# Eye Location Based on Adaboost and Random Forests

Xiangde Zhang
School of Science, Northeastern University, Shenyang, China
Email: zhangxdmath@yahoo.com.cn

Qingsong Tang, Hua Jin, Yue Qiu , Yang Guo
School of Science, Northeastern University, Shenyang, China

***Abstract*—Eye location is one fundamental but very important problem for face recognition. In this paper, we proposed a new eye location method based on Adaboost and Random Forests. The proposed method consists of three main steps. Firstly, we apply Haar features and Adaboost algorithm to extract the eye regions from a face image. Secondly, we highlight the characteristics of eyes by Gabor filter, then segment the pupil from the eye regions based on intensity information. We compute the coordinate of center of the pupil as the position of the eye. Lastly, the eye location result is judged and adjusted by symmetry-axis of the face and Random Forest. Compared with the existing eye location approaches, the proposed method use the symmetry-axis of the face and Random Forests to judge and adjust the eye location result, which enhance the accuracy of eye location remarkably. The proposed method has been tested in the CAS-PEAL-R1 database and CASIA NIR database respectively, the simulation results demonstrate that the location accuracy rate is 98.86% and 97.68% respectively.**

***Index Terms*—Adaboost, regional features, eye location, Haar feature**

## I. INTRODUCTION

Compared with other biometrics, face recognition is covert, friendly and convenient, so it has become a hot topic during the last several decades in pattern recognition, artificial intelligence, and computer vision. As eyes are the most salient and stable features in faces, Eye location is a necessary and very important part in automatic face recognition because before two face images can be compared they should be aligned in orientation and normalized in scale. The precision of eye location is crucial to the face recognition system. In addition, the accurate eye location is very important to locate other organs (such as nose, mouth etc) [1]. Because the eyes often reflect a person’s desires, needs and emotions, eye location, together with eye tracking is crucial for facial expression analysis, human computer interaction, and attentive user inter faces. Many scholars have studied eye location deeply, and some methods for this problem have been developed. Existing eye location methods can be broadly classified into two categories: the active infrared based methods and the traditional image-based passive method. The former is based on the principle of red-eye effect in flash photographs. The image based passive eye location approaches often employ the special properties of eyes in intensity, or shape of the eye. This method is relatively simple and very effective without eyeglass reflection. It can obtain high eye detection and tracking robustness and accuracy, especially indoors. But the eyeglass refection is very serious [1]. The image based passive eye location approaches often employ the special properties of eyes in intensity, or shape of the eye. This method is relatively simple and very. The image based passive eye location approaches often employ the special properties of eyes in intensity, or shape of the eye. There three main methods: projection function, template matching and Hough transform. A pupil is generally darker than surround eyeball, therefore algorithms can be designed to search for gray character, based on which projection function is proposed [2]. Hough transform is another widely used eye location method [3]. It is based on the shape feature of the iris and often works on binary valley or edge maps. Another eye location method is template matching [4]. However, these eye location methods are not precise or only effective to specifically restrained conditions, so their application is not extensively. Manly factors, such as face rotation and lighting conditions affect the performance of eye detection algorithm. But most of the existing mainly focuses on eye location from the face image of frontal view.

In this paper, a novel eye location method based on Adaboost and local features are developed. The proposed method contains three steps: (1) Eye region features are extracted based on Harr features and Adaboost. (2) Eyes are located based on local features. (3)The eye coordinate is judged and adjusted by symmetry-axis and Random. In order to enhance the accuracy of eye location, we use the summitry of the face and Random Forests to judge and adjust the eye location. Fig. 1 shows the flow chart of the proposed eye location in this paper.

Corresponding author, Guo Yang,Email:. gy_maths_pr@hotmail.com


doi:10.4304/jsw.7.10.2365-2371

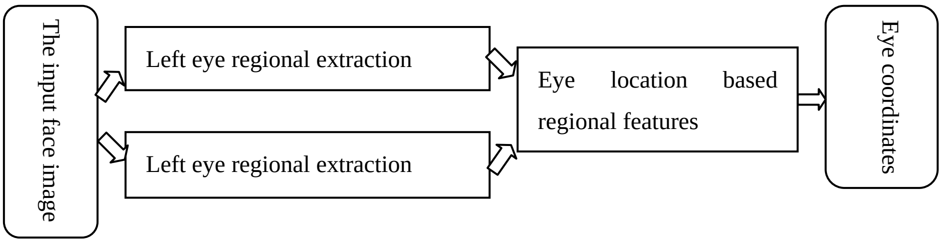



Figure 1. Flow chart of eye location

## II. Eye Location Based On Region Features

### *A. Haar Features*

Since Adaboost is a statistical method, it is important to choose effective feature for classification.Viola introduced Haar features into face detection [5], and he defined 4 elementary Haar features. Haar feature is the total gray value in the white region plus total gray value in the black region. In order to calculate Haar feature quickly, Viola given a method based on integral image. The integral image is shown in fig. 2.

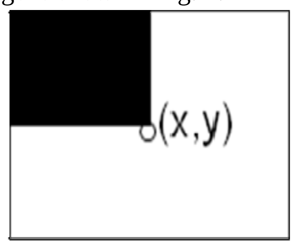



Figure 2. Schematic diagram of integral image

Let $i(x, y)$ represent gray intensity of the original image at $(x, y)$, and $ii(x, y)$ represent gray intensity of the integral image at $(x, y)$, satisfy

$$ii(x, y) = \sum_{x' < x, y' < y} i(x', y') \tag{1}$$

According to the gray distribution in eye region, we expand these elementary Haar features. Fig. 3 illustrates some Haar features used in this paper corresponding to gray distribution in eye region.

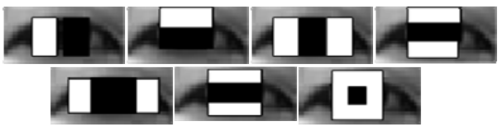


Figure 3. Haar features applied in eye location

### *B. Adaboost*

Adaboost is an algorithm which can get a stronger classifier by combination of large numbers of weak classifiers [6]. The procedure of training an Adaboost classifier is displayed as follows.

Step 1: Let $n$ be the total number of training samples and $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ be a training set,

Step 2: Let $y_i = -1, 1$, where -1 represents the negative (without eyes) and 1 represents positive (with eyes) samples, respectively. For every feature $x$, calculate the feature values of all the samples

$$F = (f_1, ..., f_n).$$

and sort feature values from small to large, we have

$$FS = (fs_1, ..., fs_n)$$

Step 3: Calculate the followed four kind of weight sum.

1) The sum weight of all the positive samples, denoted as $T\_P$.

2) The sum weight of all the negative samples, denoted as $T\_N$.

3) The sum weight of the first $k+1$ positive samples, denoted as $Pre\_P$.

4) The sum weight of the first $k+1$ negative samples, denoted as $Pre\_P$.

Step 4: Calculate the optional threshold

$$\theta_t = \begin{cases} (fs_t + fs_{t+1})/2, & t \neq n \\ fs_t + 1, & t = n \end{cases} \tag{2}$$

At the same time, calculate the classify error (i.e weighted error rate)

$$\varepsilon 1_t = Pre\_N + (T\_P - Pre\_P)$$

$$\varepsilon 2_t = Pre\_P + (T\_N - Pre\_N)$$

$$\varepsilon_t = \min(\varepsilon 1_t, \ \varepsilon 2_t)$$

Step 5: Choose the threshold with the minimize error rate denoted as $\varepsilon_t$ as the best threshold. Denote this threshold as $\theta_t$.

Step 6: Set the offset direction as.

$$p = \begin{cases} 1, & \varepsilon 1 < \varepsilon 2 \\ -1, & otherwise \end{cases}.$$

After the five steps, we get a weak classifier with minimize error rate. We can construct strong classifiers as the follow steps

Step 7: Initialize the weight

$$w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$$

where $m$ and $l$ are the number of positive samples and negative samples, respectively.

Step 8: Train $T$ weak classifiers as follows

1) Normalize the weight

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}} \tag{3}$$

for $i = 1, 2, ..., T$.

2) Train a weak classifier for every feature and

calculate its error rate on weight $w_j$

$$\varepsilon_j = \sum_{i=1}^{n} w_i \left| h_j(x_i) - y_i \right| \quad (4)$$

3) Choose the weak classifier with the minimize error rate, denote this weak classifier as $h_t$ ;

4) Update the weight of the sample

$$w_{t+1,i} = w_{t,i}\beta_t^{1-e_i} \quad (5)$$

where, $\beta_t = \dfrac{\varepsilon_t}{1-\varepsilon_t}$ ( $\varepsilon_t$ is the weighted error rate of weak classifier $h_t$ ), If $x_i$ is right classed, $e_i = 0$ , otherwise, $e_i = 1$ .

4) We get the strong classifier as follows

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2}\sum_{t=1}^{T}\alpha_t \\ -1 & otherwise \end{cases} \quad (6)$$

Where, $\alpha_t = \log\dfrac{1}{\beta_t}$

Step 9: In order to enhance the searching speed, we cascade the strong classifiers. The reasons are as following:

1) Most none eye regions are easy to be distinguished, such they can be refused by the preceding strong classifiers and need not be judged by the rear strong classifier.

2) In the training procedure, the rear strong classifier only need to treat the images which accepted by the preceding strong classifier, such the feature which used in coarse classify.

### *C Image Preprocessing*

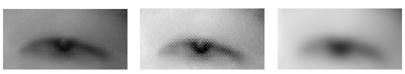
Before eye location, we need to preprocess the extracted eye images to remove the noises and get uniform gray distribution. Several filters are used to do image preprocessing such as homomorphism filter, circle filter, mean filter and Gabor filter [7-9], the first three filters are very simple, we just introduce Gabor filter briefly:

$$G(x, y, \theta, \sigma, \lambda, \varphi) = e^{\frac{-(x'^2 + y'^2)}{2\sigma^2}} \cos(2\pi\frac{x'}{\lambda} + \varphi) \,. \quad (7)$$

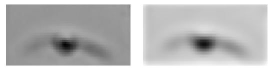
where $x' = x\cos\theta + y s\in\theta$ , $y' = -x\sin\theta + yc\,os\theta$ .

Firstly, the contrast is enhanced by homomorphism filtering. Then, we construct a circle filter to stand out the circle character, and we remove the noise by mean filtering subsequently. Lastly, the image is enhanced by a Gabor filter for its direction and frequency selectivity.

Fig.4 (a) shows the eye regions extracted by Adaboost algorithm and the filtered results by different filters. Comparing Fig.4 (a) and Fig. 4 (e), we can find the processed image highlight the pupil region, which is beneficial for eye location.



(a) (b) (c)



(d) (e)

Figure 4. The processing results of eye region: (a) Original eye region; (b) Homomorphic filtering; (c)Circle filtering; (d) Mean filtering; (e)Gabor kernel filtering

### *D Eye Location*

Because the gray intensity of pupil region is lower than its neighboring district, we can use this character to segment the pupil from eye region. In order to determine the threshold, the gray intensity of the preprocessed image is normalized into [0, 1]. The eye location procedure is as follows [10]:

Step 1: Let binaryzation threshold as $T = T\_min + T\_d$ . ( $T\_min$ is the minimum gray intensity of the image, and $T\_d$ is the step length of the threshold) .

Step 2: The image *I* is segmented to binary image $bw_0$ based on *T*, and morphologic opening and closing operation are performed on the binary image. After that, the image is filtered by a circle filter. Lastly, the position and acreage of every connected region is accounted, and all of the connected regions are arrayed according the acreage.

Step 3: Setting judgment criterion:

1) There exists at least one connected region;

2) The acreage of the biggest connected region is in a certain range;

3) The distance between the biggest connected region and the edge of the eye region is more than a certain number.

If the binary image satisfies all the three conditions, turn to step 4. Otherwise, turn to step 5.

Step 4: Determining the position of the eyes:

1) If the number of the connected region is one, then we set $bw(i, j)$ as

$$bw(i, j) = \sum_{(ii, jj)\in circularity(i, j, R)} bw0(ii, jj) \,. \quad (8)$$

After that, we set the pixel whose gray intensity is maximal on image $bw$ as the eye center. If the center position of the eye satisfies condition 1) in step 3, we think the eye is found; else we turn to step 5.

2) If the number of the connected region is more than one, we find the largest connected region and determine the eye center position as there is one connected region. If the founded eye position satisfies condition 1) in step 3, then exit the algorithm. Otherwise, we find the second largest connected region and determine the eye center position. If the founded eye position satisfies condition 1) in step3, ending. Otherwise, turn to step 5.

Step 5: Let $T := T + T\_d$ , if $T < T\_\max$ ( $T\_\max$ is the maximal threshold), then we turn to step 2.

Fig. 5 shows the binary images with increasing threshold. When $T$ =0.4, the eye position satisfying the judgment criterion is found and the eye location is finished. The eye position is shown in Fig. 6.From Fig. 6. We can see the eye location is effective.

(a) (b)

Figure 5.The binary images gotten from different threshold (a) The binary image when threshold is 0.2; (b)The binary image when threshold is 0.4

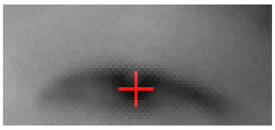
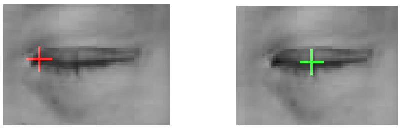


Figure 6. The result of eye location

For the closed eye image, because there is no circle geometrical characteristic, the connected region may emerge in the corner of the eye. It may result in error. In order to regulate such error, we extract a region using the eye coordinates after eye location, and then judge the eye state according to the closed eye like a rectangle. If the eye is closed, then we regulate the error eye location. Fig.6 shows the regulated results.



(a) (b)

Figure 7. The results before and after regulated: (a) before regulated; (b) after regulated;

## III. Judgment and Adjustment based on Symmetry-axis and Random Forest

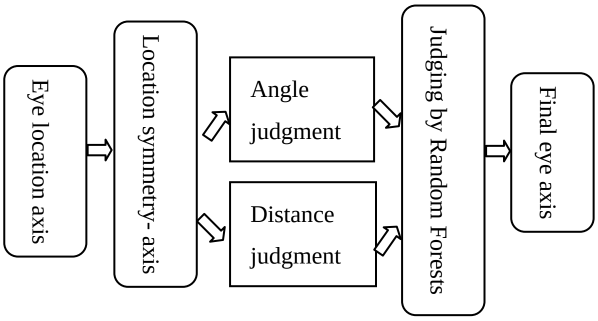



Figure 8. Flow chart of Judgment and adjustment

Eye location based on region features relies on statistical and grey intensity, but it doesn't contain the geometrical formation constraint. Hence, we use the geometrical information to judge and adjust the eye location result. The procedure is drawn in figure 8.

Since the detected face image is symmetry, it is easy to get the symmetry-axis which pass through the nose and mouth. We can judge and adjust the eye location result based on this symmetry-axis. So the eye location is more accurate. The detail procedure is described in the following A, B, C, D, E

### *A The Principle of Symmetry-axis Location [11]*

Suppose the value of a function $f(x,y)$ is 0 out of a circle whose radius is $L$, and the map of $f(x,y)$ has symmetry-axis. We perform the coordinate transformation as follows

$t = x\cos\theta + y\sin\theta,$

$s = x\sin\theta - y\cos\theta.$

In the new coordinates system, if

$\forall t,s\ f(t,s) = f(t,-s)$

$f$ is an axis-symmetric function. If

$\forall t,s\ f(t,s) = -f(t,-s)$

$f$ is an anti-symmetric function.

Using the formula of coefficient, we can have the following expression.

$$S_\theta\{f\} = \frac{1}{2}\frac{\iint f(t,s)f(t,-s)dsdt}{\int f^2(t,s)dsdt} + \frac{1}{2}. \tag{9}$$

Where $S_\theta\{f\} \in [0,1]$ is a value which represents the symmetry of $f$ .If $S_\theta\{f\} = 1$ , $f$ is a symmetric function, and if $S_\theta\{f\} = 0$ , $f$ is not a symmetric function. The more $S_\theta\{f\}$ is close to 1, the more $f$ is symmetric.

### *B The Steps of Symmetry-axis Location*

1. Illumination processing and binarization

Because illumination affects the binary image heavily, we use self-quotient [12] image (SQI) to preprocess the image.

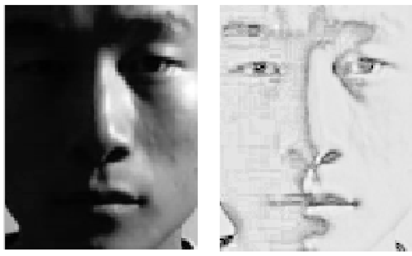


Figure 9. Face image and it's SQI

The self-quotient of image $I$ is usually defined as

$$R = \frac{I}{\hat{I}} = \frac{I}{F \times I}. \tag{10}$$

Where $\hat{I}$ is the smoothed image of $I$ , and $F$ is the kernel function for filtering. Since self-quotient imageis invariable to illumination, it can eliminate the shadow. Fig.9 shows a face image and it's SQI.

The division operation in SQI probably enlarges the noise. In order to decrease the noise in $R$ , we use a nonlinear transformation $T$ to transform $R$ to $D$ :

$$D = T(R) \tag{11}$$

In this paper, we calculate SQI as follow steps:

Step 1: Choose some smoothing kernel $G_1, G_2, \ldots, G_n$ , calculate corresponding weight $W_1, W_2, \ldots, W_n$ , and then use every weighted anisotropic filter $WG_k$ to smooth the image $I$

$$\hat{I}_k = I \oplus WG_k, k = 1, 2, \cdots, n. \quad (12)$$

Step 2: Calculate SQI of every input image and its smoothed image by

$$R_k = \frac{I}{\hat{I}_k}, k = 1, 2, \cdots, n. \quad (13)$$

Step 3: Transform the SQI by the given nonlinear function

$$D_k = T(R_k), k = 1, 2, \cdots, n. \quad (14)$$

Step 4: Calculate the weighted sum of the results from step 3 to get $R$

$$R = \sum_{k=1}^{n} m_k D_k, k = 1, 2, \cdots, n. \quad (15)$$

Where $m_1, m_2, \cdots, m_n$ are the weight value respond to each scale. In experiment, we let all them to 1.

2. Extraction symmetric information and location symmetry-axis.

We compute the center of the image and denote the center coordinates as $(m_x, m_y)$. Then, we translate the center point to get some other points as the possible center point and define a deflection range of the symmetry-axis so that we can get some possible symmetry-axes. We choose the symmetry-axis has maximal correlation coefficient as the final symmetry-axis.

*C The Principle of Random Forests*

Random Forests [13] is a combination classification algorithm. Its basic idea is combination majority weak classifiers to get a strong classifier. Random Forests is fast and easy to train. In this paper, in order to enhance the eye location accurate, we combine Random Forests and Haar feature to judge the eye location result.

A Random Forests contains $N$ decision trees. Every node of decision tree is a weak classifier. Every decision tree recurrences download to get the maximal deep. In the training process, training samples of every decision tree is a subset of the whole sample set. The decision tree searches optimal data feature and threshold to classify the sample at every node. All the decision trees compose a strong classifier, as shown in Fig.10.

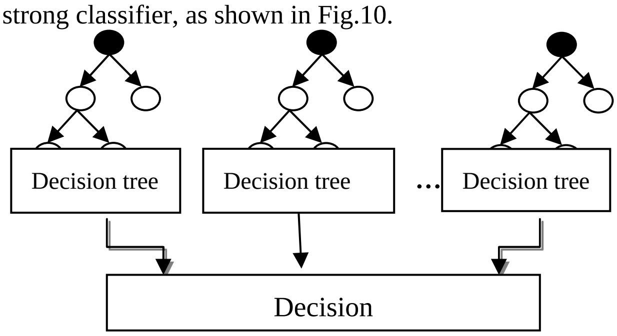



Figure10. Random Forests

The judgment of the Random Forests relies on the average of the results from all the decision trees. Input a possible eye image block $E$ into left or right Random Forests classifier, the output of the Random Forests shown as (16)

$$p_1 = (1/N)\sum_{n=1}^{N} p_{(n,p)}, p_2 = 1 - p_1 \quad (16)$$

Where $p_1$ represents the probability that the block is eye image, and $p_2$ represents the probability that the block is not eye image. $p_{(n,p)}$ represents the decision of $n$th decision tree. When $p_1 \geq 0.5$, we consider the image block as pupil. $p_1 < 0.5$, we consider the image block not pupil.

*D Training of Random Forests*

In order to train the Random Forests, we collect more than 600 face images which contain variation of illumination and expression. The eye coordinates of these images are marked by hand. We choose a region which centers on the eyes proportion to the size of the image as positive sample. The negative samples are chosen from face image near the eyes but not containing the eye or not containing the whole eye. In order to train the Random Forests, Haar feature of every image is calculated.

Since judgment the eye region is a two-class problem, every decision tree is a Binary Tree. Training the decision tree is to choose weak classifier for every node. The Haar feature which is best to classify ongoing samples is chosen as the weak classifier for the node. After that, the samples are sanded to the next level node, and the procedure above is duplicated until the correct rate requirement is met.

*E Judging and Adjusting the Eye Location Result*

The judgment and adjustment of the eye location result is done as follows.

(1) Using symmetry-axis to judge the result.

Because the symmetry-axis is perpendicular to symmetry-axis in face image, we calculate the angle of symmetry-axis and the line between the two eyes to judge the eye location result. We denote this angle as $cross\_theta$.

Step 1: If $cross\_theta > T_1$ or $cross\_theta > T_2$, where $T_1$ and $T_2$ are the given threshold, we regard the eye location is wrong.

Step 2: In order to decide which eye is wrong located, we calculate the distance from the left eye and right eye to the symmetry-axis. We denote the distance as $left\_dis$ and $right\_dis$. If $left\_dis$ or $right\_dis$ is bigger than $T_3$ or less than $T_4$, where $T_3$, $T_4$ are the given thresholds, we think respond eye is wrong located and jump to step 3, otherwise we jump to (2).

Step 3: If left eye is located wrongly and right eye is located rightly, we use the following formula to get the position of the left eye.

$$w\_eye(2) = r\_eye(2) - 2 \times mideye\_dis \times \sin(cross\_theta) \quad (13)$$

Where $w\_eye$ is left eye after revised, $r\_eye$ is the right eye coordinates and $mideye\_dis$ is the distance between eye and symmetry-axis. The procedure is similar if the left is right located and right eye is wrong located. After the eye coordinates is adjusted, we back to step 1.

The eye location result after adjustment by symmetry-axis is shown in Fig. 11.

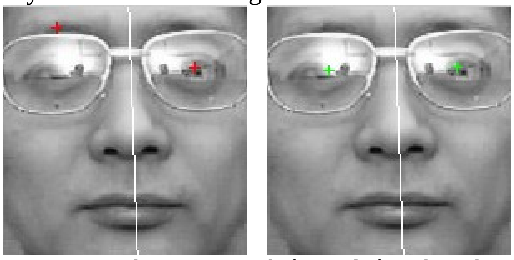


Figure11. The eye position before and after adjusted

(2) Using Random Forests to judge the result and using symmetry-axis to adjust the result

We use Random Forests to judge the extracted regions.

Step 1: Each probability for left eye and right eye block from (1) is calculated using formula (5) and two probability values denoted as $leftTemp$ and $rightTemp$ are gotten.

Step 2: Compare $leftTemp$ and $rightTemp$ to a given threshold $T_5$.

If $leftTemp \geq T_6$ and $rightTemp \geq T_6$, where $T_6$ is a given threshold we regard both eyes are right located and output the eye coordinates. If $leftTemp \geq T_6$ and $rightTemp < T_6$, we regard the left eye is right located and the right is wrong located, and jump to step 3 in (1). If $leftTemp < T_6$ and $rightTemp \geq T_6$, we regarded the left eye is located wrongly and the right eye is located rightly, and also jump to step 3 in (1). If $leftTemp < T_6$ and $rightTemp < T_6$, we regard both the two eyes are located wrongly, then we need to refer to (1) and using the eye coordinates before adjustment and structure information of face image to estimate the eye position. In experiment, we find such circumstance seldom occurs.

## IV. Experiment Result

In order to evaluate our method, we apply the algorithm to CAS-PEAL-R1databaseand CASIA NIR database. CAS-PEAL-R1database is a visible light face database supplied by Institute of Computing Technology. CASIA NIR is a near-infrared face database supplied by Institute of Automation Technology Chinese Academics of Science. Both them contain enough face images to test the effective of the algorithm.

### *A. Test Criterion of Eye Location*

Two indexes are often usd for the performance evaluation of an eye location method. One is the eye location rate; the other is the location accuracy. The former refers to the ratio of the number of images for which two eyes are correctly located to the total number of images tested, while the latter means the disparity between the manually detected eye position and the automatically detected position. In this paper, we evaluate the eye location result according to the algorithm given by Jesorsky[14]. Assuming the left and right eye coordinates from manual calibration are $E_L$ and $E_R$ respectively, while the coordinates from the proposed method are $E_L^{'}$ and $E_R^{'}$. We denote $D_L$ as the Euclidean distance between $E_L$ and $E_L^{'}$, $D_R$ as the Euclidean distance between $E_R$ and $E_R^{'}$, and $D_{LR}$ as the Euclidean distance between $E_L$ and $E_R$. The relatively error eye location is defined as follows

$$err = \frac{max(D_L, D_R)}{D_{LR}}. \tag{14}$$

The advantage of this definition is the relatively error has nothing to do with image resolution. Considering the practicality, we use $err < 0.10$ as the criterion of accurate eye location.

### *B. The Results on CAS-PEAL-R1*

CAS-PEAL-R1 includes 999594 face images captured from 1040 Chinese people. The face images in CAS-PEAL-R1 involves variety images of posture, expression, accessory, and age.

We choose Normal, Expression, Accessory, Background, Distance and Aging sub-database to test the algorithm because we can control the posture in practical application and age is not in practical application. The accurate rate of eye location is 98.86% to the 6224 right detected face images, and the results are shown in Table I. From table I, we can see the given algorithm is effective, in every sub-database, the accurate rate is more than 97%. In normal sub-database the accurate is 100%, and accessory affect eye location more than expression and pose.

TABLE I.
EYE LOCATION RESULTS ON CAS-PEAL-R1 DATABASE

| Sub-database | Right | Error | Total | Accurate rate |
|---|---|---|---|---|
| Normal | 1040 | 0 | 1040 | 100% |
| Expression | 1870 | 10 | 1880 | 99.47% |
| Accessory | 2206 | 58 | 2246 | 97.44% |
| Background | 649 | 2 | 651 | 99.69% |
| Distance | 322 | 1 | 323 | 99.69% |
| Aging | 66 | 0 | 66 | 100% |
| Total | 6153 | 71 | 6224 | 98.86% |

The results of our method and some other authors are shown in Table II. From table II, we can see our method is much more effective than other methods known.

TABLE II.
COMPARE EYE LOCATION RESULTS ON CAS-PEAL-R1 DATABASE

| method | Total number | Accurate rate |
|---|---|---|
| Ma Xiao-qiang, Fu Zhi-peng[15]. | 5768 | 93% |
| Wang Jin-fan[16]. | 4712 | 82.43% |
| Our approach | 6224 | 98.86% |

### *C The Results on CASIA NIR*

CASIA NIR contains 3940 face images from 197 people, that is, each person has 20 images. These images contain variety of pose, expression, age and accessory.

Since near-infrared image cause strong refection with glass, the face images with and without eye glass refection are very different. In order to choose more effective feature to construct classifier, we train different classifiers to such two kind samples. The accurate rate of eye location is 98.86% to the 6224 right detected face images, and the results are shown in Table III. From table III, we can see the glass reflection affect the eye location heavily. The reason is that the glasses reflection result to a big white area which probably eclipses the pupil. How to deal with such case is also a challenge problem for us.

TABLE III.
EYE LOCATION RESULTS ON CASIA-NIR DATABASE

| Sub-database | Right | Error | Total | Accurate rate |
|---|---|---|---|---|
| Without glasses | 3293 | 21 | 3314 | 99.37% |
| With glasses | 536 | 70 | 606 | 88.45% |
| Total | 3829 | 91 | 3920 | 97.68% |

The comparison between the result of Stan Z.Li [6] and our method are shown in table IV. From table IV we can see our method is more effective than Stan Z.Li's method which is the best existing method. The accuracy of proposed method is one percent higher than Stan Z.Li's method either with or without eye glass.

TABLE IV.
COMPARE EYE LOCATION RESULTS ON CASIA-NIR DATABASE

| Method | Accurate rate to face Images Without glasses | Accurate rate to face images With glasses |
|---|---|---|
| Stan Z.Li etc | 98% | 87% |
| Our approach | 99.37% | 88.45% |

## V. SUMMARY AND CONCLUSIONS

Eye location is a very important step for automatic face recognition system. It is also a very difficult problem because of its complexity. In this paper, we propose a new eye location method. We use Adaboost to extract the eye region. Different to most existing methods, after the eye is location, we use symmetry-axis and Random Forest to judge and adjust the result. The developed method has been tested in the CAS-PEAL-R1 database and CASIA NIR database separately, and the accuracy rate is 98.86% and 97.68% respectively. The experiment results also demonstrate the given method is robust for it can be apply to face images with accessories, different background, expression variety, and displacement. However, our method is not every effective to near-infrared images with glasses for their strong reflection. In such case, the accuracy decreases to 88.45%, though also higher than the existing method. In addition, the proposed method cost a little more time than some existing methods. In future, we will consider how to deal with such problem.

## ACKNOWLEDGMENTS

The experiments in this paper are performed on CAS-PEAL-R1and CASIA NIR supplied by institute of Computing Technology Chinese Academics of Science and Institute of Automation Technology Chinese Academics of Science, respectively.

## REFERENCES

[1] Fan, Y.H., Ma, J.W., "ASM and Improved Algorithm for Facial Feature Location," Journal of Computer-aided Design and Computer Graphics, vol.11, pp. 1411-1415, November, 2007.

[2] Ma, Y., Ding, X.Q., Wang, Z.E., "Robust precise eye location under probabilistic framework," Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, Beijing, 2004, pp. 339-344.

[3] Sewisy, A.A., Leberl, F., "Detection Ellipse by Finding Lines of Symmetry in the Images Via an Hough Transform Applied to Straight lines," Image and Vision Computing, vol.19 pp. 857-866, October, 2001.

[4] Zhang, B.Z., Ruan, Q., "Facial Feature Extraction Using Improved Deformable Templates," Signal Processing, 2006 8th International Conference on, Beijing, 2006, pp. 329-334.

[5] Viola, P., Jones, M., "Robust Real-Time Object Detection. International Journal of Computer Vision," vol.57, pp. 137-154, February, 2004.

[6] Li, S.Z., Zhang, Z.Q., "FloatBoost Learning and Statistical Face Detection", IEEE Transactions on pattern analysis and machine intelligence, vol. 26, pp. 1112-1123. September, 2004.

[7] Wang, Q., Yang, J.Y., "Eye Detection in Facial Images with Unconstrained Background. Journal of Pattern Recognition Research", vo1. 1, pp. 55-62, January, 2006.

[8] Park, C.W., Kwak, J.M., Park, H., et al, "An Effective Method for Eye Detection based on Texture Information", Convergence Information Technology, 2007. International Conference on. Seoul, 2007, pp.586-589.

[9] Li, K., Yan, H., "Eyes Location based on Dual-orientation Gabor Filters and Templates. Image and Signal Processing, 2009", CISP '09. 2nd International Congress on. Shenyang, pp. 1-4, October, 2009.

[10] Huang Haiying. Hu Guiming, "He Jinge. An improved geometry-based eye location method", Journal of GuangxiU niversity: Nat Sci Ed, Vol.31, pp. 214-217. September, 2009.

[11] Nahum Kiryati, Yossi Gofman, "Detecting Symmetry in Gray Level Images: The Global Optimization Approach. International Journal of Computer Vision", vol.29, pp. 29-45, January, 1998.

[12] Wang Haitao, Liu Jun, Wang Yangsheng, "Self-quotient Image", Computer Engineering, vol.31, pp.178-180, September, 1998.

[13] Bremian L., "Random forests", Machine Learning, Vol.45, pp, 5-32, January, 2001.

[14] Li, S.Z., Chu, R.F., Liao, S.C., et al, "Illumination Invariant Face Recognition Using Near-Infrared Images", IEEE Transactions on pattern analysis and machine intelligence, vol.27, pp.1-13, April, 2007.

[15] Oliver, J., Klaus, J.K., Robert, W.F., "Robust Face Detection Using the Hausdorff Distance", Lecture Notes in Computer Science, vol. 2091, pp.90-95, January, 2001.

[16] Wang, J.F., Qian, Y., Tong, W.Q., "Eyes Detection Based on Adaboost Algorithm", Modern Computer, vol.9, pp.73-75, September, 2001.