Optics and Lasers in Engineering 67 (2015) 49–56

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

Novel automatic eye detection and tracking algorithm

Kamarul Hawari Ghazali [a], Mohd Shawal Jadin [b,*], Ma Jie [a], Rui Xiao [a]

[a] Vision and Intelligent System Research Lab, Faculty of Electrical & Electronic Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
[b] Sustainable Power, Electrical and Renewable Energy Research Group, Faculty of Electrical & Electronic Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia

ARTICLE INFO

Article history:
Received 13 August 2014
Received in revised form
20 October 2014
Accepted 1 November 2014
Available online 20 November 2014

Keywords:
Eye detection
Eye tracking
Sparse matrix
Image processing

ABSTRACT

The eye is not only one of the most complex but also the most important sensory organ of the human body. Eye detection and eye tracking are basement and hot issue in image processing. A non-invasive eye location and eye tracking is promising for hands-off gaze-based human–computer interface, fatigue detection, instrument control by paraplegic patients and so on. For this purpose, an innovation work frame is proposed to detect and tracking eye in video sequence in this paper. The contributions of this work can be divided into two parts. The first contribution is that eye filters were trained which can detect eye location efficiently and accurately without constraints on the background and skin colour. The second contribution is that a framework of tracker based on sparse representation and LK optic tracker were built which can track eye without constraint on eye status. The experimental results demonstrate the accuracy aspects and the real-time applicability of the proposed approach.

1. Introduction

As one of the most vital features of face characteristics, eye and eye motion play a salient role in expressing a person's desires and needs, cognitive processes, emotional states, and interpersonal relations [1]. A great number of researches involve eye motion, such as biometric security, intelligent human–computer interface and driver fatigue detection. The challenge of eye detection and eye tracking is caused by illumination changing, in-plane rotation, out-plane rotation appearance changing and occlusion. The following parts, we will introduce the previous works about eye location and eye tracking:

1.1. Eye location

According to the survey work in [2], eye location methods are divided into 5 classes as following:

1. Shape-based approaches, which depend on the iris, the pupil and the shape of the eye.
2. Feature-based shape methods, which explore the characteristics of the human eye to identify a set of distinctive features around the eyes such as the limbs, pupil, and cornea reflections.
3. Appearance-based methods, which detect eyes directly based on the photometric appearance as characterized by the colour distribution or the filter response of the eye and its surroundings.
4. Hybrid modes, this class aims at combining the advantages of different eye models within a single system to overcome their respective shortcomings.
5. Other methods, which are not felicitous, described by the previous categories, for instance, symmetry operators, temporal information and active light.

Considering the shape of iris or pupil, the Hough transform is hired to find eye location in [3], but requires explicit feature detection. Perez et al. used thresholds of image intensities to detect and evaluate the centre of pupil, but this method is sensitive to the illumination changing [4]. Yuille et al. proposed a deformable template model which consists of two parabolas representing the eyelids and the circle for the iris [5]. This method does not consider the situation when the eye is close. A neural network eye detector is trained in [6] to detect rotated or scaled eyes; the detector can work under various light conditions, but just detects frontal view face only and needs more calculation time. Sirohey and Rosenfeld presented the methods for eye detection using linear and nonlinear filter [7]. Four Gabor wavelets were employed to detect edges of the eye; the shortage of this method is complex and time-consuming. D'Orazio et al. convolved an image with a circular filter, the value of convolution provides a candidate centre of iris in the image and usually gets wrong location in complex backgrounds [8]. Ishikawa et al. [9] proposed a method which combines the shape and appearance models through an active appearance model (AAM) [10], the shortage of this kind method is time-consuming and complex. Eye detection under an active infrared light which uses

* Corresponding author. Tel.: +60 94246056; fax: +60 94246111.
E-mail address: mohdshawal@ump.edu.my (M.S. Jadin).

http://dx.doi.org/10.1016/j.optlaseng.2014.11.003
0143-8166/

wavelength around 780–880 nm is used in [11–13]; this is an easy way but needs extra hardware support.

1.2. Eye tracking

Eye tracking is a concrete application of object tracking. Object tracking is an important issue in computer vision and a thorough survey can be found in [14]. Tracking algorithm can be generally categorized as either generative or discriminative based on their appearance models. Generative tracking algorithms usually learn a model to represent the target object and then use it to search for the image region with minimal reconstruction error. Discriminative algorithms pose the tracking problem as a binary classification task in order to find the decision boundary for separating the target object from the background [15]. Eye tracking has been found in numerous applications in multiple fields. Zhu and Ji describe a detection system based on active IR illumination for real-time gaze tracking [15]. Villanuev et al. described an eye tracking system whose objective is to identify user gaze direction [16], this needs extra hardware support. In paper [17], the authors perform face detection first before eye tracking to increase the accuracy of the eye tracking algorithm, this system ignores the status of eye closure. Hansen et al. present a new approach addressing navigation and selections in large information spaces with noisy inputs, this approach is time-consuming [18].

The main target of this work is to propose an approach using standard camera for tracking eye accurately and in real-time without constraints of eye status, background and more robust to illumination changing, in-plane rotation and out-plane rotation. The remainder of this paper is organized as follows: Section 2 discusses the proposed method. Experimental results and discussions are given in Section 3 and finally concluding remark appears in Section 4.

2. Methodology

Fig. 1 shows the framework of the proposed eye detection and tracking algorithm. After capturing the image, the first step is to search for human face in the whole image. Then the eye filter is employed to point the eye location. After successfully locating the eye location, the on-line trained classifier will be used to detect and track the eye movement. If any step lost the target, then go to the first step.

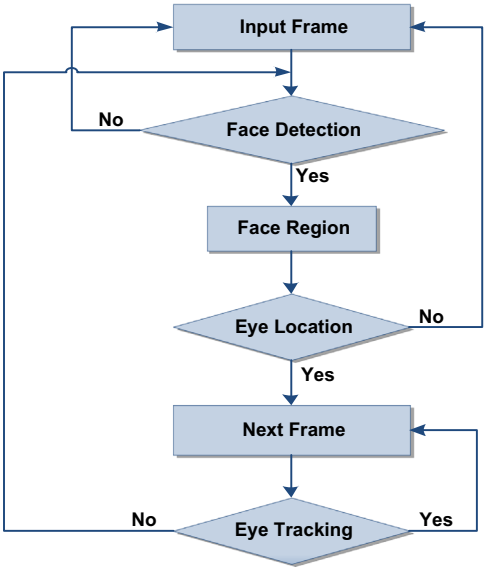

Fig. 1. The steps for eye detection and tracking system.

2.1. Face detection

In recent years, the researches on face detection have reached a great achievement, for example, colour–space based detection, neural network-based detection and feature-based detection. Considering the time consumption of algorithm, accuracy of detection algorithm and robustness of the algorithm, the Adaboost-based face detector proposed by Viola and Jones is a milestone in solving face detection tasks [19]. The basic principle of their detection algorithm is using a sub-window which can be rescaled to scan a picture or video frame to filter object and reject background based on a set of values of Harr-like features. The basic Harr-like feature is depicted in Fig. 2. The face detection procedure is shown in Fig. 3.

2.2. Eye location

Eye location is often the first and one of the most important steps for many computer vision applications such as facial recognition, facial expression recognition and fatigue detection. The accuracy of eye location will affect the subsequent processing and analysis. This work is inspired by D'Orazio et al. [8] and Hester and Casasent [20]. The proposed algorithm of eye location belongs to appearance-based methods. This novel algorithm uses a trained eye filter to convolute with frontal face image, analyse the response result to find the maximum value [21]. According to the convolution theorem, it appears that convolution in the spatial domain becomes element-wise multiplication in the frequency domain.

$$g(x,y) = f(x,y)\, h(x,y) = F^{-1}(F(u,v)H(u,v)) \tag{1}$$

where $h(x, y)$ is the eye filter kernel, $f(x, y)$ is the face image and $g(x, y)$ is the convolution result in time domain. We hope that $g(x, y)$ is synthetically generated with a bright peak at the centre of the eye and small values everywhere else. So $g(x, y)$ could be a two-dimensional Gaussian function, defined as

$$g(x,y) = e^{(x-x_0)^2 + (y-y_0)^2/\sigma^2} \tag{2}$$

where x_0 and y_0 are the locations of left eye or right eye, σ is the x and y spreads of blob. The pseudo code of build eye kernel algorithm is described as follows:

- Repeat for $i = 1, 2, \ldots, N$:
  1. Get new image and resize image to $f_i(x,y)$.
  2. Apply a random rotation up to $\pm\frac{\pi}{32}$, scales up to 1.0 ± 0.2 in the resized image.
  3. Get the location of left eye (x_{Li} and y_{Li}) and build the Gaussian for the location of left eye $g_{Li}(x,y)$.

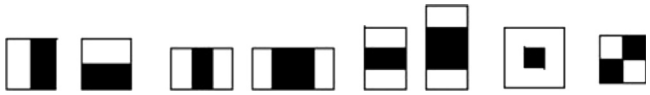

Fig. 2. The Harr-like features.

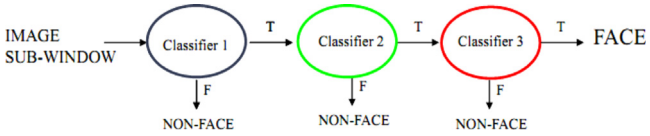

Fig. 3. The procedure of face detection.

4. Get the location of right eye (x_{Ri}, y_{Ri}) and build the Gaussian for the location of right eye, $g_{Ri}(x,y)$.
5. Convert $f_i(x,y)$ to frequency domain and get Fourier transform $F_i(u,v)$.
6. Convert $g_{Li}(x,y)$ and $g_{Ri}(x,y)$ into the frequency domain and get Fourier transform $G_{Li}(u,v)$ and $G_{Ri}(u,v)$.
7. Calculate $H_{Li} = \frac{G_{Li}(u,v)}{F_i(u,v)}$ and $H_{Ri} = \frac{G_{Ri}(u,v)}{F_i(u,v)}$.

- Output the left eye kernel $H_L(u,v) = \frac{1}{N}\sum_{i=1}^{N} H_{Li}(u,v)$.
- Output the right eye kernel $H_R(u,v) = \frac{1}{N}\sum_{i=1}^{N} H_{Ri}(u,v)$.

2.3. Eye tracker

In this paper, the eye tracking framework is composed of three parts. One is an optic flow tracker [22], another is online-trained object classifier based on compressive sensing, the last part is fusion which merges the location from the tracker and classifier and outputs the final decision. The optic flow tracker is able to adapt to change of object appearance, but it usually fails when the object of interest occlude or disappear from image frame. The online-trained classifier can get good result in object occluded or reappeared in the image sequence, but is sensitive to the dramatic appearance changing and cluttered background. So, the tracker and the classifier complement each other. The cooperation between tracker and classifier can contribute to improving system's accuracy and robustness.

2.4. Optic flow tracker

The Lucas–Kanade (LK) optical flow algorithm is an older, well-established and widely used method. The LK algorithm originally proposed in 1981 as the method is easily applied to a subset of points in the input image. The LK algorithm can be applied in a sparse context because it relies only on local information that is derived from some small window surrounding each of the points of interest. The shortcoming of using small local windows in the LK optical flow algorithm is that the large motions can move points outside of the local window and thus become impossible for LK algorithm to find. Then the pyramidal LK algorithm was developed to reduce the disadvantage mentioned before. The pyramidal LK algorithm starts from the highest level of an image pyramid which has lowest detail and working down to lower levels as described in Fig. 4.

The subset of point is collected by using corner detection. A corner can be defined as the intersection of two edges which should yield a large change in appearance. A fast corner detector named features from accelerated segment test (FAST), which was proposed by Rosten and Drummond in 2005 [23], was employed to find a corner in sub-window. The segment test criterion

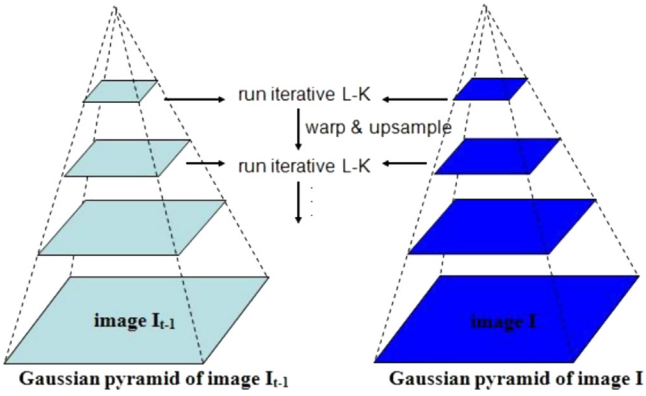

Fig. 4. The pyramidal LK optical flow.

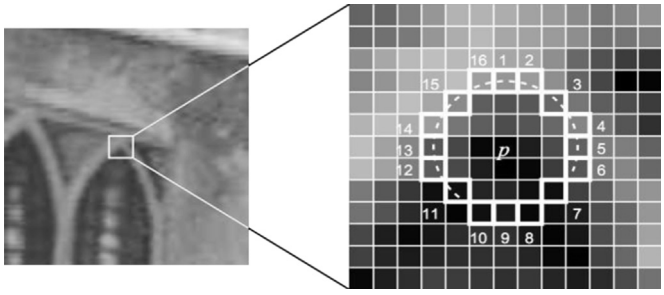

Fig. 5. Definition of corner [23].

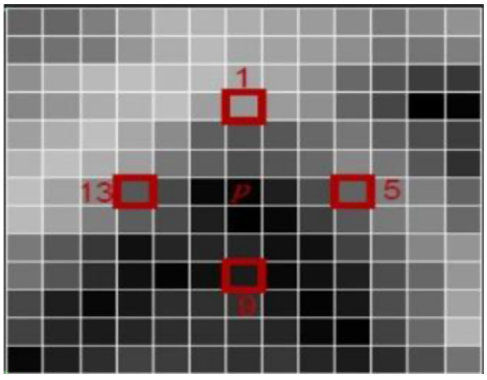

Fig. 6. Key point of corner [23].

operates by considering a circle of 16 pixels around the corner candidate p shown as Fig. 5. The circle radius is 3 pixels. The feature is detected at p if the intensities of at least 12 contiguous pixels are all above or all below the intensity of p by some threshold, t. This procedure can be optimized by examining pixels 1, 9, 5 and 13, to judge candidate pixels more quickly, because the feature of corner can only exist if three of these text pixels (1, 9, 5 and 13) as shown in Fig. 6 are all above or below the intensity of p by the threshold t.

In order to avoid the tracking failure caused by wrong position of target object, an approach named position consistency testing is developed to evaluate the reliability of the optical flow tracker. The theory of this approach describes as follows:

Let $(I_t, I_{t+1}, \ldots, I_{t+k})$ be an image sequence and P_t be a point location in time t. The object trajectory is $\overrightarrow{T_k} = (p_t, p_{t+1}, \ldots, p_{t+k})$. Tracking back from I_{t+k} to I_t to produces the backward trajectory $\overleftarrow{T_k} = (\tilde{p}_t, \tilde{p}_{t+1}, \ldots, \tilde{p}_{t+k})$. The distance between $\overrightarrow{T_k}$ and T_k is defined as dist $\left(\overrightarrow{T_k}, \overleftarrow{T_k}\right) = p_k - \tilde{p}_k$. If the distance in the x and y directions separately is smaller than the threshold θ, then the result of a tracker can be trusted.

2.5. Online-trained object classifiers

The on-line trained classifier is developed based on sparse representation which was inspired by Zhang et al., Babenko et al. and Li et al. [15,24,25]. Shannon sampling theorem is the bridge connecting the real world and digital world and the traditional approach of reconstructing signals or images from measured data. Compressive sensing surpasses the traditional limits of sampling theory, because it builds upon the fundamental fact that many signals or images can be represented by only a few non-zero coefficients in a suitable basis or dictionary [26]. Many previous studies treat tracking as a classification problem, the classifier selects interest object from background [15,24]. In this work, a set of feature vectors is abstracted using a very sparse matrix in

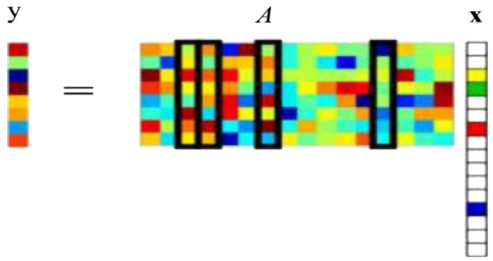

Fig. 7. The expression of compressive sensing.

reference image and adaptive classifiers are trained to select object patch from the background. The involved concept and technique will be described in detail in the following sections.

2.5.1. Sparse representation

The global appearance of an object under different illumination and viewpoint conditions is known to lie approximately in a low dimensional subspace [27]. Fig. 7 clearly depicts the concept of sparse representation. Where $y \in \Re^n$ is a lower-dimensional space; $y \in \Re^{nm}$ is a random measurement matrix and $x \in \Re^m$ is the high-dimensional image space.

The sparse matrix A (as described in Fig. 7) can be constructed according to the work of Li et al. [28]. The sparse matrix is defined as

$$r_{ij} = \sqrt{sx}\begin{cases} 1 & \text{with probability } \frac{1}{2s} \\ 0 & \text{with probability } 1-\frac{1}{s} \\ -1 & \text{with probability } \frac{1}{2s}. \end{cases} \tag{3}$$

Achlioptas used $s=1$ or $s=3$ that satisfies the Johnson–Lindenstrauss lemma [29], which governs the projection of high dimension Euclidean vectors to a lower dimensional Euclidean space $s=3$; one can achieve a threefold speedup because only one third of the data need to be processed. Li et al. mentioned that the random projections are almost as accurate as the conventional random projection where $r_{ij} \in N(0,\ 1)$ [28].

2.5.2. Multiple instance training

In the procedure of collecting positive and negative samples, the positive samples are the patches that overlap with interest object as green rectangle, and the negative samples are the patches that have a long distance to interest object as a red rectangle in Fig. 8. This method for producing positive and negative samples is proposed in MILTrack [24]. In multiple instance learning, training data has the form of $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where a bag $X_i = \{x_{i1}, \ldots, x_{im}\}$ and y_i is a bag label. The labels are defined as $y_i = \max_j \left(y_{ij}\right)$ where, y_{ij} is the instance labels, which are not known during training. Each positive or negative patch multiplies with sparse matrix to reduce the feature dimensionality. This processing can reduce the time consumption of the classifier.

2.5.3. Definition and update of classifier

In this stage, the naïve Bayes classifier [30] which updated online is built to find object patch from background patches. Each classifier h_k is composed of a Harr-like feature f_k and four parameters $(\mu_1,\ \sigma_1,\ \mu_0$ and $\sigma_0)$. h_k can be expressed as

$$h_k(x) = \log\left[\frac{p(y=1 \mid f_k(x))}{p(y=0 \mid f_k(x))}\right] \tag{4}$$

where $p(f_k(x)|y=1) \approx N(\mu_1$ and $\sigma_1)$ and $p(f_k(x)|y=0) \approx N(\mu_0$ and $\sigma_0)$ $y \in \{0, 1\}$.

The low dimensional feature representation for each patch can be denoted as $V = (v_1, \ldots, v_n)^T$, where $V \in \Re^n$ and $m >> n$. Finally, the classifier H can compose of every h_k.

$$H(v) = \log\left[\frac{\prod_{i=1}^{n} p(y=1|v_i)}{\prod_{i=1}^{n} p(y=0|v_i)}\right] = \left[\sum_{i=1}^{n} \log\frac{p(v_i|y=1)}{p(v_i|y=0)}\right] \tag{5}$$

where let $p(y=1)=p(y=0)$ and $y \in \{0, 1\}$ represents the sample label. When the classifier gets new data, the parameter (μ_1, σ_1) update following such rules:

$$\mu_i \leftarrow \alpha\mu_i + \frac{(1-\alpha)}{n}\sum_{y_k=1} v_i(k) \tag{6}$$

$$\sigma_i \leftarrow \sqrt{\alpha\sigma_i^2 + (1-\alpha)\sigma^2 + \alpha(1-\alpha)\left(\mu_i - \sum_{y_k=1} v_i(k)\right)^2} \tag{7}$$

where $i \in (0, 1)$ and $0 < \alpha < 1$ is a learning rate, $\sigma = \frac{1}{n}\sum_{y_i=i}\sqrt{v_i(k) - \frac{1}{n}\sum_{y_i=i} v_i(k)}$.

According to the previous description, the eye tracker consists of a bank of weak classifiers. When getting the eye area in first time, the positive samples which are the patches near eye area and negative samples which are far from the eye area are selected. These sample patches map to compressed domain. A feature pool is built by using Harr wavelet; each classifier is built by selecting the Harr wavelet randomly. In the next frame, the object will be

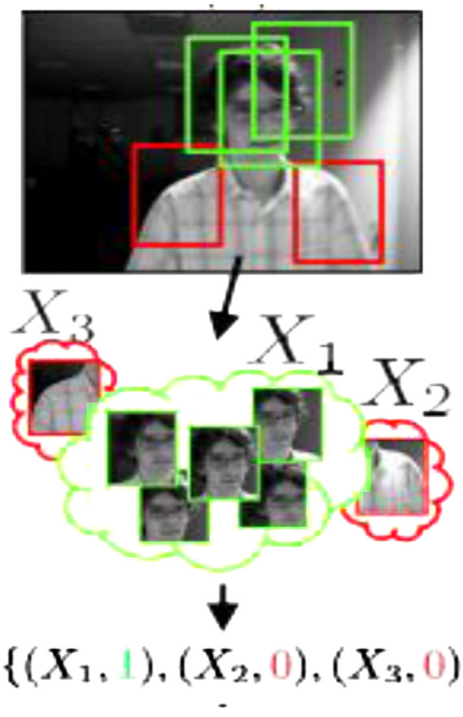

Fig. 8. Using one positive bag consisting of several images patches [24]. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

Table 1
The performance of eye location.

	Left eye	Right eye
Detection	195	194
false	5	6
Accurate rate (%)	97.5	97

searched in a circle. The centre of the circle is left up corner of the eye location in previous frame. In the tracking process, classifier is updated on line according to (3) and (4).

2.6. Fusion and validity

This integrator is used to combine the response of LK optic flow L_t and the classifier C_t into a single response. We use the predicate valid B_t to express a high degree of confidence that the final result. Let B_{t-1} be the confidence result in previous frame. The confidence rate is the overlap between new response of LK tracker and classifier and previous result B_{t-1}. The pseudo code for fusion and validity is described as follows:

Algorithm for fusion and validity

Input: L_t,C_t and B_t
Output: B_t
1: If valid (B_{t-1}) then
2: O_L=overlap (L_t, B_{t-1})
3: O_C=overlap (C_t,B_{t-1})
4: If $O_L > O_C$ then
5: $B_t = L_t$
6: Else:
7: $B_t = C_t$
8: Else:
9: $B_t = (L_t \cup C_t) \cap (L_t \cap C_t)$
10: If max (O_L,O_C) < 0.7 then
11: Restart system
12: End

3. Result and discussion

The goal of this section is to experimentally and scientifically demonstrate the validity of eye location algorithm and eye tracker. In this section, the first part of experiment is to test the performance of eye location and the other part works are assessing the performance of work frame for eye tracking. The software has been implemented by using C++ language on an Intel Core2 6400@ 2.13 GHz processor and Samsung DD2 2 G @ 667 MHz RAM. The system has been developed using CodeBlocks 10.5, OpenCV 2.3 and GCC 4.5.2.

The first experiment is to detect the accuracy of eye filters. A set of total 200 images was collected from FERET face library, Yale face library and INDIAN FACE DATABASE. All of these frontal or near frontal face pictures are not in the same illumination condition. The metric for evaluating the algorithm can be described as follows. If the centre of eye, which is computed using eye filters has displacement more than ± 1 pixels in the x and y directions from the ground true of eye centre, its status is expected to get eye location correctly. The experimental result is depicted in Table 1. Some samples of test result can be seen in Fig. 9.

The second experiment is to evaluate the runtime performance of the eye location compared with Gabor filter [31], OpenCV Cascade classifier [32]. The times to process the same picture for each of the three algorithms are described in Table 2.

The third experiment is to evaluate the performance of eye tracker, the proposed system track one eye motion in video sequences. Because there is no public video sequence for eye tracking, two self-made videos will be employed. One metric to evaluate the proposed work frame, called success rate, is denoted as follows:

$$SR = \frac{R_T \cap R_G}{R_T \cup R_G} \tag{8}$$

where R_T is the tracking bounding rectangle provided by the tracker and R_G is the ground truth bounding rectangle. If the SR is larger than

Table 2
Runtime performance.

	Gabor filter	Cascade classifier	Proposed filter
Testing time	2.86 s	83.6 ms	11.5 ms

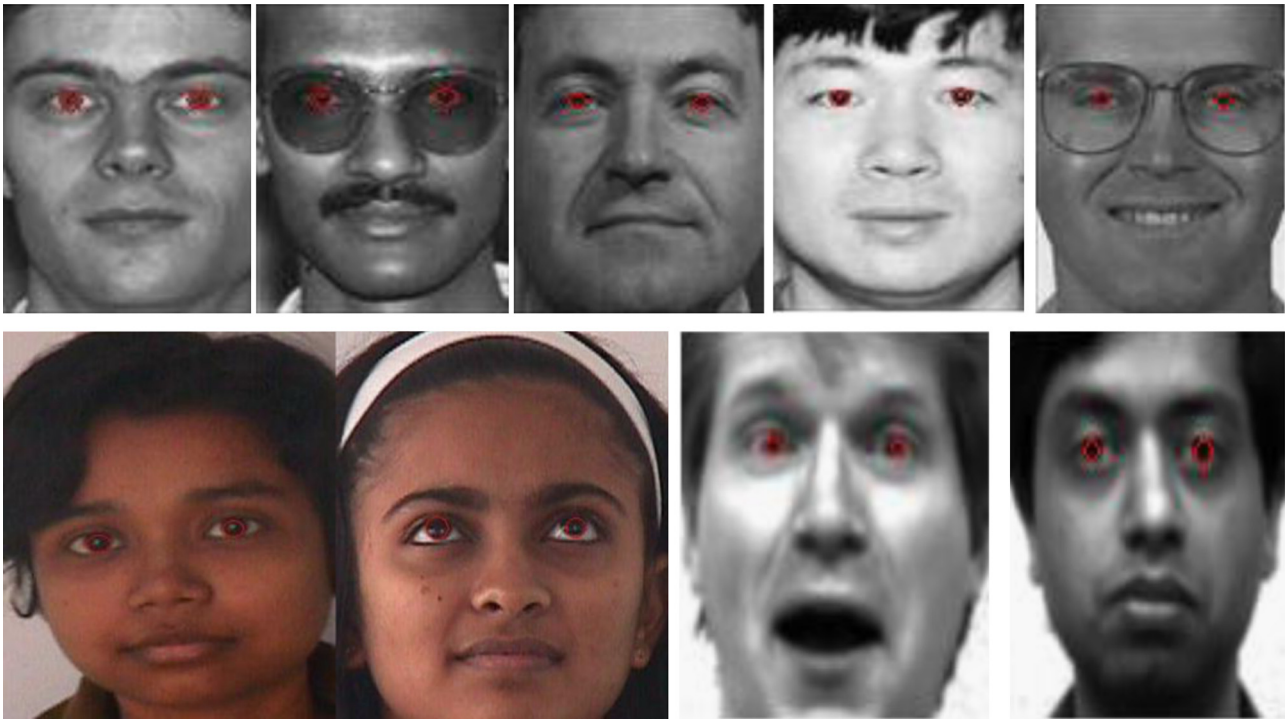

Fig. 9. Result of eye location.

0.5 in video frames, the tracking result will be considered as a success. Table 3 shows the quantitative results as described above. Fig. 10 shows some samples from video sequence.

The last experiment compares the performance between MILTrack [24] and the tracker proposed in the paper in accurate rate and frame rate. Two public testing video files were used in this experiment. Those videos can be downloaded from internet freely. Table 4 presents the accuracy rate of the algorithms. The samples for testing can be seen in Fig. 11. In Fig. 11, the blue rectangle was the result produced by MILTrack, the red rectangle was drawn by the proposed tracker.

The video sequence named David Indoor presents illumination changing, scale and pose changing. Then the Occluded Face presents poses variation and occlusion. The MILTrack runs at about 25 frames/s, and then the proposed tracker runs at 30 frames/s. According to the results of experiments, we can see that the algorithm of eye location is faster than previous algorithms and achieved a good accuracy rate (97%). Compared with [8], our approach just searches eye in the face area and in the [8] whole

Table 3
The performance of frame work.

	Video 1	Video 2
Success	85	65
false	3	2
Accurate rate (%)	96.6	97

Table 4
The rate of accuracy.

	David indoor	Occluded face	Girl
MILTrack (%)	69	99	52
Proposed tracker (%)	90	100	81

Fig. 10. Samples of eye tracking: (a) frames of video 1 and (b) frames of video 2.

Fig. 11. Samples of head tracking, the blue rectangle produce from MILTrack, the red rectangle produces from proposed tracker in this paper: (a) David Indoor, (b) occluded face and (c) girl. (For interpretation of the references to colour in this figure , the reader is referred to the web version of this article.)

image is searched to find the eye even in complex backgrounds. So our approach has a better performance than the algorithm in [8]. The difference between our approach and [20] is the method used in the training filter. In [20] the filters are trained for just eye area, and the eye filters proposed in this paper are trained on the entire face image, including the nose, mouth, "wrong eye detection" etc. One of the major drawbacks with [20] is that it has constraints with the output at a single point at the origin of the correlation plane without maximizing it. Therefore, our approach is more accurate than the method in [20]. In [24,25], an algorithm is applied to solve the L_1 minimization problems. It is a time-consuming procedure compared to our proposed approach. The performance of the proposed tracker is better than MILTrack. As the result shown, the tracking algorithm has a good performance in terms of accuracy rate which is more than 96% and can deal with the situation of eye-opening and eye closure as illustrated in Fig. 8.

4. Conclusions

In this paper, we presented a fast and accurate convolution based eye filters and innovative framework for eye tracking. The eye location detection consists of two steps: first, the human face is detected in the whole image. Then, the face area is filtered by eye filters. The maximum value of the response is the centre of the eye. In tracking part, the multiple instance learning is used to train the appearance classifiers which update on-line results in more robust tracking. A very sparse matrix is adopted to efficiently compress features for interest object and background. The performance of the proposed algorithms is measured; these results show that the eye filters can rapidly and accurately locate the eye position. The presented work frame of eye tracking has online learning ability, so it can deal with not only eye opening situation but also eye closure situation. This algorithm also gets a good result in accurate rate. There are still some cases such as object disappearing, completely occluded for a long time and large scale changing, which will make the system maybe get wrong result or lost object position. The future work will focus on figuring out those situations.

Acknowledgement

The authors gratefully acknowledge Universiti Malaysia Pahang (UMP) for their financial support and facilities.

References

[1] Underwood G. Cognitive processes in eye guidance. Oxford: Oxford University Press; 2005.
[2] Hansen DW, Ji Q. In the eye of the beholder: a survey of models for eyes and gaze. IEEE Trans Pattern Anal Mach Intell 2010;32(3):478–500.
[3] Young D, Tunley H, Samuels R. Specialised hough transform and active contour methods for real-time eye tracking. Brighton: University of Sussex, Cognitive & Computing Science; 1995.
[4] Peréz A, Cordoba ML, Garcia A, Méndez R, Munoz ML, Pedraza JL. et al. A precise eye-gaze detection and tracking system. In: Proceedings of the 11th international conference in central Europe on computer graphics, visualization and computer vision; 2003. p. 105–8.
[5] Yuille AL, Hallinan PW, Cohen DS. Feature extraction from faces using deformable templates. Int J Comput Vis 1992;8(2):99–111.
[6] Reinders MJ, Koch RWC, Gerbrands JJ. Locating facial features in image sequences using neural networks. In: Proceedings of the second international conference on automatic face and gesture recognition; 1996. p. 230–5.
[7] Sirohey SA, Rosenfeld A. Eye detection in a face image using linear and nonlinear filters. Pattern Recognit 2001;34(7):1367–91.
[8] D'Orazio T, Leo M, Cicirelli C, Distante A, An algorithm for real time eye detection in face images. In: Proceedings of the 17th international conference on pattern recognition, vol. 3; 2004. p. 278–81.
[9] Ishikawa T, Baker S, Matthews I, Kanade T. Passive driver gaze tracking with active appearance models. In: Proceedings of the 11th world congress on intelligent transportation systems; 2004.
[10] Cootes TF, Edwards GJ, Taylor CJ. Active appearance models. In: H. Burkhardt, B. Neumann (eds.), In computer vision ECCV. Berlin, Heidelberg: Springer; 1998; 484–98.
[11] Li D, Winfield D, Parkhurst DJ. Starburst: a hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In: Proceedings of the conference on computer vision and pattern recognition-workshops; 2005, 79–87.
[12] Hansen DW, Pece AE. Eye tracking in the wild. Comput Vis Image Underst 2005;98(1):155–81.
[13] Droege D, Schmidt C, Paulus D, A comparison of pupil centre estimation algorithms. In: Proceedings of the 4th conference on communication by gaze interaction COGAIN; 2008. p. 23–6. Yilmaz A, Javed O, Shah M. Object tracking: A survey. ACM Comput Surv 2006; 38(4):13.
[14] Yilmaz A, Javed O, Shah M. Object tracking: a survey. ACM Comput Surv 2006;38(4):13.
[15] Zhang K, Zhang L, Yang MH. Real-time compressive tracking. In: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (eds.), Computer vision–ECCV. Berlin, Heidelberg: Springer; 2012. p. 864–77. (b) Zhu Z, Ji Q. Eye and gaze tracking for interactive graphic display. Mach Visi Appl 2004; 15(3):139–48.
[16] Villanueva A, Cabeza R, Porta S. Gaze tracking system model based on physical parameters. Int J Pattern Recognit Artif Intell 2007;21(5):855–77.
[17] Chen YS, Su CH, Chen JH, Hung YP, Fuh CS, Chen CS. Video-based eye tracking for autostereoscopic displays. Opt Eng 2001;40(12):2726–34.
[18] Hansen DW, Skovsgaard HHT, Hansen, JP, Møllenbach E. Noise tolerant selection by gaze-controlled pan and zoom in 3D. In: Proceedings of the symposium on eye tracking research and applications; 2008. p. 205–12.
[19] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In: Conference on computer vision and pattern recognition, vol. 1; 2001. p. 511–518.
[20] Hester CF, Casasent D. Multivariant technique for multiclass pattern recognition. Appl Opt 1980;19(11):1758–61.
[21] Bolme DS, Draper BA, Beveridge JR. Average of synthetic exact filters. In: Conference on computer vision and pattern recognition; 2009.p. 2105–12.
[22] Bouguet JY. Pyramidal implementation of the Lucas Kanade feature tracker. Intel Corporation, Microprocessor Research Labs,2000, (⟨http://www.intel.com/research/mrl/research/opencv⟩).
[23] Rosten E, Drummond T. Fusing points and lines for high performance tracking. In: Proceeding of the IEEE international conference on computer vision; 2005. p. 1508–11.
[24] Babenko B, Yang MH, Belongie S. Robust object tracking with online multiple instance learning. IEEE Trans Pattern Anal Mach Intell 2011;33(8):1619–32.
[25] Li H, Shen C, Shi Q. Real-time visual tracking using compressive sensing. In: Proceedings of the conference on computer vision and pattern recognition; 2011. p. 1305–12.
[26] Davenport MA, Duarte MF, Eldar YC, Kutyniok G. Introduction to compressed sensing. In: Yonina C. Eldar, Gitta Kutyniok (eds.), Compressed sensing: theory and applications. Cambridge: Cambridge University Press; 2011.
[27] Lee KC, Kriegman D. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2005. p. 852–9.
[28] Li P, Hastie T, Church K. Very sparse random projections. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining; 2006. p. 287–96.
[29] Johnson W, Lindenstrauss J. Extensions of Lipschitz mappings into a Hilbert space. In: Proceedings of the conference in modern analysis and probability; 1984. p. 189–206.
[30] Jordan MIO. On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. Adv Neural Inf Process Syst 2002;14:841.
[31] Wiskott L, Fellous JM, Kruger N, Von Der Mals-burg C. Face recognition by elastic bunch graph matching. IEEE Trans Pattern Anal Mach Intell 1997;19(7):775–9.
[32] Viola P, Jones MJ. Robust real-time face detection. Int J Comput Vis 2004;57(2):137–54.