Pattern Recognition Letters 31 (2010) 1633–1640

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Automatic eye detection using intensity filtering and *K*-means clustering

Zhiming Qian [a,b,*], Dan Xu [b]

[a] *Chuxiong Normal University, Chuxiong 675000, China*
[b] *Department of Computer Science and Engineering, Yunnan University, Kunming 650091, China*

ARTICLE INFO

Article history:
Received 31 January 2009
Available online 17 May 2010
Communicated by T. Tan

Keywords:
Eye detection
Gabor transformation
Cluster analysis
Neighborhood operator

ABSTRACT

This paper proposes a novel eye detection method, which can locate the accurate positions of the eyes from frontal face images. The proposed method is robust to pose changes, different facial expressions and illumination variations. Initially, it utilizes image enhancement, Gabor transformation and cluster analysis to extract eye windows. It then localizes the pupil centers by applying two neighborhood operators within the eye windows. Experiments with the color FERET and the LFW (Labeled Face in the Wild) datasets (including a total of 3587 images) are used to evaluate this method. The experimental results demonstrate the consistent robustness and efficiency of the proposed method.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Eye detection is an essential initial step in many face processing applications, such as face tracking, expression analysis and face recognition. In recent years, tremendous efforts have been made in the field of face recognition to improve recognition effectiveness (Zhao et al., 2003; Phillips et al., 2005). Further, the studies (Shan et al., 2004; Wang et al., 2005; Rodriguez et al., 2006) have shown that eye localization has a considerable impact on face recognition accuracy.

General eye detection algorithms can be broadly classified into two categories: the active infrared-based approaches and the image-based passive approaches (Ji et al., 2005). The active infrared-based approaches use the physiological properties of eye pupils under infrared lightings (Haro et al., 2000; Zhu et al., 2002). These approaches can obtain highly accurate eye positions that are robust to noise. However, they require additional hardware settings at the image acquisition stage, thereby limiting their potential applications. Some of the image-based passive methods, such as projection (Zhou and Geng, 2004) and templates (Lam and Yan, 1996; D'Orazio et al., 2004), have attempted to locate the eyes using the visual appearance features of the human eye. These passive methods have produced good results in some experiments, but they also have certain limitations. For instance, facial expression, lighting condition and image resolution affect the performance of eye detection algorithms.

Besides, some image-based passive methods consider eye detection as a two-class or multi-class pattern classification problem. Adaboost and support vector machine techniques are used to address this problem (Wang et al., 2005; Tang et al., 2005; Campadelli et al., 2006). These statistical learning methods have faster speed and higher accuracy, but their training process is very time consuming. Moreover, the phenomenon of missing detection, in which an eye region is classified as a non-eye region, affects the performance of face recognition systems.

In this paper, an eye detection method based on intensity information, is presented. The proposed method has the following characteristics:

(1) It works on frontal face images within and outside the plane rotations up to 30°.
(2) It does not need training or learning in advance.
(3) It can locate the eyes from face images with variations in pose, facial expression and lighting condition.

The rest of this paper is organized as follows: Section 2 describes our eye detection algorithm; Section 3 presents the experimental results on the color FERET and the LFW (Labeled Face in the Wild) datasets; and finally, conclusions are offered in the last section.

2. The proposed method

2.1. Brief introduction

Many special eye properties (e.g., shape, texture, edge, etc.) have been used in the image-based passive eye detection approaches. Among these, intensity information is considered the most critical property. Inherently, human eyes are darker than

* Corresponding author at: Chuxiong Normal University, Chuxiong 675000, China. Fax: +86 13187669936.
E-mail address: qzhiming@126.com (Z. Qian).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.05.012

other parts of the face, and pupils are darker than the surrounding eyeballs. Based on these two cues, we adopt the coarse-to-fine strategy for eye detection. More specifically, our method contains two stages. In the first stage, image enhancement, Gabor transformation and *K*-means clustering are used to roughly localize the eyes. In the second stage, two neighborhood operators are employed to determine the precise positions of the eye centers. The process can be described as follows:

(1) *Face detection:* The input image is analyzed to determine the location and size of one or more faces (in all of the following steps, we assume that only a single face is found).
(2) *Histogram analysis:* Through a histogram analysis of the face image, the gray level range of the facial skin is found.
(3) *Image enhancement:* The grayscale image is adjusted, improving the contrast of the face image.
(4) *Gabor transformation:* The reference image is obtained by convolving the enhanced face image with Gabor wavelets.
(5) *Cluster analysis:* The reference image is analyzed using *K*-means clustering algorithm in order to discover the rough eye positions.
(6) Pupil center localization: The central position of the pupil is determined.
(7) The architecture of the proposed method is shown in Fig. 1.

2.2. Face detection

Face detection is a pattern classification method that finds the locations and sizes of human faces in arbitrary images. Over the past decade, face detection has been studied thoroughly due to its wide range of possible applications. Numerous approaches have been proposed (Yang et al., 2002), including pixel-based, parts-based, local edge features, and Haar wavelets approaches. In 2001, Viola and Jones (2001) presented a new face detection algorithm using a boosted cascade of Haar features. Due to the robustness and real-time capability of the approach, many different methods have been presented based on the enhancement of the boosting architecture. These include the improvement of AdaBoost learning (Li and Zhang, 2004), the use of new features (Dollar et al., 2007; Huang et al., 2007a), optimal feature selection (Wu et al., 2008; Brubaker et al., 2008), etc.

In this paper, all the detected face images based on Adaboost algorithm are scaled to a standard size of 130 × 150 pixels, which helps reduce the amount of translation operations and find pupil centers.

2.3. Histogram analysis and image enhancement

The eye has varied appearances due to the influence of many factors (e.g., lighting condition, expression, facial shadowing, etc.). To reduce these influences, some preprocessing methods have been proposed, and achieved good results (Tan et al., 2009; Song et al., 2006). In order to deal with similar problem, a simple and effective image preprocessing method, carried out prior to the eye detection stage, is presented. Based on the cue that eyes are darker than other parts of the face, we consider enhancing the image contrast to make the eye region more obvious than other regions of the face. First, we obtain the rough intensity value of the skin area from the face image. Second, based on the obtained skin intensity value, contrast enhancement can be employed to increase the local contrast of the eye feature.

Usually, the skin area holds a great mass of the image area in the frontal view face; hence, we select a subregion in the center of the face image as a skin sample. In order to overcome noise contamination in the subregion, the simplified local histogram analysis of the face image is used. Given a face image, we convert it to a grayscale image, after which we obtain the intensity value of the skin area based on Algorithm 1.

Algorithm 1. The simplified local histogram analysis algorithm.

Input: Grayscale face image *G*
Output: Intensity value of the skin area *si*
Step 1: Cut from the input image, and get the test window which is a rectangle (31 × 13 pixels), positioned at the center of the input image.
Step 2: Divide 256 different intensities (0–255) into 13 groups denoted as $g_1(0\text{–}19), g_2(20\text{–}39), \ldots, g_{12}(220\text{–}239), g_{13}$ (240–255). Find the maximum $m_1, m_2, \ldots, m_{12}, m_{13}$ for each group in the histogram of the input image.
Step 3: Calculate the mean intensity of the test window and find the *i*th group g_i corresponding to the mean intensity. The intensity value of the skin area *si* is equal to m_i.

After determining the intensity of the facial skin from the face image, we use contrast enhancement to adjust the contrast of the face image. Supposing that $I(i,j)$ is the intensity of a pixel at location (i,j), and *si* is the intensity of the skin area, the new grayscale image $I'(i,j)$ can be defined as:

$$I'(i,j) = \begin{cases} 0, & I(i,j) < I_{\min},\ I_{\min} = si^{r}/c_1 \\ 255, & I(i,j) > I_{\max},\ I_{\max} = si^{r}/c_2 \\ \frac{(I(i,j)-I_{\min})\cdot 255}{(I_{\max}-I_{\min})}, & I_{\min} < I(i,j) < I_{\max} \end{cases} \quad (1)$$

where γ specifies the gamma correction factor, and c_1 and c_2 are two thresholds used to control the range of intensity values in the output image. Depending on the values of c_1, c_2 and gamma ($\gamma > 1$), the nonlinear gray level transformation can weight darker out values and improve the stability against illumination variations. In practice, when γ, c_1, and c_2 are in the ranges [1.2, 1.5], [1.2, 2.4] and

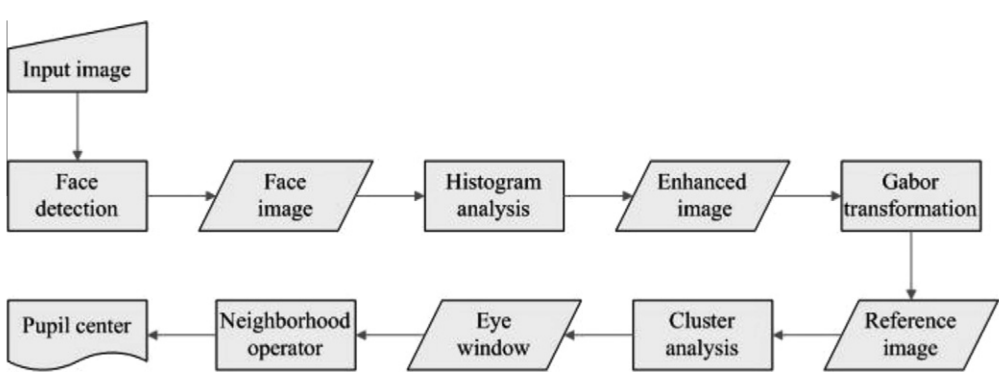

Fig. 1. Flowchart of our method.

[0.6, 1.2], respectively, this method will have a relatively better performance. In this study, we used $\gamma = 1.4$, $c_1 = 1.57$, and $c_2 = 0.785$ as the test settings. Fig. 2 shows the process of obtaining an enhanced image.

2.4. Gabor transformation

After performing image enhancement in the previous step, the influence of lighting variations, facial shadowing and expression changes for eye detection can be reduced to some degree. However, because the human eye is not a continuous region with the same intensity value, finding the eye region from the enhanced face image is not easy. To obtain a robust estimate of this region, Gabor transformation is employed by convolving the enhanced face image with Gabor wavelets.

The Gabor wavelet is well known for its effectiveness as a feature for image processing and pattern recognition (Lee, 1996). It exhibits desirable characteristics of spatial locality and orientation selectivity. The 2D Gabor wavelets can be defined as follows:

$$\psi_{\mu,v}(z) = \frac{\|k_{\mu,v}\|}{\sigma^2} e^{\left(-\|k_{\mu,v}\|^2\|z\|^2/2\sigma^2\right)}\left[e^{ik_{\mu,v}z} - e^{-\sigma^2/2}\right] \quad (2)$$

where v and μ define the scale and orientation of the Gabor wavelets, $z = (x, y)$, and $\|\cdot\|$ denotes the norm operator. $k_{v,\mu} = k_v e^{i\phi_\mu}$ is the wave vector, where $k_v = k_{\max}/f^v$ and $\varphi_u = \pi\mu/8$. $K_{\max}$ is the maximum frequency, and f is the spacing factor between kernels in the frequency domain. Suppose that $f(x, y)$ is the intensity of a face image, the convolution of $f(x, y)$ with a Gabor wavelet $\psi_{v,\mu}(x, y)$ can be defined as:

$$G(x, y, v, \mu) = f(x, y) * \psi_{v,\mu}(x, y) \quad (3)$$

where * denotes the convolution operator. To obtain multi-resolution Gabor features, five different scales and eight different orientations Gabor filters are used, with the following parameters: $v \in \{0, 1, \ldots, 4\}$, $\mu \in \{0, 1, \ldots, 7\}$, $k_{\max} = \pi/2$, $f = \sqrt{2}$, $\sigma = 2\pi$.

The Gabor representation of a face is formed from two components: the real and imaginary parts. In general, the imaginary part reflects the energy information, which is insensitive to local variations in the face image. Meanwhile, the real part reflects the contour information, which is critical to the facial feature representation of the face image. Thus, only the real part is considered in this study. From 40 Gabor-transformed face images (real part), we select three images ($\mu = 0$, $v = \{2, 3, 4\}$) to create a reference image. This selection is based on two reasons: (1) in the eye-and-brow region, orientation is the salient characteristic, which means that the signal contains more possibilities in the horizontal orientation than in other orientations; (2) low frequency Gabor filters proved to be more favorable for slowly varying intensity changes. Hence, low frequency Gabor filters are more important at the periphery of the eye, where changes in image intensity are relatively slow.

If we suppose $Greal(x, y, v, u)$ as the real part of $G(x, y, v, u)$, the reference image $Ri(x, y)$ can be defined as:

$$Ri(x, y) = q_1 Greal(x, y, 2, 0) + q_2 Greal(x, y, 3, 0) + q_3 Greal(x, y, 4, 0) \quad (4)$$

$$\begin{cases} q_1 + q_2 + q_3 = 1 \\ q_3 = 2q_2 = 4q_1 \end{cases} \quad (5)$$

where q_1, q_2, and q_3 represent weights in the image synthesis. Fig. 3 shows the creation process of the reference image. Through the Gabor transformation, the signal of the eye region can be further

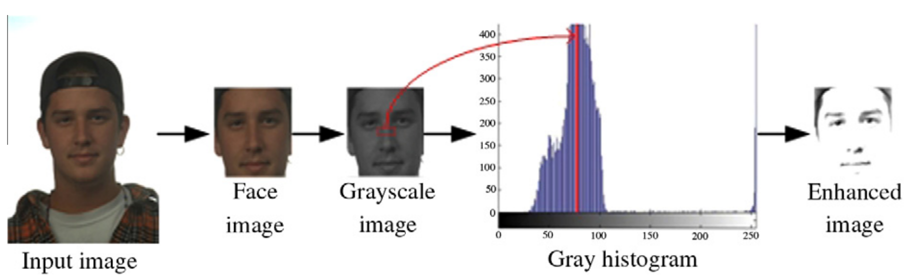

Fig. 2. The process of obtaining an enhanced image.

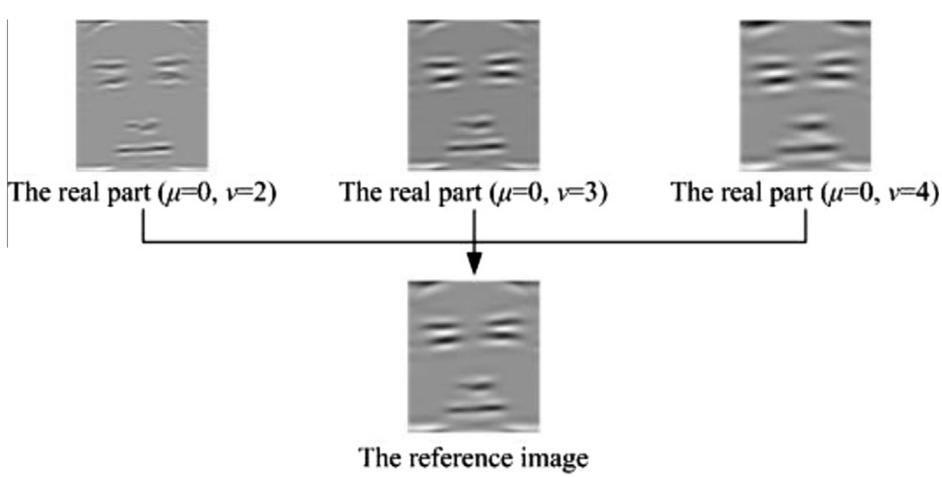

Fig. 3. The process of getting a reference image.

enhanced, and the influence of lighting variations, local shadowing and expression changes for eye detection can also be further reduced.

2.5. Cluster analysis

Based on the distribution of facial organs, the rough regions where the eyes may appear can be estimated. We suppose that the central coordinates of the eye are $e_1(x_1, y_1)$ and $e_2(x_2, y_2)$, $x_1 = 1.3w/4$, $y_1 = 1.6h/5$, $x_2 = 2.8w/4$, $y_2 = 1.6h/5$. Among them, w and h are the width and height of the face image, respectively. From this, the pretreatment windows, which can be cut from the reference image, are represented by rectangles $(0.32w \times 0.32w)$ centered at $E1$ and $E2$, respectively, as shown in Fig. 4.

In the pretreatment window, eye intensity is relatively low. Furthermore, via Gabor transformation, the intensity distribution in the eye region becomes relatively uniform. To segment the eye region, which has these intensity properties, cluster analysis is a good choice. However, because some non-eye regions (e.g., eyebrow, shadow around the eye area, etc.) and the eye region itself have similar intensity properties in some cases, it is not feasible to find the eye region by simple using cluster analysis. To solve this problem, we use both location and geometrical information to remove these non-eye regions. If $e_l(x, y)$ is the center point of the left eye region, and $e_r(x, y)$ is the center point of the right eye region; α is the angle between the horizontal direction and the line through e_l and e_r; p_l is the number of pixels in the left eye region; p_r is the number of pixels in the right eye region; w_r and h_r are the width and height of the right eye region, respectively; and w_l and h_l are the width and height of the left eye region, respectively. The following constraints are satisfied: (1) $w_l > h_l$ and $w_r > h_r$; (2) $p_l > 10$ and $p_r > 10$; (3) $\alpha < 30°$.

The following pretreatment window analysis algorithm shows the process of obtaining the eye region.

Algorithm 2. The pretreatment window analysis algorithm.

Input: Pretreatment window *PW*
Output: Eye window *EW*

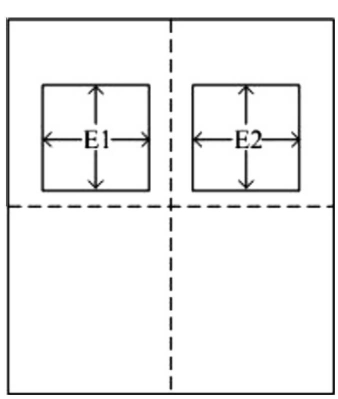

Fig. 4. The pretreatment windows.

Step 1: Partition *PW* each into three initial sets, and initialize the centroid of each set be respectively $center_1 = \min(PW)$, $center_2 = \max(PW)$, $center_3 = (center_1 + center_2)/2$.

Step 2: The new centroids $center'_1, center'_2$, and $center'_1$ are found by iterating the following two loops until convergence, which is obtained when the centroids are no longer changed.

  (2.1) For each pixel in *PW*, compute its membership in clusters by choosing the nearest centroid.

  (2.2) For each centroid, recompute its location according to members.

Step 3: The pixels associated with $center'_1$ are marked as white pixels, and the rest of the pixels are marked as black pixels. A group of white pixels forms an image patch, which will be removed if the following constraints are satisfied:

  (3.1) The width of the image patch is less than the height of the image patch.

  (3.2) The number of pixels in the image patch is less than 10.

Step 4: Calculate the numbers of image patches n, and obtain the center of each image patch $c_1(x, y), c_2(x, y), \ldots, c_n(x, y)$. Sort the center points as $c'_1(x,y), c'_2(x,y), \ldots, c'_n(x,y)$, and $center'_1 \leq \cdots \leq c'_n(y)$. One can be selected by n through the following processes.

(4.1) If $n = 1$ and $center_1 < \theta$, $center_1 = center_1 + 5$, go to Step3.
(4.2) If $n = 1$ and $center_1 \geqslant \theta$, $e(x, y) = c'_1(x, y)$.
(4.3) If $n \geqslant 2$, $e(x, y) = c'_2(x, y)$.

Step 5: Cut from the grayscale image, and obtain the eye window *EW* which is a rectangle (31 × 13 pixels), centered at $e(x, y)$.

In the foregoing algorithm, θ is a threshold which controls the value range of $center_1$'. We set $\theta = center'_1 + 10$. There are two reasons for the threshold setting. First, the eyebrow is darker than the eye in a few cases; hence, in such cases, the eye window will not be found by cluster analysis. Second, in rare cases, the eyebrow does not appear in the pretreatment window; hence, the image patch may be unique.

Here, we assume that $e_l(x, y)$ is the left eye position, and $e_r(x, y)$ is the right eye position; n_l is the number of image patches in the left pretreatment window, and n_r is the number of image patches in the right pretreatment window. If the angle between the horizontal direction and the line through $e_l(x, y)$, $e_r(x, y)$ is greater than 30°, the image patch whose center is $e_l(x, y)(n_l > n_r)$ or $e_r(x, y)(n_r > n_l)$ will be moved. The new eye position is recalculated based on Step 4 of Algorithm 2. The process of finding eye windows is shown in Fig. 5.

2.6. Pupil center localization

From the eye model shown in Fig. 6, it is clear that in an eye window, the white region around the eyeball has the strongest intensity, and the pupil has the weakest intensity. While the eye window has the size of 31 × 13 pixels, the pupil is generally larger by 20–50 pixels. Therefore, after obtaining the eye window *EW***,** the

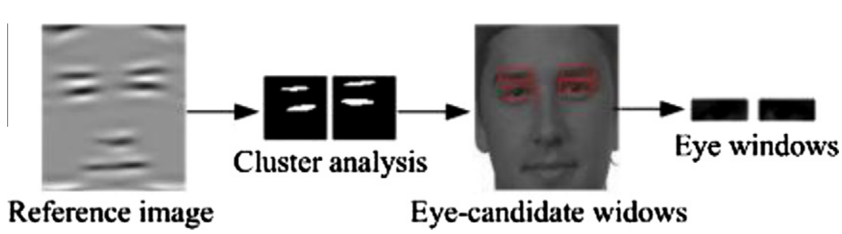

Fig. 5. The process of finding eye windows.

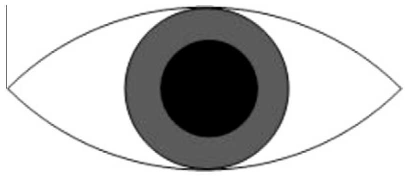

Fig. 6. Eye model used in this paper.

position of the central point of the pupil can be determined through Algorithm 3.

Algorithm 3. The pupil center localization algorithm.

Input: Eye window *EW*
Output: Pupil position $p(x, y)$
Step 1: Obtain image $NI_{(3,3)}$ by using the operator to every pixel of *EW* by adding the 3×3-neighborhood of each pixel replacing the center pixel value. The image $NI_{(5,5)}$ is obtained using the operator to every pixel of $NI_{(3,3)}$ by adding the 5×5-neighborhood of each pixel replacing the center pixel value.
Step 2: Find the minimum value $p_{\min}(x, y)$ in $NI_{(5,5)}$. The pupil position $p(x, y)$ is equal to $p_{\min}(x, y)$.

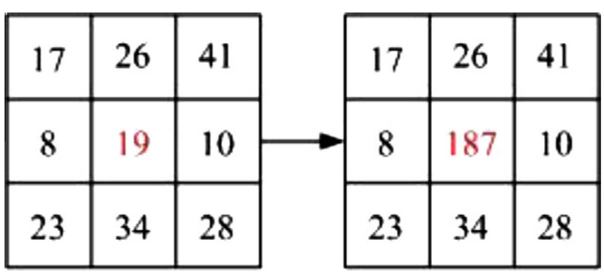

Fig. 7. The 3×3 neighborhood operator.

In the foregoing algorithm, the 3×3-neighborhood operator is used to reduce the influence of the pupillary light reflex, and the 5×5-neighborhood operator is used to locate the pupil center. The 3×3-neighborhood operator is illustrated in Fig. 7.

3. Experiments

3.1. Datasets

The proposed method is tested on two face datasets. One is a subset of the color FERET database (Phillips et al., 1998) and the other is a subset of the LFW database (Huang et al., 2007b).

The color FERET database contains a total of 11,338 face images with a resolution of 512×768 pixels. We select 2950 frontal view face images grouped into four sessions (fa 994 images, fb 992 images, dup1 736 images and dup2 228 images), which are defined in the color FERET database. These images form the first dataset. Images in this dataset contain variations in illumination, skin color and facial expression. Some images from this dataset are shown in Fig. 8(a).

The LFW database consists of 13,233 face images, which are collected from the web, with a resolution of 250×250 pixels. The images are with a large variety of lighting, expression, background, camera quality, occlusion and image noise. In contrast to the color FERET database, the LFW database stresses “real world” conditions. In other words, eye detection in the LFW database is more difficult. From the LFW database, we select 1200 images (with different facial expressions and lighting conditions, without partial occlusion) corresponding to 954 different persons. These images form the second dataset. Some images from this dataset are shown in Fig. 8(b).

3.2. Settings

We use the Viola–Jones face detector from the OpenCV library to achieve face detection. In the first dataset, 15 images are not located; in the second dataset, 8 images are not located. Hence, the

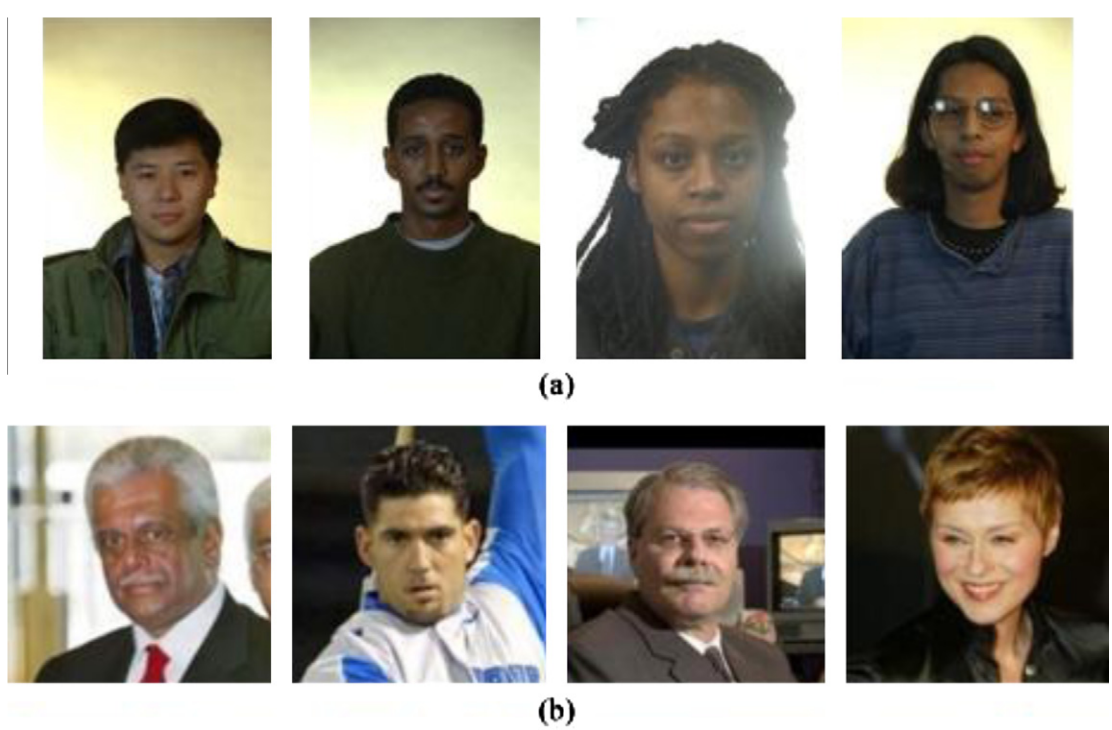

Fig. 8. Some images in the experimental datasets. (a) Sample images from the color FERET database and (b) sample images from the LFW database.

final datasets consists of 2935 face images from the color FERET database (fa 989 images, fb 989 images, dup1 731 images and dup2 226 images) and 1192 face images from the LFW database.

In order to assess the precision of eye detection, we need to adopt a suitable measure criterion. Usually, distances between the detected and the expected eye positions are adopted as measures of accuracy. One relevant factor in measuring results is the representation of distances. If the absolute distances in pixels are used, this measure criterion will rely on image resolution. In this study, we adopt the criterion described in (Jesorsky et al., 2001). The criterion is a relative *error* measure, which is resolution independent. Let d_l and d_r be the Euclidean distances between the eye centers in the ground truth and the eye centers detected by the method. The relative *error* of eye detection is then defined as:

$$error = \frac{\max(d_l, d_r)}{|d_l - d_r|} \tag{6}$$

If $error < 0.25$, the detection result is considered to be correct. Because $|d_l - d_r|$ roughly equals twice the eye width, the criterion $error < 0.25$ means that the bigger one of e_l and e_r is less than half the eye width. For face normalization, this criterion may not be very suitable. In order to further analyze the results, we compare the detection rate when the value of *error* gradually decreases from 0.25 to 0.05 with 0.05 as the interval.

Table 1
Detection results on two datasets.

Error	Subset of color FERET database				Subset of LFW database (%)
	fa (%)	fb (%)	dupl (%)	dup2 (%)	
0.25	97.8	97.6	95.9	98.2	96.1
0.2	97.5	97.1	95.1	98.2	95.6
0.15	97.0	96.7	93.6	96.9	94.3
0.1	91.8	92.1	89.6	92.5	90.6
0.05	80.6	79.2	77.6	82.3	75.1

3.3. Results and discussion

The eye detection results on two datasets are shown in Table 1. From the results, it can be seen that the proposed method can accurately locate eyes in most cases. In particular, the face images in the LFW database are with significant variability in pose, lighting and expression. Experiments on the LFW dataset show that 96.1% ($error < 0.25$) face images are correctly detected. The encouraging results on different conditions of the proposed method can be comprehensively explained. Image enhancement and Gabor transformation help reduce the influence of environmental changes, and intensity information is robust against complicated appearance changes. Usually, false detection is caused by strong interference in the eye region (e.g., the reflection of the spectacles).

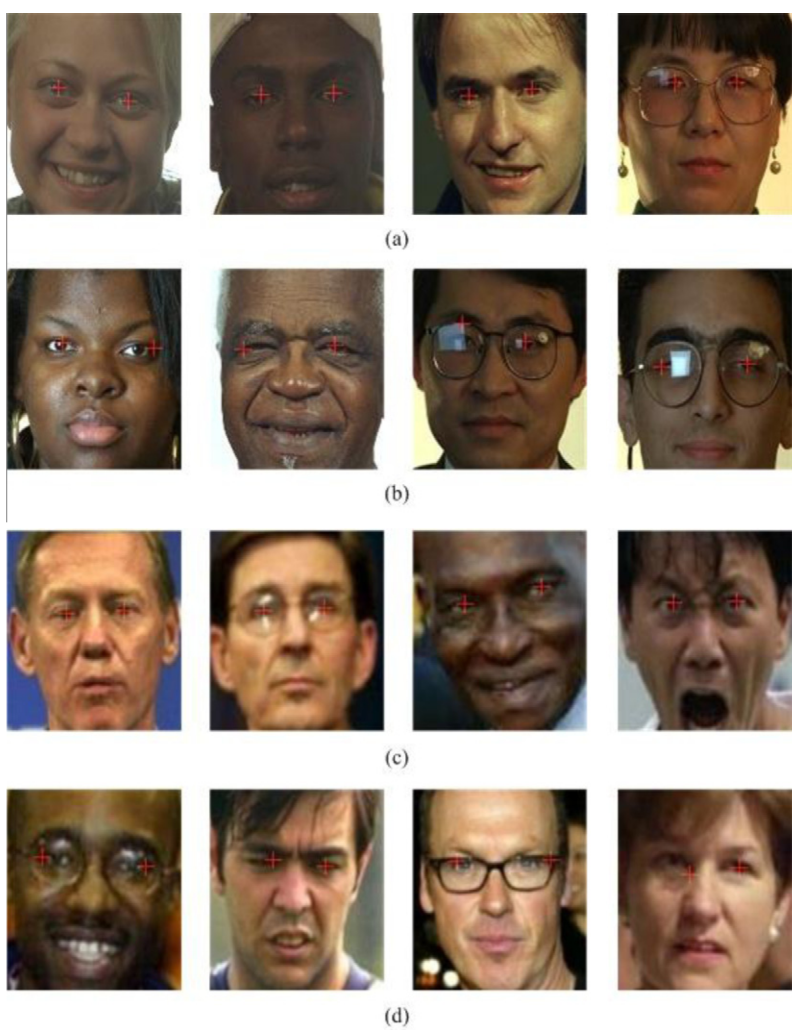

Fig. 9. Some detection results in the experimental datasets. (a) Sample eyes in color FERET that are correctly detected, (b) sample eyes in color FERET that are wrongly detected, (c) sample eyes in LFW that are correctly detected and (d) sample eyes in LFW that are wrongly detected.

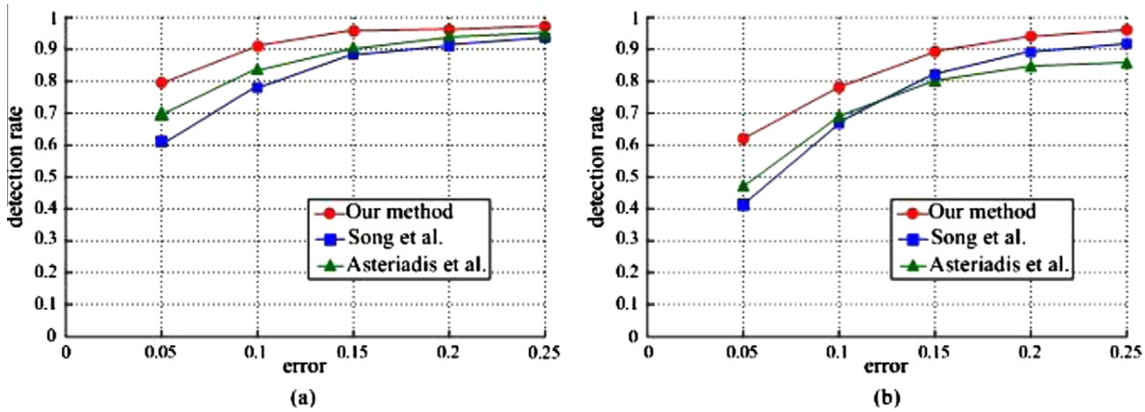

Fig. 10. Comparison of detection rates of different methods on two datasets. (a) FERET dataset and (b) LFW dataset.

In the experiments, the dup1 dataset contains 200 faces with spectacles. Among these, our method is unable to correctly locate the eyes in 26 images. This can be attributed to the fact that the reflection of spectacles changes the intensity values of pixels around the eyes, which can then lead to an erroneous analysis. Some detection results are shown in Fig. 9.

We make a comparison of detection rate with Song et al.'s (2006) and Asteriadis et al.'s (2009) methods using two face datasets. Results clearly show that our method performs better than the two other methods (Fig. 10). Especially in the LFW dataset, our method obtains a higher detection rate and detection accuracy ($0.5 < error < 0.1$). Analysis of the detection process shows that most of the incorrect detections by the two other methods are caused by image quality or lighting changes. More specifically, the method of Song et al. mainly uses reflected light dots from the eye region to locate the eye centers. If there is no light dot in the eye region, their method is unable to obtain satisfactory results. The method of Asteriadis et al. uses geometrical information to extract eye segments. When face images have much poorer quality or are under low lighting conditions, this method cannot efficiently obtain eye segments.

4. Conclusion and future work

An eye detection method based on intensity information is described in this paper. It is grounded on two important observations: the eyes are darker than other parts of the face and the pupils are darker than the surrounding eyeballs. Since these high contrast properties, our method can handle complicated variations in pose, expression and lighting. Experimental results show ($error < 0.25$) that our method achieves a detection rate of 97.2% for 2935 color FERET face images and 96.1% for 1192 LFW face images.

Because our method mainly uses intensity information, its performance is limited by the occurrences of partial occlusion. To overcome this limitation, future work should concentrate on using more comprehensive cues to predict the positions of occluded eyes. In addition, some local features have been proven to have similar properties or can help stabilize the features with Gabor filters, such as local binary decision (James and Dimitrijev, 2008) and local binary pattern (Tan and Triggs, 2007). Using these local features to improve the performance of Gabor filters is reserved for future work.

Acknowledgements

The authors would like to thank the anonymous reviewers for his detailed constructive comments, which have helped improve the paper significantly. Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counter drug Technology Development Program Office. The LFW database is provided by Huang et al. at University of Massachusetts. This work was supported by the National Science Foundation of China (No. 60663010) and the Scientific Research Fund of Yunnan Provincial Education Department (No. 09Y0404).

References

Asteriadis, S., Nikolaidis, N., Pitas, L., 2009. Facial feature detection using distance vector fields. Pattern Recognition 42 (7), 1388–1398.

Brubaker, S., Wu, J., Sun, J., Mullin, M., Rehg, J., 2008. On the design of cascades of boosted ensembles for face detection. Internat. J. Comput. Vision 77 (1–3), 65–86.

Campadelli, P., Lanzarotti, R., Lipori, G., 2006. Precise eye localization through a general-to-specific model definition. In: Proc. British Machine Vision Conf., pp. 187–197.

D'Orazio, T., Leo, M., Cicirelli, G., Distante, A., 2004. An algorithm for real time eye detection in face images. In: Proc. IEEE Conf. on Pattern Recognition, pp. 278–281.

Dollar, P., Tu, Z., Tao, H., Belongie, S., 2007. Feature mining for image classification. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, p. 164.

Haro, A., Flickner, M., Essa, I., 2000. Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 163–168.

Huang, C., Ai, H., Li, Y., Lao, S., 2007a. High-performance rotation invariant multiview face detection. IEEE Trans. Pattern Anal. Machine Intell. 29 (4), 671–686.

Huang, G., Ramesh, M., Berg, T., Learned-Miller, E., 2007b. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report, University of Massachusetts, Amherst.

James, A., Dimitrijev, S., 2008. Face recognition using local binary decisions. Signal Process. Lett. 15, 821–824.

Jesorsky, O., Kirchberg, K., Frischholz, R., 2001. Robust face detection using the Hausdorff distance. In: Bigun, J., Smeraldi, F. (Eds.), Lecture Notes in Computer Science, vol. 2091. Springer, Berlin, pp. 90–95.

Ji, Q., Wechsler, H., Duchowski, A., Flickner, M., 2005. Eye detection and tracking: Special issue. Computer Image and Vision Understanding 98 (1), 1–3.

Lam, K., Yan, H., 1996. Locating and extracting the eye in human face images. Pattern Recognition 29 (5), 771–779.

Lee, T., 1996. Image representation using 2D Gabor wavelets. IEEE Trans. Pattern Anal. Machine Intell. 18 (10), 959–971.

Li, S., Zhang, Z., 2004. Float boost learning and statistical face detection. IEEE Trans. Pattern Anal. Machine Intell. 26 (9), 1112–1123.

Phillips, P., Wechsler, H., Huang, J., Rauss, P., 1998. The FERET database and evaluation procedure for face recognition algorithms. Image Vision Comput. 16 (5), 295–306.

Phillips, P., Flynn, P., Scruggs, T., Bowyer, K., 2005. Overview of the face recognition grand challenge. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 15–24.

Rodriguez, Y., Cardinaux, F., Bengio, S., Marithoz, J., 2006. Measuring the performance of face localization systems. Image Vision Comput. 24 (8), 882–893.

Shan, S., Chang, Y., Gao, W., 2004. Curse of mis-alignment in face recognition: Problem and a novel mis-alignment learning solution. In: Proc. IEEE Conf. on Automatic Face and Gesture Recognition, pp. 314–320.

Song, J., Chi, Z., Liu, J., 2006. A robust eye detection method using combined binary edge and intensity information. Pattern Recognition 39 (6), 1110–1125.

Tan, X., Triggs, B., 2007. Fusing Gabor and LBP feature sets for kernel-based face recognition. In: Zhou, S. et al. (Eds.), Lecture Notes in Computer Science, vol. 4778. Springer, Berlin, pp. 235–249.

Tan, X., Song, F., Zhou, Z., Chen, S., 2009. Enhanced pictorial structures for precise eye localization under uncontrolled conditions. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition.

Tang, X., Ou, Z., Su, T., Sun, H., Zhao, P., 2005. Robust precise eye location by Adaboost and SVM techniques. In: Wang, J., Liao, X., Yi, Z. (Eds.), Lecture Notes in Computer Science, vol. 3497. Springer, Berlin, pp. 93–98.

Viola, P., Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 511–518.

Wang, P., Green, M., Ji, Q., Waymanm, J., 2005. Automatic eye detection and its validation. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 164–164.

Wu, J., Brubaker, S., Mullin, M., Rehg, J., 2008. Fast asymmetric learning for cascade face detection. IEEE Trans. Pattern Anal. Machine Intell. 30 (3), 369–382.

Yang, M., Kriegman, D., Ahuja, N., 2002. Detecting faces in images: A survey. IEEE Trans. Pattern Anal. Machine Intell. 24 (1), 34–58.

Zhao, W., Chellappa, R., Phillips, P., Rosenfeld, A., 2003. Face recognition: A literature survey. ACM Comput. Surv. 35 (4), 399–458.

Zhou, Z., Geng, X., 2004. Projection functions for eye detection. Pattern Recognition 37 (5), 1049–1056.

Zhu, Z., Ji, Q., Fujimura, K., 2002. Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination. In: Proc. IEEE Conf. on Pattern Recognition, pp. 318–321.