Computer Vision and Image Understanding 115 (2011) 1223–1238

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

Robust real-time multi-user pupil detection and tracking under various illumination and large-scale head motion

Chao Yan [a], Yuangqing Wang [a,*], Zhaoyang Zhang [b]

[a] Department of Electronic Science and Engineering, NanJing University, NanJing, China
[b] Key Laboratory of Advanced Display and System Application (Shanghai University), Ministry of Education, Shanghai, China

ARTICLE INFO

Article history:
Received 17 December 2010
Accepted 8 March 2011
Available online 15 March 2011

Keywords:
Eye detection
Real AdaBoost
Real support vector machine
Correlation matching
Kalman forecast

ABSTRACT

A novel approach to Robust real-time multi-user pupil detection and tracking is presented, and this kind of detection and tracking behaves well under the circumstance of various illumination or large-scale head motion. Firstly, with active IR illumination, the possible positions of human pupils are depicted according to bright pupil effect and then some image pretreatment is conducted to diminish the fake pupil positions. Secondly, other than detecting human pupils directly, human faces in the image would be detected with real AdaBoost and the detected face positions would be optimized in order to save the time of whole processing. Thirdly, based on the faces detected, human pupils would be detected with real support vector machine (real SVM) and correlation matching. At last, the human pupils detected would be tracked with Kalman forecast in order to save the detection time of next image. Results from a series of experiments show that the new method could achieve real-time (30 frame per second) with a success rate of 95% for multiple users, and it is also proved that the new method is robust for illumination variation and large-scale head motion.

1. Introduction

Pupil detection and tracking means to obtain the positions of all human pupils in an image timely. There are several items that could be used for evaluating a pupil detection and tracking algorithm, such as accuracy [1–4,9], processing speed [4–6,9], robustness level [4,7–9], user number tolerance [7,10,11]. All these items of our novel algorithm will be discussed in the part of experiments.

Pupil detection and tracking plays an significant role in many domains, such as pattern recognition [3,12,13], driver alert [4,9,14], disabled people self-service [6], human–computer interaction (HCI) [15–17], multi-user 3D display [11], and gaze tracking [18–20]. For pattern recognition, it could be used for human identification, facial expression identification and gender identification; for driver alert, it could be used to monitor drivers' eye state in order to avoid drowsy drive; for HCI, computers could generate images timely according to the viewers' eye state; for multi-user 3D display, right views and left views could be delivered into the corresponding human pupils detected; for gaze tracking, the human pupils detected together with the human corneas could depict the users' gaze, and pupil detection and tracking also could help disabled people to communicate with computers.

There are still some obstacles in front of pupil detection and tracking. Illumination variation [6,7,21,25], wearing glass [22,16,25], eyes partly occluding [22,25], face leaning [21–23] and face rotating [4,7,22] could all decrease detection accuracy rate; viewers' fast move [22,16,25] and the sharply increase of viewer number would challenge detection responding speed. In addition, the different race, gender or age [24,16] of viewers could also influence the detection robustness level. Some obstacles in front of pupil detection and tracking are shown in Fig. 1.

2. Previous work

To overcome these difficulties and obtain well-performance pupil detection and tracking systems, many approaches have been brought up in the last two decades. Generally speaking, all these approaches belong to two different categories: One analyzes the images captured under natural illumination directly, which could be finished with some software and without any illumination hardware; the other one is based on active infrared (IR) illumination, which could be used to emphasize the human pupil positions, and then this kind of methods need both software support and hardware support. The first category has no dependence on illumination hardware, so its usage is relatively simple, but it behaves less robust for illumination variation and it is computationally expensive for lack of detection beginning reference. The second category needs additional hardware support; even so, it behaves

* Corresponding author.
E-mail addresses: yanchao3756@yahoo.cn (C. Yan), yqwang@nju.edu.cn (Y. Wang).

1077-3142/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2011.03.001

Fig. 1. Some obstacles in front of pupil detection and tracking (including face out-of-plane rotating, eyes occluding, illumination interference, wearing glasses and within-plane rotating).

robust for various illumination and it has fast processing speed with detection beginning reference.

There are four classes broadly included in the first category: knowledge-based methods [26–29], feature-based methods [13,30–32], template matching methods [13,21,33] and appearance-based methods [34–36]. Knowledge-based methods mean to detect human pupils based on the relevant knowledge mankind have accumulated, such as pupil shape, eye outline, eye blink, and pupil positions in human face. Feature-based methods mean to detect human pupils based on some pupil characteristics such as gray distribution around pupil, color distribution around pupil. These characteristics could not be observed directly, but they could be obtained with machine learning. Template matching methods mean to detect human pupils by calculating the correlation coefficient of different image areas. This kind of methods demand establishing an initialized template to start the whole processing, and the initialized template is very crucial for method detection accuracy. Appearance-based methods mean to detect human pupils based on statistics analysis and sample training. This kind of methods demand colleting a large amount of simples, and the simple universality is also very significant for method detection accuracy.

For knowledge-based methods, Zhou et al. [26] proposed a generalized projection function (GPF) for eye detection, which combines integral projection function (IPF) and variance projection function (VPF). The experimental results showed that GPF was more powerful than sole VPF or sole IPF, but GPF could not detect human eyes in real time. Dobes et al. [27] presented an eye location method base on modified Hough transformation, the location correctness of which is higher than 92% on two publicly available face image databases. But this method is quite time-consuming. Torricelli et al. [28] brought up a remote eye gaze tracking method based on blink detection. The method has also been tested on a publicly available database. The obtained average true prediction rate is higher than 95% and the method could work in real time at 30 fps. How could we transform human knowledge into efficient rules in pupil detection is the key point. This transformation should be appropriate: if the rules we make are too strict, the detection correctness could not be as high as we expected; if the rules we make are too loose, the detection false alarm rate could not be as low as we expected.

For feature-based methods, Feng et al. [30] explored eye detection based on gray intensity image and this method utilized three features of human eye: low intensity, line direction of eye corners, convolution of eye variance and face image. The method obtained a good performance on MIT face database but this method would fail when there was a rotation in depth (facing downwards). In [31], a recursive nonparametric discriminant analysis feature was employed to detect multi-view human eyes. The authors believed this new feature could handle more general class distributions than Fisher discriminant analysis and this new feature had less computational complexity than traditional nonparametric discriminant analysis. Smeraldi et al. [32] applied the log-polar sampling of Gabor decomposition as a kind of feature to eye detection. This method based on frequency domain could detect human eyes in real time. But it would fail when the viewer had a quick move, and it could not be used for multiple viewers. Although feature-based methods almost need machine learning to gather features, they are extensively used for their high correctness rate.

For template matching methods, Li et al. [13] proposed a judgment system of eye or non-eye based on template matching. The system utilized five parameters to measure the similarity of input image sub-window and eye template. The experimental results showed that this method was insensitive to different intensity contrast and different illumination. But this method would fail when there was a narrow eye and this method was not in real time. In [21], two unified deformable templates are introduced for single tracking and double eyes tracking respectively, each deformable template can describe both open and closed eye states. Experimental results showed that this method could track the locations of eye/eyes accurately, but it could not deal with low resolution image sequences and bad illuminated conditions still cause tracking failures. Zhu et al. [33] developed an algorithm that utilized a template-matching technique to calculate torsional eye position. Template matching methods are usually accurate. However, this kind of methods are comparatively more computational expensive.

For appearance-based methods, Coughlin et al. [34] brought up an automated eye tracking system based on artificial neural networks (ANN), which mapped two-dimensional (2D) eye movement recordings into 2D eye positions. In [35], AdaBoost algorithm was employed to track eye regions. Combined with Lucas–Kanade–Tomasi features, this method could track the eye regions in the rotated faces. Qian and Xu [36] proposed an eye detection method based on cluster analysis. Combined with Gabor transformation, this method could locate the positions of the eyes from frontal face

images, with an accuracy rate of 96.1% on the LFW database. But experimental results showed that wearing glasses could cause failures in this method. Generally speaking, appearance-based methods firstly obtain sample distribution model or discriminant function; then this kind of methods will detect target object.

Although pupil detection based on active IR illumination appeared much later than the methods above-mentioned, this kind of methods grew much fast for their high-efficiency. IBM [37] firstly proposed pupil detection and tracking using near-infrared light source in 1998. They succeeded in pupil detection with bright pupil effect, but their detection was simple and immature. Fast head motion, illumination variation, eyes occluding, wearing glasses or makeup could all caused failures in their method. To overcome these shortcomings, many approaches have been brought up: in [25], likelihood-ratio function was combined with active IR for pupil detection and this combination could handle moderate head motion and significant lighting changes; in [7], by combining appearance-based object recognition and tracking with active IR illumination, the eye tracker could track eyes under variable and realistic lighting conditions and under various face orientations; in [38], an active eye tracker combining particle filtering, expectation maximization with active IR illumination was presented and this method exhibited robustness to light changes and camera defocusing. Active IR pupil detection takes advantage of bright pupil effect, which will be introduced in the following paragraphs in details.

3. Pupil detection and tracking

Our pupil detection algorithm combines active IR illumination and appearance-based method, and the whole process contains three main steps: face detection, pupil detection and pupil tracking. The Fig. 2 summarizes our pupil detection algorithm.

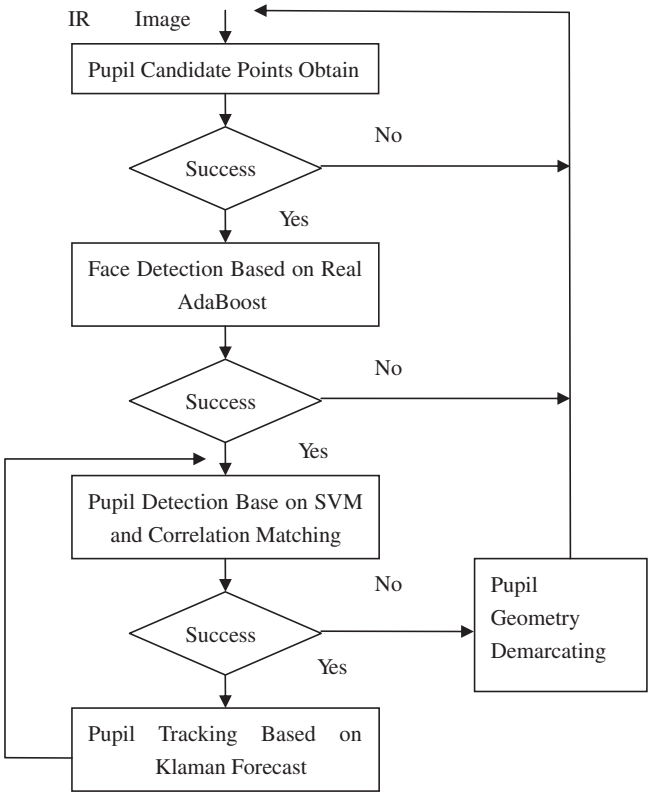


Fig. 2. Pupil detection and tracking algorithm flowchart.

3.1. Pupil candidate points obtain

To facilitate pupil detection, bright pupil effect [39–41] is taken advantage of to pre-find eyes' approximate locations. After incident ray passing thought pupil and being reflected by retina, reflected light beam would take almost the same way as incident ray does. As is shown in Fig. 3, if CCD could capture the reflected light beam, the grayscale of human pupil will much higher than the grayscale of areas around pupil. This phenomenon is called bright pupil effect. There are many factors which have influence on bright pupil effect, such as: light wavelength, distance between light source and CCD, gaze orientation, pupil size and so on. Fig. 4 demonstrates the influence on bright pupil effect by light wavelength. From this figure, we could conclude that bright pupil effect is most obvious under infrared illumination (850 nm wavelength) when other factors are equal. Bright pupil effect not only provides us detection starting reference, but also minimizes the impact of different ambient illumination conditions, ensuring image quality under varying real-world conditions including poor illumination, day and night [7]. At the same time, infrared illumination is invisible, so it will not interfere with users' work.

To utilize bright pupil effect, active infrared illumination is designed as follows: a circle of LEDs are closely around CCD camera, while two lines of LEDs are separated from CCD camera; the driving signals of these two kinds of LEDs are synchronized with odd field scanning signal and even field scanning signal respectively. So odd field image will be bright pupil image while even field image will be dark pupil image, then interlaced image would be obtained. By subtracting odd field image and even field image, owing to the great grayscale difference of pupil in these two field images, the approximate pupil locations could be gained, which we call them pupil candidate points. Illumination diagram, bright pupil image, dark pupil image and subtracted image are shown in Fig. 5.

But only subtracting two field images is far less than needed, because there will be many fake pupil candidate points as a result of edge or movement. So some image pre-processing work should be done to minimize the interference of fake pupil candidate points. Firstly, in general, several pupil candidate points gather together, so the solitary pupil candidate point would be fake and should be removed. In our algorithm, we gather the pupil candidate point numbers of every $2 * 4$-pixel area in the subtracted image. If the candidate point number of one area is less than 3, this area will be ignored; only if the candidate point number of one area is more than 2, one pupil candidate point in this area will be picked up to be genuine pupil candidate point. Secondly, if the pupil candidate points in one area are in the same line or the same row, these pupil candidate points would be edge points or movement candidate points, and this area will also be ignored. Fig. 6 illustrates the course of pupil candidate points obtain.

3.2. Face detection

Although genuine pupil candidate points provide us pupil detection reference, it is very computational expensive to detect pupils directly because there are still many fake pupil candidate points left after image pre-processing work. So face detection is introduced to further eliminate fake candidate points. Considering time complexity, we decide to use real AdaBoost, one appearance-based method, as our face detection algorithm. Though this method demands machine learning to gain strong classifiers, it is very fast to utilize real AdaBoost for face detection because real AdaBoost only concerns statistics query and statistics comparison.

By integration and training, Boosting algorithm could switch weak classification algorithm to strong classification algorithm.

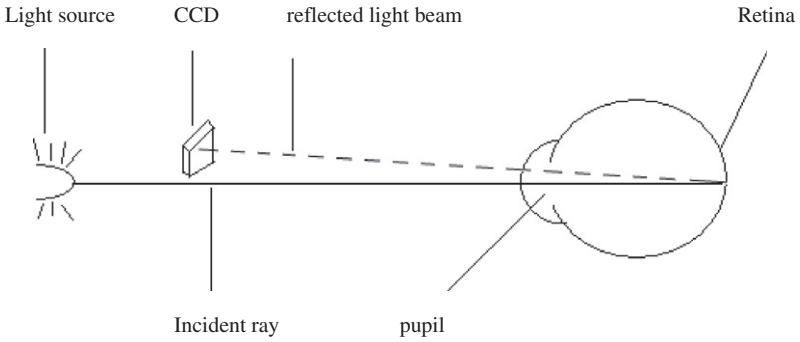


Fig. 3. Theorem diagram of bright pupil effect.

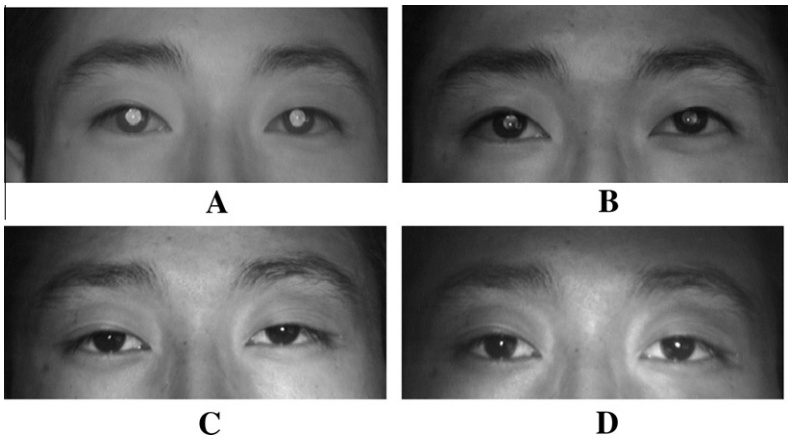


Fig. 4. Bright pupil effects under different illuminations (A) 850 nm infrared illumination, (B) 620–645 nm red light, (C) 515–530 nm green light, (D) 465–475 nm blue light).

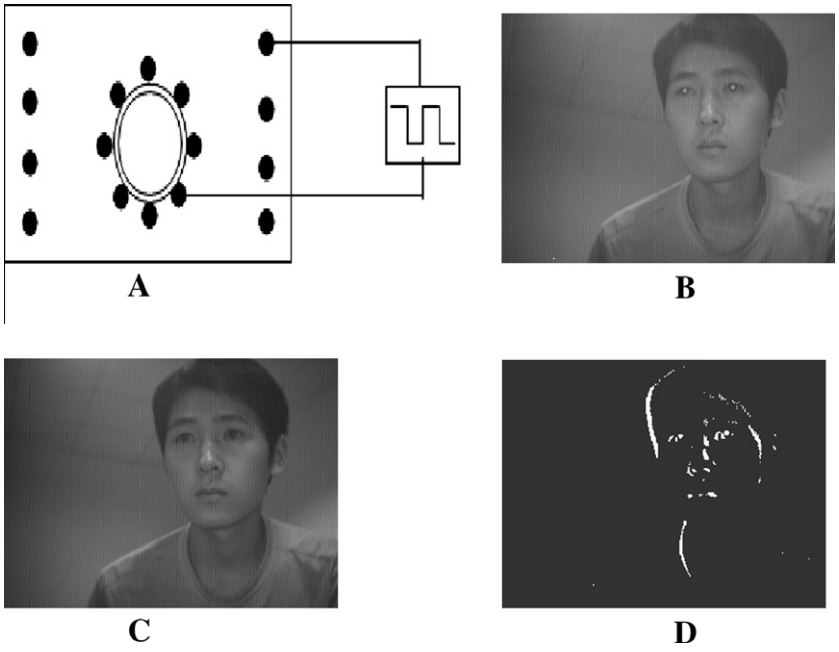


Fig. 5. (A) Active infrared illumination diagram, (B) bright pupil image, (C) dark pupil image, (D) subtracted image.

AdaBoost algorithm [42,43] is one kind of Boosting algorithm, which could be adaptive. AdaBoost algorithm is able to adaptively adjust the weight of training samples, and selects the best weak classifiers, then integrates them to become a strong classifier, in which the different weak classifiers vote respectively. According to whether the algorithm has confidence level, AdaBoost algorithm could be divided into two categories: discrete AdaBoost [44] and real AdaBoost [45]. Real AdaBoost has continuous confidence level, so it could depict classification border more accurately than discrete AdaBoost does. Hence, we chose real AdaBoost as our face

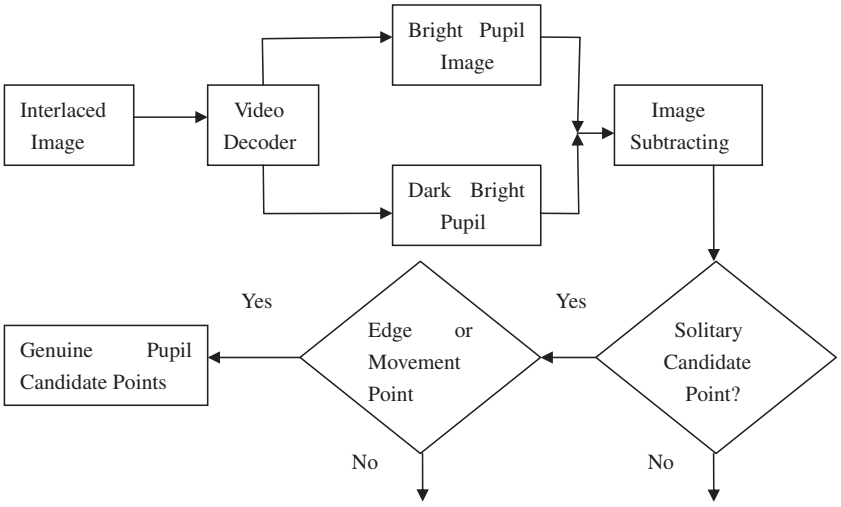


Fig. 6. Pupil candidate points obtain illustration.

detection algorithm. At the same time, we make an improvement to real AdaBoost: the smoothing factor ε will no longer be fixed at a predetermined value, and it will be changed according to the ratio of positive samples' weight and negative samples' weight. This improvement could resist over-learning more effectively.

AdaBoost algorithm is an algorithm that is based on features. Because given limited information, the recognition based on features could code the condition of special areas, and the recognition based on features is much faster than the recognition based on pixels. AdaBoost algorithm uses Haar feature. Haar feature is brought up by Viola and Jones [46], which are a kind of simple rectangle feature and has five basic feature templates, as is shown in Fig. 7. The value of Haar feature is defined as that the sum of grayscale value of the black (white) pixels subtracts the sum of grayscale value of the white (black) pixels.

These five kinds of basic Haar features are used in our face detection phase, and they behave great for front faces and leaning faces less than 30°. Nevertheless, large degree face leaning will cause failure to our face detection. So some kinds of Haar features have been taking into our consideration in order that the large leaning faces could also be detected in our face detection phase. Some of these Haar features are shown in Fig. 8, and the classifiers with these Haar features are now during the process of machine learning.

The Haar features could be placed in any size and on any position of image, so just one sub-window of an image will have a lot of Haar features. For example, a $24 * 24$ pixel image has more than 160,000 Haar features. Hence, if we calculate their feature values directly, it will cost considerable time. To solve this problem, integral image is used in calculating feature values. Value of every pixel in integral image is the sum of grayscales of the pixel's top-left pixels. For example, the value of pixel A(x, y) in integral image is:

Fig. 8. Some of improved Haar features.

$$I(x,y) = \sum_{\substack{x' \leqslant x \\ y' \leqslant y}} i(x',y')$$

$i(x', y')$ in this formula is the grayscale of pixel (x', y').

With integral image, we could calculate Haar feature values much faster. As is shown in Fig. 9, the sum of grayscales of area 1 could be obtained only with the values of pixels A, B, C, D.

$$I_{area1} = I_A + I_D - I_B - I_C$$

The calculating of rectangle Haar feature values, with integral image, is only related to the values of four rectangle's endpoint pixels and the calculating is only concerning adding and subtracting, Therefore, integral image improves the speed of face recognition largely.

Because the distances between CCD camera and different people are not the same, the faces in one image are in different sizes. In order that all faces could be detected, we prefix 8 face sizes and we firstly detect faces in the largest size, and then we will go down to detect faces in the smaller sizes until the smallest size. During this course, if a face is detected, the face area will be marked to avoid that this area will be detected for faces in the smaller sizes, which is an effective method to lower the time complexity of face

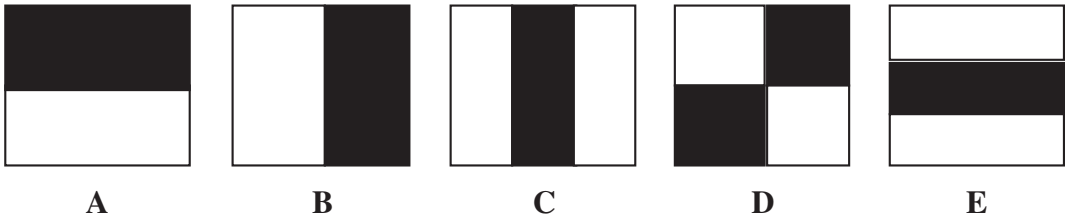


Fig. 7. Basic Haar features.

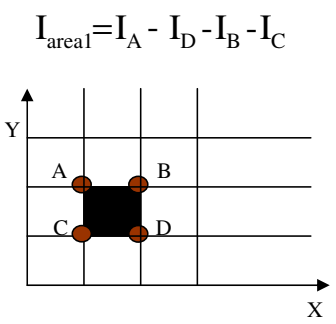


Fig. 9. Integral image.

detection algorithm. Fig. 10 illustrates the detection order in face sizes.

In the process of face detection, after an area is classified as a face, a face optimization method is applied, other than marking this area immediately. The optimization method is applied as follows: if an area is classified as a face, several other areas slightly deviated from this area will be detected with real AdaBoost too; then the AdaBoost output values of all areas are compared with each other; at last, the area with the biggest output value is deemed as face, and this area will be marked.

3.3. Pupil detection

Up to now, faces have been detected and each of them has a genuine pupil candidate point. Then candidate eye areas will be selected according to locations of these genuine pupil candidate points in faces and geometry relation between face and eye and these candidate eye areas will be detected by real support vector machine (SVM) and correlation matching. Nevertheless, SVM and correlation matching are time-consuming, so real AdaBoost is introduced once again to diminish fake candidate eye areas and leave just several candidate eye areas for SVM and correlation matching detection. The real AdaBoost classifier here is much more simple than the real AdaBoost classifier for face detection and has much less weak classifiers. Fig. 11 demonstrates this flow path.

SVM [47,48] is an excellent two-class classification algorithm which, by transforming no-linear classification issue in low dimensional space to linear classification issue in high dimensional space, finds optimal decision hyper-plane [47] to finish the classification task. The foundation of SVM is structural risk minimization. And kernel function is introduced to combining transformation from low dimensional space to high dimensional space and depicting of optimal decision hyper-plane, reducing the dimension of classification space and lowering algorithm time complexity. Polynomial

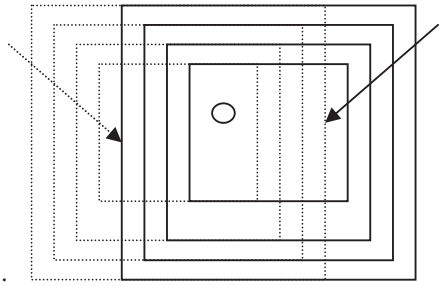


Fig. 10. Detection order in face sizes (○ represents a genuine pupil candidate point; the black frames represent faces in which the genuine pupil candidate point is right eye's pupil; the dotted frames represent faces in which the genuine pupil candidate point is left eye's pupil).

kernel [49], radial basis function (RBF) kernel [50] and Gaussian kernel [51] are three common kernel functions. Because RBF kernel is used most extensively and it has highest maturity, it is used in our eye detection method.

Features selection and support vectors selection are two significant items for SVM classification. On one hand, Haar features are still used in our eye detection algorithm since they could resist salt and pepper noise greatly. The features selection could be formulated as follows. Firstly, value range of a feature i is divided equally into M segments; secondly, $F(i)$ of this feature is calculated with following formula:

$$F(i) = \sum_{m=1}^{M} P_1^i(m) w_1^i(m) P_{-1}^i(m) w_{-1}^i(m)$$

where $P_1^i(m)$ is the probability of that positive samples' ith feature value is in segment m; $P_{-1}^i(m)$ donates the probability of that negative samples' ith feature value is in segment m; $w_1^i(m)$ represents the sum of positive samples' weights; and $w_{-1}^i(m)$ represents the sum of negative samples' weights. The weights of samples are adjusted adaptively, making SVM classifier more robust. Some Haar features selected for SVM detection are shown in Fig. 12

On the other hand, the support vectors selection could be summarized as follows. For linearly separable case, the construction of optimal decision hyper-plane could be converted to the following optimal issue:

$$\min \phi(w) = \|w\|^2,$$

and constraint condition is:

$$y_i(w * x_i + b) \geqslant 1, \quad i = 1, 2, \ldots, l$$

where w is whole vector; $\varphi(w)$ is vector function; x_i represents the ith training sample; y_i donates the class, which has only two values -1 and $+1$; and b is a constant. Then with lagrange optimization method this optimal issue could be converted to its pairing issue:

$$\max W(\alpha) = \sum_{i=1}^{l} \alpha_i \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

and constraint condition is:

$$\sum_{i=1}^{l} \alpha_i y_i = 0, \quad \alpha_i \geqslant 0,\ i = 1, 2, \ldots, l,$$

where α is lagrange multiplier. The samples corresponding to the nonzero roots of this pairing issue are support vectors. So the classification function corresponding with optimal decision hyper-plane is as follows:

$$f(x) = sgn\{(w \cdot x_i) + b\} = sgn\{\sum_i \alpha_i y_i (x_i \cdot x + b)\}$$

where x_i represents the ith support vector, and x is the sample which will be classified. For not linearly separable case, the construction of optimal decision hyper-plane could be converted to the following optimal issue:

$$\phi(w, \varepsilon) = \frac{1}{2}||w||^2 + c\left[\sum_{i=1}^{l} \varepsilon_i\right]$$

and constraint condition is:

$$y_i(w \cdot x_i + b) + \varepsilon_i \geqslant 1, i = 1, 2, \ldots, l,$$

$$f(x) = sgn\left(\sum_i \alpha_i y_i K(x_i \cdot x) + b\right)$$

where ε_i, is a nonnegative relaxation factor. With kernel function, the classification function corresponding with optimal decision hy-

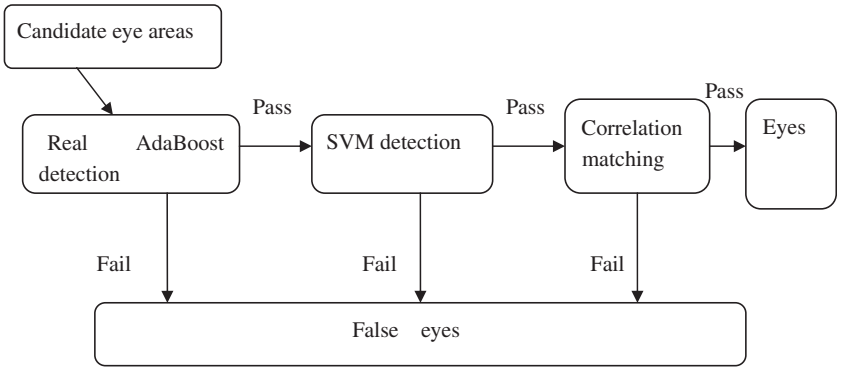


Fig. 11. Eye detection flowchart.

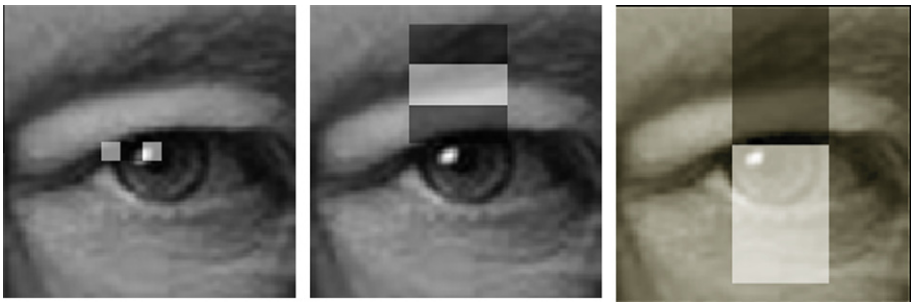


Fig. 12. Some of Haar features for SVM detection.

per-plane in a higher dimensional space could be obtained as follows:where $K(x_i, x_j)$ is the inner product of $\varphi(x_i)$ and $\varphi(x)$.

Eye "ready-for-detect" positions are composed of positions in two classes: eye positions anticipated by Kalman tracking algorithm based on the eye positions detected in the former image and eye positions in the faces newly detected. The Kalman tracking algorithm will be described in detail in the next part. Because there possibly is some deviation about the eye positions anticipated by Kalman tracking algorithm or some noise about the eye positions in the faces newly detected, not only the eye "ready-for-detect" positions are detected, but also the areas around the eye "ready-for-detect" positions. The spiral detection order is shown in Fig. 13: area zero is the eye "ready-for-detect" positions, which is detected firstly, and then the rest areas will be detected in numerical order.

Both eyes in the same face are detected with real SVM in the spiral detection order, and firstly one eye position which has passed the real SVM detection with the highest confidence level is selected for each eye in the same face. Secondly the number of eye positions which have passed the real SVM detection with confidence levels higher than 80 percents of the highest confidence level is obtained for each eye in the same face. If the number is more than 2, the eye positions corresponding with the 3 highest confidence levels will be picked up for each eye in the same face; if the number is less than 3, the eye positions with confidence levels higher than 80 percents of the highest confidence level will be picked up for each eye in the same face. Thirdly, each eye position picked up for one eye is matched with each eye position picked up for the other eye in the same face, which is called correlation matching. And then the pair of eye positions with the highest correlation coefficient will be deemed as a pair of eyes in the same face. At last, the pupil candidate points corresponding with this pair of eye positions are the pupils detected in the same face. To avoid illumination variation and movement inaccuracy, only the bright pupil field image is used in pupil detection phase.

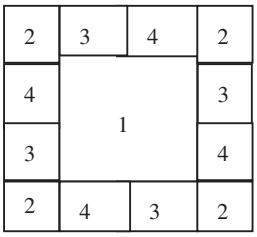


Fig. 13. Spiral eye detection order.

3.4. Pupil tracking

It is computational expensive to detect pupils, traveling one whole image, in the "candidate points-faces-pupils" detection order for every image. So after obtaining the pupils in current image, the possible pupil positions in the subsequent image are anticipated with Kalman algorithm [52,53], and the areas around these positions anticipated are detected with SVM algorithm preferentially in the subsequent image.

Simulating the motion situation of the "ready-for-detect" target in front of CCD camera, we suppose the motion of the target both in x and y axles are even-speed straight motion which is bothered by a random acceleration α. α is a random variable, $a(t) \sim N(0, \sigma_\omega^2)$ .And the motion state vector of the "ready-for-detect" target is supposed as follows:

$$X(k) = [X_{MO}(k), Y_{MO}(k), Vx(k), Vy(k)]^T.$$

where $X_{MO}(k)$ and $Y_{MO}(k)$ are the abscissa and ordinate of the "ready-to-detect" target; $Vx(k)$ and $Vy(k)$ are the speeds of the "ready-to-detect" target in x and y axles. The measure matrix is $Y(k)$:

$$Y(k) = [X_{ME}(k), Y_{ME}(k)]^T.$$

where $X_{ME}(k)$ and $Y_{ME}(k)$ are the measure abscissa and measure ordinate of the "ready-to-detect" target. So Kalman anticipation algorithm includes two models:Motion state vector model:

$$X(k+1) = A(k)X(k) + W(k),$$

where $A(k)$ is state transition matrix and $W(k)$ donates system perturbation.Measure vector model:

$$Y(k) = C(k)X(k) + M(k),$$

where $C(k)$ is the state transition matrix from current to current measurement and $M(k)$ represents measurement uncertainty. Because the motion of the target is supposed to be even-speed straight motion and $Y(k)$ only involves position, these two models could be also described with the following two matrixes:

$$\begin{bmatrix} X_{MO}(k+1) \\ Y_{MO}(k+1) \\ Vx(k+1) \\ Vy(k+1) \end{bmatrix} = \begin{bmatrix} 1,0,t,0 \\ 0,1,0,t \\ 0,0,1,0 \\ 0,0,0,1 \end{bmatrix} \begin{bmatrix} X_{MO}(k) \\ Y_{MO}(k) \\ Vx(k) \\ Vy(k) \end{bmatrix} + W(k)$$

and

$$\begin{bmatrix} X_{ME}(k) \\ Y_{ME}(k) \end{bmatrix} = \begin{bmatrix} 1,0,0,0 \\ 0,1,0,0 \end{bmatrix} \begin{bmatrix} X_{MO}(k) \\ Y_{MO}(k) \\ Vx(k) \\ Vy(k) \end{bmatrix} + M(k).$$

where t is the time interval between adjacent images; the state transition process noise covariance σ_ω^2 equals 1; $M(k)$ is normally distributed as $p(M) \sim N(0, R)$, and R represents measurement noise covariance.

For one image, the possible pupil positions anticipated with Kalman algorithm and the areas around them are detected firstly with SVM algorithm and correlation matching directly. If none pupil is detected, the whole image will be detected in the "candidate points-faces-pupils" detection order for pupils; if some pupils are detected, the face areas according to these pupils will be marked so that these areas will not be detected in the subsequent "candidate points-faces-pupils" cascade detection course. The program pseudo-code of this strategy is shown in Fig. 14.This strategy could save processing time effectively and the effectiveness of this strategy will be much higher when there are many users.

4. Experiments

4.1. Classifiers training

For face detection, a cascade AdaBoost detection system which has eight strong classifiers is obtained with machine learning. Because in realistic video captured by CCD camera, non-faces usually take much more space than faces do in the same image, we decide that the ratio of face sample number and non-face sample number is approximately 1:8. The number of face samples in our training database is 5009, and the number of non-face samples is 41,700. Besides, to solve the problem that the non-face samples, which have abundant texture, are much easier to be classified wrongly than the usual non-face samples, we make sure the non-face samples which have abundant texture are more than half of total non-face samples when we try our best to make non-face samples discrete. Some samples used in our face classifiers' training database are shown in Fig. 15. The face detection accuracy we achieve is 98.2721% and Fig. 16 illustrates the detection effect of our face detection system.

For eye detection, firstly a cascade AdaBoost pre-detection system which has just three strong classifiers is obtained. In the AdaBoost eye training set, there are 4764 eye samples and 30,000 non-eye samples. Secondly, an SVM eye detection system which involves 30 Haar features is obtained. In the SVM eye training set, there are 4764 eye samples and 11,926 non-eye samples. And there are 1687 support vectors in the SVM classifier obtained. Because only eye area contains too little information, the eye-brow area, which contains more information than only eye area does, is detected as a whole. As a result, the non-eye samples are pictures which contain both eyes and brows. Some samples used in our eye classifiers' training database are shown in Fig. 17. And the eye detection accuracy we achieve is 96.1033%.

To test our pupil detection and tracking system, a series of experiments are conducted, including pupil detection for just single people, pupil detection for multiple people, pupil detection with glasses, pupil detection with face rotation, pupil detection with occlusion, pupil detection under various illumination and pupil detection under large-scale head motion. The results of these experiments show that our pupil detection and tracking system could handle the above-mentioned situation and have a great performance in pupil detection. The experimental results and related discussion are presented respectively as follows.

```
While (image frame input)
{
  if (pupil positions anticipated  ≠0)
   { for i=1 to number of pupil positions anticipated
       {if (the position is true pupil)
           Mark related face areas
       }
     detect the unmarked areas of the image in "candidate points-faces-pupils" detection
order.
   }
  else
    detect the whole image in "candidate points-faces-pupils" detection order.
    ……
    ……
}
```

Fig. 14. Program pseudo-code of time-saving strategy with Kalman pupil tracking.

A Some face samples in training database

B Some non-face samples in training database

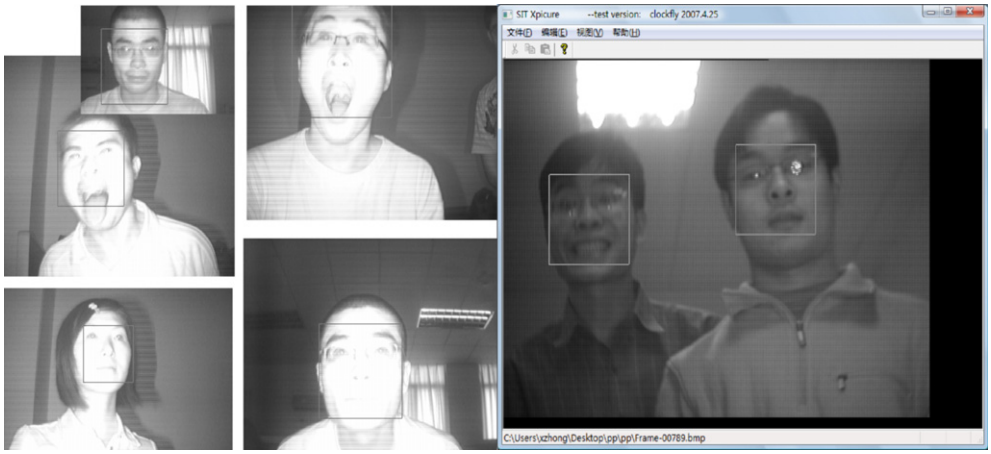
Fig. 15. Some samples used in our face classifier' training database.

4.2. Pupil detection for just a single people

The pictures in Fig. 18 are captured in three different distances between CCD camera and the man, and the distances are respectively 500 mm, 1000 mm and 2000 mm. The results of this experiment shows that despite different distances between CCD camera and the target people, our pupil detection and tracking system can manage to depict pupils correctly. In Fig. 18, the big blue rectangles demonstrate faces detected and the small blue rectangles illustrate the pupils detected.

4.3. Pupil detection for multiple people

Our pupil detection and tracking system not only can detect pupils for just a single people, but also can detect pupils for multiple people simultaneously. Fig. 19 shows some detection results of our



A Single face detection B Multiple faces detection

Fig. 16. Effect illustration of our face detection system.

pupil detection and tracking system for multiple people. The first three pictures demonstrate the pupil detection results for two people simultaneously; the last three pictures illustrate the pupil detection results for three people simultaneously. The people in these pictures are in different distances from CCD camera, and they are with different facial expression. From these detection results we could see that our pupil detection and tracking system can detect pupils for multiple people correctly and simultaneously.

4.4. Pupil detection with glasses

Wearing glasses is a great challenge for pupil detection and tracking because not only glasses lower the intensity of bright pupil effect, but also the light reflection of glasses imports noise to pupil detection and tracking. The pupil detection results of our system for people with glasses are shown in Fig. 20, and these detection results behave the robustness of our pupil detection and tracking system for people wearing glasses. However, if the light reflection covers the whole pupil area, our system will fail. So how to handle this obstacle is part of our feature work.

4.5. Pupil detection with face rotation

Face rotation is another big challenge for pupil detection and tracking and there are two kinds of face rotation: face rotation in the same plane and face rotation out of a plane. Fig. 21 shows the detection results of our system for people with face rotation. The face rotation in our experiment includes all rotating situation: rotating upside, rotating downside, rotating left in the same plane, rotating right in the same plane, rotating left out of a plane, rotating right out of a plane. The results prove that our pupil detection and tracking system are robust for face rotation. However, it will cause our system to fail when faces rotate substantially, so that how to detect pupils when faces rotate sharply is also part of our future work.

4.6. Pupil detection with occlusion

Occlusion handling is a significant issue for object detection and pattern recognition. The detection results of our system for people with occlusion are shown in Fig. 22. Although faces are occluded partially even mostly, our system can still detect pupils correctly. Considering our pupil detection method is based on face detection, we could figure out that most features in our face detection come from brow-eye area. The small blue rectangles in Fig. 22 illustrate the pupils detected.

4.7. Pupil detection under various illumination

A robust pupil detection and tracking system should resist the influence of illumination variation and perform great in any illumination situation. To test our system, an experiment is conducted as follows: firstly pupils are detected with ambient lights on; secondly pupils are detected with ambient lights off; thirdly pupils are detected with a surrounding mobile light source. Illumination variation could make the grayscale of faces change, but from the pupil detection results in Fig. 23 we could see that our system can cope with the change successfully.

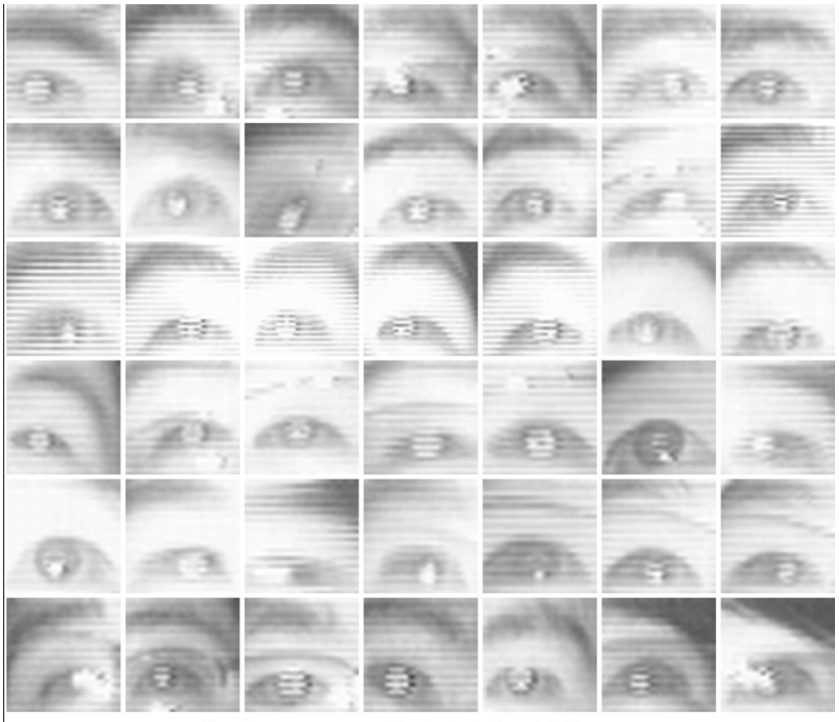
4.8. Pupil detection under large-scale head motion

A robust pupil detection and tracking system should also perform well when target people move fast and substantially. Fig. 24 demonstrates the performance of our system under large-scale head motion. The motion of target face includes moving forward, moving backward, moving upward, moving downward, moving left and moving right, and the motion is fast and large-scale. The pictures in Fig. 24 are picked up every 1 or 1.5 s from the same video. And Fig. 24 illustrates the robustness of our pupil detection and tracking system for large-scale head motion.

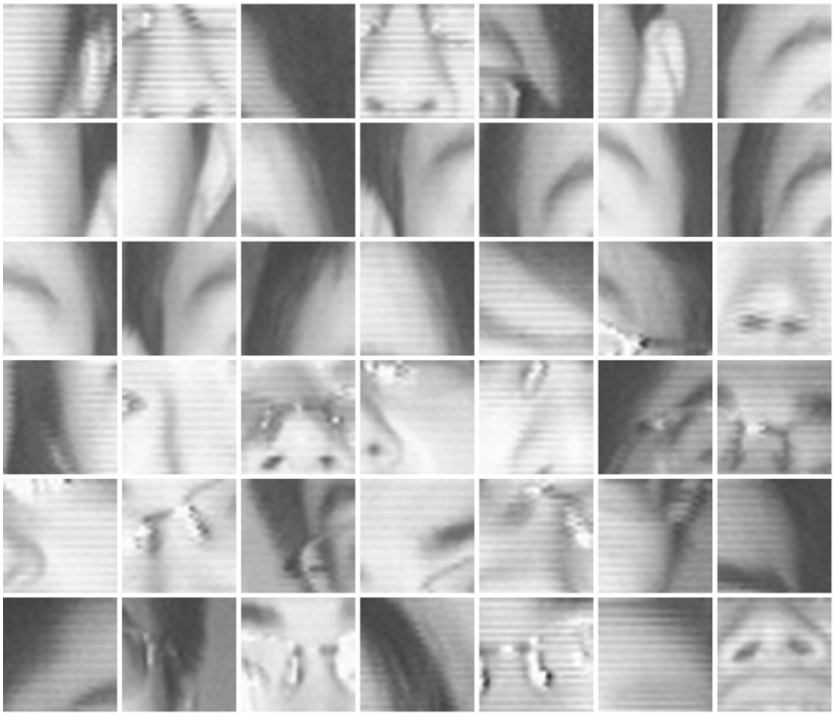
Our pupil detection and tracking system does not ask for advanced hardware. Both classifiers training and above-mentioned test experiments can be conducted on the circumstance of Windows XP, Pentium IV, 512Memory, 2.4GHZ. The resolution of test video is $640 * 480$-pixel, and the frame frequency of test video is 30 fps, satisfying the demand of real-time detection.

5. Conclusions and future goals

In this paper, a novel approach to Robust real-time multi-user pupil detection and tracking is brought up, which combines active IR illumination, real AdaBoost, real SVM, correlation matching and Kalman forecast. At the same time, a series of experiments is conducted to test this new method, and the experimental situation includes multiple people in different distances from CCD, wearing glasses, face rotation, occlusion, illumination variation and large-scale head motion. The lessons we learned from the whole research are: (1) active IR illumination not only can produce bright pupil ef-



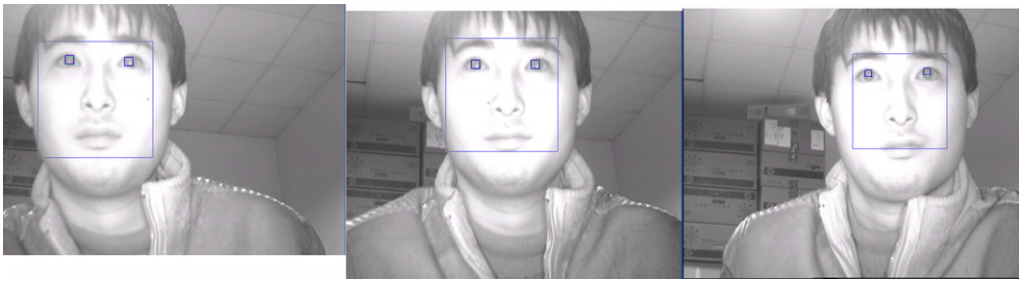
A Some eye samples in training database



B Some non-eye samples in training database

Fig. 17. Some samples used in our eye classifiers' training database.

fect to provide pupil detection reference, but also can help resist the influence of ambient light on pupil detection; (2) face-pupil detection order can eliminate fake pupil candidate points, saving much time compared with detecting pupils directly; (3) the combination of SVM and correlation matching utilizes the great classification function of SVM as well as the symmetrical relation of two eyes in the same face, performing well in pupil detection; (4) with Kalman forecast and face area marking, the time complexity of pupil detection can be decreased effectively. The experimental results show that this new method behaves great under various illumination or lager-scale head motion, and this new method can handle occlusion, face rotation, glasses interference and pupil detection for multiple people successfully. However, to improve our pupil detection and tracking system, there are still three goals we should



A The distance between CCD camera and the man is 500mm.

B The distance between CCD camera and the man is 1000mm.

C The distance between CCD camera and the man is 2000mm.

Fig. 18. Experimental results of our pupil detection and tracking system for just a single people.

Fig. 19. Experimental results of our pupil detection and tracking system for multiple people (the big blue rectangles demonstrate faces detected and the small blue rectangles illustrate the pupils detected). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 20. Experimental results of our pupil detection and tracking system for people with glasses (the big blue rectangles demonstrate faces detected and the small blue rectangles illustrate the pupils detected). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 21. Experimental results of our pupil detection and tracking system for people with face rotation (the big blue rectangles demonstrate faces detected and the small blue rectangles illustrate the pupils detected). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

reach in our future work: (1) our system should detect pupils correctly when faces rotate substantially; (2) our method should be more robust when there is sharp glasses interference; (3) to extend the applicable scope of our system, the novel method should be utilized just with embedded apparatus.

Acknowledgment

This research activity has been partially funded by Nan Jing University Postgraduate Research Innovation Fund, Chinese National

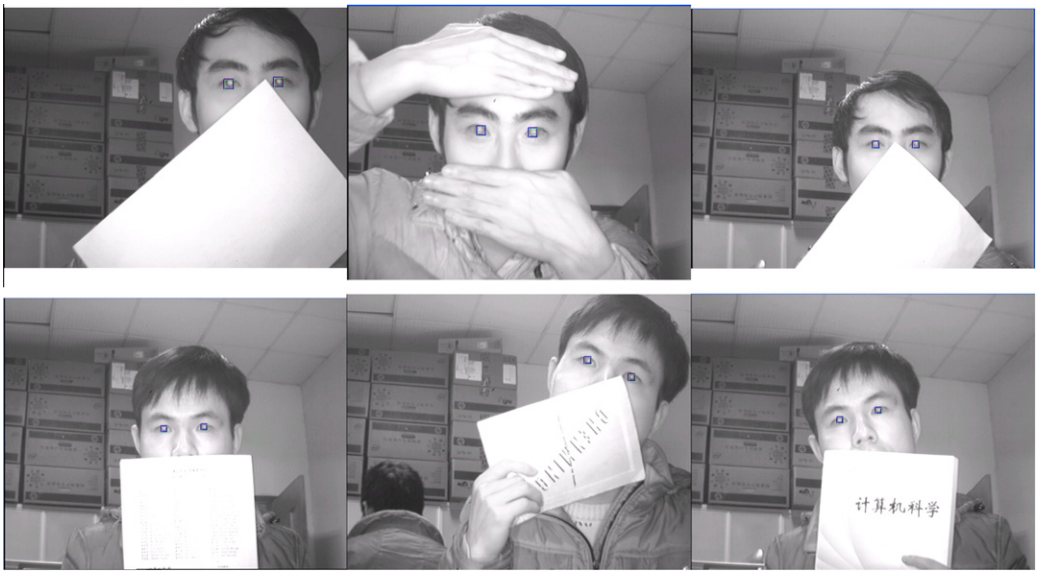


Fig. 22. Experimental results of our pupil detection and tracking system for people with occlusion.

Fig. 23. Experimental results of our pupil detection and tracking system for people under various illumination (the big blue rectangles demonstrate faces detected and the small blue rectangles illustrate the pupils detected). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Natural Science Fund Commission and Chinese Ministry of Education.

Appendix A. Specific AdaBoost training method (Real AdaBoost)

(1) Given the set of training samples $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, weak classifier space H. In the set, $x \in X$ is the sample data; $y = \pm 1$ is class label; n is the number of samples. The initialized sample weight is $D_t(i) = 1/n$, $i = 1, 2, \ldots, n$.
(2) For $t = 1, 2, \ldots, T$ (T is the number of features which are aimed to get.):
① Apply the following steps to every weak classifier in H:
(1) Divide the sample space X to $x_1, x_2, x_3, \ldots, x_n$;
(2) With the weight of training samples D_t, calculating:

$$W_k^i = P(x_i \in X_j, y_i = k) = \sum_{\substack{x_i \in X \\ y_i = k}} D_t(i), k = \pm 1$$

(3) Under the division, set the output of weak classifier as:

$$\forall x \in X_j, h(x) = \frac{1}{2} \ln\left(\frac{W_{+1}^j + \varepsilon}{W_{-1}^j + \varepsilon}\right), j = 1, 2, \ldots, m.$$

e is a tiny positive number.
(4) Calculating the initialization factor:

$$Z = 2 \sum_j \sqrt{W_{+1}^j W_{-1}^j}$$

② Select h_t in weak classifier space to minimize Z:

$$Z_t = \min_{h \in H} Z,$$

$$h_t = \operatorname*{argmin}_{h \in H} Z,$$

③ Update the weight of training samples:

Fig. 24. Experimental results of our pupil detection and tracking system for people underlarge-scale head motion (the big blue rectangles demonstrate faces detected and the small blue rectangles illustrate the pupils detected). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$D_{t+1}(i) = D_t(i)\frac{\exp[-y_i h_t(x_i)]}{Z_t\%}$$

Z_t is the initialization factor, to make D_{t+1} a probability distribution.

(3) The final strong classifier is:

$$H(x) = sign[\sum_{t=1}^{T} h_t(x) - b]$$

b is threshold which is set manually, usually 0. Similarly, we define the confidence rate of H is:

$$conf_H(x) = |\sum_t h_t(x) - b|$$

Appendix B. Specific SVM training method (RBF Kernel)

(1) Given the set of training samples $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. In the set, $x \in X$ is sample vector; $y = \pm 1$ is class label; n is the number of samples.

(2) Normalize every attribute in the sample vectors to generate standard sample set X_s.

(3) According to lattice search, Assuming there are M lattices, For $t = 1, \ldots, M$:

① Divide X_s into $X_{s1}, X_{s2}, \ldots X_{sk}$, For $e = 1, \ldots, K$:

1. Training with libSVM [54], the input data is Ct, γ_t and Xs except X_{se}, and the output data is support vector set X_{sve}, coefficient set α_e and b_e.
2. Obtain error rate err_e with sample subset X_{se}.

② Obtain the average error rate of $err_1, \ldots, err_k$.

(4) Select the C and γ corresponding to the smallest average error rate. Training with libSVM, the input data is C, γ and the whole standard sample set Xs, and the output data is support vector set X_{sv}, coefficient set α and b. The final SVM classifier could be depicted as follows:

$$f(x) = sgn\left(\sum_i \alpha_i y_i K(x_i \cdot x) + b\right)$$

.

References

[1] C. Chiang, W. Tai, M. Yang, Y. Huang, C. Huang, A novel method for detecting lips, eyes and faces in real time, Real-Time Imaging 9 (2003) 277–287.
[2] M. Khosravi, R. Safabakhsh, Human eye sclera detection and tracking using a modified time-adaptive self-organizing map, Pattern Recognition 41 (2008) 2571–2593.
[3] B. Kroon, S. Maas, S. Boughorbel, A. Hanjalic, Eye localization in low and standard definition content with application to face matching, Computer Vision and Image Understanding 113 (2009) 921–933.
[4] Q. Ji, X. Yang, Real-time eye, gaze, and face pose tracking for monitoring driver vigilance, Real-Time Imaging 8 (2002) 357–377.
[5] S. Kawato, N. Tetsutani, Detection and tracking of eyes for gaze-camera control, Image and Vision Computing 22 (2004) 1031–1038.
[6] A. Santis, D. Iacoviello, Robust real time eye tracking for computer interface for disabled people, Computer Methods and Programs in Biomedicine 96 (2009) 1–11.

[7] Z. Zhu, Q. Ji, Robust real-time eye detection and tracking under variable lighting conditions and various face orientations, Computer Vision and Image Understanding 98 (2005) 124–154.
[8] D. Yoo, M. Chung, A novel non-intrusive eye gaze estimation using cross-ratio under large head motion, Computer Vision and Image Understanding 98 (2005) 25–51.
[9] T.D. Orazio, M. Leo, C. Guaragnella, A. Distante, A visual approach for driver inattention detection, Pattern Recognition 40 (2007) 2341–2355.
[10] Tobias K. Kohoutek, Multi-user vision interface based on range imaging, in: Proc. 18th Visual Information Processing Conference, 2009.
[11] S. Phil, S. Ian, H. Klaus, B. Edward, L. Kai, B. Richard, J. Wijnand, Multi-user 3D display, in: Proceedings of Asia Display, 2007, pp. 472–477.
[12] V. Blanz, T. Vetter, Face recognition based on fitting a 3D morphable model, IEEE Trans. PAMI 25 (2003) 1063–1075.
[13] Y. Li, X. Qi, Y. Wang, Eye detection by using fuzzy template matching and feature-parameter-based judgement, Pattern Recognition Letters 22 (2001) 1111–1124.
[14] N. Edenborough, R.I. Hammound, A. Harbach, et al. Driver state monitor from Delphi, in: IEEE Computer Vision and Pattern Recognition Conference, vol. II, 2005, pp. 1206–1207.
[15] J. Wang, L. Yin, J. Moore, Using geometric properties of topographic manifold to detect and track eyes for human–computer interaction, ACM Transactions on Multimedia Computing, Communications and Applications 3 (4) (2007).
[16] B. Noureddin, P.D. Lawrence, C.F. Man, A non-contact device for tracking gaze in a human computer interface, Computer Vision and Image Understanding 98 (2005) 52–82.
[17] S. Zhai, What, s in the eyes for attentive input, Communications of the ACM 43 (3) (2003) 34–39.
[18] H. Lu, G. Fang, C. Wang, Y. Chen, A novel method for gaze tracking by local pattern model and support vector regressor, Signal Processing 90 (2010) 1290–1299.
[19] Carlos H. Morimoto, Marcio R.M. Mimica, Eye gaze tracking techniques for interactive applications, Computer Vision and Image Understanding 98 (2005) 4–24.
[20] D. Torricelli, S. Conforto, M. Schmid, T.D. Alessio, A neural-based remote eye gaze tracker under natural head motion, Computer Methods and Programs in Biomedicine 92 (2008) 66–78.
[21] Y. Li, S. Wang, X. Ding, Eye/eyes tracking based on a unified deformable template and particle filtering, Pattern Recognition Letters 31 (2010) 1377–1387.
[22] D. Torricelli, M. Goffredo, S. Conforto, M. Schmid, An adaptive blink detector to initialize and update a view-based remote eye gaze tracking system in a natural scenario, Pattern Recognition Letters 30 (2009) 1144–1150.
[23] Chern-Sheng Lin, An eye behavior measuring device for VR system, Optics and Lasers in Engineering 38 (2002) 333–359.
[24] A. Amir, L. Zimet, A.S. Vincentelli, S. Kao, An embedded system for an eye-detection senor, Computer Vision and Image Understanding 98 (2005) 104–123.
[25] D.W. Hansen, R.I. Hammoud, An improved likelihood model for eye tracking, Computer Vision and Image Understanding 106 (2007) 220–230.
[26] Z. Zhou, X. Geng, Projection functions for eye detection, Pattern Recognition 37 (2004) 1049–1056.
[27] M. Dobes, J. Martinek, D. Skoupil, Z. Dobesova, J. Pospisil, Human eye localization using the modified Hough transform, Optik 117 (2006) 468–473.
[28] D. Torricelli, M. Goffredo, S. Conforto, M. Schmid, An adaptive blink detector to initialize and update a view-based remote eye gaze tracking system in a natural scenario, Pattern Recognition Letters 30 (2009) 1144–1150.
[29] J. Wu, M.M. Trivedi, Simultaneous eye tracking and blink detection with interactive particle filters, EURASIP Journal on Advances in Signal Processing (2008).
[30] G.C. Feng, P.C. Yuan, Multi-cues eye detection on gray intensity image, Pattern Recognition 34 (2001) 1033–1046.
[31] P. Wang, Q. Ji, Multi-view face and eye detection using discriminant features, Computer Vision and Image Understanding 105 (2007) 99–111.
[32] F. Smeraldi, O. Carmona, J. Bigun, Saccadic search with Gabor features applied to eye detection and real-time head tracking, Image and Vision Computing 18 (2000) 323–329.
[33] D. Zhu, S.T. Moore, T. Raphan, Robust and real-time torsional eye position calculation using a template-matching technique, Computer Methods and Programs in Biomedicine 74 (2004) 201–209.
[34] M.J. Coughlin, T.R.H. Cutmore, T.J. Hine, Automated eye tracking system calibration using artificial neural networks, Computer Methods and Programs in Biomedicine 76 (2004) 207–220.
[35] W.O. Lee, E.C. Lee, K.R. Park, Blink detection robust to various facial poses, Journal of Neuroscience Methods 193 (2010) 356–372.
[36] Z. Qian, D. Xu, Automatic eye detection using intensity filtering and K-means clustering, Pattern Recognition Letters 31 (2010) 1633–1640.
[37] C.H. Morimoto, D. Koons, A. Amir, M. Flickner, Pupil detection and tracking using multiple light sources, Image and Vision Computing 18 (2000) 331–335.
[38] D.W. Hansen, A.E.C. Pece, Eye tracking in the wild, Computer Vision and Image Understanding 98 (2005) 155–181.
[39] H. Gu, Y. Zhang, Q. Ji, Task oriented facial behavior recognition, Computer Vision and Image Understanding 100 (2005) 385–415.
[40] W. Liao, W. Zhang, Z. Zhu, Q. Ji, W.D. Gray, Toward a decision-theoretic framework for affect recognition and user assistance, International Journal of Human–Computer Studies 64 (2006) 847–873.
[41] Q. Ji, R. Hu, 3D Face pose estimation and tracking from a monocular camera, Image and Vision Computing (2002).
[42] C. Gao, N. Sang, Q.L. Tang, On selection and combination of weak learners in AdaBoost, Pattern Recognition Letters 31 (2010) 991–1001.
[43] M. Yang, J. Crenshaw, B. Augustine, R. Mareachen, Y. Wu, AdaBoost-based face detection for embedded systems, Computer Vision and Image Understanding 114 (2010) 1116–1125.
[44] R. Nock, F. Nielsen, A real generalization of discrete AdaBoost, Artificial Intelligence 171 (2007) 25–41.
[45] R.E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated prediction, Machine Learning 37 (3) (1999) 297–336.
[46] P. Viola, M. Jones, Rapid object recognition using a boosted cascade of simple features, in: Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Kauai Hawaii, USA, 2001, pp. 905–910.
[47] X. Zhou, W. Jiang, Y. Tian, Y. Shi, Kernel subclass convex hull sample selection method for SVM on face recognition, Neurocomputing 73 (2010) 2234–2246.
[48] J. Jia, L. Cai, P. Lu, X. Liu, Fingerprint matching based on weighting method and the SVM, Neurocomputing 70 (2007) 849–858.
[49] C. Ying, B. Joseph, Identification of linear systems using polynomial kernels in the frequency domain, Journal of Process Control 12 (2002) 645–657.
[50] G.M. Foody, A. Mathur, Toward intelligent training of supervised image classifications: directing training data acquisition for SVM classification, Remote Sensing of Environment 93 (2004) 107–117.
[51] T. Wang, H. Huang, S. Tian, J. Xu, Feature selection for SVM via optimization of kernel polarization with Gaussian ARD kernels, Expert Systems with Applications 37 (2010) 6663–6668.
[52] S. Sadhu, S. Mondal, M. Srinivasan, T.K. Ghoshal, Sigma point Kalman filter for bearing only tracking, Signal Processing 86 (2006) 3769–3777.
[53] Y. Ojima, M. Kawahara, Estimation of river current using reduced Kalman filter finite element method, Computer Methods in Applied Mechanics and Engineering 198 (2009) 904–911.
[54] C.C. Chang, C.J. Lin, LIBSVM: A Library For Support Vector Machines[EB/OL], 2001. <http://www.csie.ntu.edu.tw/cjlin/libsvm>.