Where are you looking at? – Feature-Based Eye Tracking on Unmodified Tablets

Shoya Ishimaru, Kai Kunze, Yuzuko Utsumi, Masakazu Iwamura, Koichi Kise
Dept. of Computer Science and Intelligent Systems
Osaka Prefecture University
{*ishimaru,kunze*}*@m.cs.osakafu-u.ac.jp* {*utsumi,iwamura,kise*}*@m.cs.osakafu-u.ac.jp*

***Abstract*—This paper introduces our work towards implementing eye tracking on commodity devices. We describe our feature-based approach and the eye tracking system working on a commodity tablet. We recorded data of 5 subjects following an animation on screen as reference. On the assumption that the position of device and user's head is stable, the average distance error between estimated gaze point to actual gaze point is around 12.23 [mm] using user-dependent training.**

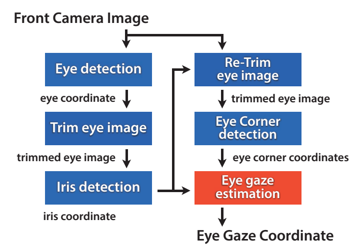

Figure 1. Overview of our proposed method and implementation.

I. Introduction

Eye gaze can give us unique insides about a person, especially as it is directly connected to cognition and other brain functions. Therefore there is more and more research focusing on eye tracking [1], [4], [2], [3]. However, the usage of existing dedicated eye trackers is still quite limited, as they are very expensive. We wonder if it is possible to use commodity tablets and phones to detect the eye gaze of the user while he's looking on the screen of the device. The contributions of the paper are as follows: (1) we present an feature-based eye tracking system working on a tablet. (2) We show that our system works with an average distance error of 12.23 [mm] for 5 users with a user dependent training phase, on the assumption that the position of device and user's head is stable.

II. Method and Implementation

We introduce our feature-based method and discuss the implementation to estimate user's eye gaze on mobile tablets. Our eye tracking system is divided into two-phases. First we detect the eye features. Second we estimate the eye gaze coordinate using linear regression.

Feature detection: In the feature detection phase, we detect the eye features (coordinates of iris and eye corners) from the image taken by front camera. The detailed flow chart is depicted in Figure1. We use the CIDetector library (an iOS built-in face detection library) as a rough detection to reduce the processing cost and then improve the iris detection by a more fine-grained method. After trimming, we detect coordinates of inner corner and outer corner using the Harris corner detection from OpenCV [5].

Eye gaze estimation: In gaze estimation phase, the coordinates of the user's gaze are calculated by regression. We use the x and y image coordinates of right eye's iris as (RI_x,RI_y) , left eye's iris as (LI_x,LI_y), right eye's inner corner as (RC_x,RC_y) and left eye's inner corner as (LC_x,LC_y). The x and y values of eye gaze coordinate are calculated in Equations (1,2).

$$x = a_1(RI_x - RC_x + LI_x - LC_x) + b_1 \qquad (1)$$

$$y = a_2(RI_y + LI_y)/2 + b_2 \qquad (2)$$

a_1, a_2 are the coefficients and b_1, b_2 are scalars. Both are estimated using regression.

III. Experimental Setup

The experiments are conducted on the Apple iPad4, with a 1 GHz dual-core A6X processor, 1.0 GB memory, and a screen which resolution is 2048 * 1536. The screen's width is 147.8[mm], the height is 19.71[mm]. The built-in front camera has a resolution of Quad-VGA (1280 * 960) as maximum. In this experiment, we used VGA (640 * 480) images with 132 dpi.

978-1-4799-2190-4/13 $26.00 © 2013 IEEE
DOI 10.1109/ACPR.2013.190

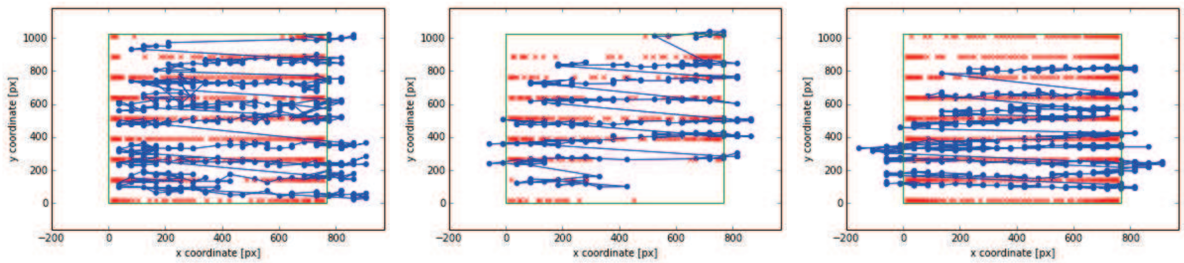

Figure 2. Ground truth and the actual gaze points as given by the animated dot for some experiments. The square is the edge of iPad screen. The 'x' marks are the coordinates of actual gaze point and 'o' marks are the coordinates of estimated gaze point by the system. as given by the animated dot for some experiments. The square is the edge of iPad screen. The 'x' marks are the coordinates of actual gaze point and 'o' marks are the coordinates of estimated gaze point by the system. the left figure shows a correct estimation, the middle depicts problems when the inner eye corner cannot be detected correctly, the right shows problems due to vertical head movement.

There are 5 test subjects with two experimental runs per participant: One for training and the other for testing. The distance between user's face to iPad is fixed to 30[cm]. To establish a reliable performance baseline, we used a portable easel to fix the iPad. For each experimental run, the subjects keep watching a dot moving on the screen. The dot moves from left to right. It jumped to next line and moved from left to right again. We record the coordinates of the dot and images from front camera. Since subjects kept watching the dot, its coordinates on screen are very close to the actual gaze point. We validated it with the SMI mobile eye tracker for a couple of trials.

IV. Results and Discussion

We evaluate our system using the distance error (Euclidean distance) between actual gaze point and the estimated gaze point. When we set the coordinates of actual gaze point as (AG_x, AG_y) and the coordinates of estimated gaze point as (EG_x, EG_y), error E [mm] is calculated as follows from Equation (3). dpi is the resolution of the taken image. 1 [inch] is 25.4 [mm].

$$E = \sqrt{(EG_x - AG_x)^2 + (EG_y - AG_y)^2} * 25.4/dpi \tag{3}$$

Figure2 shows the ground truth and the actual gaze points as given by the animated dot. The mean error for each subject are 10.74[mm], 24.04[mm], 8.36[mm], 13.47[mm] and 4.54[mm]. The average is 12.23 [mm]. There are two problems related to the experimental recordings: one related to eye corner detection, the other related to head movement (seen in Figure 2).

V. Conclusion and Future Work

In this paper, we present a method to estimate eye gaze on an unmodified tablet using only the front facing camera. We propose a feature-based system using regression. From the initial experimental result, the accuracy of our system is promising assuming fixed head and tablet positions. Most importantly, the next experiments will focus on more natural reading and tablet usage scenarios, using a stationary eye tracker for ground truth.

References

[1] R. Biedert, J. Hees, A. Dengel, and G. Buscher. A robust realtime reading-skimming classifier. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, pages 123–130. ACM, 2012.

[2] K. Kunze, A. Bulling, Y. Utsumi, S. Yuki, and K. Kise. I know what you are reading – recognition of document types using mobile eye tracking. In *in Proc. of ISWC 2013*, 2013.

[3] K. Kunze, S. Ishimaru, Y. Utsumi, and K. Kise. My reading life – towards utilizing eyetracking on unmodified tablets and phones. In *Adjunct Proceedings at UbiComp*, pages 455–464, 2013.

[4] W. Sewell and O. Komogortsev. Real-time eye gaze tracking with an unmodified commodity webcam employing a neural network. In *CHI'10 Extended Abstracts*, pages 3739–3744. ACM, 2010.

[5] V. Vezhnevets and A. Degtiareva. Robust and accurate eye contour extraction. In *Proc. Graphicon*, pages 81–84. Citeseer, 2003.