A Vision-Based Hybrid Method for Eye Detection and Tracking

Kun Mu

Department of Computer Science and Engineering, Henan Institute of Engineering, Zhengzhou 451191, China

MuKun_China@126.com

Abstract

As one of the most salient features of human face, eyes play an important role in interpreting and understanding a person's desires, needs, and emotional states. In this paper, a robust real-time "coarse to fine" method based on combining the appearance-based method and SVM approach was proposed for eye detection firstly, which combining the respective strengths of different complimentary techniques and overcoming their shortcomings. And then, a complementary tracker was designed for tracking eyes, which the eye model is continuously updated by having the eye successfully detected from the last Kalman tracker, to avoid error propagation with the CamShift tracker. Experimental results show that these enhancements have led to a significant improvement in eye tracking robustness and accuracy over existing eye trackers, especially under various conditions identified.

Keywords: *Eye detection, eye tracking, Haar-like, SVM, Kalman*

1. Introduction

With rapid development of science and technology and people demand upbeat, Human-computer interaction has become an increasingly important part of our daily lives. Numerous research topics have been explored for HCI, such as hand gesture analysis, head gesture analysis, lip movement analysis, eye gaze estimation, facial expression analysis, and other body movements analysis.

Facial features detection and tracking is significant research field in HCI and its application during these 30 years. Automatic extraction of human face and facial features (eyes, nose and mouth, for example) is an essential task in various applications, including face and iris recognition, security, surveillance systems and human computer interfacing [1-5]. Accuracy of the face recognition system depends on the accurate localization of the facial features. These facial features either is used a classifier directly or for normalizing the images. Out of several of facial features eye is one of the most important facial feature, once the eye positions are detected then the other facial features can be detected very easily.

As one of the most salient features of human face, eyes play an important role in interpreting and understanding a person's desires, needs, and emotional states. In particular, the eye-gazes, indicating where a person is looking, can reveal the person's focus of attention. There are many researchers focusing on tackling the eye detection tasks. Compared with the active IR based approaches [6], the image based passive approaches are widely used for no extra equipment is needed. Generally, approaches for eye localization can be classified into three categories: template based methods, feature based methods and appearance based methods. In the template based methods [7], a generic eye model, based on the eye shape, is designed first. Template matching is then used to search for the eyes in the image. But the locating results are heavily affected by the eye model initialization and the image contrast.

The expensive time cost also prevents its wide application. Feature based methods explore the characteristics (such as edge and intensity of iris, the color distributions of the sclera and the flesh) of the eyes to identify some distinctive features around the eyes. But if the eye is closed or partially occluded by hair or due to face orientation, it will fail. Zhou *et al*., [8] use Generalized Projection Function (GPF) to detect the eyes. The appearance based methods detect eyes based on their photometric appearance. To well represent the eyes of different subjects, these methods usually need a large amount of training data under different face orientations and different illumination conditions. ASM [9] and AAM [10] are widely used for landmarks localization. And some other works are published for eye localization. Jesorsky *et al.*, use Haussdorf distance on edge images to complete the task [11]. Chen *et al.*, use boosted detectors to detect eyes, nose and mouth corners [12]. But most work is still not fast, robust and efficient enough to integrate into functioning user-interfaces or depend on the external sensing device.

In this paper, we focus on human eye detection and tracking with the use of video camera rather than using extra measurement devices such as helmet and special sensor. A robust real-time "coarse to fine" method based on combining the appearance-based method and SVM approach was proposed for eye detection firstly, which combining the respective strengths of different complimentary techniques and overcoming their shortcomings. And then, a complementary tracker was designed for tracking eyes, which the eye model is continuously updated by having the eye successfully detected from the last Kalman tracker, to avoid error propagation with the CamShift tracker. Experiments show that these enhancements have led to a significant improvement in eye tracking robustness and accuracy over existing eye trackers, especially under various conditions identified.

2. Related Previous Works

2.1. Haar Features and Integral Image for Face Detection

The rectangular masks used for visual object detection are rectangles tessellated by black and white smaller rectangles. Those masks are designed in correlation to visual recognition tasks to be solved, and known as Haar-like wavelets. By convolution with a given image they produce Haar-like features (See Figure 1).

(a) (b) (c) (d) (e) (f)

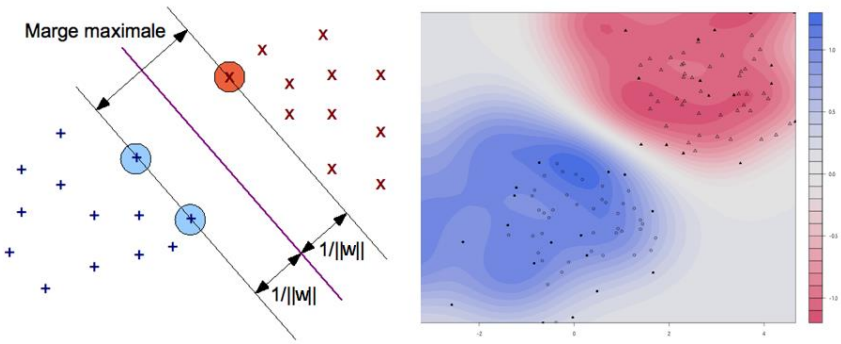
Figure 1. Face Detection Feature that Proposed by Viola: (a) and (b) are Edge Features, (c) is the Line Feature, (d) is the Diagonal Feature, (e) and (f) are Haar Features in Face Image

Viola [5] proposed a fast algorithm in which an integral image can be easily extracted from the characteristics of the local Haar features. The image coordinates of points are defined in the point image from the point at the top left. All the pixel gray values are expressed as line on the pixel gray value. Using his algorithm, a traversal of the fusion image can calculate the integral image of all points. Using the integral image, any of the original image pixel gray values within a rectangle computation is required as a constant, which can quickly calculate the characteristics of each feature value.

2.2. Support Vector Machines Approach

In machine learning, support vector machines (SVM) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, a SVM training algorithm builds a model that assigns new examples into one category or the other. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVM can efficiently perform non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces (see Figure 2).



(a) (b)

Figure 2. SVM Approach (a). Training a Support Vector Machine consists of finding the Optimal Hyperplane, that is, the one with the Maximum Distance from the Nearest Training Patterns. The Support Vectors are those (nearest) Patterns, a Distance $1/\|w\|$ from the Hyperplane. The Three Support Vectors are shown as Solid Dots (b). SVM Classification Plots

SVM is a two-class classification method that finds the optimal decision hyper-plane based on the concept of structural risk minimization [15]. Ever since its introduction, SVM has become increasingly popular.

3. Eye Detection based on Haar Feature and SVM Approach

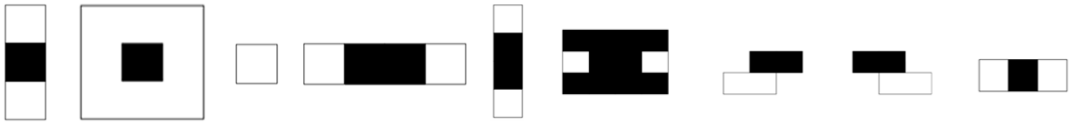
3.1. New Haar Features for Rough Eye Detection

Viola has used four features (Figure 1) for face detection and these features performed well on face. But for eye detection these features seem to insufficient, because eye has its unique characteristic different from face, for example, eye regions are darker compared to the bridge of the nose, eye regions are darker compared to the cheeks and the iris region is darker compared to the sclera. So, some new features should be proposed to work with eye detection.

In this paper, some new rectangle features are proposed for detecting eye more accurately and more adaptive to eye features. These features are proposed according to the appearance of the eye. Since eye corners usually appear darker than the neighborhood, feature g could

express this characteristic, so it should be introduced. Similarly, bright irises are often brighter than the other parts of the eye, so h is involved. Feature *i*, *j*, *l* and *m* represent the edge around the corner of the eye. Feature *k* shows the characteristic between the edge of eyeball and the eye corner. Examples of features express the eye can be seen in Figure 3 and Figure 4, and these features will improve the precision of the detection.

Integral image can be easily extracted from the characteristics of the local Haar features. The image coordinates of points are defined in the point image from the point at the top left. All the pixel gray values are expressed as line on the pixel gray value. Using the integral image, any of the original image pixel gray values within a rectangle computation is required as a constant, which can quickly calculate the characteristics of each feature value.



(f) (g) (h) (i) (j) (k) (l) (m) (n)

Figure 3. New Rectangle Features we proposed for Eye Detection

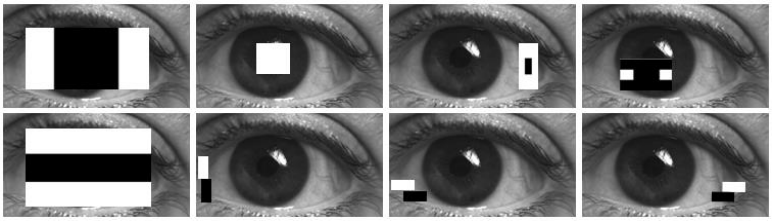


Figure 4. Some Feature Examples for Eye Detection

Within any image sub-window the total number of rectangle features is very large, far larger than the number of pixels. For an image size of 24×12, there are 32976 features originated from the rectangles in Figure 3 and 53928 features from all the rectangles in Figure 4. It is too computational to build a cascade classifier using so many features and is not necessary, in order to reduce training time, a preprocessor is used to exclude a large majority of the available features, leaving a small amount of features. In this paper 8988 optimum features are preselected from 53928 features using Adaboost, this classifier contains 7 levels, and these selected features will be used to build a cascade classifier. The detection results on face database are shown as below.

Figure 5. Some Results that is caused by the Classifier that produced by all Rectangle Features a, b, c, d, f, , h, I, j , k, l, m, n proposed by us

3.2. Eye Detection Using Support Vector Machines

After the detection of cascade classifier, some errors could come out, such as eyebrows, mouth, nares, larger or smaller eyes, to exclude these patches, some other method to classify the eye and non-eye patches that come out from the cascade classifier is necessary, and this method had better simple and fast to compute. So we introduce SVM classification algorithm for assisting to eye detection.

Support Vector Machines (SVM) is a two-class classification method that finds the optimal decision hyper-plane based on the concept of structural risk minimization. The theory of SVM can be briefly summarized as follows.

Let us consider first the simple case of linearly separable data. We are searching an optimal separating hyperplane:

$$\langle w,x\rangle + b > 0 \tag{1}$$

which minimizes the VC confidence term while providing the best generalization. The decision function is

$$f(x) = sgn(\langle w,x\rangle + b) \tag{2}$$

Geometrically, the problem to be solved is to find the hyperplane that maximizes the sum of distances to the closet positive and negative training examples. The distance is called margin and the optimal plane is obtained by maximizing $2/\|w\|$ or, equivalently, by minimizing $\|w\|^2$ subject to $y_i(\langle w,x\rangle + b) \geq 1$. Suppose now that the two classes overlap in feature space. One way to find the optimal plane is to relax the above constraints by introducing the slack variables ξ_i and solving the following problem (using 2-norm for the slack variables):

$$min_{\xi,w,b}\|w\|^2 + C\sum_{i=1}^{l}\xi_i{}^2 \tag{3}$$

$$\text{Subject to } y_i(\langle w,x\rangle + b) \geq 1 - \xi_i \ ,\forall i = 1,\dots,l \tag{4}$$

Where C controls the weight of the classification errors (C=∞ in the separable case). By introducing the lagrange multipliers, we obtain the primal and the dual Lagrangian form

$$L_p = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i^2 - \sum_{i=1}^{l}\alpha_i[y_i(\langle w,x\rangle) - 1 - \xi_i] \tag{5}$$

$$L_D = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j\, \alpha_i\alpha_j\langle x_i, x_j\rangle - \frac{1}{C}\sum_{i=1}^{l}\alpha_i \tag{6}$$

where $\alpha_i \geq 0$. The solution of the primal problem is linked to the solution of the dual by $w = \sum_i y_i\, \alpha_i x_i$.

We can express now the decision function as a function of α:

$$f(x) = sgn(\sum_{i\in S} y_i\alpha_i\, \langle x, x_i\rangle + b) \tag{7}$$

where $S = \{i|\alpha_i > 0\}$. The vector x_i, $i \in S$ are called support vectors and are the only examples from the training set that affect the shape of the separating boundary.

In practice however, a linear separating plane is seldom sufficient. To generalize the linear case one can project the input space into a higher-dimensional space in the hope of a better training-class separation. In the case of SVM this is achieved by using the so-called "kernel trick". In essence, it replaces the inner product $\langle x_i, x_j\rangle$ in (8) and (9) with a kernel

function $K\langle x_i, x_j\rangle$. As the data vectors are involved only in these inner products, the optimization process can be carried out in the feature space directly. Some on the most used kernel functions are:

$$\text{the polynomial kernel} \qquad K(x,z) = (\langle x,z\rangle + 1)^d \tag{8}$$

$$\text{the RBF kernel} \qquad K(x,z) = exp(-\|x-z\|^2/2\sigma^2) \tag{9}$$

3.3. Support Vector Machines Training State

Images of human faces are used as training data. Each training image consists of a single human face with manually located eye positions. A template is extracted around each eye, which we refer to as a training template. Two training templates are extracted for each training image, one for each eye. Positive training templates are those that corresponded to correct eye positions, we include eye images of different gazes, different degrees of opening, different face poses, different subjects, and with/without glasses. Negative training templates are those obtained by randomly selecting a sub-image in the training image that is "far enough" from a correct eye position. In our implementation "far enough" is anything further than 7 pixels in the horizontal and vertical directions from a valid eye position. The Figure 6 and Figure 7 show examples of eye and non-eye images in the training set, respectively.

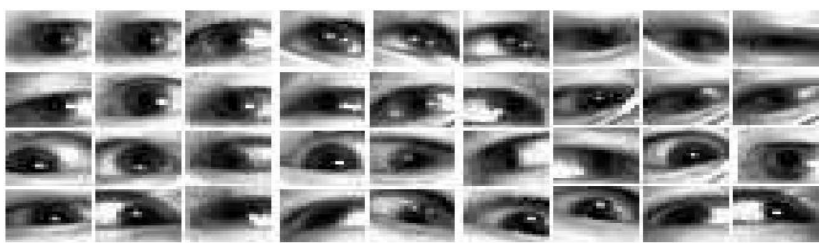


Figure 6. The Eye Images in the Positive Training Set

Figure 7. The Non-eye Images in the Positive Training Set

The training templates are 12 pixels high and 24 pixels wide. The negative set is not non-face part, but rather at an offset so that it also encompassed the other parts of the face including the nose and the mouth. It is usually difficult to recognize an eye using low quality images if the template consists of nothing but the non-face part. After finishing the above step, we get a training set, which has 682 positive images and 680 negative images. In order to obtain the best accuracy, we choose RBF kernel function for SVM classifier with $\sigma = 3$.

Eye verification with SVM works reasonably well and can generalize to people of the same race. However, for people from a race that is significantly different from those in training images, the SVM may fail and need to be retrained. SVM can work under different

illumination conditions due to the intensity normalization for the training images via histogram equalization. The eye verification results are shown as below.

Figure 8. The Eye Verification Results with SVM Classifier

4. Eye Tracking using a Combining Tracker

On the basement of work above, a robust eye detector is obtained at each frame. Given the detected eyes in the initial frames, the eyes in subsequent frames can be tracked from frame to frame. Eye tracking can be done by performing pupil detection in each frame. This brute force method, however, will significantly slow down the speed of pupil tracking, making real time pupil tracking impossible since it needs to search the entire image for each frame. This can be done more efficiently by using the scheme of prediction and detection. Kalman filtering [13] provides a mechanism to accomplish this. The Kalman pupil tracker, however, may fail if pupils are not bright enough under the different illumination conditions. In addition, rapid head movement may also cause the tracker to lose the eyes. This problem is addressed by augmenting the Kalman tracker with the Camshift tracker.

4.1. Eye Tracker with Kalman Filter

The Kalman filter uses a system's dynamics model (e.g., physical laws of motion), known control inputs to that system, and multiple sequential measurements (such as from sensors) to form an estimate of the system's varying quantities (its state) that is better than the estimate obtained by using any one measurement alone. As such, it is a common sensor fusion and data fusion algorithm.

A Kalman filter is a set of recursive algorithms that estimate the position and uncertainty of moving targets in the next time frame, that is, where to look for the targets, and how large a region should be searched in the next frame around the predicted position in order to find the targets with a certain confidence. It recursively conditions current estimate on all of the past measurements and the process is repeated with the previous a posteriori estimates used to project the new a priori estimates. This recursive nature is one of the very appealing features of the Kalman filter since it makes practical implementation much more feasible. The Kalman filter algorithm belongs to the state-space approach class of tracking algorithms. It solves the tracking problem based on the state-space equation and the measurement equation. To avoid being trapped by a local maximum, we first use one Kalman Filter to search the true maximum beyond the local one. The Kalman Filter is used to locate the start point that CamShift will search. We define the state-space equation:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ v_{x_{k+1}} \\ v_{y_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x_k \\ y_k \\ v_{x_k} \\ v_{y_k} \end{bmatrix} + W_k \tag{10}$$

And the measurement equation:

$$\begin{bmatrix} x_{ck} \\ y_{ck} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} x_k \\ y_k \\ v_{xk} \\ v_{yk} \end{bmatrix} + V_k \tag{11}$$

Where $k \geq 1$, W_k is a white Gaussian noise with diagonal variance Q. V_k is a white Gaussian noise with diagonal variance *R*. x_k, y_k is the centroid of the search window, x_{ck}, y_{ck} is the current measurement of the centroid, v_{xk}, v_{yk} is the velocity (displacement) of the pupil. T is the interval between the frames.

In addition, we use another Kalman Filter to predict the search window's width, b, and height h. We define the state-space equation:

$$\begin{bmatrix} b_{k+1} \\ h_{k+1} \\ r_{b_{k+1}} \\ r_{h_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} b_k \\ h_k \\ r_{bk} \\ r_{hk} \end{bmatrix} + U_k \tag{12}$$

And the measurement equation:

$$\begin{bmatrix} b_{ck} \\ h_{ck} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} b_k \\ h_k \\ r_{bk} \\ r_{hk} \end{bmatrix} + Z_k \tag{13}$$

Where b_k, h_k is the width and height of the search window, b_{ck}, h_{ck} is the current measurement of the width and height. r_{bk}, r_{hk} is the ratio of scale of the windows which is proportional to the scale of the pupil.

So with two Kalman Filters we can give more accurate the centroid and the size of the search window for CamShift.

4.2. Eye Tracker combining CamShif Approach

Due to the different illumination when moving the head, the eye region in the dark and bright pupil images exhibits strong and unique visual patterns such as the dark iris in the white part. This unique pattern should be utilized to track eyes in case the bright pupils fail to appear on the difference images. This is accomplished via the use of CamShift tracking algorithm. CamShift tracking is an appearance-based object tracking method. The flowchart of combining tracker is expressed as Figure 9.

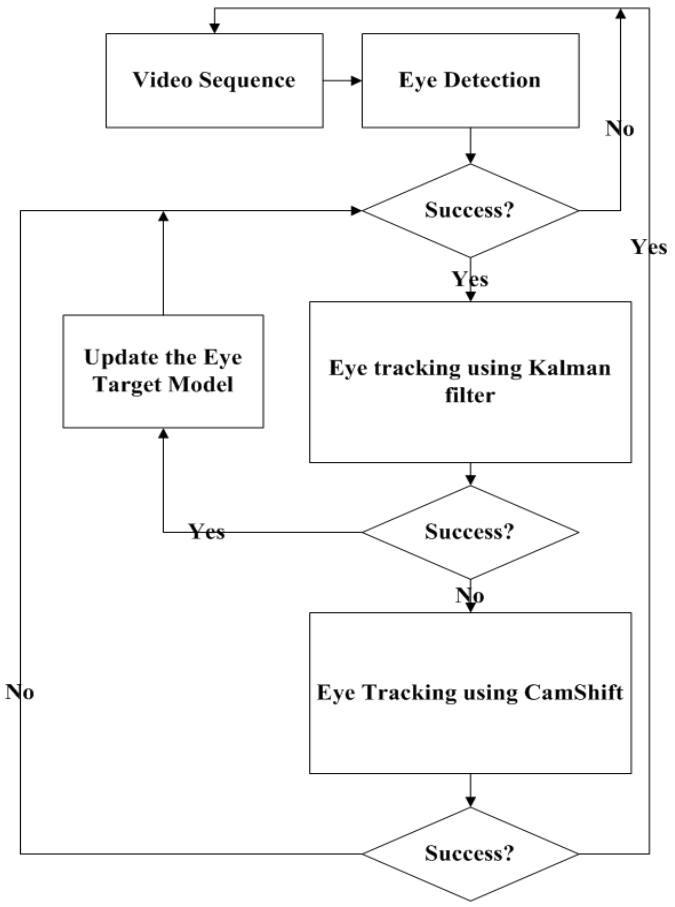


Figure 9. The Flowchart of Combining Eye Tracker we Proposed

Camshift tracking algorithm based on color performs well in solving the bottom problems of computer vision. Due to its robust and real-time quality, Camshift has become a basic tracking method which can adapt to the continuous variation of the shape and size of the target, compute fast and has strong anti-jamming capability, guaranteeing the stability and real-time of the system [14]. Camshift algorithm is a dynamic change in the distribution of the density function of the gradient estimate of non-parametric methods. It employs mean shift analysis to identify a pupil candidate region, which has the most similar appearance to the pupil model in terms of intensity distribution.

The course of algorithm is as follows:

1. Choose an initial search window W1, here, W1 is the eye detection result window we obtained above;
2. Run the MEANSHIFT algorithm;
3. Resize the search window according to the result of Step (2), and get a new window W2;
4. Use W2 as the initial search window for the next video frame and repeat the algorithm. When CAMSHIFT algorithm track a specific color object as pupil, the images do not have to calculate each frame all the pixels of the color probability distribution, just calculate pixel color probability distribution in the area that larger than the current search window. This can save a lot of computing.

In our combining eye tracker, The Kalman tracker is first initiated based on eye detection we obtained, assuming the presence of the bright pupils. When the bright pupils appear weak or disappear, the CamShift tracker is activated to take over the tracking. CamShift tracking continues until the reappearance of the bright pupils, when the Kalman tracker takes over. To prevent the CamShift tracker from drifting away, the target eye model is continuously updated by the eyes successfully detected by the Kalman tracker.

5. Experimental Results

The eye detection and tracking system we proposed in this paper is running on the hardware environment of Intel(R) Core (TM) 2 (2.93GHz), a Web camera, and the software environment of Windows 7 and Visual Studio 2008.

5.1. Experimental Results of Eye Detection

980 facial images in FERET database are select to test the eye detection experiment, including eye opened, half-closed and closed with different expressions and illumination. First, facial images come through the cascade classifier based on new Haar features we proposed, the eyes can be rough detected, the detection rate and error rate of cascade classifier for training are set as 0.99 and 0.3. After the rough detection step, some errors may come out, such as eyebrows, mouth, nares, larger or smaller eyes, to exclude these patches, so we introduce the geometric characters for further process. The detection results are described as Figure 10 and Table 1, eyes are detected more accurately and pupil locations are received at the same time.

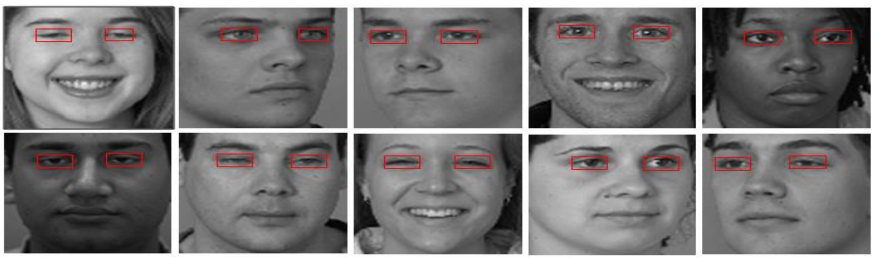


Figure 10. Experimental Results of Eye Detection on FERET Database using our Method

Moreover, we collected 156 images by web camera under different illumination, with\without glasses and head rotation for testing with the size of 320*240. The experimental data and results are described as Figure 11, and the average detection time is 67ms. The method we proposed shows that it is more accurate and robust.

Figure 11. Eye Detection under Different Circumstances

Table 1. The Eye Detection Results Data

	Two eyes are detected	One eye is detected	Two eye are missed	False drop
Original Rectangle feature	89.6%	5.3%	5.1%	27.9%(273/980)
Proposed method	97.6%	2.1%	0.3%	3.1%(30/980)

5.2. Experimental Results of Combining Eye Tracker

After eye detection, we test eye tracking effect by using combining eye tracker. The results are demonstrated as Figure 12 and Table 2. Testers are tested with opened, closed and occluded eyes due to face rotations. 400 frames per testers are extracted during test for eye tracking under different conditions.

Table 2. Tracking Results Statistics Comparison

	Kalman Tracker	CamShift tracker	Combining tracker
Left Eye(Open)	72.57%	71.89%	97.39%
Left Eye(Closed)	0%	0%	86.92%
Right Eye(Open)	72.83%	72.14%	97.84%
Right Eye(Closed)	0%	0%	86.74%
Left Eye(Occluded)	0%	0%	97.22%
Right Eye(Occluded)	0%	0%	97.67%

Figure 12. Eye Tracking by Combining Eye Tracker we Proposed

6. Conclusion

In this paper, a new method for eye detection based on Haar features and SVM classifier is proposed. We use Viola's conception, and add some new rectangle features to construct a cascade classifier for rough eye detection. The results from the classifier usually have some errors, such as eyebrows, mouth, nares, larger or smaller eyes, so SVM classifier are introduced to assistant to detect the eye location accurately. And then, a combining eye tracker to overcome the effect of eye closure and external

illumination by combining Kalman filter with CamShift algorithm is presented. This eye tracker can robustly track eyes under variable and realistic lighting conditions and under various face orientations. The experiment results on FERET database and web camera are excellent and robust in real time.

References

[1] P. Campadelli, R. Lanzarotti and G. Lipori, "Eye localization for face recognition", RAIRO Theoretical Informatics and Applications, no. 40, **(2006)**, pp. 123-139.
[2] C. Huang, H. Ai, Y. Li, and S. Lao, "High-Performance Rotation Invariant Multiview Face Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 4, **(2007)** April, pp. 671-686.
[3] O. Jesorsky, K. J., Kirchberg and R. W., "Robust face detection using the hausdorff distance", Lecture Notes in Computer Science, **(2001)**, pp. 212-227.
[4] Y. Ma, X. Ding, Z. Wang and N. Wang, "Robust precise eye location under probabilistic framework" Proceeding of IEEE Int'l Conf. Automatic Face and Gesture Recognition, **(2004)**
[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", Proceeding of IEEE Conf. of Computer Vision and Pattern Recognition, **(2001)**.
[6] Z. Zhu and Q. Ji, "Real-time eye detection and tracking under various light conditions", Proceedings of ACM SIGCHI Symposium on Eye Tracking Research and Applications, **(2002)** New Orleans.
[7] A. Yuille, P. Hallinan and D. Cohen, "Feature extraction from faces using deformable templates", International Journal of Computer Vision, **(1992)**, pp. 99-111.
[8] Z. Zhou and X. Geng, "Projection Functions for Eye Detection", Pattern Recognition, **(2004)**.
[9] T. F. Cootes, C. J. Taylor, D. H. Cooper and J. Graham, "Active Shape Models-Their Training and Application, Computer Vision and Image Understanding, **(1995)**, pp. 38-59.
[10] T. F. Cootes, G. J. Edwards and C. J. Taylor, "Active appearance models", 5th European Conference on Computer Vision, Berlin, **(1998)**, pp. 484-498.
[11] O. Jesorsky, K. J. Kirchberg and R. W. Frischholz, "Robust Face Detection Using the Hausdorff Distance", The 3rd International Conference on Audio and Video-Based Biometric Person Authentication, Verlag Berlin Heidelberg, **(2001)**, pp. 91-95.
[12] L. Chen, L. Zhang, L. Zhu, M. Li and H. Zhang, "A Novel Facial Feature Localization Method Using Probabilistic-like Output", Asian Conference on Computer Vision, **(2004)**, pp. 1-10.
[13] A. Blake, R. Curwen and A. Zisserman, "A framework for spatio-temporal control in the tracking of visual contours", International Journal of Computer Vision, vol. 11, no. 2, **(1993)**, pp. 127-145.
[14] W. Xu and E.-J. Lee, 'Continuous Gesture Trajectory Recognition System Based on Computer Vision", Appl. Math. Inf. Sci., Special Issue 2, no. 6, **(2012)**, pp. 339-346.
[15] C. Cortes and V. Vapnik, Support vector networks. Machine Learning, Vol. 20, pp. 273–297 **(1995)**

Author

Kun Mu received his B. S. and M. S. in Kiev University of science and technology in 2004 and 2007; currently, he is a lecture of School of Computer Science and Engineering, Henan Institute of Engineering, Zhengzhou, China. His main research fields are the nonlinear control theory and application research and computer control and automation.