camera cannot be used for calculating gaze position from an acquired eye image because the spatial resolution is too low. Commonly a front camera has VGA resolution (640×480) because the camera is configured for video calls. Although the rear view camera has greater spatial resolution, it cannot capture images of the smartphone user's eye. To overcome the resolution problem, a small 1 MP (Mega-Pixels) USB webcam is attached to the top of the smartphone, which is combined with a PC for real-time eye image processing to calculate gaze position.

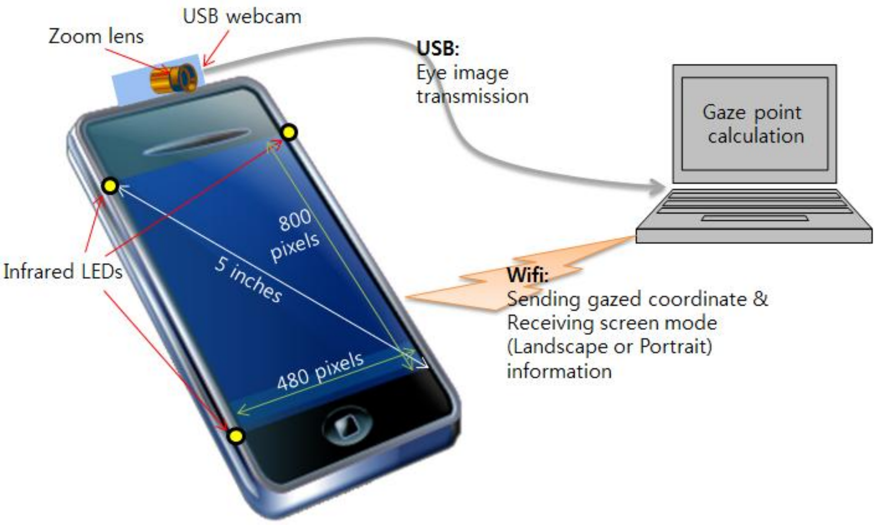


Fig. 1. Concept of smartphone based eye tracking device

Next, we designed a zoom lens in order to acquire a magnified eye image. The zoom lens has 3.2 times magnification ratio compared with the default configured lens. Since the default lens of the USB webcam is suitable to capture wide view landscapes, the resolution of the captured eye region is not enough to analyze the gaze position in the case of capturing a facial image by using the lens.

Table 2. Eye tracking accuracy prediction in various cases

	640 × 480 with default lens	**640 × 480 with zoom lens**	**1280 × 720 with default lens**	**1280 × 720 with zoom lens**
Iris diameter	23pixels	73pixels	57pixels	232pixels
Pupil diameter	5~18pixels	15~58pixels	11~45pixels	46~186pixels
Specular reflection size	< 1pixel	1pixel	< 1pixel	7pixels
Size of specular reflective rectangle	4(H) × 6(V) pixels^2	12(H) × 20(V) pixels^2	9(H) × 16(V) pixels^2	38(H) × 64(V) pixels^2
Predicted error	133pixels (4°)	40pixels (1.2°)	50pixels (1.5°)	12pixels (0.4°)

To confirm the need for a high-resolution camera and a zoom lens, we predicted gaze point resolutions in the case of both VGA and 1MP cameras, with or without a zoom lens as shown in **Table 2**. To predict errors as shown in Table I, we used a quantitative model when the viewing distance was 300mm as shown in **Fig. 2**.

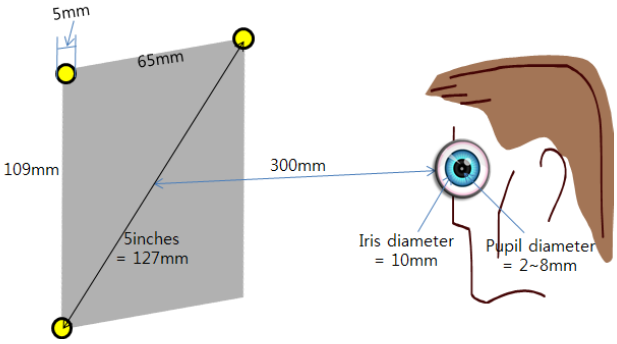


Fig. 2. Used smartphone based model for predicting eye tracking accuracy

The used smartphone was diagonally 5 inches and the aspect ratio was 3:5 (480pixels×800pixels). The diameter of each IR-LED was 5mm. On the basis of human anatomy, the diameter of the iris and pupil is 10mm and 2~8mm [20], respectively. Note that the appropriate image resolution of infrared vision based iris recognition requires an iris diameter of more than 200 pixels [21]. Many eye tracking methods use this level of resolution. Since the radius of the convex cornea surface of the human eye is 7.8mm, we could estimate the specular reflective IR-LEDs' size and the size of the specular reflective rectangle based on the triangular similarity whose base and height were 5 inches and 307.8mm, respectively. Also, since the focal length of the 640×480 camera with the default lens was 680 pixels and the magnification ratio of the zoom lens was 3.2 times compared with the default lens, we could calculate the iris and pupil diameters, specular reflections and their rectangle sizes in the captured image.

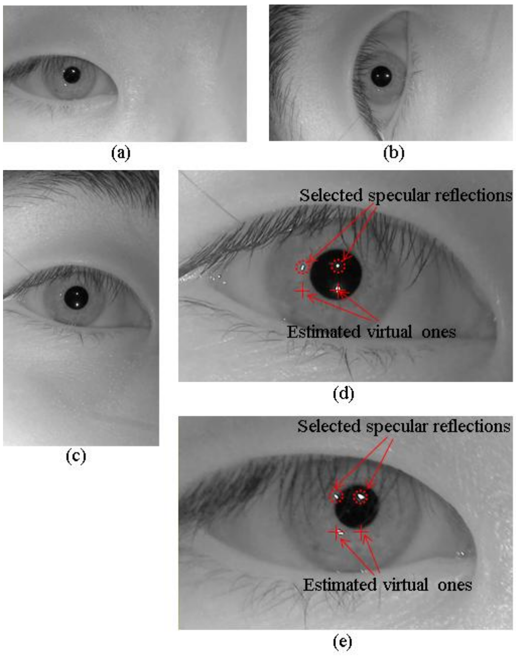


Fig. 3. Captured images and specular reflections. (a) Captured eye image in portrait mode. (b) Captured eye image in landscape mode. (c) 90° Clockwise rotated image of (b). (d) Cropped eye region of portrait mode eye image. (e) Cropped eye region of landscape mode eye image

Consequently, we predicted the average case error between the gazed position and calculated position based on the size of specular reflective rectangle. Even if the 640×480 with zoom lens has reasonable accuracy (error = 1.2°), the case has a fatal problem because the specular reflection size is just 1 pixel. This means that the reflection can disappear from the captured image if the viewing distance becomes a little distant. Therefore a camera including a 1280×720 zoom lens is the most suitable combination to implement eye tracking on a smartphone.

As shown in Figs. 1 and 2, three IR-LEDs were attached in our system. In case of portrait screen mode, specular reflections caused by upper two IR-LEDs were used for calculating gaze position. Namely, the position of virtual lower two specular reflections are estimated based on the screen aspect ratio. Similarly, specular reflections caused by the left two IR-LEDs are adopted in case of landscape screen mode. **Fig. 3** shows such two cases.

2.2. Algorithm

To calculate gaze position on a smartphone display, several steps are performed as shown in **Fig. 4**.

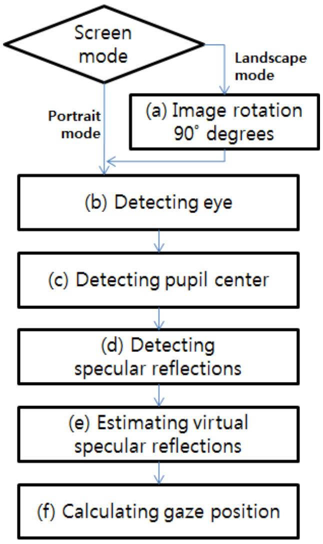


Fig. 4. Flow chart of proposed eye tracking method.

Firstly, the screen mode should be considered because the principal point is automatically changed by the built-in gyroscopic sensor depending on whether the phone is used in landscape or portrait orientations as shown in **Fig. 5**. Since we used the eye image in portrait mode as the default, the clockwise rotated eye image in landscape mode should be rotated counterclockwise as in **Fig. 4** (a).

Then the eye region is detected before performing pupil detection in order to reduce false pupil regions, as in **Fig. 4** (b). In our used camera, since face region is not perfectly included in the captured image, an eye detection method is directly performed instead of considering a separate face detection procedure. To detect eye region, the widely used Adaboost method is used [22]. In order to minimize processing time, the original eye image of 1280 pixel × 720 pixel is reduced to 128 pixel × 72 pixel because only the rough eye position is necessary to define the pupil candidate region. On the basis of the detected eye region center, a pupil candidate region of 256 pixel × 256 pixel are defined as shown in **Fig. 6**.

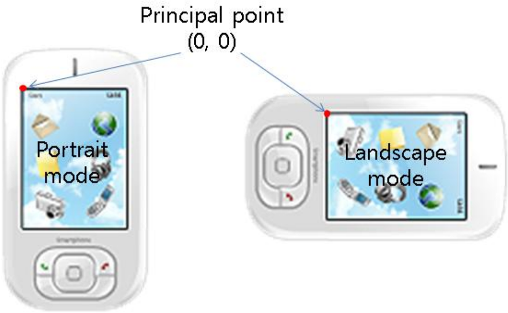


Fig. 5. Display modes of smartphone and their principal points.

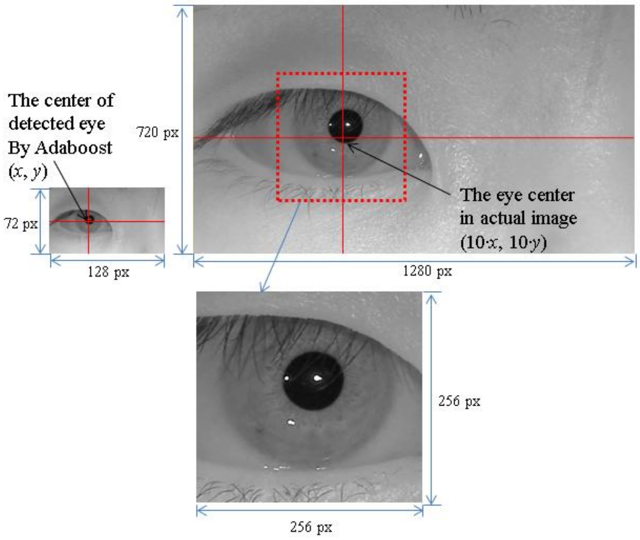


Fig. 6. Example of eye detection and cropped pupil candidate region

In the pupil candidate region, pupil center detection is performed as in **Fig. 4** (c). To detect pupil center, several steps are performed as shown in **Fig. 7**. In the steps, binarization is firstly performed as shown in **Fig. 7** (b) because the pupil region in an infrared image is comparatively darker than other regions. To perform the binarization, threshold value is determined by using Gonzalez's automatic decision method [20]. After binarization, component labeling is performed in order to separate each combined region and give different labels to each region [20]. Among the regions, the pupil region is determined by size and ratio filtering based on anatomical pupil size and pupil's circular shape. As shown in Table 1, the theoretical pupil diameter was already calculated as 46 pixels ~ 186 pixels. Therefore, the range of pupil size can be defined as about 1661 pixels2 ~ 27158 pixels2. The aspect ratio of each region is calculated for ratio filtering based on the aspect ratio of a pupil's circular shape closes to 1. That is, among regions in the range of 1661 pixels2 ~ 27158 pixels2, the thing having an aspect ratio of closest to 1 is determined as pupil as **Fig. 7** (c). Consequently, the calculated center of gravity of the region is determined as the pupil center position as shown in **Fig. 7** (d).

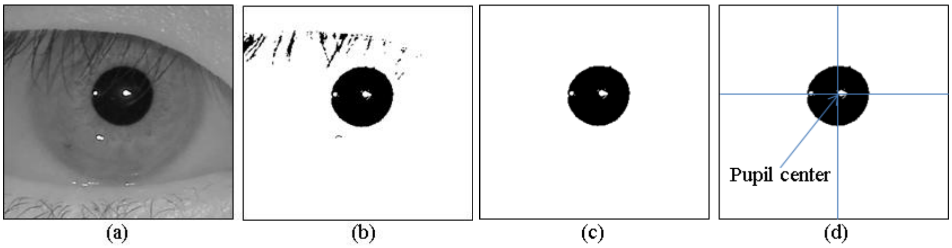


Fig. 7. Sequential steps for detecting pupil center (a) Cropped pupil candidate image (b) After binarization (c) Size and ratio filtering result after component labeling (d) Calculated center of gravity

After obtaining the pupil center position, centers of specular reflections should be detected to calculate eye gaze position on the smartphone screen. In a previous work, it was shown that specular reflection can be modeled on the basis of a convex mirror model because the human cornea has a convex shape and its surface is reflective and uniform [23]. Specular reflection is always generated on the straight line connecting the light source's center and the convex sphere's center as shown in **Fig. 8** [23]. Also, the pupil center is always positioned away at angle Kappa [2] from the gaze vector. Therefore, our proposed method is almost unaffected by the geometric relation between the human facial position and smartphone screen as shown in **Fig. 8**. While a large amount of facial movement can cause gaze error due to transformation of the camera's perspective [1], this factor can be ignored in our method because such facial movement is not allowed for acquiring an eye image in a smartphone attached camera based scenario.

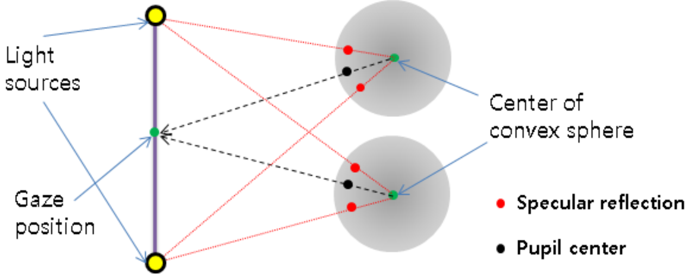


Fig. 8. Almost constant geometric relation between pupil center and specular reflection center in cases of different cornea convex sphere positions caused by facial movement

To detect specular reflection regions, a local square region defined as 120 pixels × 120 pixels on the basis of the already estimated "Size of specular reflective rectangle" of Table I is used. Then, binarization is performed in order to highlight very bright specular reflection regions as shown in **Fig. 9** (b). Here a threshold value of 230 is used. Then, component labeling is adopted to separate each specular reflection as shown in **Fig. 9** (c). Consequently, each center of specular reflection is obtained by calculating centers of gravity as shown in **Fig. 9** (d).

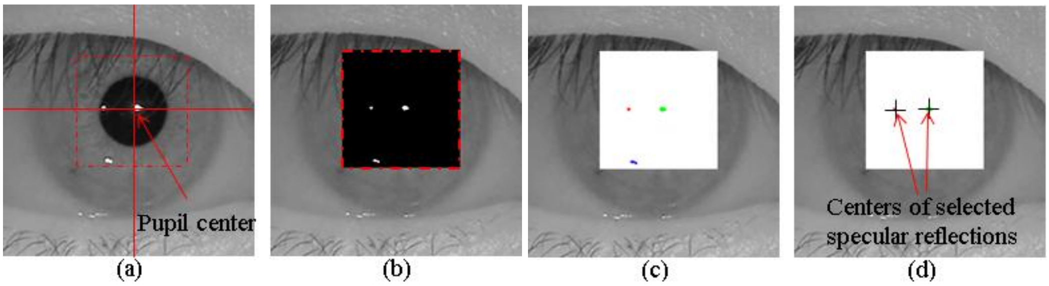


Fig. 9. Steps for detecting specular reflections (a) Defined square region for detection specular reflections (b) After local binarization (c) After component labeling (d) Obtained each center of specular reflections

Only the upper two specular reflections are used to calculate gaze position, because the lower specular reflection can be occluded by the user's hand. Therefore, the positions of remaining two specular reflections should be estimated in order to know the specular reflective rectangle. If the two specular reflection centers are (x_{LU}, y_{LU}) and (x_{RU}, y_{RU}), the remaining two positions are determined as (x_{LD}, y_{LD}) and (x_{RD}, y_{RD}) based on the following equations. In equation (1), R_y: R_x means the aspect ratio of the screen which corresponds to 5:3 in our used smartphone.

$$(x_{LD}, y_{LD}) = (x_{LU}, y_{LU} + \frac{R_y}{R_x}(x_{RU} - y_{LU})) \tag{1}$$

$$(x_{RD}, y_{RD}) = (x_{RU}, y_{LD}) \tag{2}$$

Based on the obtained pupil center and four positions of specular reflections, we can easily calculate gaze position on the smartphone screen. Since the origin point of screen (0, 0) is automatically changed to the top-left position in both cases of portrait and landscape modes, the screen mode is not considered in this step. If the pupil center position is (P_x, P_y), gaze position (G_x, G_y) on the smartphone screen can be calculated using the following equations. In equation (3), X_{res} and Y_{res} are horizontal and vertical resolutions of the smartphone screen, respectively.

$$(G_x, G_y) = (X_{res} \cdot \frac{P_x - x_{LU}}{x_{RU} - x_{LU}}, Y_{res} \cdot \frac{P_y - y_{LU}}{y_{LD} - y_{LU}}) \tag{3}$$

3. Results and Discussion

To perform experiments, we made an eye tracking mock-up device as shown in **Fig. 10** then Android based test application software was implemented. The software receives and interprets network data of gaze position and visualizes a cursor position determined by the user's gaze position. To examine the performance of our proposed eye tracking system, we performed several kinds of test.

Fig. 10. Used mock-up device for eye tracking

Firstly, the average RMS (Root Mean Square) error was measured for 10 subjects. In the test, each subject gazed at 18 pre-defined reference points 5 times as shown by the red points in **Fig. 11**. In the result, the average RMS error was about 31.4 pixels that correspond to horizontally 25 pixels and vertically 19 pixels, respectively. **Fig. 11** shows three examples of actual calculated gaze positions against reference points.

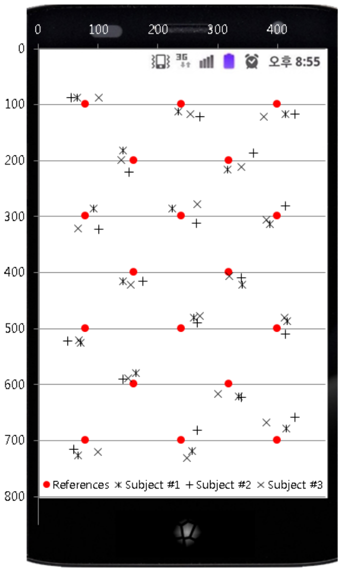
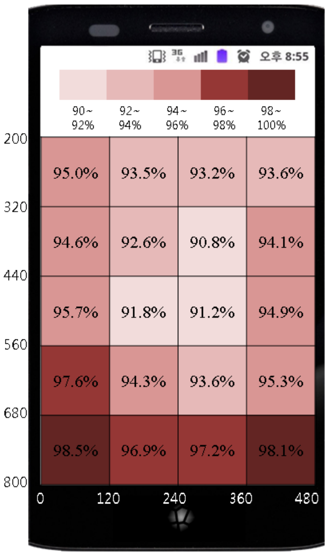


Fig. 11. Examples of calculated gaze positions against reference positions

Secondly, the hit ratio in each 5×4 region on smartphone screen was measured. Conventionally, icons are arranged in a 2-dimensional array pattern therefore the hit ratio of each icon region is more meaningful than the previous measured RMS error. **Fig. 12** shows the average hit ratio of 10 subjects. Each subject gazed at randomly announced number positions 100 times. In the bottom region, the reason for comparatively high hit ratios was because the calculated Y-axis gaze positions of greater than 800 were regarded as 799. Consequently, the average hit ratio of all regions was about 94.6% which is acceptable in terms of icon navigation in a smartphone user interface.



95.0%	93.5%	93.2%	93.6%
94.6%	92.6%	90.8%	94.1%
95.7%	91.8%	91.2%	94.9%
97.6%	94.3%	93.6%	95.3%
98.5%	96.9%	97.2%	98.1%

Fig. 12. The average hit ratio of 5×4 region

Lastly, we focused on additional functions of the proposed method which are automatic blink detection and pupil size measurement. In our proposed algorithm, detecting pupil region is necessary to obtain pupil center position. The blink detection can be performed on the basis of successive obtained pupil size values, because the pupil region is occluded in the case of a closed eye state. Successive pupil size values can be obtained in the case of open eye state. In research fields of human-computer interface and ergonomics, the average blink rate and the pupil size variation have been used as important metrics to quantitatively measure human emotional states [24].

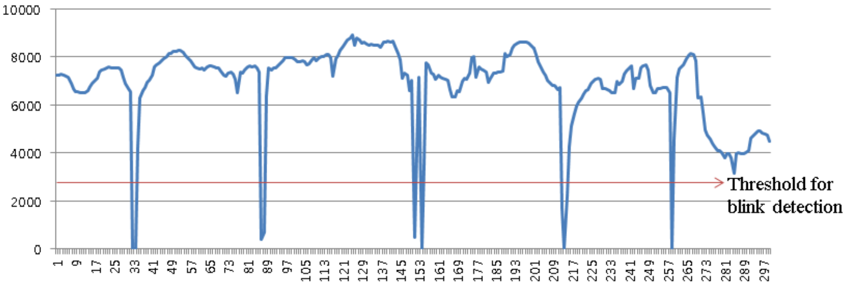


Fig. 13. Successive pupil sizes over a 20 second period

Therefore, we performed simple tests by analyzing 5 minute eye image videos of 5 persons in order to measure blink detection accuracy. In the results, the average blink detection rate was 97.7% (255 / 261) compared with manually detected location in time. Then, to measure the accuracy of pupil size detection, we randomly selected 100 open eye image frames then manually counted the number of pixels of pupil region by using the Adobe Photoshop CS3's magic wand selection tool. The result was that the average absolute difference between automatically detected and manually detected pupil sizes was 109 pixels. Since the average pupil size of the manually detected method was 6988 pixels, the error rate was about 2.99% (209 pixels / 6988 pixels). **Fig. 13** shows example of successive pupil sizes over a 20 second period.

4. Conclusions

In this paper, we proposed a new vision based eye tracking method which is suitable for smartphones. To supplement hardware weaknesses in terms of the camera device in smartphones, a mock-up prototype was designed and implemented by including infrared LEDs, camera, and a compact zoom lens. By analyzing the geometric relation between pupil center and infrared specular reflection positions, continuous eye gaze positions were calculated in real-time. Through several experiments, we validated performance of the eye tracking mounted smartphone. Also, we confirmed that additional functions such as blink detection and pupil size measurement were accurately performed in our system. However, our proposed method could not be operated in some cases such as wearing glasses and no eye region in captured image.

Our proposed method can be applied into hand disabled person, busy both hand situation such as surgeon in surgery and keystroke. Also, usability evaluation for smartphone user may be possible by using our gaze tracking, pupil size and blink measurement functions.

In future works, we will try to implement eye tracking methods in smartphones by using the default built-in frontal camera through performing an image super-resolution scheme. And, a new eye tracking method without extra LEDs will be studied based on user dependent calibration for defining the geometric relation between eye and scree regions. Also, we will define and evaluate effective gazing interactions which substitute many kinds of convenient interaction scheme used in touch based interface. Furthermore, gaze position, blink rate, and pupil size data acquired from our system will be applied to many kinds of usability evaluation systems which are necessary in making user friendly services or products.

References

[1] Y. J. Ko, E. C. Lee and K. R. Park, "A robust gaze detection method by compensating for facial movements based on corneal specularities," *Pattern Recognition Letters,* vol. 29, no. 10, pp. 1474-1485, 2008.
http://dx.doi.org/10.1016/j.patrec.2008.02.026

[2] E. C. Lee and K. R. Park, "A robust eye gaze tracking method based on virtual eyeball model," *Machine Vision and Applications,* vol. 20, no. 5, pp. 319-337, 2009.
http://dx.doi.org/10.1007/s00138-008-0129-z

[3] T. D. Andrew, S. Vinay, R. Tim, K. G. Anand, J. M. Brian and K. Barbara, "Binocular eye tracking in virtual reality for inspection training," *in Proc. of the 2000 symposium on Eye tracking research & applications,* pp. 89-96, 2000.
http://dl.acm.org/citation.cfm?id=355031

[4] A. W. Paul, M. A. Gretchen and A. B. Barbara, "Instructor use of eye position based feedback for pilot training," in *Proc. of the Human Factors and Ergonomics Society Annual Meeting,* vol. 42, no. 20, pp. 1388-1392, 1998.
doi: 10.1177/154193129804202005

[5] H. Heo, E. C. Lee, C. J. Kim, K. R. Park and M. C. Whang, "A realistic game system using multi-modal user interface," *IEEE Transactions on Consumer Electronics,* vol. 53, no. 6, pp. 1364-1372, 2010.
http://dx.doi.org/10.1109/TCE.2010.5606271

[6] H. G. Joseph, J. S. Mark, L. Marion, S. Neil and M. W. Anna, "Eye tracking in web search tasks:

design implications," in *Proc. of the 2000 symposium on Eye tracking research & applications,* pp. 51-58, 2000.
http://dl.acm.org/citation.cfm?id=507082

[7] T. D. Andrew, "Eye Tracking Methodology: Theory and Practice," *Springer,* 2007.

[8] C.Colombo and A. Del. Bimbo, "Interacting through eyes," *Robotics and Autonomous Systems,* vol. 19, no. 3-4, pp. 359-368, 1997.
http://dx.doi.org/10.1016/S0921-8890(96)00062-0

[9] Y. Ebisawa, "Improved video-based eye-gaze detection method," *IEEE Transactions on Instrumentation and Measurement,* vol. 47, no. 4, pp. 948-955, 1998.
http://dx.doi.org/10.1109/19.744648

[10] D. H. Yoo, J. H. Kim, B. R. Lee and M. J. Chung, "Non-contact eye gaze tracking system by mapping of corneal reflections," in *Proc. of Fifth IEEE International Conference on Automatic Face Gesture Recognition,* pp. 101-106, 2002.

[11] S. W. Shih and J. Liu, "A novel approach to 3-D gaze tracking using stereo cameras," *IEEE Transactions on Systems Man and Cybernetics Part B,* vol. 34, no. 1, pp. 234-245, 2004.
http://dx.doi.org/10.1109/TSMCB.2003.811128

[12] T. Ohno, N. Mukawa and A. Yoshikawa, "FreeGaze: a gaze tracking system for everyday gaze interaction," in *Proc. of the 2000 symposium on Eye tracking research & applications,* pp. 125-132, 2002.
http://dl.acm.org/citation.cfm?id=507098

[13] E. C. Lee, K. R. Park, M. C. Whang and J. Park, "Robust gaze tracking method for stereoscopic virtual reality system," *Lecture Notes in Computer Science,* vol. 4552, pp. 700-709, 2007.
http://dx.doi.org/10.1007/978-3-540-73110-8_76

[14] C. W. Cho, J. W. Lee, E. C. Lee and K. R. Park, "A robust gaze tracking method by using frontal viewing and eye tracking cameras," *Optical Engineering,* vol. 48, no. 12, pp. 127202-1-127202-15, 2009.
http://dx.doi.org/10.1117/1.3275453

[15] L. Kristian, "Measuring gaze point on handheld mobile devices," in *Proc. of ACM CHI 2004 Conference on Human Factors in Computing Systems,* pp. 1556, 2004.
http://dl.acm.org/citation.cfm?id=986132

[16] D. Heiko, D. L. Alexander and A. Schmidt, "Eye-gaze interaction for mobile phones," in *Proc. of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology,* pp. 364-371, 2007.
http://dl.acm.org/citation.cfm?id=1378122

[17] O. Gustav and K. Lundin, "Eye movement study of reading text on a mobile phone using paging, scrolling, leading, and RSVP," in *Proc. of the 6th International Conference on Mobile and Ubiquitous Multimedia,* pp. 176-183, 2007.
http://dl.acm.org/citation.cfm?id=1329493

[18] R. H. Thomas, E. Eva and L. O. Andreas, "Use your head: exploring face tracking for mobile interaction," in *Proc. of the SIGCHI conference on Human Factors in computing systems,* pp. 845-850, 2006.
http://dl.acm.org/citation.cfm?id=1125617

[19] E. C. Lee, Y. J. Ko and K. R. Park, "Gaze tracking based on active appearance model and multiple support vector regression on mobile devices," *Optical Engineering,* vol. 48, no. 7, pp. 077002-1-077002-11, 2009.
http://dx.doi.org/10.1117/1.3158992

[20] R. C. Gonzalez and R. E. Woods, "Digital Image Processing, 2nd Edition," *Prentice-Hall Inc,* 2002.

[21] J. R. Matey, O. Naroditsky, K. Hanna, R. Kolczynski, D. J. LoIacono, S. Mangru, M. Tinker, T. M. Zappia and W. Y. Zhao, "Iris on the move: acquisition of images for iris recognition in less constrained environments," *Proceedings of the IEEE,* vol. 94, no. 11, pp. 1936-1947, 2006.
http://dx.doi.org/10.1109/JPROC.2006.884091

[22] W. Peng, M. B. Green, J. Qiang and J. Wayman, "Automatic eye detection and its validation," in *Proc. Computer Vision and Pattern Recognition - Workshops,* pp. 164, 2005.

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1565482&tag=1

[23] E. C. Lee and S. H. Son, "Anti-spoofing method for iris recognition by combining the optical and textural features of human eye," *KSII Transactions on Internet and Information Systems,* vol. 6, no. 9, pp. 2424-2441, 2012.
http://dx.doi.org/10.3837/tiis.2012.09.026

[24] P. H. John, W. A. Allan and R. Peter, "Eye-gaze control of multimedia systems," in *Proc. of the Sixth International Conference on Human-Computer Interaction,* pp. 37-42, 1995.
http://dx.doi.org/10.1016/S0921-2647(06)80008-0

Eui Chul Lee received a BS degree in Software from Sangmyung University, Seoul, South Korea in 2005. He received MS and Ph.D. degrees in Computer Science from Sangmyung University in 2007 and 2010, respectively. He was Researcher in Division of Fusion and Convergence of Mathematical Sciences at the National Institute for Mathematical Sciences from March 2010 to February 2012. Since March 2012, he has been Assistant Professor in the Department of Computer Science at Sangmyung University, Seoul, Korea. His research interests include computer vision, image processing, pattern recognition, Brain Computer Interface (BCI), and Human Computer Interface (HCI).

Min Woo Park received a BS degree in Computer Science from Sangmyung University, Seoul, South Korea in 2012. Currently, he has been master candidate course student of Department of Computer Science from Sangmyung University since September 2012. His research interests include Pattern Recognition and Human Computer Interface (HCI).