Eye Gaze Tracking System Using .Net

Madhu M Nayak[1], Usha Rani J[2], Anandhi G[3]
Department of CSE
[1, 2, 3]Assistant Professor,GSSIETW, Mysuru

Abstract-*This paper addresses the eye gaze tracking problem using a low-cost and more convenient web camera in a desktop environment, as opposed to gaze tracking techniques requiring specific hardware, e.g., infrared high-resolution camera and infrared light sources, as well as a cumbersome calibration process. In the proposed method, we first track the human face in a real-time video sequence to extract the eye regions. Then, we combine intensity energy and edge strength to obtain the iris center and utilize the piecewise eye corner detector to detect the eye corner. We adopt a sinusoidal head model to simulate the 3-D head shape, and propose adaptive weighted facial features embedded in the pose from the orthography and scaling with iterations algorithm, whereby the head pose can be estimated. Finally, the eye gaze tracking is accomplished by integration of the eye vector and the head movement information. Experiments are performed to estimate the eye movement and head pose on the BioID dataset and pose dataset, respectively. In addition, experiments for gaze tracking are performed in real-time video sequences under a desktop environment. The proposed method is not sensitive to the light conditions. Experimental results show that our method achieves an average accuracy of around 1.28◦ without head movement and 2.27◦ with minor movement of the head.*

Keywords-Desktop environment, gaze tracking, head pose, illumination change, web camera.

I. INTRODUCTION

Eye gaze tracking has many potential attractive applications including human–computer interaction, virtual reality, and eye disease diagnosis. For example, it can help the disabled to control the computer effectively [1]. In addition, it can support controlling the mouse pointer with one's eyes so that the user can speed up the selection of the focus point. Moreover, the integration of user's gaze and face information can improve the security of the existing access control systems. Eye gaze has been used to study human cognition [2], memory [3] and multi-element target tracking task [4]. Along this line, eye gaze tracking is closely related with the detection of visual saliency, which reveals a person's focus of attention.

To accomplish the task of gaze tracking, a number of approaches have been proposed. The majority of early gaze tracking techniques utilize intrusive devices such as contact lenses [5] and electrodes [6], requiring physical contact with the users; such a method causes a bit of discomfort to users. Tracking the gaze with a head-mounted device such as headgear [7], [8] is less intrusive, but is inconvenient from a practical viewpoint. In contrast, video-based gaze tracking techniques that could pro-vide an effective nonintrusive solution are more appropriate for daily usage.

The video-based gaze approaches commonly use two types of imaging techniques: *infrared imaging* and *visible imaging*. The former needs infrared cameras and infrared light sources to capture the infrared images, while the latter usually utilizes high-resolution cameras for images (see Fig. 1). As infrared-imaging techniques utilize invisible infrared light sources to obtain the controlled light and a better contrast image, it can reduce the effects of light conditions, and produce a sharp contrast between the iris and pupil (i.e., bright-dark eye effect), as well as the reflective properties of the pupil and the cornea (PCCR) [9]–[12]. As a result, an infrared imaging-based method is capable of performing eye gaze tracking. Most of video-based approaches belong to this class. Unfortunately, an infrared-imaging-based gaze tracking system can be quite expensive. Other shortcomings include: 1) An infrared-imaging system will not be reliable under the disturbance of other infrared sources; 2) not all users produce the bright-dark effect, which can make the gaze tracker fail; and 3) the reflection of infrared light sources on glasses is still an issue.

Compared with the infrared-imaging approaches, visible-imaging methods circumvent the aforementioned problems without the need for the specific infrared devices and infrared light sources. They are not sensitive to the utilization of glasses and the infrared sources in the environment. Visible-imaging methods should work in a natural environment, where the ambient light is uncontrolled and usually results in lower contrast images. The iris center detection will become more difficult than the pupil center detection because the iris is usually partially occluded by the upper eyelid.

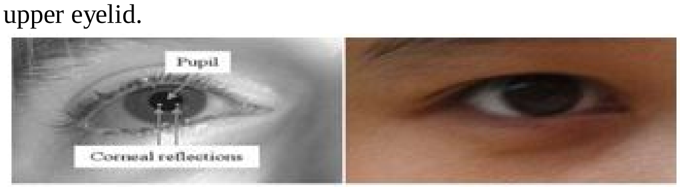

Fig. 1. (a) Image under infrared light [12]. (b) Image under visible light.

In this paper, we concentrate on visible-imaging and present an approach to the eye gaze tracking using a web camera in a desktop environment. First, we track the human face in a real-time video sequence to extract the eye region. Then, we combine intensity energy and edge strength to locate the iris center and utilize the piecewise eye corner detector to detect the eye corner. To compensate for head movement causing gaze error, we adopt a sinusoidal head model (SHM) to simulate the 3-D head shape, and propose an *adaptive-weighted facial features embedded inthe POSIT algorithm* (AWPOSIT), whereby the head pose can be estimated. Finally, eye gaze tracking is performed by the integration of the eye vector and head movement information.

II. RELATED STUDY

This section overviews feature-based and appearance based visible imaging gaze tracking methods. Feature-based gaze tracking relies on extracting the features of the eye region, e.g., the iris center and iris contour, which provide eye movement information. Zhu and Yang [13] performed feature extraction from an intensity image. The eye corner was extracted using a preset eye corner filter and the eye iris center was detected by the interpolated Sobel edge magnitude. Then, the gaze direction was determined through a linear mapping function. With that system, users must keep their head stable because the gaze direction is sensitive to the head pose. Valenti *et al.* [14] computed the eye location and head pose, and combined them. Torricelli *et al.* [15] utilized the iris and corner detection methods to obtainthe geometric features, which are mapped to screen coordinates by the general regression neural network (GRNN). In general, the estimated accuracy of the system relies heavily on the input vector of GRNN, and will deteriorate with error in any element of the input vector. Ince and Kim [16] developed a low-cost gaze tracking system, which utilized shape and intensity-based deformable eye pupil center detection and movement decision algorithms. Their system performed on low-resolution video sequences, but the accuracy was sensitive to head pose.

Appearance-based gaze tracking does not extract the features explicitly, but instead utilizes the whole image to estimate the gaze. Along this line, Sugano *et al.* [17] have presented an online learning algorithm within the incremental learning framework for gaze estimation, which utilized the user’s operations (i.e., mouse click) on the PC monitor. At each mouse click, they created a training sample with the mouse screen coordinate as the gaze label associated with the features (i.e., head pose and eye image). Therefore, it was cumbersome to obtain a large number of samples. In order to reduce the training cost, Lu *et al.* [18] have proposed a decomposition scheme, which included the initial estimation and subsequent compensations. Hence, the gaze estimation could perform effectively using the training samples. Nguyen [19] first utilized a new training model to detect and track the eye, and then employed the cropped image of the eye to train Gaussian process functions for gaze estimation. In their applications, a user has to stabilize the position of his/her head in front of the camera after the training procedure. Similarly, Williams *et al.* [20] proposed a sparse and semi-supervised Gaussian process model to infer the gaze, which simplified the process of collecting training data. However, many unlabeled samples are still utilized. Lu *et al.* [21] have proposed an eye gaze tracking system based on a local pattern model (LPM) and a support vector regressor (SVR). This system extracts texture features from the eye regions using the LPM, and feeds the spa-tial coordinates into the SVR to obtain a gaze mapping function. Lu *et al.* [22] introduced an adaptive linear regression model to mapping function, which describes the relationship between the eye vector and the gaze point on the screen. In Phases 1 and 2, a calibration process computes the mapping from the eye vector to the coordinates of the monitor screen. Phase 3 entails the head infer the gaze from eye appearance by utilizing fewer training samples.

III. PROPOSED METHOD

The most notable gaze features in the face image are the iris center and eye corner. The eyeball moves in the eye socket when looking at different positions on the screen. The eye corner can be viewed as a reference point, and the iris center in the eyeball changes its position that indicates the eye gaze. Therefore, the gaze vector formed by the eye corner and iris center contains the information of gaze direction, which can be used for gaze tracking. However, the gaze vector is sensitive to the head movement and produces a gaze error, while the head moves. Therefore, the head pose should be estimated to compensate for the head movement.

Our three-phase feature-based eye gaze tracking approach uses eye features and head pose information to enhance the accuracy of the gaze point estimation (see Fig. 2). In Phase 1, we extract the eye region that contains the eye movement information. Then, we detect the iris center and eye corner to form the eye vector. Phase 2 obtains the parameters for the mapping function, which describes the relationship between the eye vector and the gaze point on the screen. In Phases 1 and 2, a calibration process computes the mapping from the eye vector to the coordinates of the monitor screen. Phase 3 entails the head pose estimation and gaze point mapping. It combines the eye vector and head pose information to obtain the gaze point.

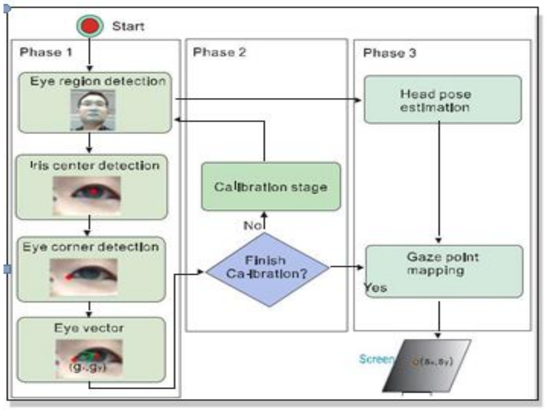

Fig 2. Three-phase feature-based Eye Gaze tracking method.

A. Eye Region Detection

To obtain the eye vector, the eye region should be located first. Traditional face detection approaches cannot provide accurate eye region information in uncontrolled light and with free head movement. Therefore, an efficient approach should address illumination and pose problems. Here, we present a two-stage method to detect the eye region.

In the first stage, we utilize local sensitive histograms [24] to cope with various lighting. Compared with normal intensity histograms, local sensitive histograms embed spatial information and decline exponentially with respect to the distance to the pixel location where the histogram is calculated. Examples of the utilization of local sensitive histograms are shown in Fig. 3, in which three images with different illuminations have been transformed to ones with consistent illumination via the local sensitive histograms.

In the second stage, we adopt the active shape model (ASM) [26] to extract facial features on the gray image, through which the illumination changes are eliminated.

Fig. 3.(a) Input images [25]. (b) Results using local sensitive histograms

The eye region can be extracted using an improved version of ASM. In Fig. 4, the eye region in each image is detected under the different illumination and head pose.

Fig. 4. (a) ASM results on the gray image. (b) Mapping ASM results to the original images and extracting the eye region

B. Eye Features Detection

In the eye region, the iris center and eye corner are the two notable features, by which we can estimate the gaze direction. Accordingly, the following two sections focus on the detection of iris center and eye corner.

1) Iris Center Detection: Once the eye region is extractedusing the previous steps, the iris center will be detected in the eye region. We first estimate the radius of the iris. Then, a combination of intensity energy and edge strength information is utilized to locate the iris center.

In order to estimate the radius, we first use the L_0 gradient minimization method [27] to smooth the eye region, which can remove noisy pixels and preserve the edges at the same time. Subsequently, a rough estimation of the iris center can be ob-tained by the color intensity. Then, a canny edge detector is used on the eye regions. There exist some invalid edges with short length. Hence, a distance filter is applied to remove the invalid edges that are too close to or too far away from the rough center of the iris. Random sample consensus (RANSAC) is utilized to estimate the parameters of the circle model for the iris. The radius r of the iris can be calculated after the RANSAC is applied to the edge points of the iris.

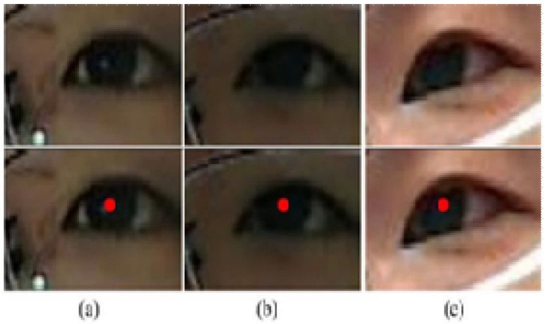

Fig. 5. First row shows the different eye regions, while the second row gives the detection results of the iris center

Fig. 5 illustrates the results of iris center detection, in which Fig. 5 (a)–(c) are in the same video sequence. That is, Fig. 5(a) is the first frame, where the iris center could be accurately detected using the proposed algorithm. Therefore, we obtain the radius of the iris which is taken as prior knowledge of the iris detection in the following frames. Accordingly, supposing the radius of the iris does not change with respect to the large distance between the user and he computer screen, we can detect the iris center of eye images as shown in Fig. 5(b) and (c).

2) Eye Corner Detection: Usually, the inner eye corner is viewed as a reference point for gaze estimation because it is insensitive to facial expression changes and eye status [28], and is more salient than the outer eye corner. Therefore, we should detect the inner eye corner to guarantee the gaze direction accuracy. We propose a piecewise eye corner detector based on the curvature scale space and template match rechecking method. We perform the procedures on the smoothed eye image
mentioned previously. A canny operator is used to generate the edge map; then edge contours are extracted from the edge map and small gaps are filled.

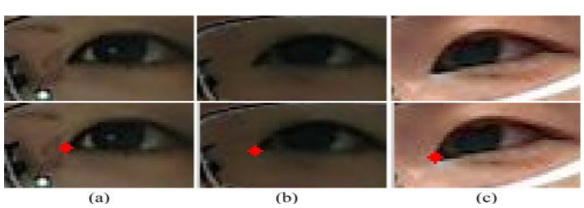

Fig. 6. First row: eye regions, and the second row: eye corner detection results.

C. Eye Vector and Calibration

When we look at the different positions on the screen plane, while keeping our head stable, the eye vector is defined by the iris center p_{iris} and eye corner p_{corner}, i.e., $g = p_{corner} - p_{iris}$. It provides the gaze information to obtain the screen coordinates by a mapping function. A calibration procedure is to present the user with a set of target points at which to look, while the corresponding eye vectors are recorded. Then, the relationship between the eye vector and the coordinates on the screen is determined by the mapping function. Different mapping functions can be utilized such as the simple linear model [13], SVR model [21], and polynomial model [29]. In practice, the accuracy of the simple linear model is not sufficient and SVR model requires more calibration data. Fortunately, the second-order polynomial function is a good compromise between the numbers of calibration points and the accuracy of the approximation [30]. In our calibration stage, the second-order polynomial function is utilized and the user is required to look at nine points (see Fig. 8); the eye vectors are computed and the corresponding screen positions are known. Then, the second-order polynomial can be used as mapping function, which calculates the gaze point on the screen, i.e., scene position, through the eye vector

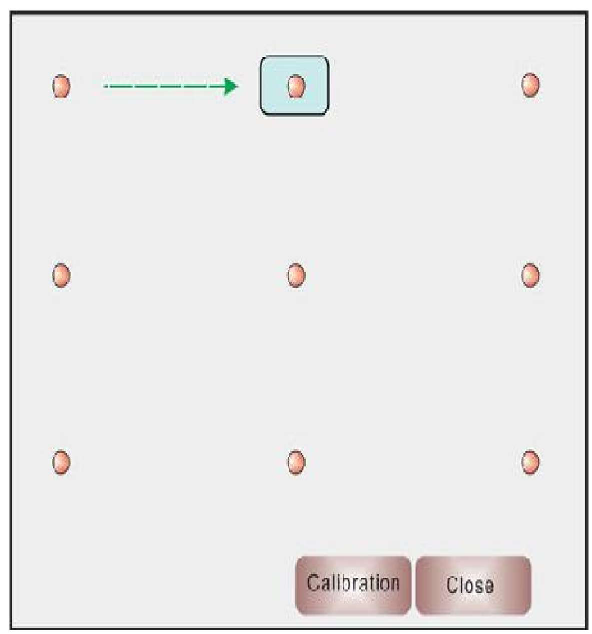

Fig. 7 . Nine positions on the screen (1024 × 1280 pixels).

$$u_x = a_0 + a_1 g_x + a_2 g_y + a_3 g_x g_y + a_4 g_x^2 + a_5 g_y^2$$
$$u_y = b_0 + b_1 g_x + b_2 g_y + b_3 g_x g_y + b_4 g_x^2 + b_5 g_y^2$$

where (u_x, u_y) is the screen position, (g_x, g_y) is the eye vector,$(a_1, \ldots, a_5)$ and $(b_1, \ldots, b_5)$ are the parameters of mapping function that can be solved using the least square method. We have quantified the projection error on the computer screen, and found that a pixel deviation of the iris center or the eye corner would lead to approximately 100 pixels deviation on the screen. Accordingly, utilizing the mapping function, the user's gaze point can be calculated efficiently in each frame.

D. Head Pose Estimation

This module handles the facial feature tracking and the head pose estimation approach in video sequences. The pose estimation approaches uses a stereo camera and 3-D data for head shape or limited head rotation. Usually, the human head can be modeled as an ellipsoid or a cylinder for simplicity, with the actual width and radii of the head for measures. Some use the cylindrical head model (CHM) to estimate head pose , which can perform in real time and track the state of the head roughly. The system utilize the 3-D head shape because the ellipsoid and cylinder do not highlight facial features. The SHM can better approximate the shape of different faces.

IV. CONCLUSION

A model for gaze tracking has been constructed using a web camera in a desktop environment. Its primary novelty is using intensity energy and edge strength to locate the iris center and utilizing the piecewise eye corner detector to detect the eye corner. Further, we have proposed the AWPOSIT algorithm to improve the estimation of the head pose. Therefore, the combination of the eye vector formed by the eye center, the eye corner, and head movement information can achieve improved accuracy and robustness for the gaze estimation. The experimental results have shown the efficacy of the proposed method.

REFERENCES

[1] A. T. Duchowski, “A breadth-first survey of eye-tracking applications,” Behavior Res. Methods, Instrum., Comput., vol. 34, no. 4, pp. 455–470, 2002.

[2] S. P. Liversedge and J. M. Findlay, “Saccadic eye movements and cognition,” Trends Cognitive Sci., vol. 4, no. 1, pp. 6–14, 2000.

[3] M. Mason, B. Hood, andC.N. Macrae, “Look into my eyes:Gaze direction and person memory,” Memory, vol. 12, no. 5, pp. 637–643, 2004.

[4] Z. Kang and S. J. Landry, “An eye movement analysis algorithm for a multielement target tracking task: Maximum transition-based agglomerative hierarchical clustering,” IEEE Trans. Human Mach. Syst., vol. 45, no. 1, pp. 13–24, Feb. 2015.

[5] D. A. Robinson, “A method of measuring eye movemnent using a scleral search coil in a magnetic field,” IEEE Trans. Bio-med. Electron., vol. 10, no. 4, pp. 137–145, Oct. 1963.

[6] A. E. Kaufman, A. Bandopadhay, and B. D. Shaviv, “An eye tracking computer user interface,” in Proc. IEEE Symp. Res. Frontiers Virtual Reality, 1993, pp. 120–121.

[7] J. S. Babcock and J. B. Pelz, “Building a lightweight eyetracking headgear,” in Proc. Symp. Eye Tracking Res. Appl., 2004, pp. 109–114.

[8] K. Takemura, K. Takahashi, J. Takamatsu, and T. Ogasawara, “Estimating 3-D point-of-regard in a real environment using a head-mounted eye-tracking system,” IEEE Trans. Human Mach. Syst., vol. 44, no. 4, pp. 531–536, Aug. 2014.

[9] J. J.Cerrolaza, A.Villanueva, and R. Cabeza, “Taxonomic study of polynomial regressions applied to the calibration of video-oculographic systems,” in Proc. Symp. Eye Tracking Res. Appl., 2008, pp. 259–266.

[10] A. Haro, I. Essa, and M. Flickner, “A non-invasive computer vision system for reliable eye tracking,” in Proc. Extended Abstracts Human Factors Comput. Syst., 2000, pp. 167–168.

[11] D. W. Hansen and A. E. Pece, “Eye tracking in the wild,” Comput. Vis. Image Understanding, vol. 98, no. 1, pp. 155–181, 2005.

[12] E. D. Guestrin and E. Eizenman, “General theory of remote gaze estimation using the pupil center and corneal reflections,” IEEE Trans. Biomed. Eng., vol. 53, no. 6, pp. 1124–1133, Jun. 2006.

[13] J. Zhu and J. Yang, “Subpixel eye gaze tracking,” in Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recog., 2002, pp. 124–129.

[14] R.Valenti, N. Sebe, and T. Gevers, “Combining head pose and eye location information for gaze estimation,” IEEE Trans. Image Process., vol. 21, no. 2, pp. 802–815, Feb. 2012.

[15] D. Torricelli, S. Conforto, M. Schmid, and T. DAlessio, “A neural-based remote eye gaze tracker under natural head motion,” Comput. Methods Programs Biomed., vol. 92, no. 1, pp. 66–78, 2008.

[16] I. F. Ince and J. W. Kim, “A 2D eye gaze estimation system with lowresolution webcam images,” EURASIP J. Adv. Signal Process., vol. 2011, no. 1, pp. 1–11, 2011.

[17] Y. Sugano, Y. Matsushita, Y. Sato, and H. Koike, “An incremental learning method for unconstrained gaze estimation,” in Proc. Comput. Vis., 2008, pp. 656–667.

[18] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, “Learning gaze biases with head motion for head pose-free gaze estimation,” Image Vis. Comput., vol. 32, no. 3, pp. 169–179, 2014.

[19] B. L. Nguyen, “Eye gaze tracking,” in Proc. Int. Conf. Comput. Commun.Technol., 2009, pp. 1–4.

[20] O.Williams, A. Blake, and R. Cipolla, “Sparse and semi-supervised visual mapping with the S3GP,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog., 2006, pp. 230–237.

[21] H.-C. Lu, G.-L. Fang, C. Wang, and Y.-W. Chen, “A novel method for gaze tracking by local pattern model

and support vector regressor," Signal Process., vol. 90, no. 4, pp. 1290–1299, 2010.

[22] F. Lu, Y. Sugano, T. Okabe, and Y. Sato, "Adaptive linear regression for appearance-based gaze estimation," IEEE Trans. Pattern Anal. Mach.Intell., vol. 36, no. 10, pp. 2033–2046, Oct. 2014.

[23] F. Lu, T. Okabe, Y. Sugano, and Y. Sato, "A head pose-free approach for appearance-based gaze estimation," in Proc. Brit. Mach. Vis. Conf., 2011, pp. 1–11.

[24] S. He, Q. Yang, R.W. Lau, J.Wang, andM.-H. Yang, "Visual tracking via locality sensitive histograms," in Proc. IEEE Conf. Comput. Vis. Pattern Recog., 2013, pp. 2427–2434.

[25] K.-C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 5, pp. 684–698, May. 2005.

[26] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," Comput. Vis. Image Understanding, vol. 61, no. 1, pp. 38–59, 1995.

[27] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L0 gradient minimization,"ACM Trans. Graph., vol. 30, no. 6, pp. 174.1–174.11, Dec. 2011.

[28] Y.-l. Tian, T. Kanade, and J. F. Cohn, "Dual-state parametric eye tracking," in Proc. IEEE Int. Conf. Autom. Face Gesture Recog., 2000, pp. 110–115.

[29] Z. R. Cherif, A. Nait-Ali, J. Motsch, and M. Krebs, "An adaptive calibration of an infrared light device used for gaze tracking," in Proc. Inst. Meas. Technol. Conf., 2002, pp. 1029–1033.

[30] J. Sigut and S.-A. Sidha, "Iris center corneal reflection method for gaze tracking using visible light," IEEE Trans. Biomed. Eng., vol. 58, no. 2, pp. 411–419, Feb. 2011