

A REAL-TIME VIDEO-BASED EYE TRACKING APPROACH FOR DRIVER ATTENTION STUDY

Xianping FU, Ying ZANG, Hongbo LIU*

*School of Information Science and Technology
Dalian Maritime University
Dalian 116026, China
e-mail: fxp@dlmu.edu.cn, lhb@dlut.edu.cn*

Communicated by Steve Maybank

Abstract. Knowing the driver's point of gaze has significant potential to enhance driving safety, eye movements can be used as an indicator of the attention state of a driver; but the primary obstacle of integrating eye gaze into today's large scale real world driving attention study is the availability of a reliable, low-cost eye-tracking system. In this paper, we make an attempt to investigate such a real-time system to collect driver's eye gaze in real world driving environment. A novel eye-tracking approach is proposed based on low cost head mounted eye tracker. Our approach detects corneal reflection and pupil edge points firstly, and then fits the points with ellipse. The proposed approach is available in different illumination and driving environment from simple inexpensive head mounted eye tracker, which can be widely used in large scale experiments. The experimental results illustrate our approach can reliably estimate eye position with an accuracy of average 0.34 degree of visual angle in door experiment and 2–5 degrees in real driving environments.

Keywords: Eye-tracking, driver attention, corneal refraction, random sample consensus

1 INTRODUCTION

The analysis of driver attention has long been a popular field of research in light of the potential for safety improvements [1, 2, 3]. Driver's eye gaze has been most

* Corresponding author

recently conducted as driver workload metrics [4], and as a proxy for driver attention [5, 6, 7]. Despite active research and significant progress in the last 30 years, eye detection and tracking remains challenging due to the individuality of eyes, occlusion, scale variability, location, and light conditions in real driving environment [8, 9]. Although eye tracking has been deployed in a number of research systems and to a smaller degree of consumer products, eye tracking has not reached its full potential. The primary obstacle to integrating these techniques into large scale usage is that they have been either too invasive or too expensive for routine use. Eye-tracking systems can be divided into remote and head mounted systems [10]. Each type of system has its respective advantages. For example, remote systems are not as intrusive but are not as accurate or flexible as head mounted systems [11]. In driving environments, the driver's view field is extremely wide; except looking ahead, looking left and right for the attractive objects aside the road drivers sometimes even look back for reversing or monitoring a following vehicle. Therefore, for the remote system, it is difficult to make a scene camera with so large view field available and to calibrate eye gaze with such a large scale head movement.

To address driver's visual distraction problem in large view field (almost 360 degrees), the designed system uses a head mounted eye tracker. Given this advance, the most significant remaining obstacle is the cost and flexibility. In the recent years, the price of high-quality digital camera technology has dropped precipitously and new technology made the camera lighter and more flexible than before. Some software implementations are integrated with specialized digital processors in cameras to obtain high-speed performance, which makes them more convenient and executable for head mounted device to be used in real driving environment.

Therefore it is possible to develop a widely available, reliable and high-speed eye-tracking algorithm that runs on general embedded computing hardware in order to integrate eye tracking into everyday driver's attention study [12]. Towards this goal, we have developed a hybrid eye-tracking algorithm that integrates feature-based and model-based approaches and made its implementation available for low cost device.

The main contribution of this paper is focused on the two parts. Firstly, we develop an eye-tracking algorithm which has improved performance in pupil center and corneal reflection detection. It can enhance the performance of head mounted eye gaze tracking system. The pupil contour and corneal reflection are detected by feature based method, and then the pupil location, shape and size are calculated by ellipse fitting method. Secondly, we proposed a novel calibration method in car. The scene camera with infrared illumination, four infrared reflection labels are fixed on rear view mirror, center console, left side mirror and right side mirror by help of a sticker. The calibration is implemented and updated every time when driver looks at these labels over a 200 ms dwell time. A proposed method is more flexible than laser pointer calibration method on biopic telescope aiming point tracking [13]. The similar method is implemented in door simulation to test the real time performance.

The rest of the paper is organized as follows. Related works about eye tracking algorithms and applications are reviewed in Section 2. Section 3 introduces the

structure of our proposed head mounted eye tracking system. The proposed hybrid algorithms are presented in Section 4. In Section 5, our experimental results and some discussion are illustrated, and finally conclusions are given in Section 6.

2 RELATED WORKS

Eye tracking technology has been available for many years using a variety of methods, such as Purkinje-reflection based, contact-lens based eye coil systems, electro-oculography, and corneal reflection [14]. In recent years, head-mounted and remote camera-based systems have been developed to allow more natural and less cumbersome methods of gaze tracking. They make it possible to collect the real-time video record for the eye movement. Eye-tracking algorithms can be classified into two approaches: feature-based and model-based approaches [15, 16, 17]. Feature-based approaches detect and localize image features related to the position of the eye [12]. Feature-based approaches have in common that a threshold is needed to decide when a feature is present or absent. The determination of an appropriate threshold is typically left as a free parameter that is adjusted by the user. The detected eye features vary widely across algorithms but most often rely on intensity levels or intensity gradients. For example, in infrared images created with the dark-pupil or bright-pupil technique, an appropriately set intensity threshold can be used to extract the region corresponding to the pupil. The pupil center can be taken as the geometric center of this identified region. The intensity gradient can be used to detect the limbus in visible spectrum images or the pupil contour in infrared spectrum images. An ellipse can then be fitted to these feature points.

On the other hand, model-based approaches do not explicitly detect features but rather find the best fitting model that is consistent with the image. For example, integral differential operators can be used to find the best-fitting circle or ellipse for the limbus and pupil contour [18]. This approach requires an iterative search of the model parameter space that maximizes the integral of the derivative along the contour of the circle or ellipse. The model-based approach can provide a more precise estimate of the pupil center and pupil contour than a feature-based approach given that a feature defining criteria is not applied to the image data. However, this approach requires searching a complex parameter space that can be fraught with local minima [19]. Thus gradient techniques cannot be used without a good initial guess for the model parameters. Thus, the gain in accuracy of a model based approach is obtained at a significant cost in terms of computational speed and flexibility. Notably however, the use of multi-scale image processing methods in combination with a model-based approach holds promise for real time performance.

Infrared spectrum imaging is commonly used in eye tracking. Infrared imaging eliminates uncontrolled specular reflection by actively illuminating the eye with a uniform and controlled infrared light not perceivable by the user. Infrared eye tracking typically utilizes either bright-pupil or dark-pupil techniques. Bright-pupil techniques illuminate the eye with a source that is on or very near the axis of the

camera. The result of such illumination is that the pupil is clearly demarcated as a bright region due to the photo reflective nature of the back of the eye. Dark-pupil techniques illuminate the eye with an off-axis source such that the pupil is the darkest region in the image, while the sclera, iris and eyelids all reflect relatively more illumination. In either method, the first-surface specular reflection of the illumination source off the cornea (the outer-most optical element of the eye) is also visible. A further benefit of infrared imaging is that the pupil, rather than the limbus, is the strongest feature contour in the image (Figure 2); both the sclera and the iris strongly reflect infrared light while only the sclera strongly reflects visible light. Tracking the pupil contour is preferable given that the pupil contour is smaller and more sharply defined than the limbus. The vector between the pupil center and the corneal reflection that is a white dot on cornea is typically used as the dependent measure rather than the pupil center alone. This is because the vector difference is insensitive to slippage of the head mounted device – both the camera and the source move simultaneously. Furthermore, due to its size, the pupil is less likely to be occluded by the eyelids.

In this paper, we investigate a novel algorithm on infrared spectrum imaging techniques and extend these techniques to visible spectrum imaging as well. The dark pupil techniques are considered.

3 SYSTEM STRUCTURE

We implement an eye-tracking algorithm with images captured from head mounted system. There are two cameras in this head mounted system, one is infrared camera with IR (Infra Red) illumination source and IR filter, which face driver's eye. The other is scene camera fixed on glasses frame. The structure of this head mounted system is shown in Figure 1. Common sun glasses frame and cheap compact camera with 640×480 resolution are used, with infrared illumination by 850 nm LED fixed beside the eye camera. The IR filter is a 820–890 nm band pass filter.

4 EYE-TRACKING ALGORITHM

In this section, we propose an eye-tracking algorithm that combines feature-based and model-based approaches to achieve a good trade-off between run-time performance and accuracy for dark-pupil infrared illumination. The goal of the algorithm is to extract the location of the pupil center and the corneal reflection position so as to relate the vector difference between these measures to coordinates in the scene image. Li et al. proposed a pupil feature detection approach, namely “starburst” [20]. Our algorithm improved their work. The improvements consist in using the horizontal and vertical projections of the binary image to estimate the pupil center at the first and key frame, which is different from Li's random guess. The corneal reflection is eliminated from the image, and the pupil edge points are located using an iterative feature-based technique with only eight rays from estimated



Fig. 1. The low cost head mounted eye gaze tracker in our driving attention study



Fig. 2. Corneal reflection of dark pupil effect with different pupil position in our head mounted eye gaze tracker (the corneal reflection is the brightest point in the image and pupil is the darkest region)

pupil center. Before ellipse fitting, a proximity based approach is used to eliminate outliers. An ellipse is fitted to a subset of the detected inliers edge points using the Random Sample Consensus (RANSAC) paradigm. The best fitting parameters from this feature based approach are then used to initialize a local model based search for the ellipse parameters that maximize the fit to the image data.

4.1 Noise Reduction

Due to the uninformed illumination in real driving environment and the use of a low-cost camera in this head mounted eye tracker, we need to begin by reducing the noise present in the images. We reduce the shot noise by applying a 5×5 Gaussian filter with a standard deviation of 2 pixels.

4.2 Corneal Reflection Detection

As illustrated in Figure 2, the corneal reflection corresponds to one of the brightest regions in the eye image, and the round shape and size of the corneal reflection is almost fixed when the distance from IR camera to cornea is established after hardware setup. Thus the corneal reflection can be obtained through pixel intensity threshold and geometrical character.



Fig. 3. The distortion shape of corneal reflection. If the ratio of distortion of the brightest region is larger than this one it is not considered as corneal reflection

Note that because the cornea extends approximately to the limbus, we can limit our search for the corneal reflection to a square region of interest with a small window of 160×120 pixels. To begin, the threshold is used to produce a binary image in which only values above this threshold are taken as corneal reflection candidates. However, a constant threshold across observers and even within observers is not optimal. Therefore an adaptive threshold which decreases from the brightest pixel intensity value in each frame is used to localize the corneal reflection [20]. Given its small size, the corneal reflection is approximately a circle in the image. Within these corneal reflection candidates, only those ratios between width and height of candidate blobs less than 2 will be processed due to the round shape property of corneal reflection (as shown in Figure 3). Our corneal reflection detection algorithm is given below (Algorithm 1).

Algorithm 1 Corneal Reflection Detection Algorithm.

01. Input image;
 02. Threshold \leftarrow brightest pixel;
 03. Do
 04. $i = i + 1$;
 05. $threshold = threshold - 1$;
 06. Image binaryzation by threshold;
 07. $s(i) \leftarrow$ the area of largest blob/the average area of blobs
 07. when the absolute of width/height of blobs less than 2;
 08. While ($s(i) < s(i - 1)$)
 09. Output the center coordination of corneal reflection.
-

In this algorithm, threshold will decrease from the brightest to lower intensity. When the brightest threshold is adopted, it is likely that the largest candidate region is attributable to the corneal reflection, as other specular reflections tend to be quite small and located off the cornea as well as near the corner of the image where the eyelids meet. The ratio between the area of the largest candidate and the average area of other regions is calculated as the threshold is lowered. At first, the ratio will increase because the corneal reflection will grow in size faster than other areas. Note that the intensity of the corneal reflection monotonically decreases towards its edges, explaining this growth. A lower threshold will, in general, also induce an increase in false candidates. The ratio will begin to drop as the false candidates become more prominent and the size of the corneal reflection region becomes large. The highest ratio is taken as optimal threshold.

The location of the corneal reflection is then given by the geometric center (x_c, y_c) of the largest region in the image using the adaptively determined threshold. While the approximate size of the corneal reflection can be derived using the threshold region from the localization step, this region does not typically include the entire profile of the corneal reflection. To determine the full extent of the corneal reflection, we assume that the intensity profile of the corneal reflection follows a bivariate Gaussian distribution. If the radius r where the average decline in intensity is maximal is related to the radius with maximal decline for a Gaussian (i.e. a radius of one standard deviation), the full extent of the corneal reflection as $2.5r$ to capture 99% of the corneal reflection profile is taken.

Radial interpolation is then used to remove the corneal reflection. First, the central pixel of the identified corneal reflection region is set to the average of the intensities along the contour of the region. Then for each pixel between the center and the contour, the pixel intensity is determined via linear interpolation. An example of this process can be seen in Figure 4 (compare Figures 4 a) and 4 b)).

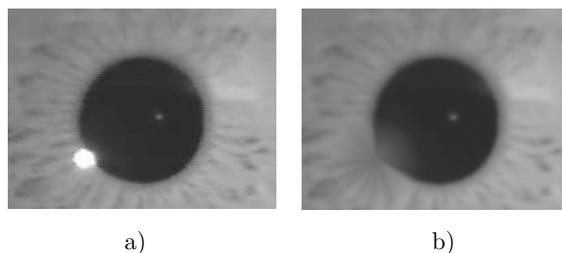


Fig. 4. The corneal reflection and removal by Gaussian with a radius of one standard deviation. The corneal reflection is treated as $2.5r$ to capture 99% of the corneal reflection profile.

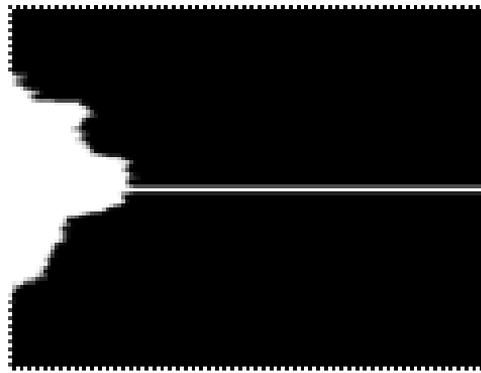
4.3 Pupil Edge Points Detection

We have improved the feature-based method [20] to detect the pupil contour with small neighborhood and fixed eight rays. The best guess of the pupil center is implemented on horizontal and vertical projection, as the pupil is the darkest region in the input eye image and corneal reflection size is much smaller than the pupil, the projections on horizontal and vertical have trough on the curve which can be treated as an estimated pupil center. The horizontal and vertical projection results are shown in Figure 5. In local region, the estimated pupil center should be located around the center of the image. Therefore in Figure 5 b), the wave trough m and n will be removed and wave trough k will be treated as horizontal position of the pupil center. The initial pupil center position is shown in Figure 5 c). Our algorithm to detect pupil contour features is given below (Algorithm 2).

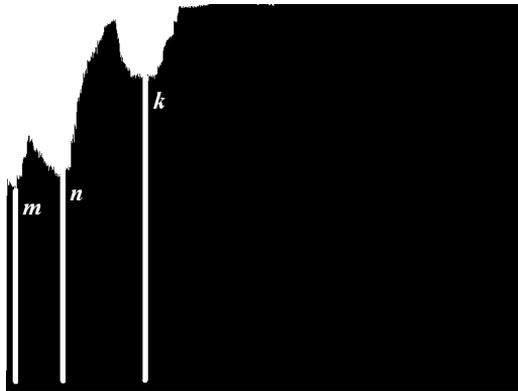
Algorithm 2 Pupil Contour Features Detection Algorithm.

01. Input image;
 02. $E_{pc} \leftarrow$ Projection center as estimated pupil center;
 03. $\alpha = 0$;
 04. Do
 05. $\alpha = \alpha + 45$;
 06. $P_E \leftarrow$ Intensity derivatives as estimated pupil contour points;
 06. on the rays from E_{pc} ;
 07. $\beta = 0$;
 08. Do
 09. $\beta = \beta + 5$;
 10. $[P_c] \leftarrow$ Intensity derivatives as estimated pupil contour points
 10. on rays from P_E ;
 11. While ($\beta < 360$)
 12. While ($\alpha < 360$)
 13. Output the center coordination of corneal reflection.
-

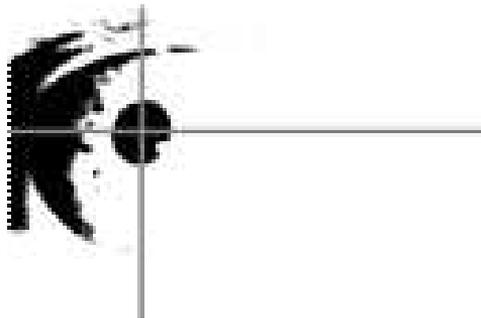
For each frame, a location is chosen that represents the best guess of the pupil center in the frame. For the first frame and key frame this can be taken as the trough value of the projection image. The pupil shape is also considered as circle and its size is limited within a reasonable scope. For subsequent frames, the location of the pupil center from the previous frame is used. Because the pupil contour frequently occupies very little of the image, instead of applying edge detection to the entire eye image or to a region of interest around the estimated pupil location, we detect pupil edges along a limited number of rays that extend from a central best guess of the pupil center. The proposed method to detect pupil center is shown in Figure 6. After the pupil center is calculated, in the next frame, the neighborhood region of 160×120 pixels is used to calculate pupil center rather than the whole frame.



a)



b)



c)

Fig. 5. The horizontal and vertical projection results to locate the initial pupil center position

When the pupil center is estimated, rays from this center are used to estimate pupil contour edge points which are intensity leap point along rays. The rays from estimated pupil center are limited on eight directions with equal angle step, which can be shown in Figures 6 a) and 6 d). In Figure 6 a) there is a good estimated pupil center, so the eight rays can reach the proper pupil edge. Figure 6 d) shows a wrong estimated pupil center, the estimated pupil center is outside of the pupil, so only two from the eight rays can reach the pupil edge. Because the horizontal and vertical projection results are used to locate the initial pupil center position within neighbor region of 160×120 , almost all the estimated pupil center is inside the pupil. This method takes advantage of the high-contrast elliptical profile of the pupil contour present in images taken with infrared illumination using the dark-pupil technique.

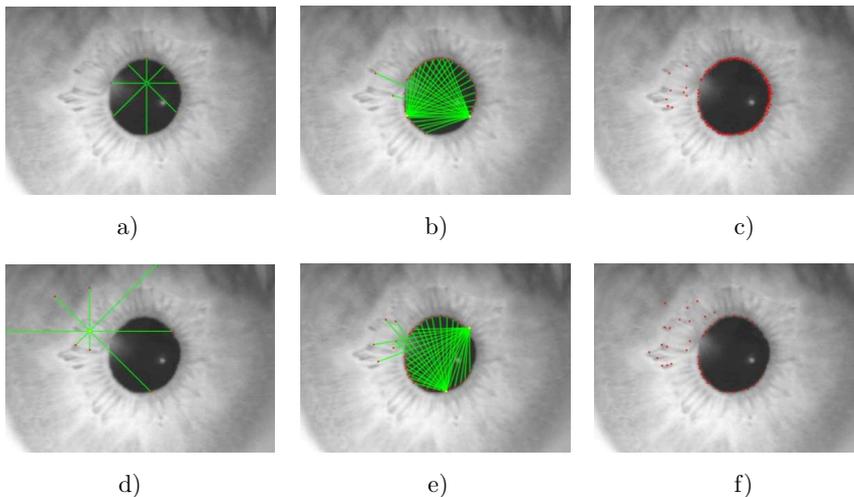


Fig. 6. Detected pupil contour using two step method: The first step is drawing eight ray radiation from estimated pupil center. The second step is drawing rays from the detected pupil edge points. a) The good estimated pupil center, eight rays from the estimated pupil center can reach the pupil edge. b) From the eight detected pupil edge points after a), second round of rays from the pupil edge points is used to detect pupil edge; in this figure, only two groups of rays from pupil edge points are shown. There are eight groups of rays together. c) The detected pupil contours. d) The wrong estimated pupil center. e) Two groups of rays from detected pupil edge points are good enough to detect the pupil edge points. f) The detected pupil edge points begin with the wrong estimated pupil center.

Next, the eight derivatives rays from estimated pupil center, extending radial rays away from this starting point, are independently evaluated pixel by pixel until a threshold θ ($\theta = 20$) is exceeded. Given that we are using the dark-pupil technique, only positive derivatives (increasing intensity as the ray extends) are considered. When this threshold is exceeded, a feature point is defined at that location and the

processing along the ray is halted. If the ray extends to the border of the image, no feature point is defined. The eight candidate feature points of the initial rays are shown in Figure 6 a).

For each of the eight candidate feature points whose distance from starting point is less than 100, the above-described feature detection process is repeated backwards from the feature points. However, rays are every 5 degrees and are limited to $\gamma = \pm 50$ degrees around the ray that originally generated the feature point. The motivation for limiting the return rays in this way is that if the candidate feature point is indeed on the pupil contour (as shown in Figure 6 b)), the returning rays will generate additional feature points on the opposite side of the pupil such that they are all consistent with a single ellipse (i.e. the pupil contour).

The two-stage feature detection process improves the robustness of the method to poor initial guesses for the starting point. This is a problem when an eye movement is made as the eye can rapidly change positions from frame to frame. This is especially true for images obtained at low frame rates. For example, such a case is shown in Figure 6 d). However, the feature points are biased to the side of the pupil contour nearest to the initialization point. The second iteration of the ray process would minimize this bias, the computational burden is affordable with the two iterations and thus the strategy would be efficient. At this point an ellipse could be fitted to the candidate points.

The detected feature locations for the second group of rays are shown in Figures 6 b) and 6 e). When the initial guess is a good estimate of the pupil center, for example during eye fixations which occupy the majority of the frames, only a single iteration is required.

4.4 Ellipse Fitting

There are two phases to get the pupil contour by ellipse fitting based on the detected pupil edge points. The first is outlier elimination algorithm; the other is model-based ellipse fitting algorithm.

Before fitting these data, it is desirable to eliminate the outliers first. To this end, we classify a group of unlabeled data into two classes. One of them consists of data that can be fitted well by an ellipse, and the other consists of data that can be classified as outliers. Inliers are those sample points for which the algebraic distance to the ellipse is less than some threshold. In other words, it is a two-class classification problem with prior knowledge on one of the classes. This threshold is derived from a probabilistic model of the error expected based on the nature of our feature detector. The outlier elimination algorithm is a proximity-based algorithm which is based on algebra graphic theory to eliminate distant, isolated outliers.

The model-based algorithm can fit an ellipse model to the inliers selected after the first phase. Now that the first phase eliminates most of the outliers, model-based algorithm can be effectively applied to fit an ellipse model, which tests all the data points with respect to ellipse model and classifies the points that saliently deviate from ellipse as outliers and then classify other points as inliers.

4.4.1 Outlier Elimination

An inlier is a sample in the data attributable to the mechanism being modeled whereas an outlier is a sample generated through error and is attributable to another mechanism not under consideration. In our application, inliers are all of those detected feature points that correspond to the pupil contour and outliers are feature points that correspond to other contours, such as that between the eyelid and the eye.

Assume that we have K pupil contour points $f_i = [x_i, y_i]$, $i = 1, 2, 3, \dots, K$, where K means N or M . N points are from an ellipse with small amounts of noise (inliers), M points are randomly scattered in the plane (outliers). We can make further assumptions about the data points: Average distances between inliers are smaller than those between inliers and outliers, and inliers are the majority ($> 50\%$) of the data set.

$$g_i = \min\{D(f_i, f_{j \neq i})\}, \quad i, j \in K \quad (1)$$

where D is the distance between the neighbors of detected pupil feature points. Construct an adjacency graph based on proximity which calculates the distance between the neighbors of detected pupil feature points. The major component is considered as composed of inliers, other small components are considered as composed of outliers.

4.4.2 Model-Based Algorithm

Given a set of candidate feature points, the next step of the algorithm is to find the best fitting ellipse. In two-dimensional space, specifically, let $\vec{p}_1, \vec{p}_2, \dots, \vec{p}_N$ be a set of N points, $\vec{p}_i = [x_i, y_i]^T$. Let $\vec{t} = [x^2, xy, y^2, x, y, 1]^T$, then we have the function

$$F(\vec{p}, \vec{v}) = \vec{t}^T \vec{v} = ax^2 + bxy + cy^2 + dx + ey + f = 0, \quad (2)$$

the implicit equation of the generic ellipse, characterized by the parameter vector $\vec{v} = [a, b, c, d, e, f]^T$. The task is to find the parameter vector \vec{v}_0 , associated with the ellipse which fits $\vec{p}_1, \dots, \vec{p}_N$ best in the least squares sense, as the solution of the objective

$$\min_{\vec{v}} = \sum_{i=1}^N [D(\vec{p}_i, \vec{v})]^2 \quad (3)$$

where $D(\vec{p}_i, \vec{v})$ is a suitable distance.

We can achieve this goal by running an algorithm similar to RANSAC, which is an effective technique for model fitting in the presence of a large but unknown percentage of outliers in a measurement sample. However, RANSAC has been shown to be inappropriate when the percentage of outliers is high and the number of parameters in the model is large; it is computationally unacceptable when the number of parameters and the portion of outliers are large. The random sample consensus

is determined by Equation (4)

$$P = 1 - (1 - w^n)^k \quad (4)$$

where P is the probability of finding the correct model after running RANSAC for k times; w is the portion of the inliers; n is the minimum number of data points needed to fit a model. Assume $w = 0.5$, to guarantee $P = 0.99$, $n = 5$, $k = 146$ to fit an ellipse. Fitting algorithm itself becomes computationally expensive when n is large [21].

However, since we have greatly decreased the percentage of outliers in the remaining data set by employing the outlier detection algorithm based on proximity, it is now feasible to run a RANSAC-type algorithm. On the other hand, RANSAC admits the possibility of outliers and only uses a subset of the data to fit the model. In detail, RANSAC is an iterative procedure that selects many small but random subsets of the data, uses each subset to fit a model, and finds the model that has the highest agreement with the data set as a whole. The subset of data consistent with this model is the consensus set.

First, we use the entire set of inliers selected by the first stage algorithm to fit an initial model, instead of randomly choosing the minimum number of points as in the original RANSAC, since the remaining outliers represent just a small percentage and are close to the inliers. Moreover, since our initialization is not random, it is unnecessary to run RANSAC repeatedly many times.

The following procedure is repeated R times. First, five samples are randomly chosen from the detected feature set given that this is the minimum sample size required to determine all the parameters of an ellipse. Singular Value Decomposition (SVD) on the conic constraint matrix generated with normalized feature-point coordinates is used to find the parameters of the ellipse that perfectly fit these five points. If the parameters of the ellipse are imaginary, the ellipse center is outside of the image, or the major axis is greater than two times the minor axis, five different points are randomly chosen until this is no longer the case. Then, the number of candidate feature points in the data set that agree with this model (i.e. the inliers) are counted. After the necessary number of iterations, an ellipse is fitted to the largest consensus set (shown in Figure 7).

4.5 Mapping and Calibration

In order to calculate the point of gaze of the user in the scene image, a mapping between locations in the scene image and an eye-position measure (e.g., the vector difference between the pupil center and the corneal reflection) must be determined. The typical procedure in eye-tracking methodology is to measure this relationship through a calibration procedure. During calibration, the user is required to look at a number of scene points for which the positions in the scene image are known. While the user is fixating each scene point $s = (x_s, y_s, 1)$, the eye position $e = (x_e, y_e, 1)$ is measured (note the homogeneous coordinates). In this paper, the calculation

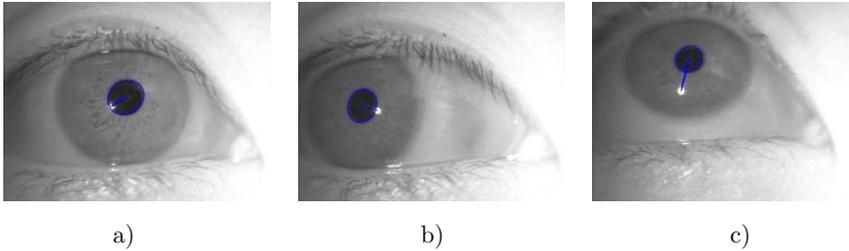


Fig. 7. Ellipse fitting result and the eye-position measure. The calibration is based on the mapping between locations in the scene image and the vector from pupil center to the corneal reflection. Different pupil positions are shown in these figures

is based on floating calibrator method [22], in which the light spot from a head mounted laser pointer projected on a wall while the head is scanning is recorded by the scene camera, in synchronization with the infrared eye camera. The difference is that the calibrators (infrared labels) are fixed on several locations such as left side mirror, right side mirror, rear view mirror, center console. The driver will look at these calibrators to accomplish calibration procedure before experiment is started every time. Interpolation is performed within target position where no samples were taken. Thus, non-linear interpolation error can be minimized, even for wide-range tracking. We generate the mapping between the two sets of points using a linear homographic mapping.

The calibration result can be updated every time the driver looks at these calibration labels during experiments. Therefore when the glasses slide on the nose bridge or the eyes are squinting due to lighting changes or seating position is changed, the calibration will be updated when the driver looks at side mirror, rear view mirror or center console.

5 EXPERIMENTAL RESULTS AND DISCUSSIONS

Eye-tracking evaluation was conducted in order to validate the performance of the algorithm. Two groups of experiments are implemented, one using door simulation where the calibration is more delicate with floating calibrators and illumination is good; the other is real driving environment where calibration is based on four labels and illumination is not uniform. The smoothing buffer size of the gaze data is 4 frames, that is, every gaze piece of data is related with last three ones. The resolution of eye image is 640×480 . Scene camera is used to capture calibration points (laser dots or infrared labels). The mapping relationship between eye camera, scene camera and real world is calibrated when the system is set up (when the cameras are installed on glasses frame).

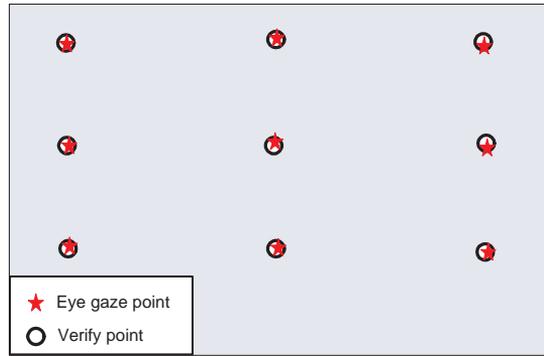
During indoor experiment, the frame rate is 25 frames per second with 2.4 GHz Intel CPU, 4GB RAM PC when the image resolution is 640×480 . Video was recorded from the head mounted eye tracker described in Section 3 while three

subjects viewed a movie trailer projected on a white wall. Prior to viewing the trailer, the subjects placed a laser pointer on their head mounted tracker and scanned randomly on the white wall; the subjects will gaze at this moving laser dot. The distance from the wall is approximately 300 cm. The laser dots on the white wall can be automatically detected using image processing method to be treated as floating calibrators. The evaluation was conducted twice for each user. After viewing the movie trailer, the evaluation is implemented, nine dots are projected on the wall, the subjects fixed these dots and calibrated eye gaze positions are calculated. The evaluation result is shown in Figure 8. The average of error is 0.34 degree.

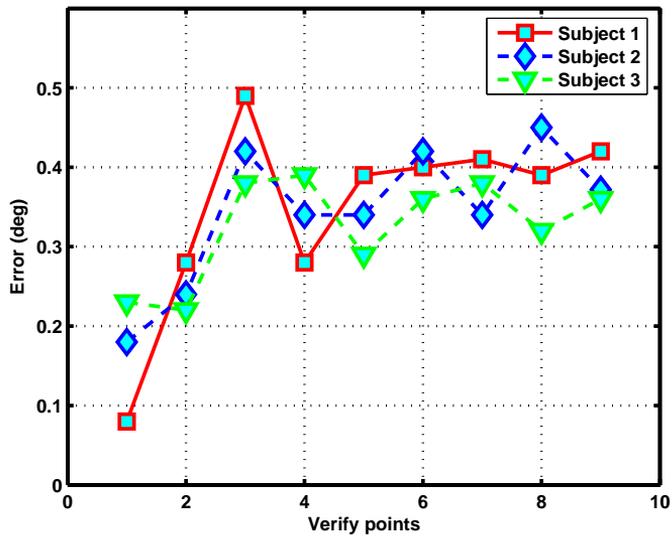
For the real driving environment, a Car PC with 1.5 GHz CPU, 2 GB RAM is used; the frame rate is 15 frames per second of 640×480 resolution. The four different shapes infrared reflect labels are fixed on two side mirror, rear view mirror and center console. The shapes of the four labels are cross, dot, vertical and horizontal line. The scene camera can separate the four labels with the help of different shape and infrared light reflection. These labels are used as calibrators. The calibration procedure is not only implemented before the experiment, but can also be used to improve eye gaze during the experiment, especially when the glasses slide on the nose bridge, the eyes are squinting due to lighting changes or seating position change. We use dwell time – if the user continues to look at the labels over 200 ms, i.e. the scene camera focuses on the target labels over 200 ms, the labels are recorded as calibrators. In this case, we think the driver is paying attention to the label. Such a long dwell time is used to ensure that an inadvertent fixation will not be made by simply “looking around” on the labels. We compare the tracking results under different conditions in real-world driving. The pupil center and corneal reflection detection and ellipse fitting results are shown in Figures 9 and 10. In some frames the pupil detection failed. The detection rate is 96.31 % in all the performed 23 500 frames. We manually verified these gaze positions from scene video and notes, the calibration points and tracking average error in different illumination conditions are shown in Figure 11 and Table 1 with the mean and standard deviations. The average error is 2.95 degrees in usual light conditions, and 4.81 degrees in sunlight. The average error is 2.48 degrees at night; therefore sunlight makes tracking accuracy degrade and the best experimental result is at night with infrared illumination.

6 CONCLUSIONS

In this paper, we focused on eye-tracking approaches of driver attention. A novel eye-tracking algorithm was proposed to collect driver’s eye gaze in real world driving study with for a low cost head mounted tracker. Both corneal reflection location and pupil contour are detected through adaptive feature-based techniques. Horizontal and vertical projection of binary image is used to estimate pupil center, then eight radial rays from this center to reach the pupil edge are iterated to get the pupil edge points. After outliers elimination, the RANSAC paradigm is applied to maximize the accuracy of ellipse fitting in the presence of gross feature-detection



a)



b)

Fig. 8. Verification of the proposed low cost head mounted eye gaze tracking. a) 9 check points in the scene image (circles) and tracking results (stars), b) Tracking errors for the 9 points. The average error is 0.34 degree.

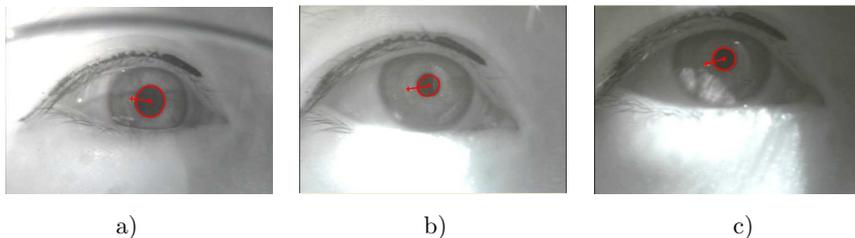


Fig. 9. Experimental results under sunlight. The pupil center, corneal reflection and ellipse fitting result still usable.

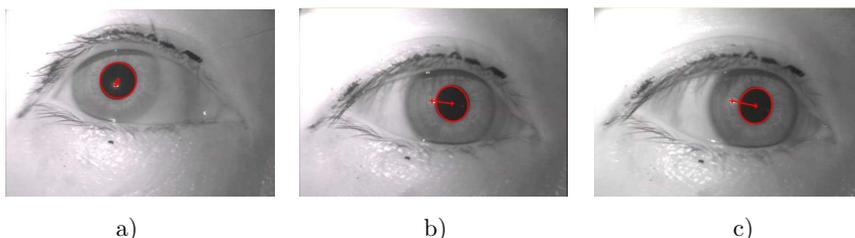
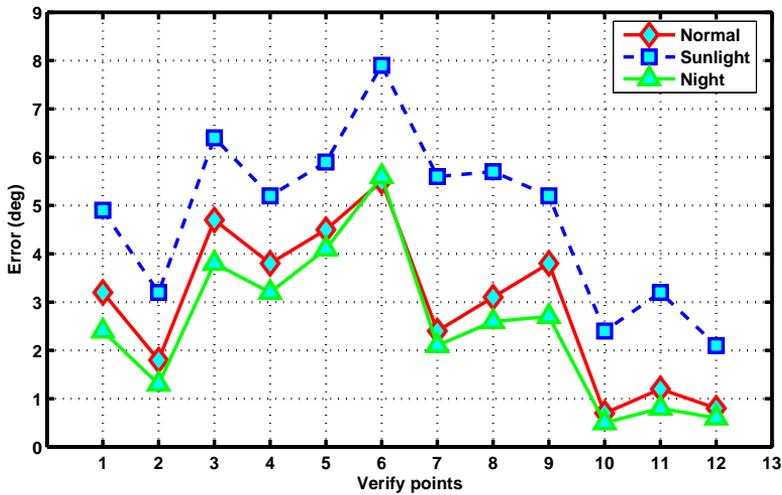
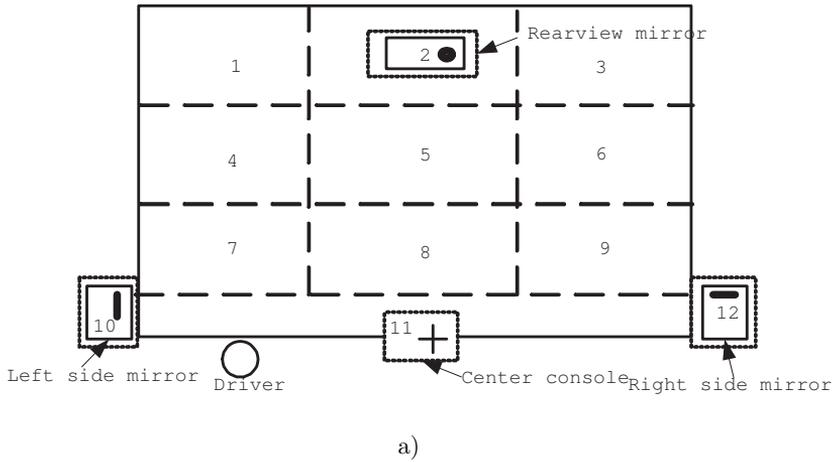


Fig. 10. Experimental results at night. When surrounding light is weak, the infrared illumination can make pupil center, corneal reflection and ellipse fitting result very accurate.

errors. Finally, a model-based approach is applied to further refine the fit. We conducted a validation study which indicates that the algorithm performs well on video obtained from the low-cost head mounted eye tracker. The average error of the verification experiments for three subjects is 0.34 degree in door experiment. In

Point	Original		Estimated Gaze		Sunlight		Normal		Night	
	X	Y	X	Y	Mean	STD	Mean	STD	Mean	STD
1	45	24	36.03	13.40	4.90	2.31	3.20	1.95	2.40	1.01
2	130	8	132.26	3.82	3.20	2.01	1.80	1.56	1.30	0.26
3	225	9	223.40	13.68	6.40	3.02	4.70	2.97	3.80	1.03
4	30	57	35.73	59.68	5.20	2.96	3.80	2.68	3.20	0.57
5	131	58	133.24	56.26	5.90	2.31	4.50	2.14	4.10	1.04
6	225	58	230.56	63.11	7.90	3.18	5.50	2.98	5.60	1.89
7	33	129	41.63	128.47	5.60	2.13	2.40	1.95	2.10	0.78
8	122	139	126.49	141.27	5.70	3.15	3.10	3.02	2.60	1.20
9	242	142	242.89	143.67	5.20	2.61	3.80	1.85	2.70	1.06
10	28	134	133.24	56.26	2.40	1.45	0.70	0.45	0.50	0.21
11	95	67	230.56	63.11	3.20	2.56	1.20	0.95	0.80	0.12
12	86	95	41.63	128.47	2.10	1.94	0.80	1.31	0.60	0.20

Table 1. Experimental results in different real driving environment



b)

Fig. 11. Experimental result in real driving environment. a) Different shape infrared reflection labels are attached on rear view mirror, center console, left and right side mirror. Positions 1–12 are used as check points. b) Tracking errors for 12 check points for different conditions. The average error is 2.95 degrees in usual light conditions, 4.81 degrees under sunlight and 2.48 degrees at night.

real world driving environment, the average error is 2.48–4.81 degrees in different illumination conditions.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant Nos. 60873054, 61073056, 61173035), the Fundamental Research Funds for the Central Universities (Grant No. 2011QN031, 2011JC006), Liaoning Education Department Research Fund(L2010061), Dalian Science and Technology Fund (Grant No. 2010J21 DW006), and America Research to Prevent Blindness International Research Scholar Award (2010).

REFERENCES

- [1] KANDIL, F.—ROTTER, A.—LAPPE, M.: Car Drivers Attend to Different Gaze Targets when Negotiating Closed Vs. Open Bends. *Journal of Vision*, Vol. 10, 2010, No. 4, pp. 1–11.
- [2] DOSHI, A.—TRIVEDI, M.: On the Roles of Eye Gaze and Head Dynamics in Predicting Driver's Intent to Change Lanes. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 10, 2009, No. 3, pp. 453–462.
- [3] BURGUILLO, J.—RODRIGUEZ, P.—COSTA, E.—GIL, F.: History-Based Self-Organizing Traffic Lights. *Computing and Informatics*, Vol. 28, 2009, No. 2, pp. 157–168.
- [4] REIMER, B.: Impact of Cognitive Task Complexity on Drivers' Visual Tunnelling. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2138, 2009, No. 1, pp. 13–19.
- [5] HAMMOUD, R.: *Passive Eye Monitoring: Algorithms, Applications and Experiments*. Springer Verlag 2008.
- [6] YAO, K.—LIN, W.—FANG, C.—WANG, J.—CHANG, S.—CHEN, S.: Real-Time Vision-Based Driver Drowsiness/Fatigue Detection System. *Proceedings of 2010 IEEE Vehicular Technology Conference (VTC 2010-Spring)*, 2010, pp. 1–5.
- [7] MIYAJI, M.—KAWANAKA, H.—OGURI, K.: Driver's Cognitive Distraction Detection Using Physiological Features by the AdaBoost. *Proceedings of 12th International IEEE Conference on Intelligent Transportation Systems*, Missouri 2009, pp. 1–6.
- [8] LIU, R.—YUAN, B.: Automatic Eye Feature Extraction in Human Face Images. *Computing and informatics*, Vol. 20, 2001, No. 3, pp. 289–301.
- [9] LU, Y.—ZHOU, J.—YU, S.: A Survey of Face Detection, Extraction and Recognition. *Computing and informatics*, Vol. 22, 2003, No. 2, pp. 163–195.
- [10] YAMAZOE, H.—UTSUMI, A.—YONEZAWA, T.—ABE, S.: Remote and Head-Motion-Free Gaze Tracking for Real Environments With Automated Head-Eye Model Calibrations. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Anchorage 2008, pp. 1–6.

- [11] ZHU, Z.—JI, Q.: Novel Eye Gaze Tracking Techniques Under Natural Head Movement. *IEEE Transactions on Biomedical Engineering*, Vol. 54, 2007, No. 12, pp. 2246–2260.
- [12] MIYAKE, T.—ASAKAWA, T.—YOSHIDA, T.—IMAMURA, T.—ZHANG, Z.: Detection of View Direction with a Single Camera and Its Application Using Eye Gaze. *Proceedings of the 35th Annual Conference of IEEE Industrial Electronics*, Porto 2009, pp. 2037–2043.
- [13] FU, X.—LUO, G.—PELI, E.: Telescope Aiming Point Tracking Method for Biopic Driving Surveillance. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 18, 2010, No. 6, pp. 628–636.
- [14] VILLANUEVA, A.—CABEZA, R.: Evaluation of Corneal Refraction in a Model of a Gaze Tracking System. *IEEE Transactions on Biomedical Engineering*, Vol. 55, 2008, No. 12, pp. 2812–2822.
- [15] HANSEN, D.—JI, Q.: 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, No. 3, pp. 478–500.
- [16] LI, Y.—WANG, S.—DING, X.: Eye/Eyes Tracking Based on a Unified Deformable Template and Particle Filtering. *Pattern Recognition Letters*, Vol. 31, 2010, No. 11, pp. 1377–1387.
- [17] AMIRI, A.—FATHY, M.: Video Shot Boundary Detection Using Generalized Eigenvalue Decomposition and Gaussian Transition Detection. *Computing and Informatics*, Vol. 30, 2011, No. 3, pp. 595–619.
- [18] DOSHI, A.—TRIVEDI, M.: Investigating the Relationships Between Gaze Patterns, Dynamic Vehicle Surround Analysis, and Driver Intentions. *Proceedings of the 2009 IEEE Intelligent Vehicles Symposium*, Xi'an 2009, pp. 887–892.
- [19] SHA, S.—JIANER, C.—SANDING, L.: A Fast Matching Algorithm Based on K-Degree Template. *Proceedings of the 4th International Conference on Computer Science and Education*, Nanning 2009, pp. 1967–1971.
- [20] LI, D.—WINFIELD, D.—PARKHURST, D.: Starburst: A Hybrid Algorithm for Video-Based Eye Tracking Combining Feature-Based and Model-Based Approaches. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, 2005, pp. 79–79.
- [21] YU, J.—ZHENG, H.—KULKARNI, S.—POOR, H.: Outlier Elimination for Robust Ellipse and Ellipsoid Fitting. *Proceedings of the 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Dutch Antilles 2009, pp. 33–36.
- [22] FU, X.—LUO, G.—PELI, E.: Tracking Telescope Aiming Point for Biopic Driving Surveillance. *Proceedings of International Conference on Image Processing, Computer Vision, and Pattern Recognition*, Las Vegas: WORLDCOMP 2009.



Xianping Fu received the Ph. D. degree in communication and information systems from Dalian Maritime University, Dalian, China, in 2005. Now he is Professor at Information Science and Technology College, Dalian Maritime University, Dalian, China. From 2008 to 2009 he was a Postdoctoral Fellow at Schepens Eye Research Institute, Harvard Medical School, Boston, MA. His research interests include perception of natural scenes in engineering systems, including multimedia, image/video processing, and object recognition.



Ying ZANG received her B. Sc. degrees in computer science and technology of Liao-Ning University (China) in 2004 and her M. Sc. degree in computer science and technology of Dalian Maritime University in 2010. Her research interests include digital image processing and pattern recognition.



Hongbo Liu is a Professor at the School of Information Science and Technology at Dalian Maritime University, Dalian, China. He received Ph. D. degrees in computer science from Dalian University of Technology. His research interests include system modelling and optimization involving soft computing, probabilistic modeling, cognitive computing, machine learning, data mining, etc. In the last several years, he has been engaged in research and teaching on design and analysis of computer algorithms, agent system, intelligent robotics, brain and cognition, etc. He actively participates and organises international conferences and workshops.