Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 26 (2005) 2252–2261
Pattern Recognition Letters
www.elsevier.com/locate/patrec

A robust method for eye features extraction on color image

Zhonglong Zheng [a,b,*], Jie Yang [a], Limin Yang [a]

[a] *Department of Automation, Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, P.O. Box 280, No. 1954 Huashang Road, Shanghai 200030, China*
[b] *Institute of Information Science and Engineering, Zhejiang Normal University, Jinhua 321004, China*

Received 15 May 2004; received in revised form 11 January 2005
Available online 10 August 2005

Communicated by I.K. Sethi

Abstract

This paper presents a robust algorithm to extract eye features, including pupil center and radius, eye corners and eyelid contours, from frontal face images. These features are very useful cues for applications like face recognition, facial expression recognition and 3D face modeling from 2D images. With the assumption that the rough eye window is known, the proposed method integrates color information, Gabor features and the mutual localization relationship between different features. Different from other methods focusing on gray scale, our method detects and estimates pupil center in H channel of HSV color space, and then pupil radius is estimated and refined. For eye-corner localization, a Gabor eye-corner filter is constructed to detect the corner point. It is more robust than projection methods or edge detection methods. Based on the previous results, eyelid curves are simply fitted by spline function. The proposed algorithm has been tested on our SJTU database which contains more than 700 pictures. The experiment results show very good robustness and accuracy.

Keywords: Eye feature detection; Principal component analysis; Gabor feature space; Eye-corner filter

1. Introduction

It is essential to detect facial features and their contours when creating the automatic face analysis and processing systems. The results of detection could be used for the model-based image coding (Zheng et al., 1999), the recognition of a face and facial expression (Brunelli and Poggio,

* Corresponding author. Address: Department of Automation, Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, P.O. Box 280, No. 1954 Huashang Road, Shanghai 200030, China. Fax: +86 21 62933739.
E-mail address: zhonglong@sjtu.edu.cn (Z. Zheng).

0167-8655/$ - see front matter
doi:10.1016/j.patrec.2005.03.033

1992), 3D face recognition (Blanz and Vetter, 2003) and so on.

Eye feature detection plays an important role in facial feature detection because eye features are the very salient facial features. As a result, eye feature detection usually is the first step in a face recognition system. The eye features include pupil centers and their radiuses, eye corners and eyelid contours. Among these features, pupil centers and radiuses are the easy features to be detected and estimated. While the precise corner location is crucial in 3D face recognition (Blanz and Vetter, 2003) on the one hand, and determines the approximate eyelid contour location on the other hand. As we know, deformable contour model is a well-known method to extract object contour in computer vision. Lam and Yan (1996) and Yin and Basu (1999) made an attempt to utilize such algorithm in eye contour extraction. Because of the limitations of deformable contour model under conditions in (Vezhnevets and Degtiareva, 2003), it is not optimal and stable in eye contour extraction. The main reason is that deformable models need careful formulation of the energy term and close model initialization in order to avoid unwanted contour results.

To overcome the limitations of deformable models, researchers pay more attention on several landmark points of eyes other than extracting the complete continuous eye contour. Then the eye contour can be fitted by mathematical functions (Feng and Yuen, 2001; Goto et al., 2002; Vezhnevets and Degtiareva, 2003). This results in better robustness, but less accuracy. For this reason, this paper also adopts such strategy.

The paper is organized as follows. Based on the assumption that the rough eye window has been detected, Section 2 describes our method in detail: pupil center and radius, eye corners and eyelid contours are precisely detected and estimated step by step. The experimental results are listed in Section 3. Section 4 is the conclusion of the paper.

2. Extraction of eye features

Many eye feature detection algorithms have been developed in the last decade. Most of them work on the gray images. Since color information could provide extra cues for detection and recognition, why not use it? In this paper, the input image for the proposed method is color image and each of them contains a single eye. It is assumed in this paper that the approximate location of eye had been known from previous rough eye detection step. And the eye image is cropped to a uniform scale.

The proposed method can be divided into three steps.

- Pupil center detection and radius estimation.
- Eye-corner localization.
- Eyelid contour fitting.

2.1. Pupil center detection and radius estimation

In this subsection, we mainly deal with detecting the pupil center and estimating its radius. Pupil center is located in H channel of HSV color space. After that the radius of the pupil is estimated and then refined by a simple search algorithm.

2.1.1. Pupil center detection

Pupil center detection is the first step of the algorithm. It is worth noting that pupil center detection is performed in H channel of HSV color space. The HSV color space (hue, saturation, value) is often used by people because it is more consistent with how people experience color better than the RGB color space is. Fig. 1 shows the original image and corresponding H channel image. It is very interesting to notice that in H channel the pupil is the brightest region comparing with its neighborhoods. Because of pupil's physiological property and its response to light, this is robust under different illuminations. So it is easy to locate the brightest area just by simple integral projection method. The result of locating pupil center is shown in Fig. 2.

The location of the pupil center can be derived from the vertical integral projection and horizontal integral projection in Fig. 2. The peak points of the projection curves are corresponding to the coordinates of the pupil center. The result is pretty good (the pink cross in the last eye image of Fig. 2

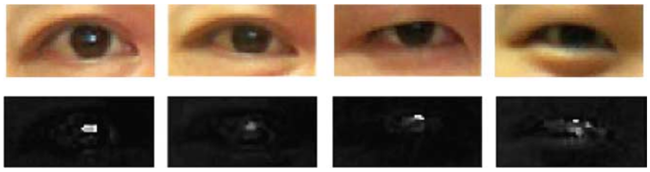


Fig. 1. Original image and corresponding H channel image.

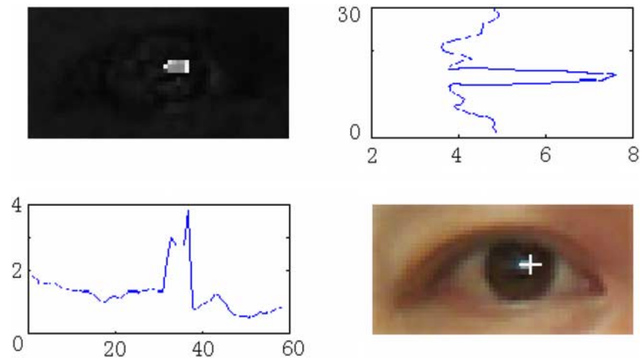


Fig. 2. Location of pupil center by integral projection.

denotes the coordinates of pupil center). This method at least has two advantages: simple and more robust. In the process of finding the pupil center, we only need get the maximum value of the projection curve and its corresponding coordinate. We need not preset a threshold to get the location because the maximum value is the target to find. As we all know a threshold in an algorithm usually leads to poor performance because it is sensitive to the selection of sample sets.

However, the location of pupil center is not very precise. We still need to refine it.

2.1.2. Refining process and radius estimation

From the above method, the approximate coordinates of pupil center is known. In fact, it can be refined by searching the border of the eye ball. Obviously, the shape of the eye ball is almost circular and is darker than the background. So we can search a circle in which the region is darker than the background. Vezhnevets and Degtiareva (2003) adopted the algorithm (Ahlberg, 1999):

$$f_\Theta(x,y,r) = \int_{\theta\in\Theta} I(x + r\cos\theta, y + r\sin\theta)\,\mathrm{d}\theta \qquad (1)$$

and made a slight change to estimate the radius of the eye ball. The most likely radius will be the value with large $\frac{\mathrm{d}}{\mathrm{d}r}f_\Theta$. We realized this method and found that it is not very robust. The reason is that we should set a threshold for $\frac{\mathrm{d}}{\mathrm{d}r}f_\Theta$ in order to determine the radius. While the threshold for $\frac{\mathrm{d}}{\mathrm{d}r}f_\Theta$ may be different from one eye image to another. And even for the same eye image under variant illumination, the threshold can not keep the same. So it is hard to determine an optimal global threshold for the dataset.

In this paper, we describe a search strategy based on gray level to estimate the pupil radius. Once the approximate location of pupil center is given, a circle with fixed radius is established. It can be called the location initialization of the eye ball. Then the following searching strategy is divided into two steps. The first step is circle shift. The operation of this step is to shift the circle in a given neighborhood to obtain a location on which the mean gray level of pixels in the circle is the lowest. Before calculating the mean gray value of pixels in the circle, a middle filter is applied to the eye image to eliminate the highlight of pupil. The second step is expansion or shrinking

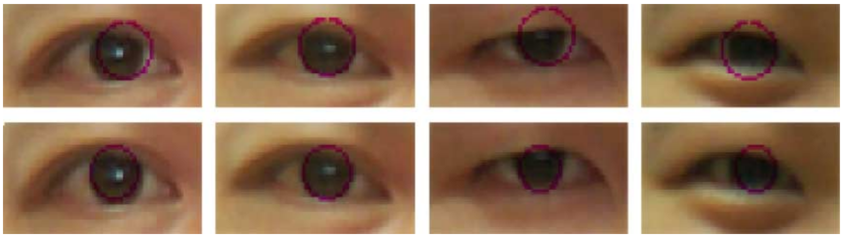


Fig. 3. Location of eyeball before and after the search strategy.

of the circle. After the first step, the pupil center is nearly at the center of the circle. The imperfection is that the radius of the circle may not be accurate. So it is necessary to expand or shrink the circle for different eyeball sizes. The rule of expansion or shrinking is also to minimize the mean gray value of pixels in the circle. In Fig. 3, the first row is the initial location of eye ball after pupil center detection. The second row is the final location of eye ball after our search strategy. More results are listed in Fig. 10. The experimental results show the accuracy and robustness of the search strategy.

2.2. Eye-corner detection

The eye corner is the key cue which determines the location and orientation of the eye. One way to detect eye corner is based on gradient information (Yulle et al., 1992). If the image is free of noise, the method based on gradient information is accurate and simple. In fact, the real image is more or less contaminated by noise. This therefore leads to a huge amount of spurious edges and possible absence or discontinuity of significant edges. Under such condition, the precise location of eye corner is hard to find or absolutely lost. Variance projection (VPF) method (Feng and Yuen, 1998) improved the traditional integral projection function (IPF)

$$\sigma_v^2(x) = \frac{1}{y_2 - y_1} \sum_{y=y_1}^{y_2} [I(x, y_i) - V_{\mathrm{mean}}(x)]^2 \tag{2}$$

$$\sigma_h^2(y) = \frac{1}{x_2 - x_1} \sum_{x=x_1}^{x_2} [I(x_i, y) - H_{\mathrm{mean}}(y)]^2 \tag{3}$$

where V_{mean} and H_{mean} are the mean values of traditional integral projection function. The comparison of IPF and VPF for a neat image is shown in Fig. 4. It can be seen that the VPF directly reflects the variation in vertical direction. However, we find that VPF shows low reliability in general case because of so many changes along a vertical column or a horizontal row in a real eye image. Hybrid projection function (HPF) which combined the IPF and VPF together (Zhou and Geng, 2004) was used to detect eye corner and border.

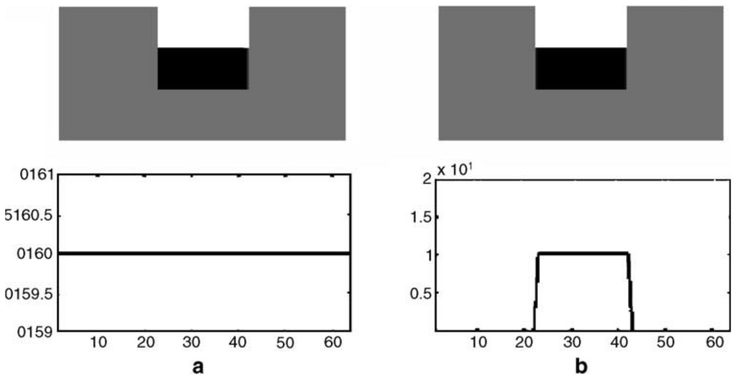


Fig. 4. (a) IPF and (b) VPF for a neat image.

Still HPF suffers from its weakness. Eye corner was detected by looking for brightness valley points of luminance values instead of edge points (Vezhnevets and Degtiareva, 2003). Before searching the brightness valley points, a low pass filter is applied to the image for noise alleviation. So if the pre-processing step is not perfect, the searching process will get into embarrassment and the searching result will be doubtful.

In this paper, we propose an eye-corner filter using Gabor feature space for eye-corner detection. Gabor wavelet is a powerful tool in image feature extraction (Huang and Wechsler, 1999; Lee, 1996; Porat and Zeevi, 1988; Ville et al., 2004). The Gabor wavelet can be defined as follows (Ladws et al., 1993):

$$\psi_{\mu,v}(z) = \frac{\|k_{\mu,v}\|^2}{\sigma^2} e^{(-\|k_{\mu,v}\|^2\|z\|^2/2\sigma^2)}[e^{ik_{\mu,v}z} - e^{-\sigma^2/2}] \tag{4}$$

where μ and v define the orientation and scale of the Gabor kernels, and the wave vector $k_{\mu,v}$ is defined as

$$k_{\mu,v} = k_v e^{i\phi_\mu} \tag{5}$$

where $k_v = k_{\max}/f^v$ and $\phi_\mu = \pi\mu/8$. $k_{\max}$ is the maximum frequency and f is the spacing factor between kernels in the frequency domain.

Once given these parameters, the Gabor feature representation of an image $I(z)$ is

$$G_{\mu,v}(z) = I(z) * \psi_{\mu,v}(z) \tag{6}$$

where $z = (x, y)$ and $*$ is the convolution operator. Fig. 5 is the Gabor feature representation of the first eye image in Fig. 1 at five different scales and 8 orientations.

Eye corner is one of the most salient feature points in an eye image in the sense of structure information. It is obvious that eye-corner point is the intersection of the two eyelid curves and the end point of eyelid curves. Such topological structure is exclusive in an eye image. Furthermore, the structure is highlighted in Gabor feature space at certain scale and orientation. Motivated by these properties, the paper describes an eye-corner filter based on Gabor feature space. The filter is constructed by coefficients at some scales and orientations in Gabor feature space (Ville et al., 2004). For example, the filter for eye corner near the bridge of the nose is denoted by a 5×5 mask. The center of the mask is corresponding to the eye corner which is manually located. The other elements of the mask are corresponding to the neighbors of the eye corner. The value of the elements in the mask is determined by its Gabor representation. Let $I(x, y)$ be an eye image, $C(x, y)$ be the 5×5 patch image centered at eye corner and $G_{\mu,v}(z)$ be the Gabor representation of $C(x, y)$

$$G = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,8} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,8} \\ \cdots & \cdots & \cdots & \cdots \\ g_{5,1} & g_{5,2} & \cdots & g_{5,8} \end{pmatrix} \tag{7}$$

at five scales and eight orientations. For illumination invariance, G can be normalized as

$$G' = \frac{G}{\sqrt{\sum_{i,j}|g_{i,j}|^2}} = \begin{pmatrix} g'_{1,1} & g'_{1,2} & \cdots & g'_{1,8} \\ g'_{2,1} & g'_{2,2} & \cdots & g'_{2,8} \\ \cdots & \cdots & \cdots & \cdots \\ g'_{5,1} & g'_{5,2} & \cdots & g'_{5,8} \end{pmatrix} \tag{8}$$

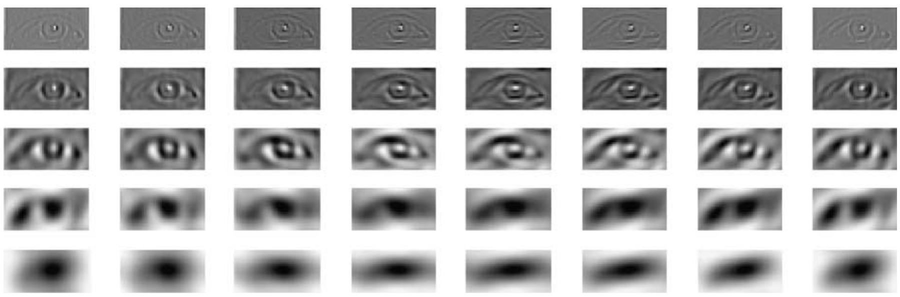


Fig. 5. Gabor feature representation of an eye image.

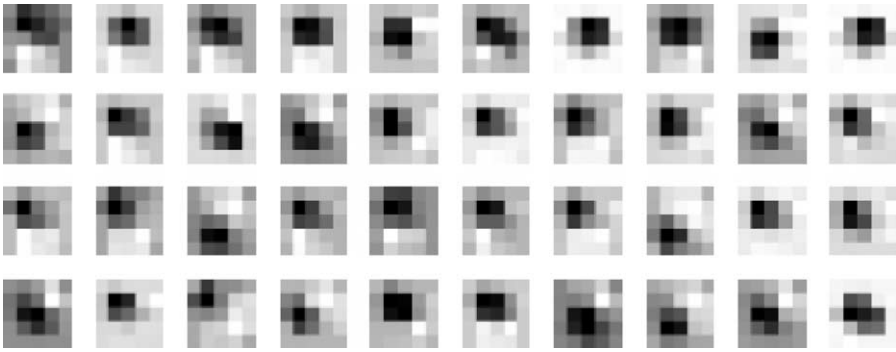


Fig. 6. Gabor representation of eye-corner images.

Let f be the mean value of the first two rows coefficients of G'

$$f = \frac{1}{n}\sum_{i,j} g_{i,j} \tag{9}$$

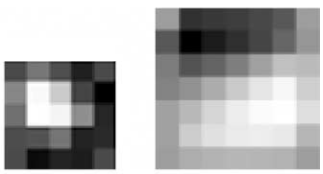
where $n = 16$, $i = 1,2$ and $j = 1,2,\ldots,8$. Some Gabor representations, f of $C(x,y)$, are shown in Fig. 6.

To construct an eye corner (near the bridge of nose) filter, we select 80 corner images described by $C(x,y)$ above. The final eye corner (near the bridge of nose) filter is constructed by calculating the average of f

$$F_{5\times 5} = \frac{1}{N}\sum_{i=1}^{N} f_i \tag{10}$$

where $N = 80$. $F_{5\times 5}$ is shown in Fig. 7(a).

It is obvious that the structure of another eye corner is a little more complex than the one near the bridge of nose. So the size of this filter is 7×7 in order to contain more information of the corner structure. $F_{7\times 7}$ is shown in Fig. 7(b). The color of Fig. 7 is in reverse for the reason of illustration.



a b

Fig. 7. Gabor eye-corner filters: (a) $F_{5\times 5}$ filter, (b) $F_{7\times 7}$ filter.

To detect eye corner in an eye image, one need to convolve the Gabor representation of the eye image (the selection of scale and orientation is the same as Gabor representation of eye-corner images, and normalized by Eqs. (8) and (9)) with certain filter. In order to reduce the computational cost, the convolution operation could be calculated on part of the image because the center and radius of the eyeball is given by previous algorithm. The corner detection results are shown in Fig. 8. From Fig. 8, we can see that the corner point in (b) and (c), denoted by a red[1] circle, is easy to find. Experimental results demonstrate the robustness and accuracy of this corner filter.

2.3. Eyelid curve fitting

In this step, we only need to find some medial points to fit the curve. It is advisable to detect the points along the column through the center of pupil in the eye image, denoted by red circle in Fig. 9(a). We use a 7×7 window to filter the eye image on gray level: the minimum value in the window is subtracted from the other pixels in order to make the edge more salient. By searching the peak points of column line in Fig. 9(a), we find the medial points. Fitting the eyelid curve along corner points and medial points by spline function, we depict the eyelid curve in Fig. 9(b).

[1] For interpretation of color in Fig. 8, the reader is referred to the web version of this article.

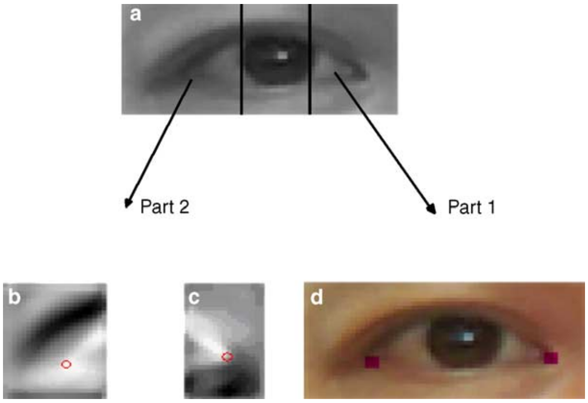


Fig. 8. Corner detection: (a) an eye image on gray level, (b) convolution of Gabor representation of part 2 in (a) with $F_{7\times7}$, (c) convolution of Gabor representation of part 1 in (a) with $F_{5\times5}$, (d) final corner detection result.

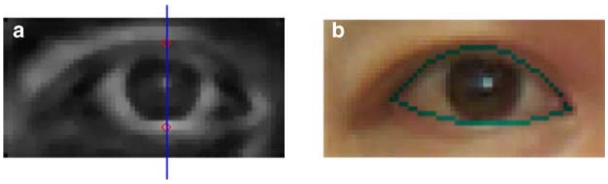


Fig. 9. Medial point detection and eyelid curve fitting.

3. Experimental results

After obtaining the eye-corner filter $F_{5\times5}$ and $F_{7\times7}$, we apply them to our test dataset which contains more than 700 images. The images are taken under different lighting conditions and some have a slight rotation. Fig. 10 shows part of the experimental results. Note that $F_{5\times5}$ and $F_{7\times7}$ above are constructed by right eye image samples. So when applied to detecting eye corners of left eyes, both of them should be mirrored.

To evaluate the performance of our algorithm, we carried out the eye features detection experiment with the whole homebrew database. The detection performance is evaluated based on subjective judgment since no ground truth is available. In other words, it is hard to say which pixel is corresponding to the exact eye corner. The situation is the same for pupil center detection. We have also attempted to perform some quantitative evaluations about the performance of the algorithm. We randomly selected 400 images from our database and hand-localized pupil centers and eye corners on them. They serve as ground truth. A circularity confidence interval, centered at the hand-localized pixel with 5 pixel radius, is defined. Let $H(i,j)$ denote hand-localized pixel and $E(i,j)$ the detected pixel. The distance Dis of H and E is defined as $\|H-E\|_2$. Then the accuracy of the algorithm is defined as

$$A=\begin{cases}\left(1-\dfrac{\mathrm{Dis}}{5}\times 0.5\right)\times 100\%, & \mathrm{Dis}<=5\\ 0, & \mathrm{Dis}>5\end{cases} \tag{11}$$

The satisfactory factor is set to 0.5 in Eq. (11). That is to say, the accuracy is 50% if the detected pixel position is on the boundary of the confidence interval. If it is out of the confidence interval, the accuracy is set to 0. Meanwhile, the accuracy is 100% if the detected pixel and hand-localized one are at the same position. The detection perfor-

Fig. 10. Part of experimental results.

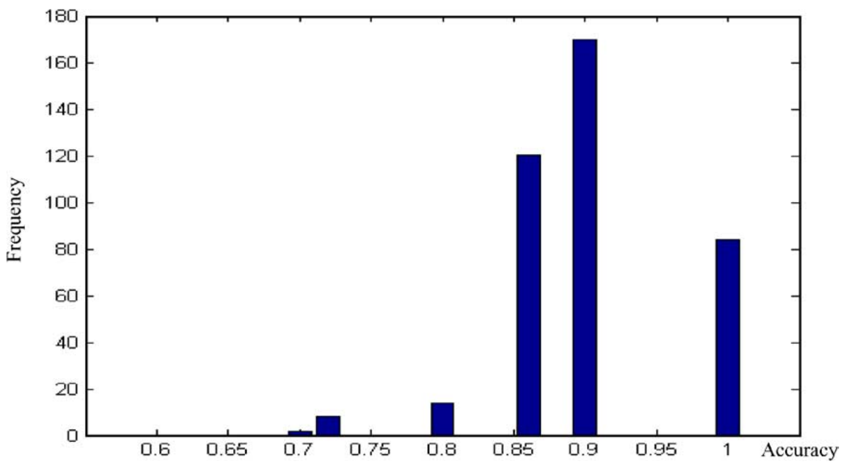


Fig. 11. Detection performance of pupil center.

mances of pupil center and eye corners are illustrated in Figs. 11–13.

The detection accuracy of pupil centers and eye corners is pretty satisfactory. In most cases, the

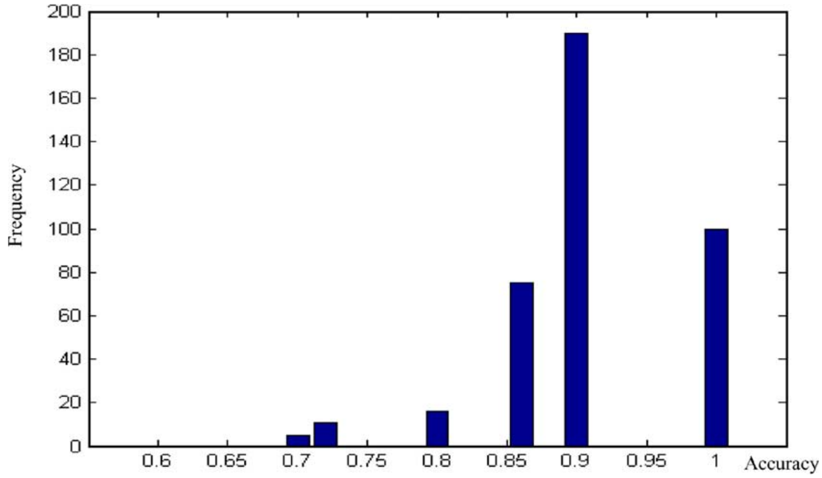


Fig. 12. Detection performance of eye corners near the bridge of nose.

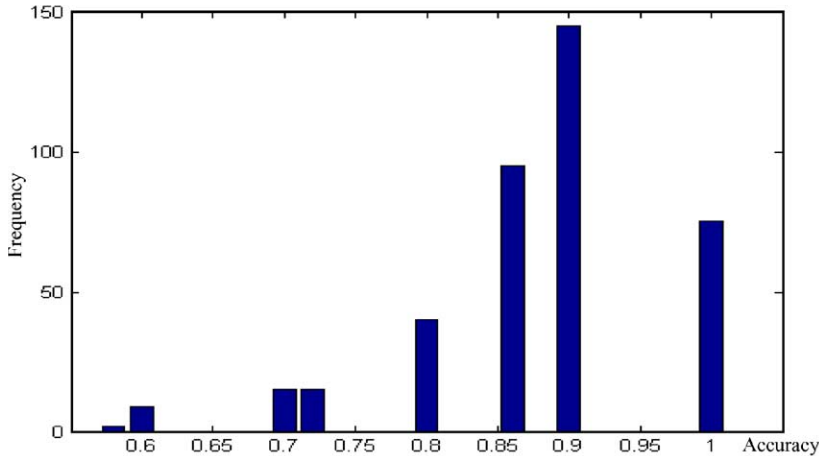


Fig. 13. Detection performance of eye corners far from the bridge of nose.

position of detected pixel is within a 3-pixel confidence interval of the "ground truth". Relatively speaking, the detection performance of eye corners far from the bridge of nose is inferior to the one near the bridge of nose. The reason is that the structure information of eye corners near the bridge of nose is more salient.

The algorithm process is step by step, that is to say, the previous detection result is essential to the next step. So if the first step is not accurate, the following detection may be failed. Fortunately, pupil center detection and radius estimation achieve high accuracy. Therefore, the proposed method demonstrates good performance. When the image resolution is too small or the eye is almost closed, the method will lose its robustness and accuracy.

4. Conclusion

A robust and accurate algorithm for eye feature extraction is reported in the paper. First, the center of pupil is detected in H channel of HSV color space, and the radius of eyeball is estimated and refined. Second, based on the result obtained from first step, eye corner is detected by the proposed

Gabor eye-corner filter constructed in Gabor feature space. Finally, based on the results obtained from the previous steps, eyelid curve is fitted by spline function. The proposed methods have shown good performance in terms of robustness and accuracy, while being simple in implementation.

Experimental results show that the following situation may lead to inaccuracy.

- The eye is closed or almost closed.
- The eye is covered by something opaque.
- The image resolution is too low.

To overcome the limitations above, the future work will be concentrated on improving the accuracy and robustness of the algorithm.

Acknowledgment

The authors wish to acknowledge that this work is supported by Fundamental Project of Shanghai under grant number 03DZ14015. The authors would like to thank Mr. Yitan Zhu for useful discussion.

References

Ahlberg, J., 1999. A system for face localization and facial feature extraction. Tech. Rep. LITH-ISY-R-2172, Linkoping University.

Blanz, V., Vetter, T., 2003. Face recognition based on fitting a 3D morphable model. IEEE Trans. PAMI 25 (9), 1063–1075.

Brunelli, R., Poggio, T., 1992. Face recognition through geometrical features. In: 2nd European Conf. on Computer Vision, pp. 792–800.

Feng, G.C., Yuen, P.C., 1998. Variance projection function and its application to eye detection for human face recognition. Patter Recognition Lett. 19, 899–906.

Feng, G.C., Yuen, P.C., 2001. Multi-cues eye detection on gray intensity image. Pattern Recognition 34 (5), 1033–1046.

Goto, T., Lee, W.-S., Magnenat-Thalmann, N., 2002. Facial feature extraction for quick 3D face modeling. Signal Process.: Image Comm. 17, 243–259.

Huang, J., Wechsler, H., 1999. Eye detection using optimal wavelet packets and RBFs. Internat. J. Pattern Recognition Artif. Intell. (IJPRAI) 13 (6), 1009–1026.

Ladws, M., Vorbruggen, J.C., Buhmann, J., Lange, J., 1993. Distortion invariant object recognition in the dynamic link architecture. IEEE Trans. Comput. 42, 570–582.

Lam, K.M., Yan, H., 1996. Locating and extracting the covered eye in human face image. Pattern Recogntion 29 (5), 771–779.

Lee, T.S., 1996. Image representation using 2D Gabor wavelets. IEEE Trans. PAMI 18 (10), 959–971.

Porat, M., Zeevi, Y.Y., 1988. The generalized Gabor scheme of image representation in biological and machine vision. IEEE Trans. PAMI 10 (4), 452–468.

Vezhnevets, V., Degtiareva, A., 2003. Robust and accurate eye contour extraction. In: Proc. Graphicon, pp. 81–84.

Ville, K., Kamarainen, J.-K., Kalviainen, H., 2004. Simple Gabor feature space for invariant object recognition. Patter Recognition Lett. 25, 311–318.

Yin, L., Basu, A., 1999. Realistic animation using extended adaptive mesh for model based coding. In: Proc. Energy Minimization Methods in Computer Vision and Pattern Recognition, pp. 315–318.

Yulle, A., Hallinan, P., Cohen, D., 1992. Feature extraction from faces using deformable templates. Internat. J. Comput. Vision 8 (2), 99–111.

Zheng, N., Song, W., Li, W., 1999. Image coding based on flexible contour model. Machine Graphics Vision 1 (8), 83–94.

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