Cursor Control by Point-of-Regard Estimation for a Computer With Integrated Webcam

Stefania Cristina, Kenneth P. Camilleri

Department of Systems and Control Engineering
University of Malta, Malta
Email: stefania.cristina@um.edu.mt
kenneth.camilleri@um.edu.mt

***Abstract*—The problem of eye-gaze tracking by videooculography has been receiving extensive interest throughout the years owing to the wide range of applications associated with this technology. Nonetheless, the emergence of a new paradigm referred to as pervasive eye-gaze tracking, introduces new challenges that go beyond the typical conditions for which classical video-based eye-gaze tracking methods have been developed. In this paper, we propose to deal with the problem of point-of-regard estimation from low-quality images acquired by an integrated camera inside a notebook computer. The proposed method detects the iris region from low-resolution eye region images by its intensity values rather than the shape, ensuring that this region can also be detected at different angles of rotation and under partial occlusion by the eyelids. Following the calculation of the point-of-regard from the estimated iris center coordinates, a number of Kalman filters improve upon the noisy point-of-regard estimates to smoothen the trajectory of the mouse cursor on the monitor screen. Quantitative results obtained from a validation procedure reveal a low mean error that is within the footprint of the average on-screen icon.**

Keywords–Point-of-regard estimation; Eye-gaze tracking; Iris center localization.

I. Introduction

The idea of estimating the human eye-gaze has been receiving increasing interest since at least the 1870s [1], following the realization that the eye movements hold important information that relates to visual attention. Throughout the years, efforts in improving eye-gaze tracking devices to minimize discomfort and direct contact with the user led to the conception of videooculography (VOG), whereby the eye movements are tracked remotely from a stream of images that is captured by digital cameras. Eye-gaze tracking by VOG quickly found its way into a host of applications, ranging from human-computer interaction (HCI) [2], to automotive engineering [3][4]. Indeed, with the advent of the personal computer, eye-gaze tracking technology was identified as an alternative controlling medium enabling the user to operate the mouse cursor using the eye movements alone [2].

Following the emergence and widespread use of highly mobile devices with integrated imaging hardware, there has been an increasing interest in mobile eye-gaze tracking that blends well into the daily life setting of the user [5]. This emerging interest led to the conception of a new paradigm that is referred to as *pervasive tracking*, which term refers to the endeavor for tracking the eye movements continuously in different real-life scenarios [5]. This notion of pervasive eye-gaze tracking is multi-faceted, typically characterized by different aspects such as the capability of a tracking platform to permit tracking inside less constrained conditions, to track the user remotely and unobtrusively and to integrate well into devices that already comprise imaging hardware without necessitating hardware modification.

Nevertheless, this new paradigm brings challenges that go beyond the typical conditions for which classical video-based eye-gaze tracking methods have been developed. Despite considerable advances in the field of eye-gaze tracking as evidenced by an abundance of methods proposed over the years [6], video-based eye-gaze tracking has been mainly considered a desktop technology, often requiring specific conditions to operate. Commercially available eye-gaze tracking systems, for instance, are usually equipped with high-grade cameras and actively project infra-red illumination over the face and the eyes to obtain accurate eye movement measurements. In utilizing specialized hardware to operate, active eye-gaze tracking fails to integrate well into devices that already comprise imaging hardware, while its usability is constrained to controlled environments away from interfering infra-red sources. On the other hand, passive eye-gaze tracking which operates via standard imaging hardware and exploits the appearance of the eye without relying on specialized illumination sources for localization and tracking, provides a solution that promises to integrate better into pervasive scenarios.

Nonetheless, utilizing existing passive eye-gaze tracking methods to address the challenges associated with pervasive tracking, such as the measurement of eye movement from low-quality images captured by lower-grade hardware, may not necessarily be a suitable solution. For instance, existing shape-based methods that localize the eye region inside an image frame by fitting curves to its contours, often require images of suitable quality and good contrast in which the boundaries between different components such as the eyelids, the sclera and the iris are clearly distinguishable [7]–[10]. Similarly, feature-based methods that search for distinctive features such as the limbus boundary [11][12], necessitate these features to be clearly identifiable. It has been reported in [13], that appearance-based methods relying on a trained classifier, such as a Support Vector Machine (SVM), to estimate the 2D point-of-regard directly from an eye region image without identifying its separate components, perform relatively well on lower-quality images as long as the training data includes images of similar quality as well. However, this performance

Copyright (c) IARIA, 2014. ISBN: 978-1-61208-354-4

usually comes at the cost of lengthy calibration sessions that serve to gather the user-dependent data that is required for training [14][15]. Moreover, recent attempts to track the eye-gaze on mobile platforms by existing eye-gaze tracking methods [16][17], have reported undesirable constraints such as a requirement for close-up eye region images [18] and lengthy calibration sessions [16].

In light of the challenges associated with pervasive tracking, we propose a passive eye-gaze tracking method to estimate the point-of-regard (POR) on a monitor screen from lower-quality images acquired by an integrated camera inside a notebook computer. To localize the iris center coordinates from low-resolution eye region images while the user sits at a distance from the monitor screen, we propose an appearance-based method that localizes the iris region by its intensity values rather than the shape. In addition, our method ensures that the iris region can be located at different angles of rotation and under partial occlusion by the eyelids, and can be automatically relocated after this has been entirely occluded during blinking. Following iris localization, the iris center coordinates extracted earlier are mapped to a POR on the monitor screen via linear mapping functions that are estimated through a brief calibration procedure. A number of Kalman filters finally improve upon the noisy POR estimates to smoothen the trajectory of the mouse cursor on the monitor screen.

This paper is organized as follows. Section II describes the details of the proposed passive eye-gaze tracking method. Section III presents and discusses the experimental results, while Section IV draws the final remarks which conclude the paper.

II. Method

The following sections describe the stages of the proposed method, starting off with eye region detection and tracking up to the estimation of the POR onto the monitor screen.

A. Eye Region Detection

The estimation of the POR on the monitor screen requires that the eye region is initially detected inside the first few image frames. Searching for the eye region over an entire image frame can be computationally expensive for a real-time application and can lead to the occurrence of several false positive detections. Therefore, prior to detecting the eye region, the bounding box that encloses the face region is detected first such that this constrains the search range for the eye region, reducing the searching time as well as the possibility of false positives. The eye region is subsequently detected within the area delimited by the boundaries of the face region.

Given the real-time nature of our application, we chose the Viola-Jones algorithm for rapid detection of the face and eye region [19]. Within the Viola-Jones framework, features of interest are detected by sliding rectangular windows of Haar-like operators over an image frame, subtracting the underlying image pixels that fall within the shaded regions of the Haar-like operators from the image pixels that fall within the clear regions. Candidate image patches are classified between positive and negative samples by a cascade of weak classifiers arranged in order of increasing complexity. Every weak classifier is trained to search for a specific set of Haar features by a technique called boosting, such that each stage processes the samples that pass through the preceding classifier and rejects the negative samples as early into the cascade as possible to ensure computational efficiency.

The face and eye region detection stages in our work utilize freely available cascades of classifiers that come with the OpenCV library [20], which had been previously trained on a wide variety of training images such that detection generalizes well across different users. Since the training data for these classifiers was mainly composed of frontal face and eye region samples, the user is required to hold a frontal head pose for a brief period of time until the face and eye regions have been successfully detected. In case multiple candidates are detected by the face region classifier, the proposed method chooses the candidate that is closest to the monitor screen characterized by the largest bounding box, and discards the others.

B. Eye Region Tracking

To allow for small and natural head movement during tracking without requiring the uncomfortable use of a chin-rest, the initial position of the eye region detected earlier needs to be updated at every image frame to account for its displacement in the x- and y-directions. While performing eye region detection on a frame-by-frame basis would be a possible solution to estimate the eye region displacement through an image sequence, such an approach would be sub-optimal in terms of computational efficiency for a real-time application. Therefore, assuming gradual and small head displacement, the eye region is tracked between successive image frames by template matching, using the last known position of the eye region inside the previous image frame to constrain the search area inside the next frame.

A template image of the eye region is captured and stored following earlier detection of this region by the Viola-Jones algorithm. The template image is then matched to the search image inside a window of fixed size, centered around the last known position of the feature of interest. Template matching utilizes the normalized sum of squared differences (NSSD) as a measure of similarity, denoted as follows,

$$NSSD(x,y) = \frac{\sum_{x',y'}[T(x',x') - I(x+x',y+y')]^2}{\sqrt{\sum_{x',y'} T(x',y')^2 \sum_{x',y'} I(x+x',y+y')}} \tag{1}$$

where T denotes the template image and I denotes the search image. A NSSD value of zero represents a perfect match between the template and search image, whereas a higher value denotes increasing mismatch between the two images. This permits the identification of the new position of the feature of interest, which is specified by the location inside the search image that gives the minimum NSSD value after template matching.

C. Iris Center Localization

The movement of the eyes is commonly represented by the trajectory of the iris or pupil center in a stream of image frames [6], and hence the significance of localizing the iris or pupil center coordinates after the eye region has been detected. Given the small footprint of the eye region inside the image space, we opt to localize the iris center coordinates rather than the pupil, since the iris occupies a larger area inside the eye region and can be detected more reliably.

While there exist different methods that permit localization of the iris region inside an image frame, not all of these methods are suitable for localizing the iris region from low-resolution images, especially if fine details such as the contours of different components of the eye [7]–[10] need to be clearly distinguishable. We propose an appearance-based method that segments the iris region via a Bayes' classifier to localize it. The Bayes' classifier is trained during an offline training stage to classify between iris and non-iris pixels based on their red channel value in the RGB color space. During tracking, intensity values of pixels residing within the eye region are classified as belonging to the iris region if their likelihood exceeds a pre-defined threshold value, θ:

$$\frac{p(x_r(i,j) \mid \varpi_{iris})}{p(x_r(i,j) \mid \varpi_{non-iris})} \geq \theta \tag{2}$$

where $p(x_r(i,j) \mid \varpi_{iris})$ denotes the class-conditional probability of observing a red-band measurement at pixel *(i, j)* knowing it belongs to the iris class, while $p(x_r(i,j) \mid \varpi_{non-iris})$ denotes the class-conditional probability of observing the same red-band measurement at pixel *(i, j)* knowing it belongs to the non-iris class. The resulting binary image contains a blob of pixels that belongs to the iris region, whose center of mass is taken to represent the iris center coordinates. In case the eyebrow is also mistakenly classified as belonging to the iris region due to the resemblance in color with dark irises, the blob of pixels that is closer to the center of the eye region is considered to represent the iris.

The Bayes' classifier had been previously used for skin region segmentation in images [21], but to our knowledge it has never been adopted to the problem of iris region localization for eye-gaze tracking until our work. Preliminary results have shown this method to be suitable in localizing the iris region from low-quality images, owing especially to the fact that the proposed localization method depends upon statistical color modeling rather than geometrical information. Another advantage that is also related to its independency from geometrical information is the ability to locate the iris region at different angles of rotation and under partial occlusion by the eyelids. The main downside of this method is its susceptibility to illumination variations, which problem is however alleviated by training the Bayes' classifier on iris and non-iris pixels acquired under different illumination conditions.

D. POR Estimation

Having determined the iris center coordinates, the final stage seeks to map these coordinates to screen coordinates in order to estimate the user's POR on the monitor screen.

For simplicity, we assume the iris center in the image space to displace along a flat plane, such that we can define a linear mapping relationship between the image and screen coordinates as follows,

$$(\mathbf{x}_s^{(3)} - \mathbf{x}_s^{(1)}) = \frac{(\mathbf{x}_s^{(2)} - \mathbf{x}_s^{(1)})}{(\mathbf{x}_i^{(2)} - \mathbf{x}_i^{(1)})}(\mathbf{x}_i^{(3)} - \mathbf{x}_i^{(1)}) \tag{3}$$

where $\mathbf{x}_s^{(1)}$ and $\mathbf{x}_s^{(2)}$ denote the screen coordinates of two calibration points respectively, whereas $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$ denote the corresponding iris center coordinates inside the eye region which are estimated while the user fixates at the two calibration points. During tracking, the mapping function in (3) computes the displacement in screen coordinates between the new POR $\mathbf{x}_s^{(3)}$ and the calibration point $\mathbf{x}_s^{(1)}$, following the estimation of the displacement in image coordinates between the new iris center location $\mathbf{x}_i^{(3)}$ and the previously estimated $\mathbf{x}_i^{(1)}$. In order to compensate for the assumption of planar iris movement, the monitor screen is divided into four separate quadrants by strategically placed calibration points as illustrated in Figure 1, such that each quadrant is assigned different parameter values that best describe the linear mapping between the image-to-screen coordinates.

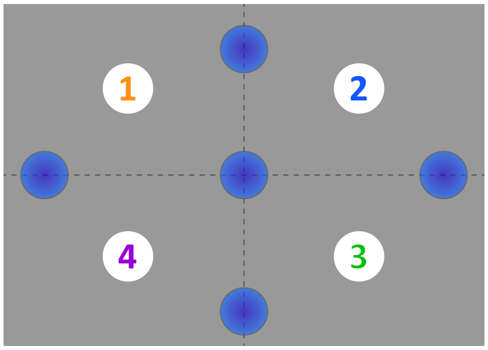

Figure 1. Strategically placed calibration points divide the screen display into four separate quadrants.

To alleviate the issue of noisy iris center estimations from low-quality images, and hence smoothen the trajectory of the mouse cursor on the monitor screen after mapping the iris center coordinates to a POR, we propose to use a Kalman filter to improve upon these noisy measurements. Indeed, the Kalman filter is an algorithm that recursively utilizes noisy measurements observed over time to produce estimates of desired variables that tend to be more accurate than the single measurements alone [22]. We define the Kalman filter parameters for our specific application of smoothing the mouse cursor trajectory as follows:

State Vector: We define the state vector $\mathbf{x}_{k+1}$ as,

$$\mathbf{x}_{k+1} = [\Delta x_s \quad \Delta y_s]^T \tag{4}$$

where Δx_s denotes the horizontal on-screen displacement, $(x_s^{(3)} - x_s^{(1)})$, and similarly for the vertical on-screen displacement, Δy_s.

Transition Matrix: Assuming the eye movement during tracking to consist of fixation periods and smooth movement between one visual stimulus and another, we represent the transition matrix A_{k+1} by a simple linear model of the ideal mouse cursor trajectory during fixations and shifts between visual stimuli as follows,

$$A_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

Measurement Vector: In our work, the measurement vector $\mathbf{z}_{k+1}$ holds the estimated displacement of the iris center in image coordinates and is defined as,

$$\mathbf{z}_{k+1} = [\Delta x_i \quad \Delta y_i]^T \tag{6}$$

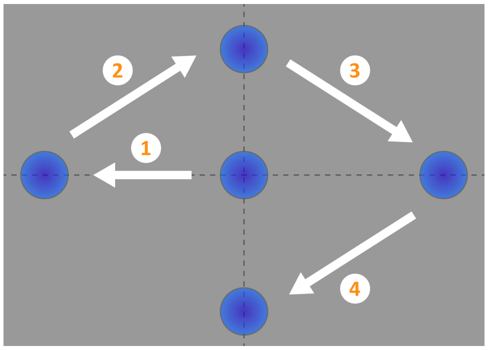

Figure 2. Five visual stimuli were displayed in succession on the monitor screen during a brief calibration procedure in order to collect image-screen coordinate pairs.

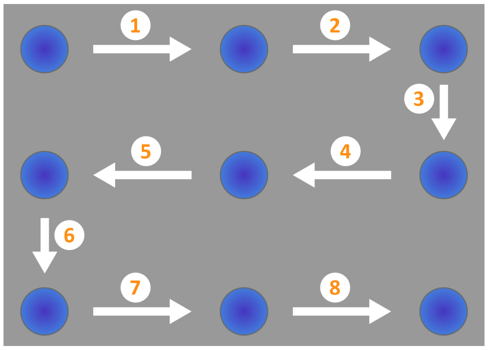

Figure 3. A validation session consisting of nine visual stimuli displayed in succession served to calculate the error of the estimated PORs.

where Δx_i represents the horizontal image displacement, $(x_i^{(3)} - x_i^{(1)})$, and similarly for the vertical image displacement, Δy_i.

Measurement Matrix: The measurement matrix defines the relationship that maps the true state space onto the measurement. In our work, the values that populate the measurement matrix can be derived from (3), such that this matrix maps the screen coordinates onto the image coordinates,

$$H_{k+1} = \begin{bmatrix} \frac{(x_i^{(2)} - x_i^{(1)})}{(x_s^{(2)} - x_s^{(1)})} & 0 \\ 0 & \frac{(y_i^{(2)} - y_i^{(1)})}{(y_s^{(2)} - y_s^{(1)})} \end{bmatrix} \tag{7}$$

Measurement Noise and Process Noise: The measurement noise is represented by vector, $\mathbf{v}_{k+1} = [v_{k+1}^x v_{k+1}^y]$, characterized by standard deviations δ_{v_x} and δ_{v_y} in the x- and y-directions respectively, and similarly the process noise is represented by vector, $\mathbf{w}_{k+1} = [w_{k+1}^x w_{k+1}^y]$, characterized by standard deviations δ_{w_x} and δ_{w_y} in the respective x- and y-directions. The process noise is taken to represent the characteristics inherent to the visual system itself, such that the standard deviations δ_{w_x} and δ_{w_y} are therefore set to a low value to model the small, microsaccadic movements performed by the eye during periods of fixation. Values for the standard deviations, δ_{v_x} and δ_{v_y}, that adequately smooth the mouse cursor trajectory after the estimation of noisy iris center measurements were found experimentally.

Separate Kalman filters are assigned to every screen quadrant, with each filter being characterized by a different measurement matrix corresponding to the screen quadrant for which it is responsible. During tracking, all Kalman filters are updated online to produce an estimate of the POR following the estimation of the iris center coordinates, such that the on-screen position of the mouse cursor can subsequently be updated according to the Kalman filter estimate that corresponds to the quadrant of interest. In updating the Kalman filters at every time step, we ensure a smooth hand over between one filter and another as the mouse cursor trajectory crosses over adjacent screen quadrants.

III. Experimental Results and Discussion

To evaluate the proposed eye-gaze tracking method, a group of five participants consisting of two females and three males with a mean age of 38.2 and standard deviation of 15.9, were recruited for an experimental session. All participants were proficient computer users without any prior experience in the field of eye-gaze tracking, except one who was already accustomed to the technology. The experimental procedure was carried out on a 15.6” notebook display while each participant was seated inside a well-lit indoor environment at an approximate distance of 60 cm from the monitor screen and the camera. Image data was acquired by the webcam that was readily available on-board the notebook computer.

Following detection and tracking of the eye region, and iris center localization, each participant was requested to sit through a brief calibration procedure that served to estimate the mapping functions required to transform the iris center coordinates into a POR on the monitor screen. During the calibration procedure, the participants were instructed to fixate at five visual stimuli appearing in succession on the monitor screen as shown in Figure 2, and requested to minimize their head movement such that pairs of image-screen coordinates were collected. The five visual stimuli were positioned strategically in order to divide the screen into four separate quadrants, as illustrated in Figure 1. A different mapping function was estimated for each quadrant according to the relationship between the image and screen coordinates collected earlier.

Every participant was then requested to sit through a validation procedure that served to estimate the error between the estimated POR and ground truth data. The validation procedure consisted of nine visual stimuli which were evenly spread throughout the monitor screen and displayed in succession as shown in Figure 3. The participants were instructed to move the mouse cursor with their eyes as close to each visual stimulus as possible and hold its position for a brief period of time such that the on-screen coordinates of the mouse cursor were recorded, as shown in Figure 4 for one of the participants. During the validation procedure, the participants were allowed a small degree of head movement and were requested to displace their head in a natural way. Table I displays the mean and standard deviation of the error in pixels for each participant in the x- and y-directions.

By analyzing the results in Table I, it can be observed that in all cases the mean and standard deviation of the error in the x-direction exceeds the error in the y-direction. The main source for this discrepancy in error relates to inaccuracies in estimating the iris center coordinates. For instance, it was noted that despite retaining similar surrounding conditions between different participants to reduce any bias in the results, the

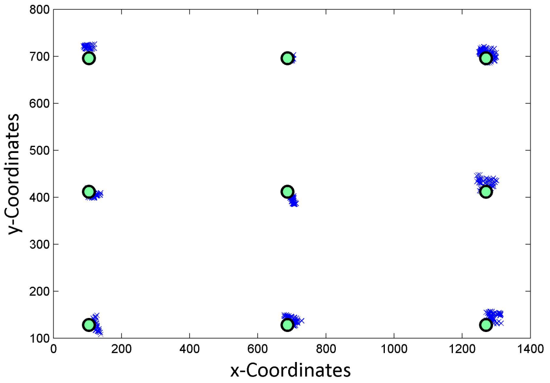

Figure 4. Validation result for one of the participants showing the displayed visual stimuli (green) and the estimated on-screen PORs (blue).

TABLE I. MEAN AND STANDARD DEVIATION OF THE ERROR IN PIXELS IN THE X- AND Y-DIRECTIONS, OF THE ESTIMATED ON-SCREEN POR COORDINATES.

Participant Number	Mean (x, y)	Standard Deviation (x, y)
1	(62.47, 19.63)	(130.89, 14.14)
2	(41.88, 13.63)	(132.76, 10.82)
3	(88.37, 45.62)	(177.96, 79.82)
4	(47.21, 16.89)	(107.75, 24.10)
5	(36.57, 17.10)	(109.19, 34.81)

accuracy of the collected data was subject to the anatomical characteristics of the participants. The protruding ridge of the brow above the eye was more accentuated for some participants rather than others, and this tended to create a dark shadow around the inner corner of the eye which was at times incorrectly segmented along with the iris region. Dark colored pixels belonging to the eyelashes were also often segmented with the iris region as shown in Figure 5, due to their close resemblance in color to dark brown irises. This erroneous inclusion of pixels which do not belong to the iris region served to shift the center of mass of the segmented blob of pixels horizontally towards the inner or outer eye corners, away from the true iris center. It was found that even a seemingly trivial error of a few pixels in the estimation of the iris center coordinates inside the image frame, could result in a significant error in the estimation of the POR at an approximate distance of 60 cm from the monitor screen.

The main source for the error in the y-direction could be the less than ideal positioning of the webcam at the top of the monitor screen, in relation to the positioning of the eyes as the user sits in front of the display. Indeed, commercial systems usually place the tracking device below the monitor screen in order to capture a better view of the visible portion of the eyeball that is not concealed below the eyelid. Being situated at the top of the screen, the webcam that is utilized in our work captures a smaller portion of the iris especially when the user gazes downwards, partially occluding the iris region below the eyelid and potentially introducing an error in the estimation of the iris center coordinates. It is, however, worth noting that the proposed method for iris region segmentation was equally capable of detecting the iris region under partial occlusion by the eyelids as shown in Figure 6, and therefore suitably alleviated the issue of the less than ideal positioning of the webcam.

In order to put the error values tabulated in Table I into context, the mean error in pixels across all participants was calculated and compared to the resolution of the computer screen. Given that the resolution of the monitor screen is equal to 1366 × 768 pixels, a mean error value of (55.30, 22.57) constitutes around 4% and 3% of the screen resolution in the horizontal and vertical directions respectively. Also, at a distance of 60 cm away from the monitor screen, a mean error of (55.30, 22.57) pixels corresponds to (1.46°, 0.71°) in visual angle. If the average on-screen icon is taken to have an average size of 45 × 45 pixels, the mean error that is achieved through the proposed method can be considered to be within the footprint of the average on-screen icon and therefore applicable to an HCI scenario.

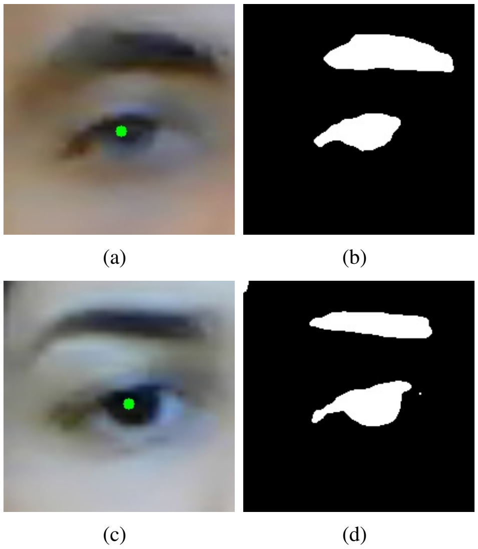

Figure 5. Inclusion of eyelashes in the segmented iris regions in Figures (b) and (d) corresponding to the eye region images in Figures (a) and (c).

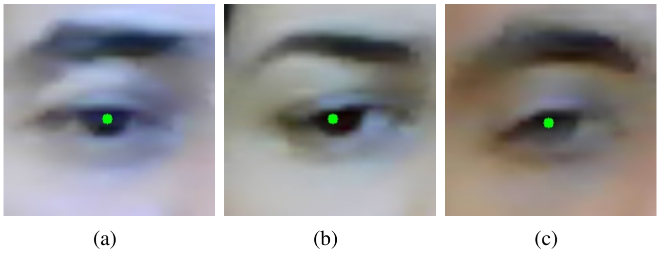

Figure 6. The proposed method for iris region segmentation was capable of localizing the iris center coordinates marked in green, under partial occlusion of the iris region by the eyelids.

IV. CONCLUSION

In this paper, we have proposed a passive eye-gaze tracking method to estimate the POR on a monitor screen from low-quality image data acquired by an integrated camera inside a notebook computer. Following eye region detection and tracking, we proposed an appearance-based method which allows the localization of the iris center coordinates from low-resolution eye region images. The iris center coordinates were subsequently mapped to a POR on the monitor screen by linear mapping functions, following which each POR estimate was improved by Kalman filters to smoothen the mouse cursor trajectory.

The experimental results obtained following a validation procedure revealed a noticeable discrepancy between the error in the x- and y-directions, with the error in the x-direction being the dominant between the two. The source for this error was observed to be the incorrect segmentation of pixels belonging to shadow or artifacts, such as the eyelashes, along with the iris region, producing a horizontal shift away from the true iris center. Nonetheless, the proposed method for iris region segmentation was capable of detecting the iris region under partial occlusion by the eyelids, especially when the user gazes downwards, permitting the estimation of the POR in less than ideal conditions. It is noteworthy to mention that despite the availability of low-resolution eye region images, the proposed method achieved a relatively low mean error of (1.46°, 0.71°) in visual angle. Future work aims to compensate for head movement inside the mapping functions, which map the iris center coordinates to a POR on the monitor screen, in order to permit larger head movement during tracking.

Acknowledgment

This work forms part of the project *Eye-Communicate* funded by the Malta Council for Science and Technology through the National Research & Innovation Programme (2012) under Research Grant No. R&I-2012-057.

References

[1] R. J. K. Jacob, “What you look at is what you get: eye movement-based interaction techniques,” in Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, 1990, pp. 11–18.

[2] J. L. Levine, “An eye-controlled computer,” in IBM Thomas J. Watson Research Center Res. Rep. RC-8857, Yorktown Heights, N.Y., 1981.

[3] S. J. Lee, J. Jo, H. G. June, K. R. Park, and J. Kim, “Real-Time Gaze Estimator Based on Driver’s Head Orientation for Forward Collision Warning System,” IEEE Transactions on Intelligent Transportation Systems, vol. 12, 2011.

[4] M. Shahid, T. Nawaz, and H. A. Habib, “Eye-Gaze and Augmented Reality Framework for Driver Assistance,” Life Science Journal, vol. 10, 2013, pp. 1571–1578.

[5] A. Bulling, A. T. Duchowski, and P. Majaranta, “PETMEI 2011: The 1st International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction,” in Proceedings of the 13th International Conference on Ubiquitous Computing: UbiComp 2011, 2011.

[6] D. W. Hansen and Q. Ji, “In the Eye of the Beholder: A Survey of Models for Eyes and Gaze,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, 2010, pp. 478–500.

[7] R. Valenti, A. Lablack, N. Sebe, C. Djeraba, and T. Gevers, “Visual Gaze Estimation by Joint Head and Eye Information,” in 20th International Conference on Pattern Recognition, Aug. 2010, pp. 3870–3873.

[8] R. Valenti, N. Sebe, and T. Gevers, “Combining Head Pose and Eye Location Information for Gaze Estimation,” IEEE Transactions on Image Processing, vol. 21, 2012, pp. 802–815.

[9] W. Haiyuan, Y. Kitagawaa, and T. Wada, “Tracking Iris Contour with a 3D Eye-Model for Gaze Estimation,” in Proceedings of the 8th Asian Conference on Computer Vision, Nov. 2007, pp. 688–697.

[10] T. Moriyama, T. Kanade, J. Xiao, and J. F. Cohn, “Meticulously Detailed Eye Region Model and Its Application to Analysis of Facial Images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, 2006.

[11] F. Timm and E. Barth, “Accurate Eye Centre Localisation by Means of Gradients,” in Proceedings of the Sixth International Conference on Computer Vision Theory and Applications, Mar. 2011, pp. 125–130.

[12] E. G. Dehkordi, M. Mahlouji, and H. E. Komleh, “Human Eye Tracking Using Particle Filters,” International Journal of Computer Science Issues, vol. 10, 2013, pp. 107–115.

[13] J. Mansanet, A. Albiol, R. Paredes, J. M. Mossi, and A. Albiol, “Estimating Point of Regard with a Consumer Camera at a Distance,” Pattern Recognition and Image Analysis, 2013, pp. 881–888.

[14] W. Sewell and O. Komogortsev, “Real-Time Eye Gaze Tracking With an Unmodified Commodity Webcam Employing a Neural Network,” in Proceedings of the 28th International Conference Extended Abstracts on Human Factors in Computing Systems, Apr. 2010, pp. 3739–3744.

[15] R. Stiefelhagen, J. Yang, and A. Waibel, “Tracking Eyes and Monitoring Eye Gaze,” in Proceedings of the Workshop on Perceptual User Interfaces, Oct. 1997, pp. 98–100.

[16] C. Holland and O. Komogortsev, “Eye Tracking on Unmodified Common Tablets: Challenges and Solutions,” in Proceedings of the Symposium on Eye Tracking Research and Applications, Mar. 2012, pp. 277–280.

[17] K. Kunze, S. Ishimaru, Y. Utsumi, and K. Kise, “My Reading Life - Towards Utilizing Eyetracking on Unmodified Tablets and Phones,” in Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing, 2013, pp. 283–286.

[18] Y. Zhang, A. Bulling, and H. Gellersen, “Towards pervasive eye tracking using low-level image features,” in Proceedings of the Symposium on Eye Tracking Research and Applications, 2012, pp. 511–518.

[19] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jul. 2001, pp. 1231–1238.

[20] “OpenCV,” 2014, URL: http://http://opencv.org/ [accessed: 2014-04-21].

[21] V. Vezhnevets, V. Sazonov, and A. Andreeva, “A Survey on Pixel-Based Skin Color Detection Techniques,” in Proceedings of the GraphiCon, 2003, pp. 85–92.

[22] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” Journal of Basic Engineering, 1960, pp. 35–45.

View publication stats