Eye Gaze Tracking With a Web Camera in a Desktop Environment

Yiu-ming Cheung, *Senior Member, IEEE*, and Qinmu Peng, *Member, IEEE*

***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 an 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.**

***Index Terms*—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 multielement 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.

Manuscript received May 16, 2014; revised August 13, 2014, December 16, 2014, and January 28, 2015; accepted January 29, 2015. Date of publication March 5, 2015; date of current version July 11, 2015. This work was supported by the Faculty Research Grants of Hong Kong Baptist University, Hong Kong, under Projects FRG2/14-15/075 and FRG1/14-15/041, and by the National Science Foundation of China under Grant: 61272366. Recommended for publication by Associate Editor I. Cheng. *(Corresponding author: Yiu-ming Cheung).*

Y.-m. Cheung is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, and with United International College, 519085 Zhuhai, China, and also with HKBU Institute of Research and Continuing Education, Shenzhen, Guangdong, China (e-mail: ymc@comp.hkbu.edu.hk).

Q. Peng is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR (e-mail: pqinmu@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2015.2400442

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 provide 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.

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 in the POSIT algorithm* (AWPOSIT), whereby the head pose can

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

be estimated. Finally, eye gaze tracking is performed by the integration of the eye vector and head movement information. The main contributions of this paper are as follows.

1) The proposed approach can tolerate illumination changes and robustly extract the eye region, and provides an accurate method for the detection of the iris center and eye corner.
2) A novel weighted adaptive algorithm for pose estimation is proposed to address pose estimation error; thus, improving the accuracy of gaze tracking.

The remainder of this paper is organized as follows: Section II describes related study. The details of the proposed gaze tracking method are presented in Section III, which includes eye feature detection, calibration, and pose estimation. In Section IV, we empirically study the performance of the proposed approach. Section V is the conclusion.

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 obtain the 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 semisupervised 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 spatial coordinates into the SVR to obtain a gaze mapping function. Lu *et al.* [22] introduced an adaptive linear regression model to infer the gaze from eye appearance by utilizing fewer training samples.

In summary, the appearance-based methods can circumvent the careful design of visual features to represent the gaze. Instead, they utilize the entire eye image as a high-dimensional input to predict the gaze by a classifier. The construction of the classifier needs a large number of training samples, which consist of the eye images from subjects looking at different positions on the screen under the different conditions [17], [19], [21], [23]. These techniques generally have reduced requirements for the image resolution, but they are sensitive to head motion and light changes, as well as the number of training samples. In contrast, the feature-based methods are able to extract the salient visual features to denote the gaze, e.g., [13], [16], which yield acceptable gaze accuracy even with slight changes of illumination, but are not tolerant to head movement. The study in [14] and [15] estimates the gaze by taking into account the head movement to compensate for the gaze shift when the head moves.

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

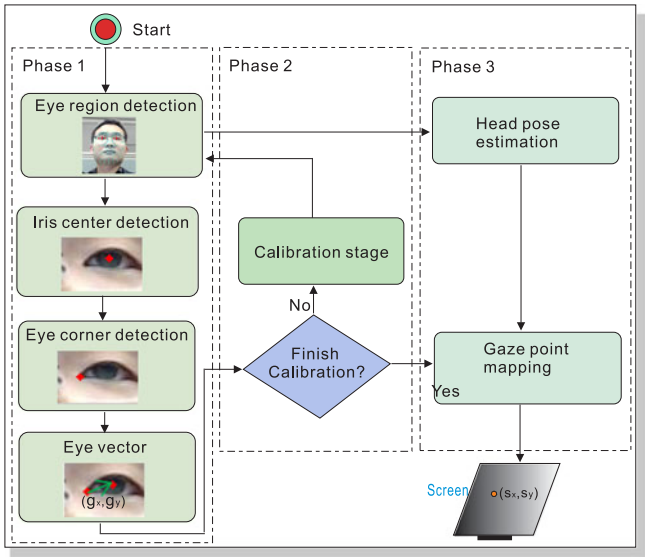


Fig. 2. Three-phase feature-based eye gaze tracking method.

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.

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. The details of facial feature extraction using ASM are as follows.

1) *Feature Selection*: The obvious features, each of which is denoted as (x_i, y_i), are selected and expressed as a vector $\mathbf{x} = (x_1, \cdots x_n, y_1, \cdots, y_n)^T$. Thus, a face shape is described by a set of n landmark points.

2) *Statistical Shape Model*: A set of landmark points (i.e., training images) should be aligned to analyze and synthesize new shapes to those in the training set. It uses the principal component analysis method

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{Pb} \tag{1}$$

(a) (b)

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

where $\bar{\mathbf{x}}$ is the mean shape, $\mathbf{P}$ contains the top t eigenvectors corresponding to the largest eigenvalues, and $\mathbf{b} = (b_1, b_2, \ldots, b_n)^T$ with b_i being the shape parameter which is restricted to $\pm 3\sqrt{\lambda_i}$ for the purpose of generating a reasonable shape.

3) *Fitting*: We make model shapes fit the new input shape by translation t, rotation θ, and scaling s, that is

$$\mathbf{y} = T_{\mathbf{X},t,s,\theta}(\bar{\mathbf{x}} + \mathbf{Pb}) \tag{2}$$

where $\mathbf{y}$ is a vector containing the facial features. 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.

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 extracted using 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 obtained 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

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

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.

Finally, we combine the intensity energy and edge strength to locate the iris center. We denote the intensity energy and the edge strength by E_1 and E_2, respectively

$$E_1 = \Sigma(I * S_r) \tag{3}$$

$$E_2 = \sqrt{g_x^2 + g_y^2} \tag{4}$$

where I is the eye region, and S_r is a circle window with the same radius as the iris. g_x and g_y are the horizontal and vertical gradients of the pixel, respectively. In order to detect the iris center, we should minimize the intensity energy in the circle window and maximize the edge strength of iris edges. The trade-off is controlled by the parameter τ. That is

$$(x_c, y_c) = \min_{(x,y)}\{E_1(x,y) - \tau \cdot \left(\int_{-\pi/5}^{\pi/5} E_2(x,y) \cdot ds + \int_{4\cdot\pi/5}^{6\cdot\pi/5} E_2(x,y) \cdot ds \right)\} \tag{5}$$

where (x_c, y_c) is the coordinate of the iris center. The integral intervals are $[-\frac{1}{5}\pi, \frac{1}{5}\pi]$ and $[\frac{4}{5}\pi, \frac{6}{5}\pi]$, because these ranges of the iris edge do not overlap with the eyelids. Further, the arcs of the iris edges correspond to the same range of ones in a circle with radius r. We compute the integral by the sum of the edge strength of each pixel located on the arcs. 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 the computer screen, we can detect the iris center of eye images as shown in Fig. 5(b) and (c).

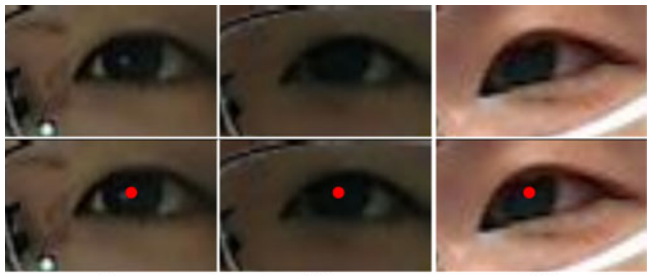


Fig. 5. First row shows the different eye regions, while the second row gives the detection results of the iris center.

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. The definition of curvature for each point μ is given as

$$k(\mu) = \frac{\Delta x_\mu \Delta^2 y_\mu - \Delta^2 x_\mu \Delta y_\mu}{[(\Delta x_\mu)^2 + (\Delta y_\mu)^2]^{1.5}} \tag{6}$$

where $\Delta x_\mu = (x_{\mu+l} - x_{\mu-l})/2$, $\Delta y_\mu = (y_{\mu+l} - y_{\mu-l})/2$, $\Delta^2 x_\mu = (\Delta x_{\mu+l} - \Delta x_{\mu-l})/2$, $\Delta^2 y_\mu = (\Delta y_{\mu+l} - \Delta y_{\mu-l})/2$, and l is a small step. The curvature of each segment is calculated under the different scales depending on the mean curvature (k_ori) of the original segment. The scale parameter σ of Gaussian filter $G = exp(-x^2/\sigma^2)$ is set at $\sigma^2 = 0.3 * k\_\text{ori}$. We consider the local maxima as initial corners, whose absolute curvature should be greater than a threshold, which is twice the neighboring local minima. Then, we remove the T-junction point when it is very close to the other corners. In addition, we calculate the angle for each corner. The angle of the candidate inner eye corner falls into a restricted range $[120°, 250°]$ because the eye corner is the intersection of the two eyelid curves. Hence, the true candidate eye inner corners are selected based on this condition. Then, we use the eye template, which is generated from the training eye images to find the best matching corner as the inner eye corner. To construct the corner template, 20 inner eye patches are selected from eye images, collected from ten males and ten females of different ages. The size of each patch

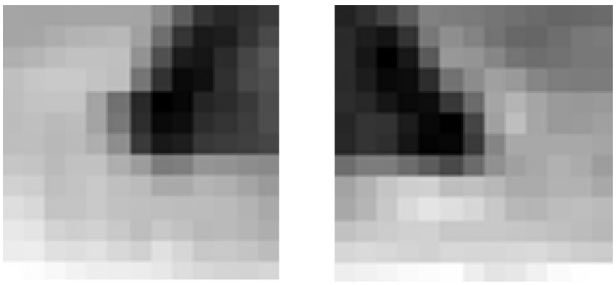


Fig. 6. Inner eye template, where (a) left eye corner template, and (b) right eye corner template.

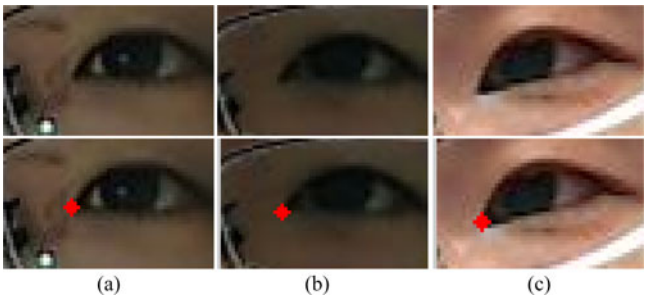


Fig. 7. First row: eye regions, and the second row: eye corner detection results.

is 13×13, and the center of each patch corresponds to the eye corner which is manually marked. The inner eye template is constructed by the average of 20 patches, as shown in Fig. 6.

Finally, a template matching method is used to locate the eye corner with the best response. The measure uses the normalized correlation coefficient

$$\wp = \frac{\sum_{x,y}(I(x,y)-\bar{I})(T(x,y)-\bar{T})}{\{\sum_{x,y}(I(x,y)-\bar{I})^2 \sum_{x,y}(T(x,y)-\bar{T})^2\}^{0.5}} \quad (7)$$

where $I(x,y)$ is the eye image, $\bar{I}$ is the mean value, $T(x,y)$ is the template, and $\bar{T}$ is the mean value. The corner detection results are shown in Fig. 7.

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 $p_{\text{\_iris}}$ and eye corner $p_{\text{\_corner}}$, i.e., $g = p_{\text{\_corner}} - p_{\text{\_iris}}$. 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 number 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

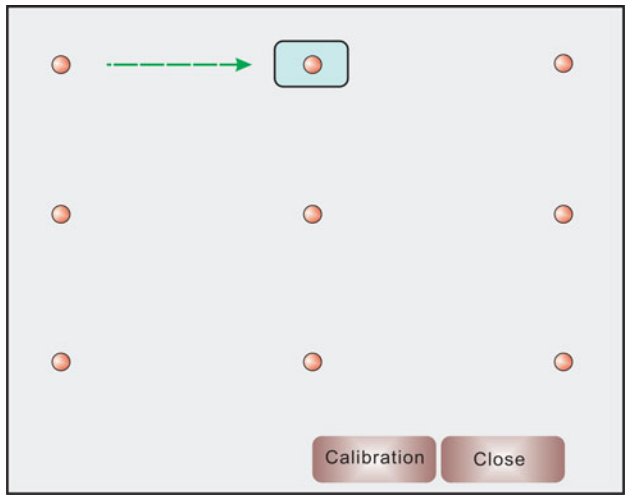


Fig. 8. Nine positions on the screen (1024×1280 pixels).

$$\begin{aligned} u_x &= a_0 + a_1 g_x + a_2 g_y + a_3 g_x g_y + a_4 g_x^2 + a_5 g_y^2 \\ u_y &= b_0 + b_1 g_x + b_2 g_y + b_3 g_x g_y + b_4 g_x^2 + b_5 g_y^2 \end{aligned} \quad (8)$$

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 section elaborates on facial feature tracking and the head pose estimation algorithm in video sequences. Prior head pose estimation approaches mostly use a stereo camera [21], [31], accurate 3-D data for head shape [32], or limited head rotation [33]. Real-time solutions are limited by the complex representations or accurate initialization for head models [14]. Usually, the human head can be modeled as an ellipsoid or 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 [34]–[36], which can perform in real time and track the state of the head roughly.

To improve the estimation of head pose, we utilize an SHM to simulate 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. 2-D facial features can be

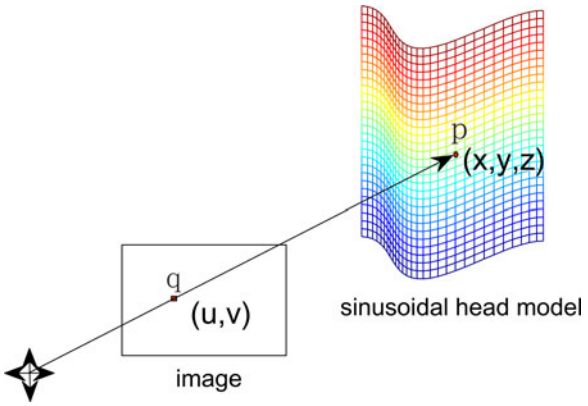


Fig. 9. Perspective projection of 3-D point p onto the image plane.

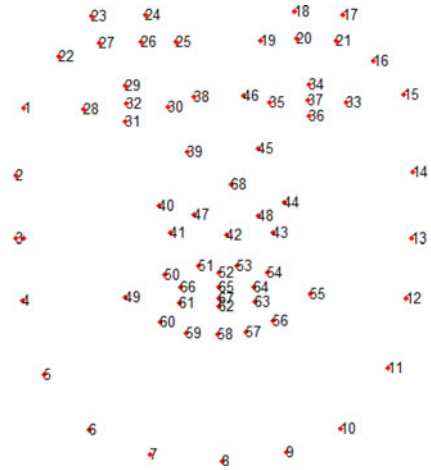


Fig. 10. Locations of the facial features.

related to 3-D positions on the sinusoidal surface. When the 2-D facial features are tracked in each video frame, the 2-D–3-D conversion method can be utilized to obtain the head pose information. Pose from orthography and scaling with iterations (POSIT) [37] is a 2-D–3-D conversion method to get the pose (i.e., rotation and translation) of a 3-D model given a set of 2-D image and 3-D object points. For a better estimation of the head pose, we propose the AWPOSIT algorithm because the classical POSIT algorithm estimates the pose of the 3-D model based on a set of 2-D points and 3-D object points by considering their contribution uniformly. As for the 2-D facial features, they have different significance with respect to reconstructing the pose due to their reliability. If some features are not detected accurately, the overall accuracy of the estimated pose may decrease sharply with the classical POSIT algorithm. The proposed AWPOSIT can obtain more accurate pose estimation using key feature information. The implementation details follow.

The SHM assumes that the head is shaped as a 3-D sine (see Fig. 9) and the face is approximated by the sinusoidal surface. Hence, the motion of the 3-D sine is a rigid motion that can be parameterized by the pose matrix $\mathbf{M}$ at frame F_i . The pose matrix includes the rotation matrix $\mathbf{R}$ and translations matrix $\mathbf{T}$ at the ith frame, i.e.

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{T^T} & \mathbf{1} \end{bmatrix} = [M_1|M_2|M_3|M_4] \tag{9}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix, and $\mathbf{T} \in \mathbb{R}^{3\times1}$ is the translation vector with $\mathbf{T} = (t_x^i, t_y^i, t_z^i)^T$, and M_1 to M_4 are column vectors. The head pose at each frame is calculated with respect to the initial pose, and the rotation and translation matrix can be set at 0 for the initial frame (i.e., standard front face). The ASM model is used on the initial frame to obtain the 2-D facial features. Then, these features are tracked using the Lucas–Kanade (LK) feature tracker [38] algorithm in the subsequent frames over time. Since these facial features are related to the 3-D points on the sinusoidal model, the movements of which are regarded as summarizing the head motion, we utilize the perspective projection through the pinhole camera model for establishing the relation between the 3-D points on the sinusoidal surface and their corresponding projections on the 2-D image plane. Fig. 9 shows the relation between the 3-D point $p = (x, y, z)^T$ on the sinusoidal surface and its projection point $q = (u, v)^T$ on the image plane, where u and v are calculated by

$$\begin{aligned} u &= f\frac{x}{z} \\ v &= f\frac{y}{z} \end{aligned} \tag{10}$$

with f being the focal length of the camera.

As 2-D facial features have different significance with respect to reconstructing the pose information, we consider two factors to weigh the facial features: 1) robustness of facial features, and 2) normal direction of the facial features in the 3-D surface. The first factor w_1 assigns the values to facial features to indicate their importance. Specifically, strong features should be assigned much larger weights because they can provide more reliable information for the pose estimation. These features (see Fig. 10) are grouped into six classes, each of which obtains the different weight according to its robustness in experiment:

1) cheek points $w_1(1:15) = 0.011$;
2) eyebrow points $w_1(16:27) = 0.017$;
3) eye points $w_1(28:37) = 0.011$;
4) nose points $w_1(38:48) = 0.026$;
5) mouth points $w_1(49:67) = 0.011$;
6) nose tip point $w_1(68) = 0.03$.

The second factor utilizes the normal direction of the facial feature to weigh its contribution. The normal direction can be estimated by the previous pose. Let the unit vector $\vec{h}$ stand for the normal direction of the initial front face pose. Each facial feature point has a normal vector $\vec{b}_i$, and $w_{2i} = \frac{\vec{h}\cdot\vec{b}_i}{|\vec{h}|\cdot|\vec{b}_i|}$ denotes the significance of the ith facial feature. $\tilde{w}_i = w_{1i} \cdot w_{2i}$ denotes the total weight for the ith feature. Then, $\tilde{w}_i$ is normalized to obtain the weight w_i, i.e., $w_i = \frac{\tilde{w}_i}{\sum_i \tilde{w}_i}$.

The 2-D facial points is denoted as P_{2D} and the 3-D points on the sinusoidal model is denoted as P_{3D}. The AWPOSIT algorithm is given in Algorithm 1.

Algorithm 1: M = AWPOSIT(P_{2D}, P_{3D}, w, f)

Input: P_{2D}, P_{3D}, w and f.
1: $n = size(P_{2D}, 1)$; $c = ones(n, 1)$
2: $u = P_{2D_x}/f$; $v = P_{2D_y}/f$
3: $H = [P_{3D}, c]$; $O = pinv(H)$
4: Loop
5: $J = O \cdot u$; $K = O \cdot v$
6: $Lz = 1/(\sqrt{(1/\|J\|) \cdot (1/\|K\|)})$
7: $M_1 = J \cdot Lz$; $M_2 = K \cdot Lz$
8: $R_1 = M_1(1:3)$; $R_2 = M_2(1:3)$
9: $R_3 = \frac{R_1}{\|R_1\|} \times \frac{R_2}{\|R_2\|}$
10: $M_3 = [R_3; Lz]$
11: $c = H \cdot M_3/Lz$
12: $uu = u$; $vv = v$
13: $u = c \cdot w \cdot P_{2D_x}$; $v = c \cdot w \cdot P_{2D_y}$
14: $e_x = u - uu$; $e_y = v - vv$
16: if $\|e\| < \epsilon$ then
17: $M_4 = [M_1(4), M_2(4), Lz, 1]^T$; Exit Loop
18: end if
19: end Loop
Output: M.

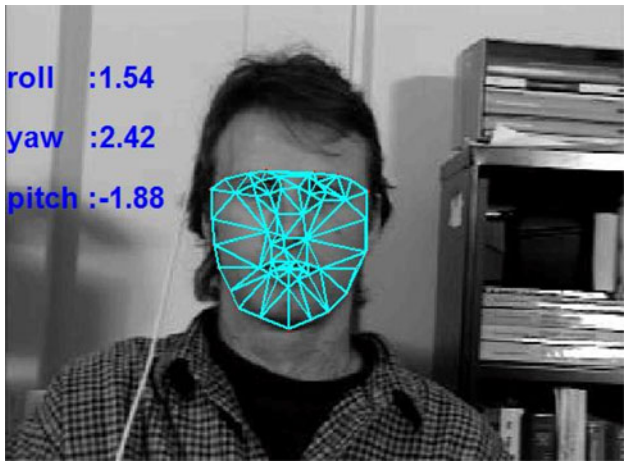


Fig. 11. Example of pose estimation.

The tracking mode takes the value of the global head motion by 2-D facial features on the initial front face. Then, these features are tracked using the LK tracker algorithm and AWPOSIT determines the pose information in the video frames. If AWPOSIT fails to converge, the tracking mode stops and a reinitialization detects the 2-D facial features again. Then, tracking mode can resume. Fig. 11 shows an example for the head pose estimation, in which the 3-D rotation angles (i.e., yaw, pitch, roll) can be obtained from the rotation matrix $\mathbf{R}$.

When the head pose algorithm is available, we can compensate for the gaze error by the head movement. It estimates the head pose and computes the corresponding displacement $(\Delta u_x, \Delta u_y)$ caused by the head movement. Suppose that the initial 3-D coordinate of the head is denoted as (x_0, y_0, z_0), and its position of projection on the image plane is (u_0, v_0). The coordinate of the head is (x', y', z') when head movement occurs. The corresponding parameters $\mathbf{R}$ and $\mathbf{T}$ are estimated by the AWPOSIT

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{R} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \mathbf{T}. \tag{11}$$

Therefore, the displacement $(\Delta u_x, \Delta u_y)$ can be calculated by

$$\begin{aligned} \Delta u_x &= f\frac{x'}{z'} - u_0 \\ \Delta u_y &= f\frac{y'}{z'} - v_0. \end{aligned} \tag{12}$$

The eye vector is extracted and the calibration mapping function is adopted to obtain the gaze point (u_x, u_y) on the screen. Combining the gaze direction from the eye vector and the displacement from the head pose, the final gaze point (s_x, s_y) can be obtained with

$$\begin{aligned} s_x &= u_x + \Delta u_x \\ s_y &= u_y + \Delta u_y. \end{aligned} \tag{13}$$

The implementation steps of the proposed system is summarized in Algorithm 2.

Algorithm 2: Pseudocode of eye gaze tracking system

Initialization:
– Extracting 2-D facial features using ASM
– Initialize the 3-D SHM P_{3D}
and head pose M
– Get calibration mapping function
Tracking the gaze through all the frames:
1: **for** t = 1 to all Frames do
2: Extract the eye region
3: Detect the iris center $p_{\text{\_iris}}$
4: Detect the eye inner corner $p_{\text{\_corner}}$
5: Eye vector is obtained: $g = p_{\text{\_corner}} - p_{\text{\_iris}}$
6: Get static gaze point (u_x, u_y) by mapping function
7: Track the face features P_{2D}
8: Obtain the feature weight w and head pose
M = AWPOSIT(P_{2D}, P_{3D}, w, f)
9: Get the displacement $(\Delta u_x, \Delta u_y)$
10: Obtain the final gaze point (s_x, s_y)
11: **end for**

IV. Evaluation

An evaluation of eye feature detection, head pose estimation, and gaze estimation is presented.

A. Eye Center Detection

The detection of the eye center is a difficult task in eye features detection. The accuracy of eye center detection directly affects the gaze estimation. To evaluate the detection accuracy of the eye center by the proposed algorithm, the dataset BioID [39], which consists of 1521 grayscale images collected from 23 subjects under the different illumination and scale changes, is utilized.

TABLE I
PERFORMANCE OF DIFFERENT METHODS—BIOID DATASET

Different methods	Accuracy ($e \leq 0.05$)	Accuracy ($e \leq 0.1$)
Campadelli *et al.* [40]	62.00%	85.20%
Niu *et al.* [41]	75.00%	93.00%
Valenti *et al.* [14]	86.09%	91.67%
Proposed method	87.21%	93.42%

TABLE II
PERFORMANCE OF FOUR METHODS—BOSTON UNIVERSITY HEAD POSE DATASET

Rotation angles	Sung *et al.* [35]	An and Chung [42]	Valenti *et al.* [14]	Proposed method
Roll	3.1	3.22	3.00	2.69
Yaw	5.4	5.33	6.10	4.53
Pitch	5.6	7.22	5.26	5.48

Fig. 12. Examples of the results on the BioID dataset [39].

In some cases, the eyes are closed and hidden by glasses. The ground truth of the eye center is provided in the dataset.

To measure the accuracy, the normalized error e proposed by Jesorsky *et al.* [39] is used

$$e = \frac{\max(\mathrm{d_{left}}, \mathrm{d_{right}})}{d} \tag{14}$$

where d_{left} and d_{right} are the Euclidean distance between the estimated eye center and the ones in the ground truth, and d is Euclidean distance between the eyes in the ground truth.

Table I shows the results compared with other methods for the normalized error smaller than 0.05 and 0.1. In the case of accurate location of the iris region (i.e., $e \leq 0.1$), the proposed method outperforms the others. In the more accurate cases (i.e., $e \leq 0.05$), the proposed method achieves superior accuracy compared with the other methods. Fig. 12 shows example results of the iris center on the BioID dataset. The proposed method can work on different conditions, such as changes in pose, illumination, and scale. In most of the closed eyes and glasses cases, it can still roughly estimate the iris center due to the robust detection of the eye region. Some failures occur due to higher angle head poses because the ASM cannot extract the facial features.

B. Head Pose Estimation

Since eye gaze is determined by the eye vector and head movement, the head pose estimation is utilized to compensate for the eye gaze so that the gaze error can be reduced. Boston University [36] provides a head pose dataset for performance estimation. Generally, the pose estimation error is measured by the root-mean-square error for the three rotation angles (i.e., pitch, yaw, and roll).

Table II presents results from the evaluation of pose estimation with approaches proposed by An and Chung [42], Sung *et al.* [35], and Valenti *et al.* [14]. An and Chung [42] used 3-D ellipsoidal model to simulate the head and obtain the pose information. Sung *et al.* [35] proposed to combine the active appearance models and the CHM to estimate the pose. Similar to this study, Valenti *et al.* [14] presented a hybrid approach combining the eye location cue and CHM to estimate the pose. In [14], the results are comparable with [35]. The proposed method achieves improved accuracy for the head pose using the SHM and adaptive-weighted POSIT.

Fig. 13(a)–(c) shows three tracking examples of the head movement with pitch, yaw, and roll head rotation. Each example of pose tracking is performed on a video sequence consisting of 200 frames. Fig. 13(d)–(f) shows the estimated head rotation angles and the ground truth.

C. Gaze Estimation

The eye gaze tracking system includes a web camera to acquire image sequences (see Fig. 14). It utilized a Logitech web camera with a resolution of 960×720 pixels, which was installed below the computer monitor. The computer hardware configuration was an Intel Core (TM) i7 CPU 3.40 GHz. With the approximate $70\,\mathrm{cm}$ distance between a subject and the screen, each subject sat in front of the computer screen to make his/her head fully captured by the camera. Every subject looked at nine target points on the screen 40 times, which was usually completed in less than 15 min. The experiments were performed

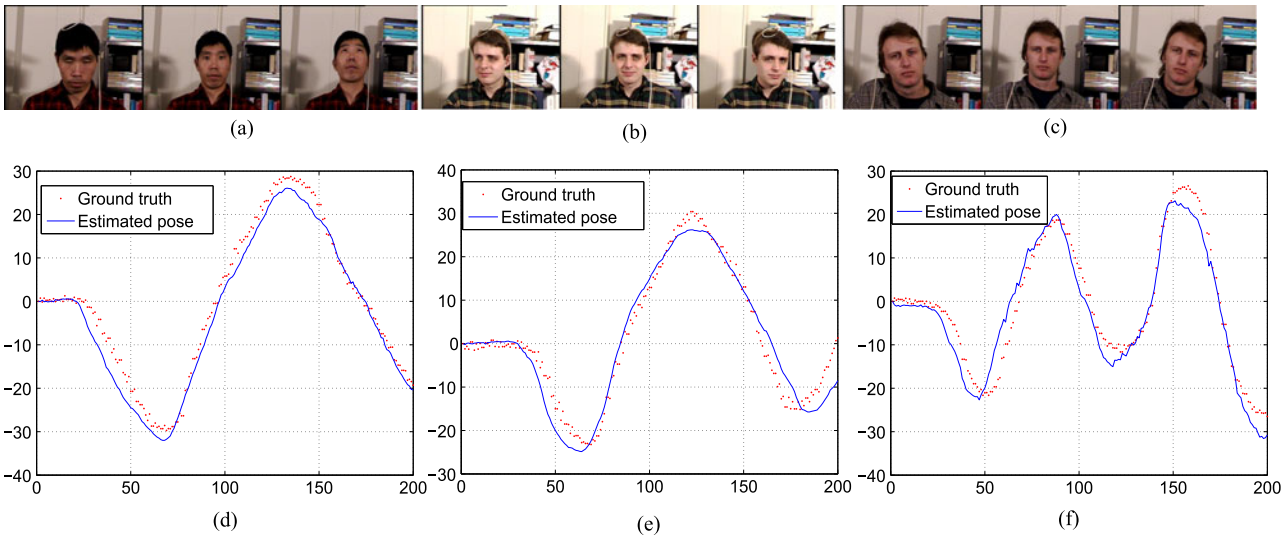


Fig. 13. Examples of head movement in the Boston University head pose dataset (a) Pitch (llm6) (b) Yaw (jim1) (c) Roll (jam1) (d) Estimated pitch (e) Estimated yaw (f) Estimated roll.

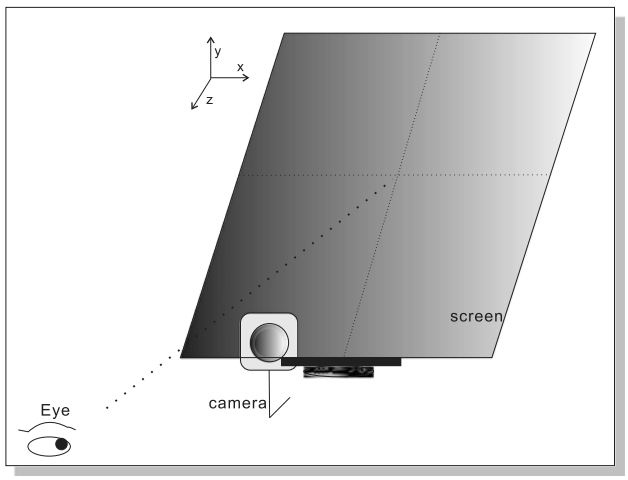


Fig. 14. Setup of the gaze tracking system with screen dimensions of 1024×1280 pixels.

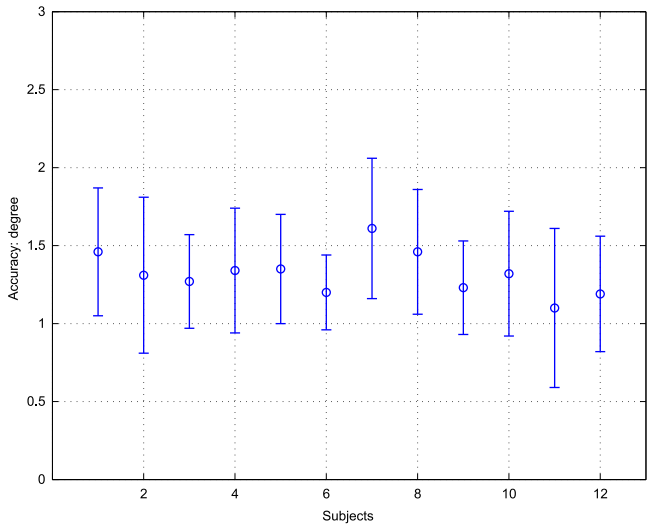


Fig. 15. Average accuracy with standard deviation by subject without head movement.

in a desktop environment, in which the light could come from the fluorescents, LEDs, or sunlight.

Performance of the proposed system includes gaze tracking without head movement and gaze tracking with head movement. The accuracy of the eye gaze tracking system is measured by angular degree (A_{dg})

$$A_{dg} = arctan\left(\frac{A_d}{A_g}\right) \tag{15}$$

where A_d is the distance between the estimated gaze position and the real observed position, and A_g represents the distance between the subject and the screen plane. The smaller A_{dg} is, the higher is the accuracy of the eye gaze tracking.

1) Gaze Tracking Without Head Movement: Twelve subjects, four who wore glasses, participated with different illumination conditions. The subjects were requested to look at different positions on the screen. The estimated gaze points were recorded. We computed the angular degree with respect to the target point positions. Fig. 15 shows the average accuracy for subjects. The gaze accuracy varied due to the different characteristics of the eyes, head movement, and the sitting position.

We compared the proposed method with Williams *et al.* [20], Lu *et al.* [22], Valenti *et al.* [14], Zhu and Yang [13], and Nguyen [19] in Table III. As Williams *et al.* [20] and Lu *et al.* [22] are appearance-based methods, they, as well as Zhu and Yang [13] and Nguyen [19], can estimate the gaze provided that the head is fixed. The proposed method is robust to light conditions and capable of detecting eye features (see Fig. 16). The accuracy of the proposed tracking system is about 1.28°, which outperforms Valenti *et al.* [14], Zhu and Yang [13], and Nguyen [19], but not Williams *et al.* [20] and Lu *et al.* [22]. Nevertheless, the proposed model is robust against the light changes, while Williams *et al.*

TABLE III
PERFORMANCE OF DIFFERENT METHODS WITHOUT HEAD MOVEMENT

Different method	Accuracy (degree)	Robust to light changes
Zhu and Yang [13]	1.46	Yes
Valenti *et al.* [14]	2.00	Yes
Nguyen [19]	2.13	No
Williams *et al.* [20]	0.83	No
Lu *et al.* [22]	0.97	No
Proposed method	1.28	Yes

Fig. 16. Eye features (in the top right corner of the image) can be detected when the light changes a lot. The image source is [25].

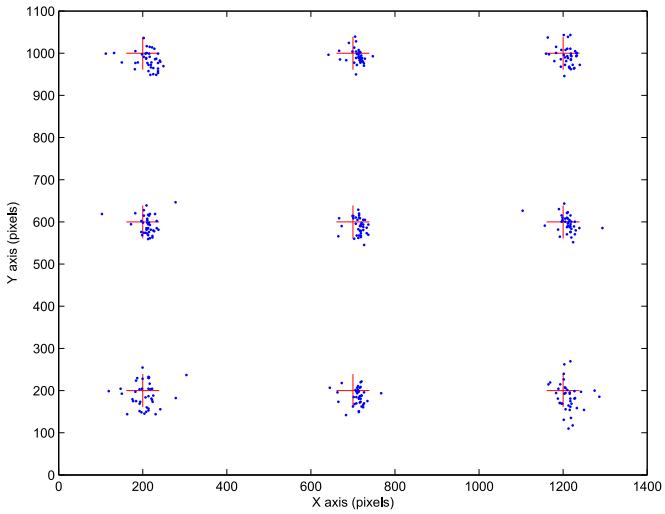


Fig. 17. Points of gaze without head movement are marked as the dot points, while the target point is marked as a cross. The x-axis and y-axis correspond to the screen coordinates.

[20] and Lu *et al.* [22] are not. Further, the proposed model does not need the training samples for gaze estimation. By contrast, the models of Williams *et al.* [20] and Lu *et al.* [22] need 91 and nine training samples, respectively.

The points of gaze on the screen are shown in Fig. 17. In most cases, the gaze error in the y-direction is larger than that in the x-direction because part of the iris is occluded by the eyelids, resulting in an accuracy reduction for the y-direction. Another reason is that the range of eye movement in the y-direction is smaller than that in the x-direction. Therefore, the eye motion in the y-direction is considered as a minor movement that is more difficult to detect.

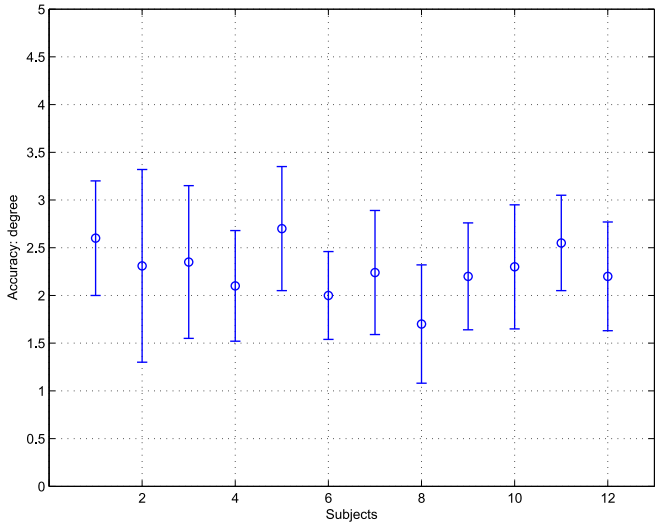


Fig. 18. Average accuracy with standard deviation by subject with head movement.

TABLE IV
PERFORMANCE OF DIFFERENT METHODS WITH HEAD MOVEMENT

Different method	Accuracy (degree)	Robust to light changes	Range of head motion (*degree*)
Torricelli *et al.* [15]	2.40	No	15.5×15.5×4.1
Valenti *et al.* [14]	2–5	Yes	16 ×15.5 ×6.2
Sugano *et al.* [17]	4.0	No	11.4×10.2×1.1
Lu *et al.* [18]	2–3	No	18.2×15.2×10.7
Proposed method	2.27	Yes	15.5×15.5 × 5

2) Gaze Tracking With Head Movement: In this experiment, the subjects are allowed to move their heads while gazing at the points on the screen. Fig. 18 shows the average accuracy by subject. The error with head movement is much larger than that with the head fixed. The increased error is largely caused by the head pose estimation and more difficulty in detecting the eye features on the nonfront face. To avoid the large gaze error, as a rule of thumb, the head movement is limited in a range, approximately $\pm 15.5^\circ$, $\pm 15.5^\circ$, and $\pm 5^\circ$ rotation deviation around the x-, y-, and z-axes; the x-translation, y-translation, and z-translation is about ± 12, ± 12, and ± 10mm, respectively. If the rotation and translation are beyond the limited range, the gaze error could lead to hundreds of pixels deviation. From a practical viewpoint, the range of head movement is sufficient for subjects to cover their gaze points on the screen.

Table IV presents the performance of the proposed method with the head movement in comparison with Torricelli *et al.* [15], Valenti *et al.* [14], Sugano *et al.* [17], and Lu *et al.* [18]. It is difficult to utilize the same dataset to evaluate the performance of different methods. Nevertheless, we have explicitly shown the range of head motion we utilized in each method, and compared them under similar experimental conditions for a fair comparison.

The accuracy of the proposed gaze tracking system is about 2.27°. The work by Valenti *et al.* [14] obtained accuracy

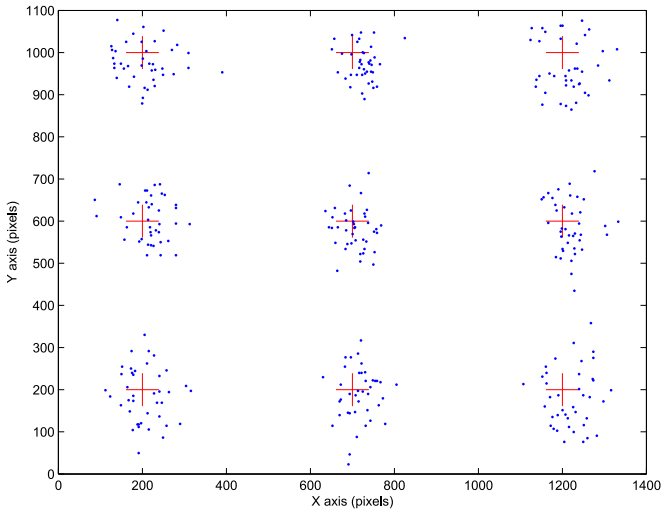


Fig. 19. Points of gaze with head movement are marked as the dot points, while the target point is marked as a cross. The x-axis and y-axis correspond to the screen coordinates.

between 2° and 5°, in which the best result, i.e., 2°, was obtained when a subject keeps the head stable, while the worst result was about 5° with large head movement. Lu *et al.* [18] obtained a result comparable with the proposed method. This method is appearance based, which needs more training samples. In this example, it includes 33 static images and a 5-s video clip, which is somewhat inconvenient in use. The gaze accuracy in [17] is not high even after using 1000 training samples. In contrast, the proposed gaze system utilizes a web camera capturing the face video and works well in a desktop environment. Nevertheless, there still exist failure cases with our proposed system. For example, one case is that the gaze direction is not consistent with the head pose direction, i.e., we turn our head, but look in the opposite direction. Another case is facial expression, e.g., laugh, which causes a large deviation in the locations of the facial features. Under the circumstances, the projection error on the screen will be more than a hundred pixels. Nevertheless, we are able to easily circumvent these cases and utilize the proposed system.

The points of gaze on the screen are shown in Fig. 19. The accuracy in the y-direction is greater than in the x-direction. The gaze error is not uniform on the screen. Instead, the accuracy toward the screen edge increases slightly. As the eyeball moves to the edge of the eye socket when we look at the screen edge points, the iris is overlapped by the eyelids. As a result, the detection accuracy of the gaze vector will deteriorate.

V. 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

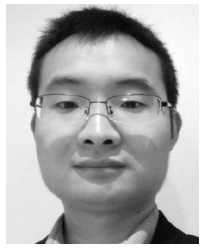
[1] A. T. Duchowski, "A breadth-first survey of eye-tracking applications," *Behavior Res. Methods, Instrum., Comput.*, vol. 34, no. 4, pp. 455–470, 2002.
[2] S. P. Liversedge and J. M. Findlay, "Saccadic eye movements and cognition," *Trends Cognitive Sci.*, vol. 4, no. 1, pp. 6–14, 2000.
[3] M. Mason, B. Hood, and C. 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 low-resolution 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 S^3GP," 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, and M.-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 L_0 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.
[31] L. Morency, P. Sundberg, and T. Darrell, "Pose estimation using 3D view-based eigenspaces," in *Proc. IEEE Int. Workshop Anal. Model. Faces Gestures*, 2003, pp. 45–52.
[32] P. Martins and J. Batista, "Accurate single view model-based head pose estimation," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recog.*, 2008, pp. 1–6.
[33] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, Apr. 2000.
[34] J. Xiao, T. Moriyama, T. Kanade, and J. F. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *Int. J. Imaging Syst. Technol.*, vol. 13, no. 1, pp. 85–94, 2003.
[35] J. Sung, T. Kanade, and D. Kim, "Pose robust face tracking by combining active appearance models and cylinder head models," *Int. J. Comput. Vis.*, vol. 80, no. 2, pp. 260–274, 2008.
[36] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, Apr. 2000.
[37] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comp. Vis.*, vol. 15, no. 1–2, pp. 123–141, 1995.
[38] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," Microprocessor Research Lab, Intel Corp., Tech Rep. no. 5(2001), pp. 1–10, 2001.
[39] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the Hausdorff distance," in *Proc. Audio Video-Based Biometric Person Authentication*, 2001, pp. 90–95.
[40] P. Campadelli, R. Lanzarotti, and G. Lipori, "Precise eye localization through a general-to-specific model definition," in *Proc. Brit. Mach. Vis. Conf.*, 2006, pp. 187–196.
[41] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao, "2D cascaded adaboost for eye localization," in *Proc. 18th Int. Conf. Pattern Recog.*, 2006, pp. 1216–1219.
[42] K. H. An and M. J. Chung, "3D head tracking and pose-robust 2D texture map-based face recognition using a simple ellipsoid model," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2008, pp. 307–312.

Yiu-ming Cheung (SM'06) received the Ph.D. degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, in 2000.

He is currently a Full Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His research interests include machine learning, information security, image and video processing, and pattern recognition.

Dr. Cheung is a Senior Member of the Association for Computing Machinery.



Qinmu Peng (M'09) received the B.E. degree from the North China Electric Power University, Baoding, China, in 2008, and the M.E. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2011. He is currently working toward the Ph.D. degree with the Department of Computer Science, Hong Kong Baptist University, Hong Kong.

His main research interests include image processing, pattern recognition, and machine learning methods in computer vision.