

Image processing method for eyes location based on segmentation texture

Fei Ni, Zhuang Fu*, QiXin Cao, YanZheng Zhao

State key laboratory of Vibration, Shock & Noise at Shanghai JiaoTong University, Shanghai, China

Received 20 June 2007; received in revised form 29 November 2007; accepted 29 November 2007

Available online 5 December 2007

Abstract

Image processing is a very important way of sensing environment. In a facial image, to detect eyes localization should be significant for image understanding and recognition. In this paper, differing from most methods based on color, edge and shape, we present a new method based on segmentation texture, which only depend on features of partial and whole textures. As result of it, the method puts up great robustness in different light, noise, tilted head direction and low cost of calculation. Firstly, a facial image is divided into several square blocks at vertical and horizontal orientation. By employing DFFT (Discrete Fast Fourier Transform), texture of every block is calculated. Textures of these blocks make up segmentation texture, and then some textures features are evaluated. Subsequently, the Hough Transform is involved to detect the facial region in a low-resolution image (27×25). Finally, within facial region, the horizontal projection of texture features is computed to locate eye vertical position, and then, to analyze textures of row eye located, the horizontal position of eyes will be obtained. Experiments demonstrate the effectiveness and reliability of this method.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Image processing; Human eye location; Segmentation texture; Discrete Fast Fourier Transform

1. Introduction

Machine vision should be a direct method for getting information of surroundings. For explaining, sorting, abstracting and comprehending these images, some appropriate image processing methods must be brought forward. As a crucial branch of it, the human eyes location problem has attracted significant interests in the last decades [1–3]. Eye detection is often the first step in numerous applications, such as video surveillance, human computer interface, face recognition, and image database management [1]. Eyes represent most essential and important physical information of face. In fact, according to physiology, as if a basic point of position, eyes are closely connected to other parts of face, such as nose, ears, mouth and eyebrows. Thereby, it should be easy to address these parts once eyes are spotted. More important, as a reference mark, eye's size can be employed to recognize characters of face features. For instance, if a person's mouth is 1.2 times wider than one's eye, it can be concluded that the person has a big mouth. So, differing from

most reports, in which a mean face is expected as reference mark [4–6], we consider that the size of one's own eye should be better yardstick.

Many efforts have been addressed to seek for eyes. A pupil generally is darker than surround eyeball, therefore algorithms can be designed to search for gray character [7,16]. For more effective, gray balance or contrast enhancement can be employed. Based on it, precedential knowledge on the facial feature arrangement will be helpful to detect several candidates of eyes [8,17,18]. But, the color of pupil must sharply affect the work of algorithms, as well as lighting condition. In addition, the eyebrows should be similar to eyes at gray level. So, it is difficult to choose a correct one from these eyes candidates. For solving these problems, an artificial template could be involved. So, in [9], the correlation coefficient between the template and the eye image can be calculated to decide which eye pair is. In addition, in the interest of reducing disturbance of color, Hough Transform is proposed to find the circle shape of the eye irises and eyelids [2]. As we know, the Hough Transform should lead to heavy load of calculation. Furthermore, NN can be counted on to solve problem well. Bianchini and Sarti [1] refer to the eyes possess strong horizontal and vertical edges, the exploitation of gradient features is particularly suited to repre-

* Corresponding author. Tel.: +86 21 34206071; fax: +86 21 34206071.
E-mail address: zhfu@sjtu.edu.cn (Z. Fu).

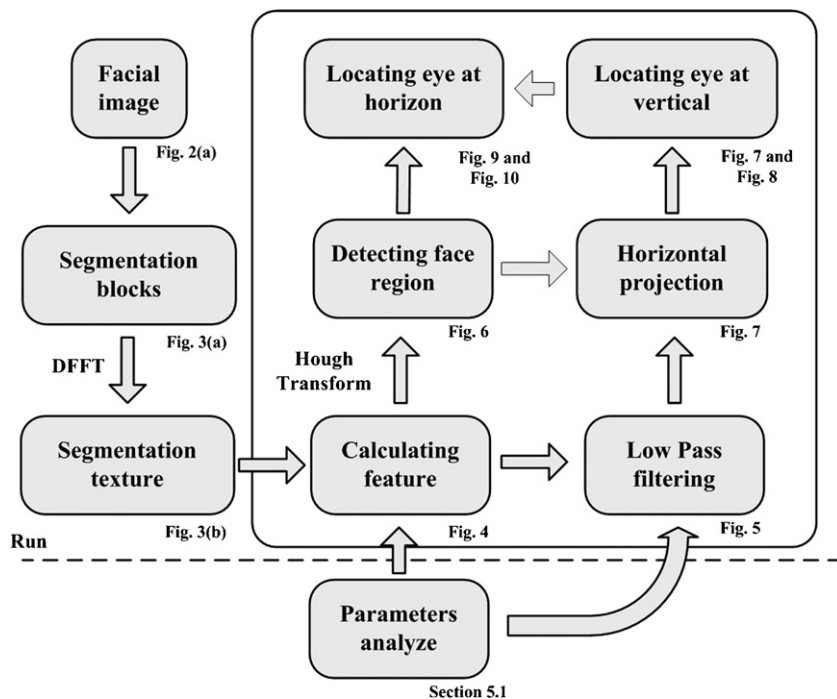


Fig. 1. Overall algorithm.

sent the image content. Therefore, a neural autoassociators can be trained to detect eye region by gradient features. In order to extract gradient features, Sobel filter is utilized. More recently, Gabor wavelets techniques [10], where Gabor wavelet-based linear filters are used for eye corner detection, and non-linear (Gaussian) filters are used for eye location. Moreover, the fractal model provides an excellent representation of the ruggedness of natural surfaces and has served as an image analysis tool for a variety of applications [3,11,12]. The fractal dimension, which is the most commonly used fractal feature, has been applied to measure the irregularity of texture images and to describe these images. Lin et al. [3] used the fractal dimension to form eye pairs and verify them in human facial images. The lacunarity [13], which is a high-order fractal parameter, can assess the largeness of gaps or holes in images and describe the distribution of gaps within an image/region. In [11], a method to estimate lacunarity, which is based on a rectangular box instead of a square box, is presented. All these methods mentioned above closely depend on edge, shape, and model of skin color. There are many challenges to improve their robustness. So, in this paper, a method that based on segmentation texture feature is present. Even if no edge detect, color model or training samples are involved, the algorithm can work well under various situation.

This paper is organized as follows: The proposed overall algorithm is described in Section 2. In Section 3, segmentation texture and detecting face will be achieved. The eyes location is discussed in Section 4. In Section 5, some parameters that mentioned in Sections 3 and 4 will be analyzed, and AR, private database are utilized to perform experiments. Finally, conclusion is drawn in Section 6.

2. System overview

In this paper, a novel method for addressing eyes, which is based on segmentation texture, is presented. The proposed eyes location method has three steps. In the first step, symmetry axis is estimated and obtains the tilted angle of head. In the 2nd step, the facial image is divided into several square blocks at vertical and horizontal orientation. Texture of every block is calculated, and these textures make up segmentation texture. In this phase, more important, some stabile texture features, which can represent some features of frequency domain, should be extracted. After these features are mapped into lower resolution image, Hough Transform is employed to detect facial region. Finally, in every row, to sum up these features updated according to tilted angle will gain horizontal projection. This projection would indicate the eye's position at vertical. In the row where eyes located, textures of blocks would be analyzed to fix horizontal positions of eyes. Fig. 1 shows the overall algorithm.

So, the method presented in this paper has several advantages described as follows:

1. It is not necessary to build up model of skin and eye color.
2. Edge detect is not required, therefore, recognition will not be distinctly affected by lumination and noise. In AR database, to reduce or increase V component of HSV up to 20% will not lead to failure recognition.
3. DFFT is employed to reveal texture features. The cost of calculation is acceptable.
4. When head direction is tilted, instead of rotating the image that will result in more cost of calculation, to rotate a lower resolution 2-D matrix tilted angle will achieve the recogni-



Fig. 2. To extract texture of block: (a) window in facial image, (b) blocks, and (c) textures of blocks.

- tion. Tilted angle of head direction is allowed within range: -90° – 90° .
5. Do not need training and precedential knowledge.
 6. Can deal with different scale images (eye high is 10–40 pixels) without modifying parameters.

For testing our algorithm, the AR database [14] is utilized. In addition, at 2006 China International Industry Fair (CIIF), the recognition system of human face is exhibited and many visitors face photos are collected to verify the algorithm. So these photos make up our private database including 204 front face photos.

3. Calculating segmentation texture and detecting face region

3.1. Facial symmetry axis

Because of the consistent fabric of human face, estimating symmetry axis will be greatly helpful to filter some candidates and excrescent region. Generally, the principle moments can estimate symmetry axis well. In [15], the method for detecting direction of facial images based on PCA is presented. It is pointed out that in the binary image of a human face, the first principal axis shows the direction of the head while the second principal axis indicates the direction of the line joining the two eyes [11]. Because facial image is a spatial shape at 2 dimensions, we can reduce the PCA to principal inertia. The axis along down direction of head direction will display fewer eigenvalue than other axis joining direction of eyes. So, the tilted angle: θ_h and symmetry axis position: S_h at horizon can be worked out. Assume I is inertia matrix, and it can be determined by:

$$I = \begin{bmatrix} I_{xx} & -I_{xy} \\ -I_{yx} & I_{yy} \end{bmatrix} \tag{1}$$

$$\begin{cases} I_{xx} = \mu_{02}, \\ I_{yy} = \mu_{20}, \\ I_{xy} = I_{yx} = \mu_{11} \end{cases} \tag{2}$$

$$\mu_{ij} = \sum_{x,y} (x - g_x)^i \cdot (y - g_y)^j \tag{3}$$

where μ_{ij} is center inertia of the power of i at x and power of j at y ; g_x, g_y are barycentre. The eigenvector v and eigenvalue λ can be computed as follows:

$$[\lambda \quad v] = \text{eig}(I) \tag{4}$$

where $\text{eig}(x)$ stand for calculating eigenvector and eigenvalue of I . Because I is a real symmetric matrix, the Jacobi method should be appropriate to solve Eq. (4). Obviously, λ is a 2×2 diagonal matrix; v is a 2×2 matrix. In λ , without loss of generality, we can assume that $\lambda_1 \leq \lambda_2$, then the corresponding eigenvector of λ_1 : v_h represents the head-up direction. The tilted angle: θ_h can be calculated by:

$$\theta_h = a \tan(v_h) \tag{5}$$

In this paper, the original image is not necessary to be rotated because θ_h would not affect texture distribution. In a nutshell, the rotated image will not result in the rotation of texture distribution derived from DFFT. However, the tilted angle should be considered in when horizontal projection is calculated. Once the symmetry axis is gained, these eyes candidates will be described as following rules [11]:

1. Two eyes should be found at two sides of the symmetry axis.
2. The distance between each eye to the symmetry axis should be within a certain range in pixel numbers. Such as, in AR, the

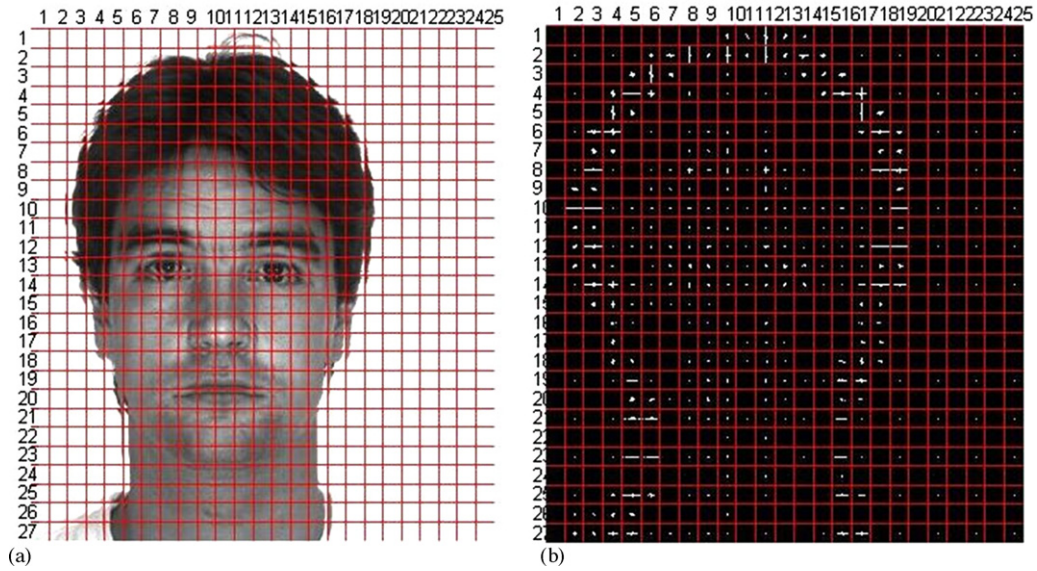


Fig. 3. Calculating segmentation texture: (a) divide the image into 27×25 blocks, $\delta = 20$ and (b) texture of every block.

distance is around 30 pixels. In privacy database, the distance is around 20 pixels. Generally, based on the outside contour, the distance can be computed approximately.

3.2. Calculating segmentation texture by DFFT

These textures of eyes are special enough to distinguish them from face [19]. And, in order to obtain ample texture features, it is required that the high of eye involve 10 pixels at least. Generally, a simple way to reveal texture is to employ DFFT [20–22]. So, a window, which is height and width both are 32

pixels, is masked in a facial image. Within the window, all pixels are called as a “block”; the block will be transformed by DFFT (see Fig. 2).

$$F = \wp(T) \tag{6}$$

where $\wp(x)$ is DFFT of 2-D matrix, T is 32×32 matrix, every element of T is corresponding to gray level of block; F is a complex matrix, its element $f(i, j)$ indicates the value of frequency domain:

$$f(i, j) \in F \quad i, j = 1, 2, \dots, 32 \tag{7}$$

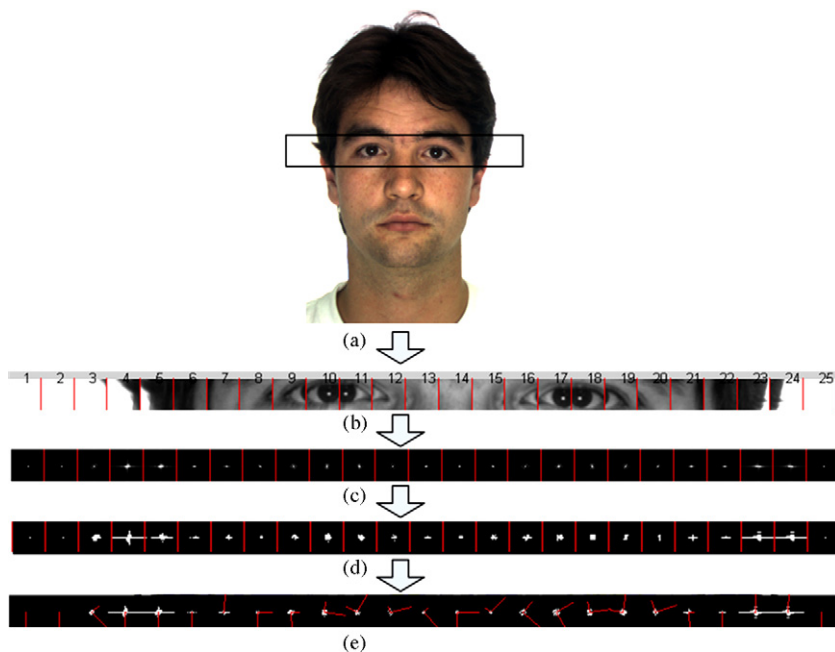


Fig. 4. To compute inertia axes: (a) facial image, (b) 25 blocks, window size = 32×32 , $\delta = 20$, (c) texture, (d) binarization, and (e) estimate inertia axes.

Now, these blocks have been converted into frequency domain from spatial domain. The modulus $R(i, j)$ and phase $\theta(i, j)$ of $f(i, j)$ can be evaluated by Eqs. (8) and (9):

$$R(i, j) = \sqrt{(f(i, j) \cdot \text{real})^2 + (f(i, j) \cdot \text{imag})^2} \quad (8)$$

$$\theta(i, j) = a \tan \left(\frac{f(i, j) \cdot \text{imag}}{f(i, j) \cdot \text{real}} \right) \quad (9)$$

where $f(i, j) \cdot \text{real}$ and $f(i, j) \cdot \text{imag}$, respectively, are real and imaginary part of $f(i, j)$.

Obviously, $R(i, j)$ make up a 32×32 real matrix: R^l . Usually, the modulus indicates value and direction of frequency, it can be regarded as block texture. For more visual comprehension, the R^l should be proportionally reduced into the gray level range by Eq. (10). Assume the maximum in R^l is R_{\max} , the minimum is R_{\min} :

$$R(i, j) = \text{Int} \left(\frac{R(i, j)}{100S} \right) \in R^l \quad i, j = 1, 2, 3 \dots 32 \quad (10)$$

where $\text{Int}(x)$ stands for the nearest integer of x ; $S = (R_{\max} - R_{\min})/255$. The texture of block has been showed in Fig. 2(c). The window size and shift space of step: δ is alterable. Because the eye height is almost 20 pixels, a 32×32 pixels window, which shifts 20 pixels at each step, is masked on facial image. For covering total facial region, the step number is 25 at horizon and 27 at vertical. So, facial image is divided into 675 blocks. In other words 675 R^l make up 27×25 matrix: G called as segmentation texture, in which texture of block that located i th row and j th column is denoted by $R^l_{i,j}$. For giving prominence to the texture, binarization is involved to handle $R^l_{i,j}$. The binarization result of $R^l_{i,j}$ is indicated by $R^b_{i,j}$. The threshold for binarization: t_1 should be 25 now, further discussion about t_1 can be found in Section 5.1. The binarized segmentation texture and blocks are showed in Fig. 3.

Texture features indicate some stable characters that can represent texture of given region. In Fig. 3(c), the texture of eye region approximately is an ellipse whose major axis is horizontal or vertical. So, in $R^b_{i,j}$, by employing method described in Section 3.1, the principal inertia of every block can be obtained. For instance, the row where eye located is extracted to calculate inertia axes (see Fig. 4). Fig. 4(e) shows two inertia axes of every block.

So, the texture features of a block can be summarized as follows:

1. Eigenvalue of major axis of texture: E_{major} .
2. Eigenvalue of minor axis of texture: E_{minor} .
3. The proportion between the number of white points: n_w and total points: n_t in $R^b_{i,j}$: $\tau = n_w/n_t$.
4. The high frequency proportion ϕ_x :

$$\phi_x = t/n_w$$

where t is the number of white points, and the distance between these white points and the center of $R^b_{i,j}$ is greater than x . In

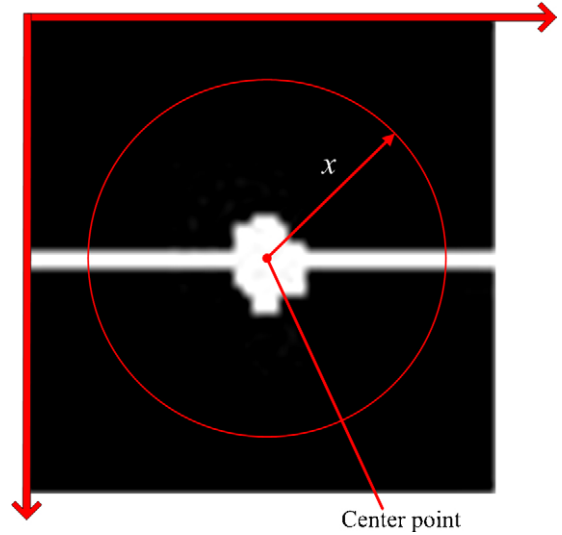


Fig. 5. To calculate ϕ_x in a block texture.

Fig. 5, t means the number of white points that located outside red circle with radii: x .

Especially, within a block, E_{major} represents the level of gray variety at some directions; more important, τ can indicates the whole gray variety. These blocks of eyes or edges will show greater τ . ϕ_x indicates the proportion that frequency is higher than x . Generally, hair edges should have greater ϕ_x , on the contrary, ϕ_x will be zero when block locate smooth region.

3.3. Using E_{major} to detect face region

To detect face region will greatly be propitious to dwindle projection region. In a general way, eyes should locate middle of face at vertical. By experiments, E_{major} of texture features can be used to detect outside outline of face well even the face is masked by glasses, sunglasses, a scarf or under different light direction. For example, in Fig. 6, a face with glasses and scarf is picked up to explain the detection processing. E_{major} is mapped in Fig. 6(b), and according to Fig. 6(b), the binarization image: B is showed in Fig. 6(c) under threshold $t_2 = 1000$, in Section 5.1, we will analyze how to determine t_2 . Thereafter, B is rotated with $-\theta_h$. Obviously, the rotated image will lead to some blank points that do not exist before rotation. Zero will be filled in these points. After rotating, according to Hough Transform, an ellipse should be recognized in binarization image by Eq. (11). Generally, the face of human is an ellipse with major axis at vertical.

$$\left. \begin{aligned} [a \ b \ \text{laxis} \ \text{saxis}] &= \text{TH}(B) \\ \text{laxis} &\geq \text{saxis} + 2 \end{aligned} \right\} \quad (11)$$

where a, b are, respectively, horizontal and vertical center coordinates of ellipse; $\text{laxis}, \text{saxis}$ stand for, respectively, value of ellipse major and minor axes; $\text{TH}(x)$ means the ellipse Hough Transform for binarization image: x . The ellipse is showed in Fig. 6(d). Based on this ellipse, in original image, corresponding

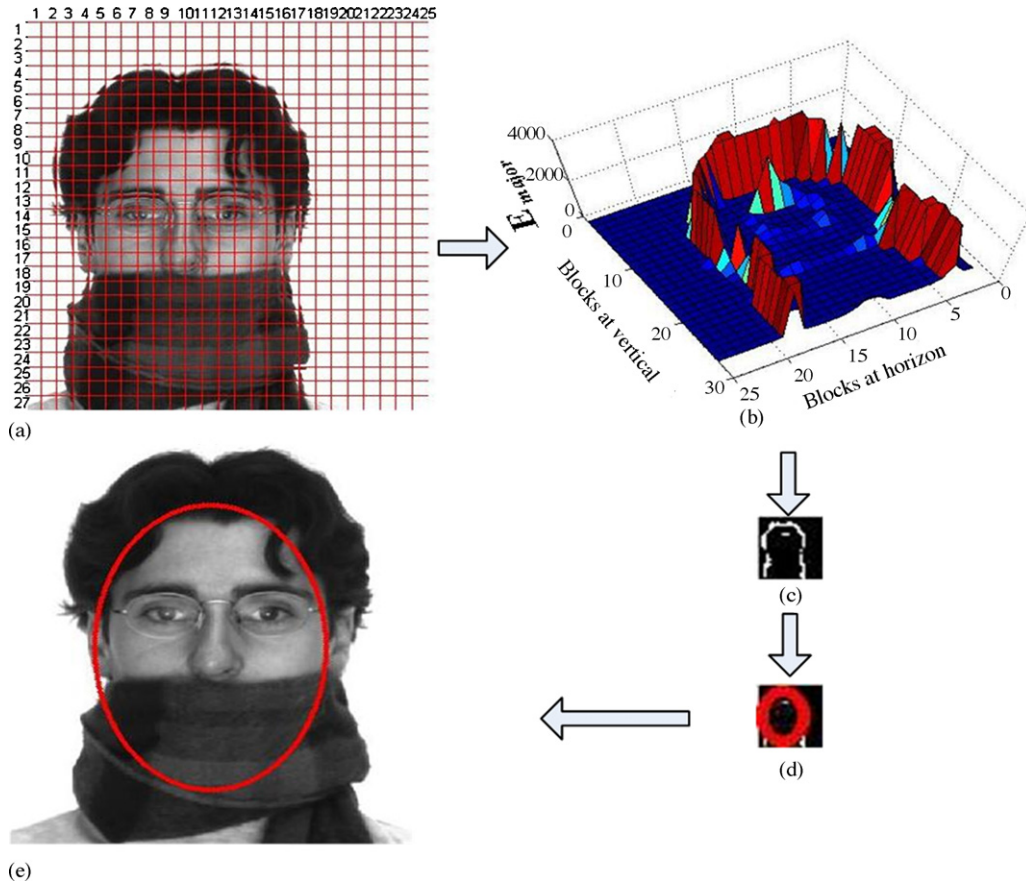


Fig. 6. Detecting face region by major axis distribution: (a) segmentation blocks, (b) map of E_{major} , (c) binarization image of E_{major} , (d) detect ellipse in binarization image, and (e) the face region in original image.

parameters of ellipse can be derived from:

$$\left. \begin{aligned} a_o &= (a + 1) \times \delta \\ b_o &= (b - 1) \times \delta \\ \text{shortaxis} &= (\text{saxis} - 1) \times \delta \\ \text{longaxis} &= (\text{laxis} - 1) \times \delta \end{aligned} \right\} \quad (12)$$

where, in original image, a_o, b_o are, respectively, horizontal and vertical center coordinates of ellipse; longaxis, shortaxis are, respectively, ellipse major and minor axes. The ellipse will be rotated with θ_h , and the face region is showed in Fig. 6(e).

4. Estimating eyes position

4.1. Calculating eyes position at vertical

Once the face region is detected, eye must be involved in this region. Because the row where eyes located has more texture variety, τ can reflect eye row well. The distribution of τ is mapped in Fig. 7(b), and its gray level is showed in Fig. 7(c). In accordance with B , the Fig. 7(b) will be rotated with $-\theta_h$, and then zero would initialize these blank points. At last, in the rotated image, the sum of τ that lie at i th row would be indicated

by τ_i^{row} , and it can be calculated by Eq. (13).

$$\left. \begin{aligned} \tau_i^{row} &= \sum_{i,j \in F} \tau_{i,j} \\ F &= \left\{ (i, j) \mid \frac{(j-b)^2}{\text{saxis}^2} + \frac{(i-a)^2}{\text{laxis}^2} < 0.8, \phi_{t4}(i, j) > t_3 \right\} \end{aligned} \right\} \quad (13)$$

where $i = 1, 2, 3, \dots, \text{Int}(27(\cos(\theta_h) + \sin(\theta_h)))$; $j = 1, 2, 3, \dots, \text{Int}(25(\cos(\theta_h) + \sin(\theta_h)))$; τ_{ij} is one of τ that located i th row and j th column; $i, j \in F$ means blocks with $\phi_{t4}(i, j) > t_3$ and blocks located outside F will be excluded. t_3 and t_4 will be discussed in Section 5.1, now to assume $t_3 = 0.001, t_4 = 13$. The Horizontal projection: τ_i^{row} is plotted in Fig. 7(d). Obviously, the maximum τ_i^{row} indicate the row where eye located. Using max to denote the row of maximum.

Because eye's height is around 10–40 pixels, usually, there are sub-maximums values at two sides of peak. The 3 rows are called as eye row. There are two possible eye row distributions (see Fig. 8). At vertical direction, the position of top eye line: e_{top} and bottom eye line: e_{bottom} can be evaluated by:

1. (14) If $\tau_{\max-1}^{row} < \tau_{\max+1}^{row}$ then $e_{top} = (\max - 1) \times \delta - 5, e_{bottom} = e_{top} + 32$.
2. (15) Else if $\tau_{\max-1}^{row} \geq \tau_{\max+1}^{row}$ then $e_{top} = (\max - 1) \times \delta - 20, e_{bottom} = e_{top} + 32$.

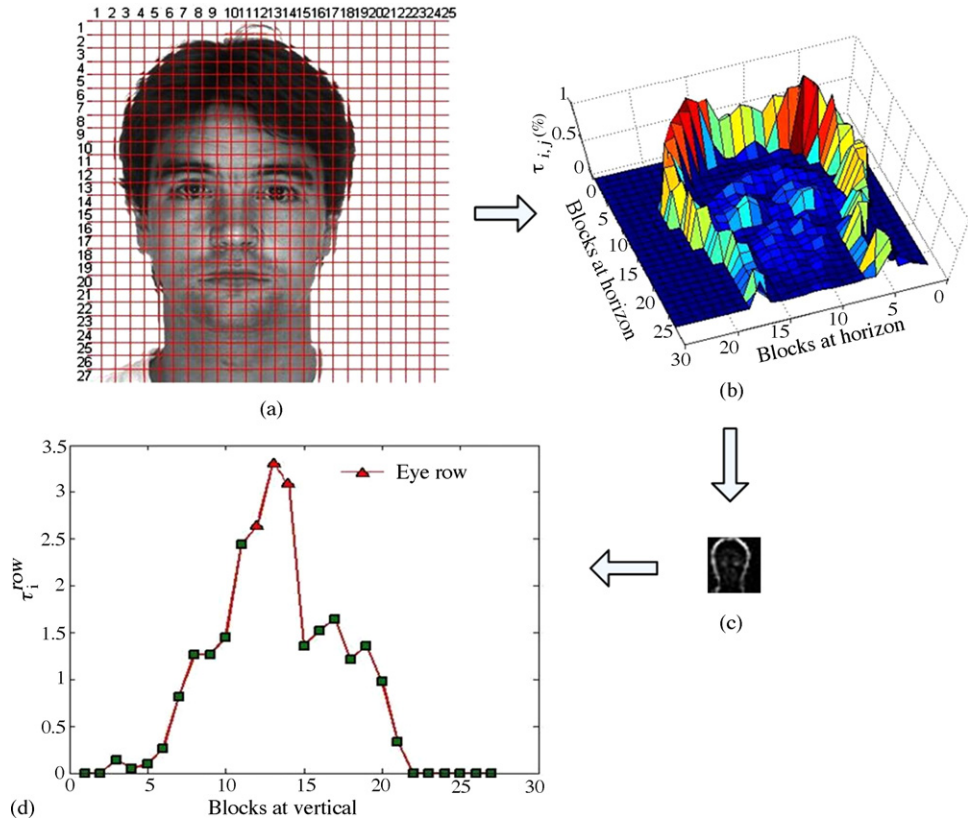


Fig. 7. Horizontal projection: (a) segmentation blocks, (b) map of τ , (c) gray level image of τ , and (d) τ_i^{row} distribution.

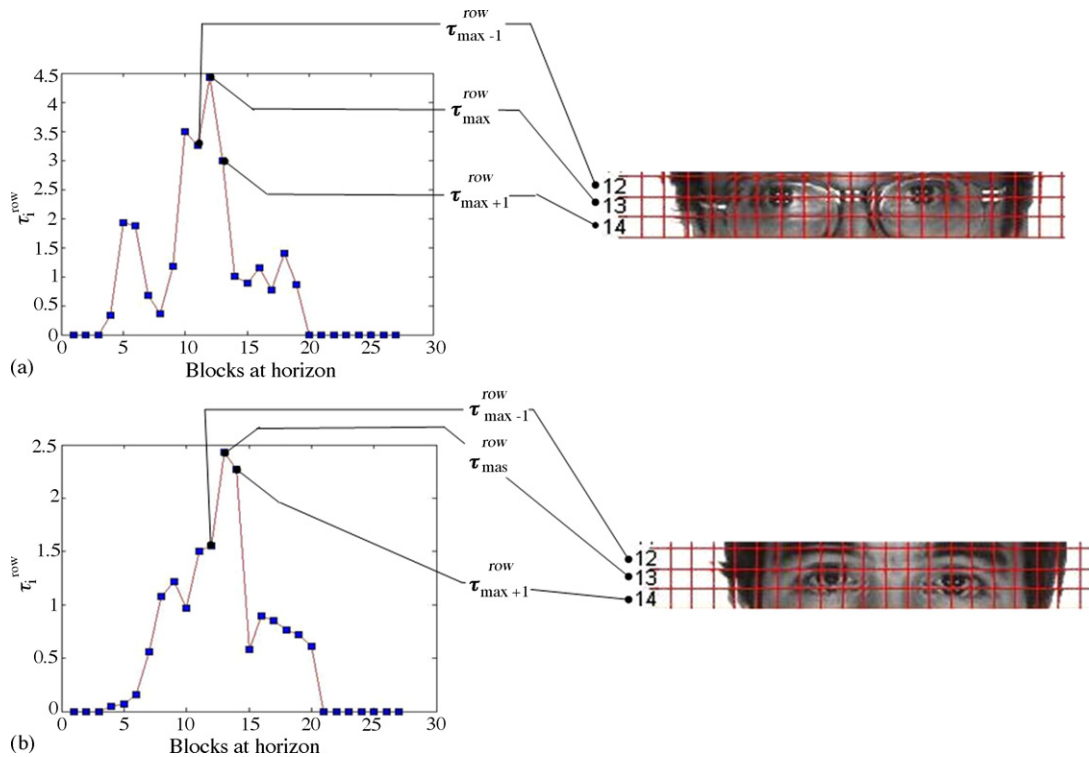


Fig. 8. Two possible τ_i^{row} distribution of eye row, (a) $\tau_{\text{max}-1}^{\text{row}} \geq \tau_{\text{max}+1}^{\text{row}}$, (b) $\tau_{\text{max}-1}^{\text{row}} < \tau_{\text{max}+1}^{\text{row}}$.

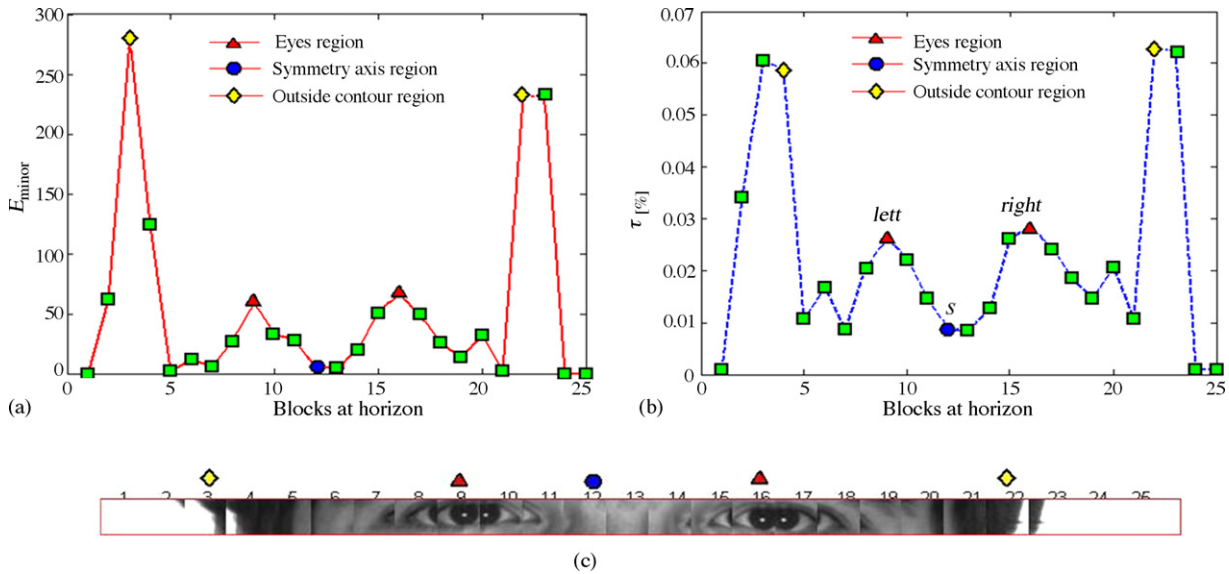


Fig. 9. Locating eyes at horizon when Eq. (16) is satisfied: (a) E_{minor} distribution, (b) τ distribution, and (c) a series of blocks in E_h .

Obviously, e_{top} and e_{bottom} should be sloped by θ_h in original image.

4.2. Calculating eyes position at horizon

Because of eyes' symmetry, eyes horizontal position should locate both sides of symmetry axis. At vertical, the region between e_{top} and e_{bottom} , marked as E_h , would be employed DFFT to extract texture (see Fig. 3). E_{minor} and τ of every block are plotted in Fig. 9(a and b) and Fig. 10(a and b). According to S_h , symmetry axis region can be marked by circle point in Figs. 9 and 10.

There are two possible distributions of τ . They will lead to different ways to estimate eyes position at horizon.

1. Because a face wearing a glasses or sunglasses will result in more gray variety at these regions of symmetry axis and around it. If the condition: Eq. (16) is satisfied, the face may not wear a glasses or sunglasses greatly probably.

$$\tau_s < \tau_k \quad (k = \text{max} - 3, \text{max} - 4, \text{max} + 3, \text{max} + 4) \quad (16)$$

where s means the position of symmetry axis. On account of an eye width will occupy 30 pixels at least, at horizon, the left line: e_{left} and right line: e_{right} of left eye can be determined by (see Fig. 9, left=9):

$$e_{\text{left}} = (\text{left} - 1) \times \delta - 5 \quad e_{\text{right}} = e_{\text{left}} + 40 \quad (17)$$

where left is the position of a relatively greater point at left side of s . In a similar way, the left and right line of right eye:

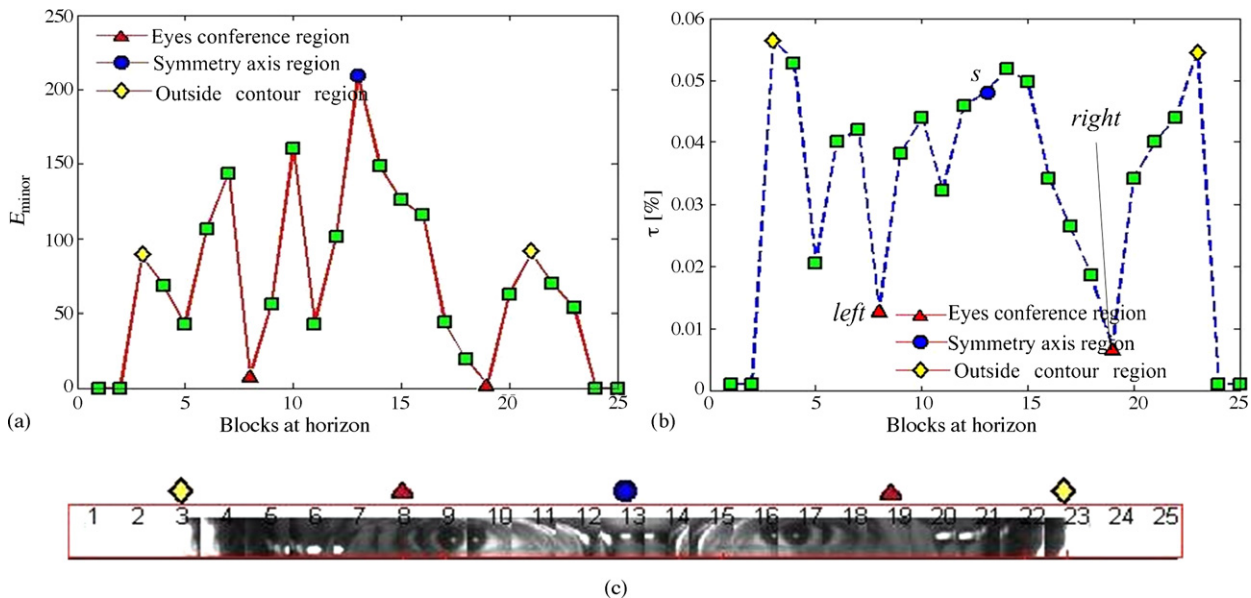


Fig. 10. Locating eyes at horizon when Eq. (19) is satisfied: (a) E_{minor} distribution, (b) τ distribution, and (c) a series of blocks in E_h .

e_{rleft} and e_{rright} can be computed by:

$$e_{\text{rright}} = (\text{right} + 1) \times \delta + 5 \quad e_{\text{rleft}} = e_{\text{rright}} - 40 \quad (18)$$

where right is the position of a relatively greater point at right side of s .

- If Eq. (19) is satisfied, then, a glasses or sunglasses should be worn.

$$\tau_s > \tau_k \quad (k = \max -3, \max -4, \max +3, \max +4) \quad (19)$$

Fig. 10 shows the distribution satisfying the Eq. (19).

Differing from previous situation, the left is not eye region any more but eye conference region, which is the position of a relative fewer point at left side of s . As such, right is the position of a relatively fewer point at right side of s . So, these eye lines at horizon can be evaluated by:

$$e_{\text{lleft}} = (\text{left} - 1) \times \delta - 20 \quad e_{\text{lright}} = e_{\text{lleft}} + 40 \quad (20)$$

$$e_{\text{rright}} = \text{right} \times \delta + 20 \quad e_{\text{rleft}} = e_{\text{rright}} - 40 \quad (21)$$

Now, the left eye can be enveloped by a rectangle that is made up of e_{top} , e_{bottom} , e_{lleft} and e_{lright} . The right eye can be enveloped by a rectangle that is made up of e_{top} , e_{bottom} , e_{rright} and e_{rleft} . Obviously, in original image, these parameters will be sloped by θ_h .

5. Parameters analyses and experiments

5.1. Parameters analyses

There are four important thresholds t_1 , t_2 , t_3 and t_4 in Sections 3 and 4. t_1 is a basic threshold for binarization, it will directly impact other three parameters. As discussing above, t_2 will be with responsibility for extracting outside outline. The low pass filter will be determined by t_3 and t_4 , and appropriate low pass filter will get rid of influence of rough edges furthest. So, to select feat parameters will effectively improve robustness. In AR database, 10 samples with different hair color and eye color under normal lumination (section 1) can be picked up. These samples build up sample group: SG^1 , and corresponding to these samples, left light on samples (section 5) are extracted to make up sample group SG^2 . In these two sample groups, left

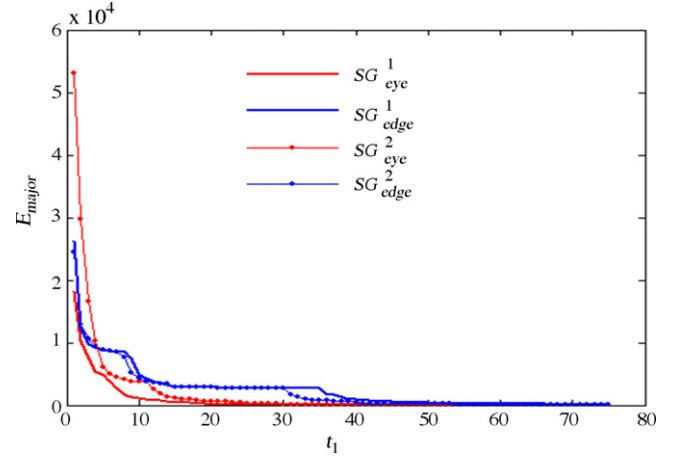


Fig. 11. E_{major} of SG^1_{eye} , SG^1_{edge} , SG^2_{eye} , SG^2_{edge} under different t_1 .

eye and edge (such as hair edges or glasses edges around eyes) regions would be marked manually. In SG^1 and SG^2 , we can average eye and edge regions, respectively. As a result of it, the texture of average eye and average edge regions of SG^1 can be denoted by SG^1_{eye} and SG^1_{edge} , respectively. Similarly, SG^2_{eye} and SG^2_{edge} should be achieved. These parameters can be evaluated by following steps:

- To evaluate t_1 and t_2 . In Fig. 11, E_{major} of SG^1_{eye} , SG^1_{edge} , SG^2_{eye} , SG^2_{edge} is plotted under different t_1 .

In Fig. 11, when $30 > t_1 > 20$, in these two sample groups, E_{major} of edge region reaches a platform around 3000. Synchronously, E_{major} of eye region will show a platform around 300. So, according to Eq. (22), eye and edge regions can be right distinguished with high robustness.

$$t_2 = 1000, \quad 30 > t_1 > 20 \quad (22)$$

- Find t_3 and t_4 . Obviously, t_1 will react on t_3 and t_4 directly. Thereby, to map their relationships are necessary. Therefore, basing on SG^1_{eye} and SG^1_{edge} of simple group SG^1 , the relationships between t_1 , t_3 and t_4 are mapped in Fig. 12(a and b). In eye region, when $t_1 > 10$ and $t_4 \geq 13$, t_3 almost is zero

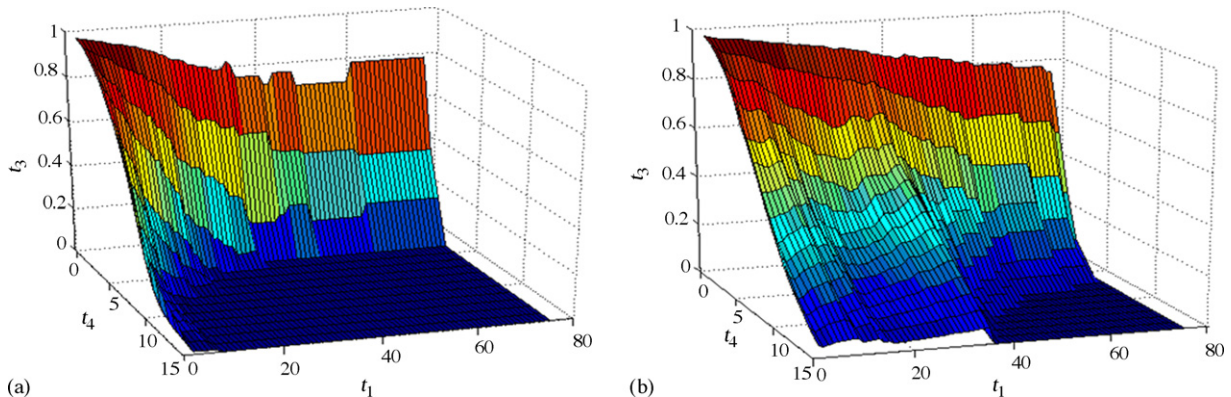


Fig. 12. Relationship between t_1 , t_3 and t_4 in simple group SG^1 , (a) SG^1_{eye} , (b) SG^1_{edge} .

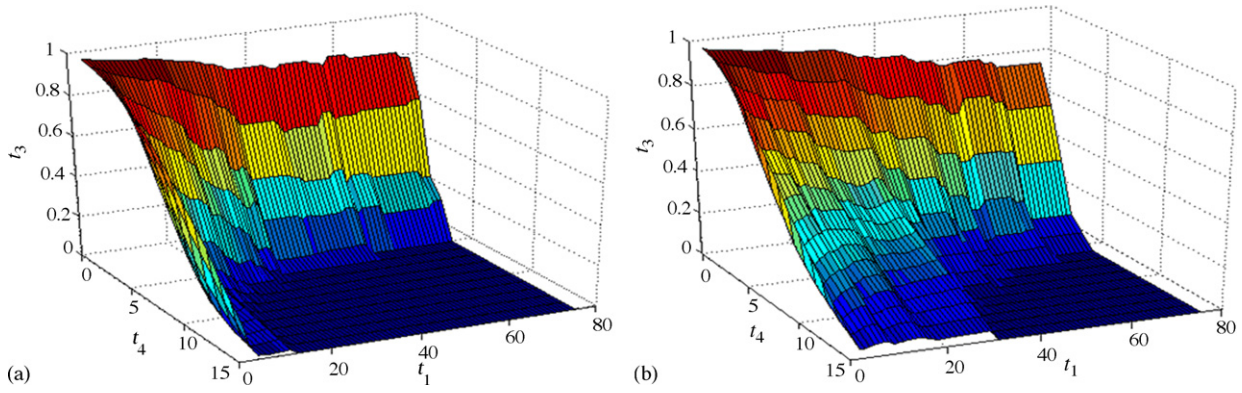


Fig. 13. Relationship between t_1 , t_3 and t_4 in simple group SG^2 , (a) SG^2_{eye} , (b) SG^2_{edge} .

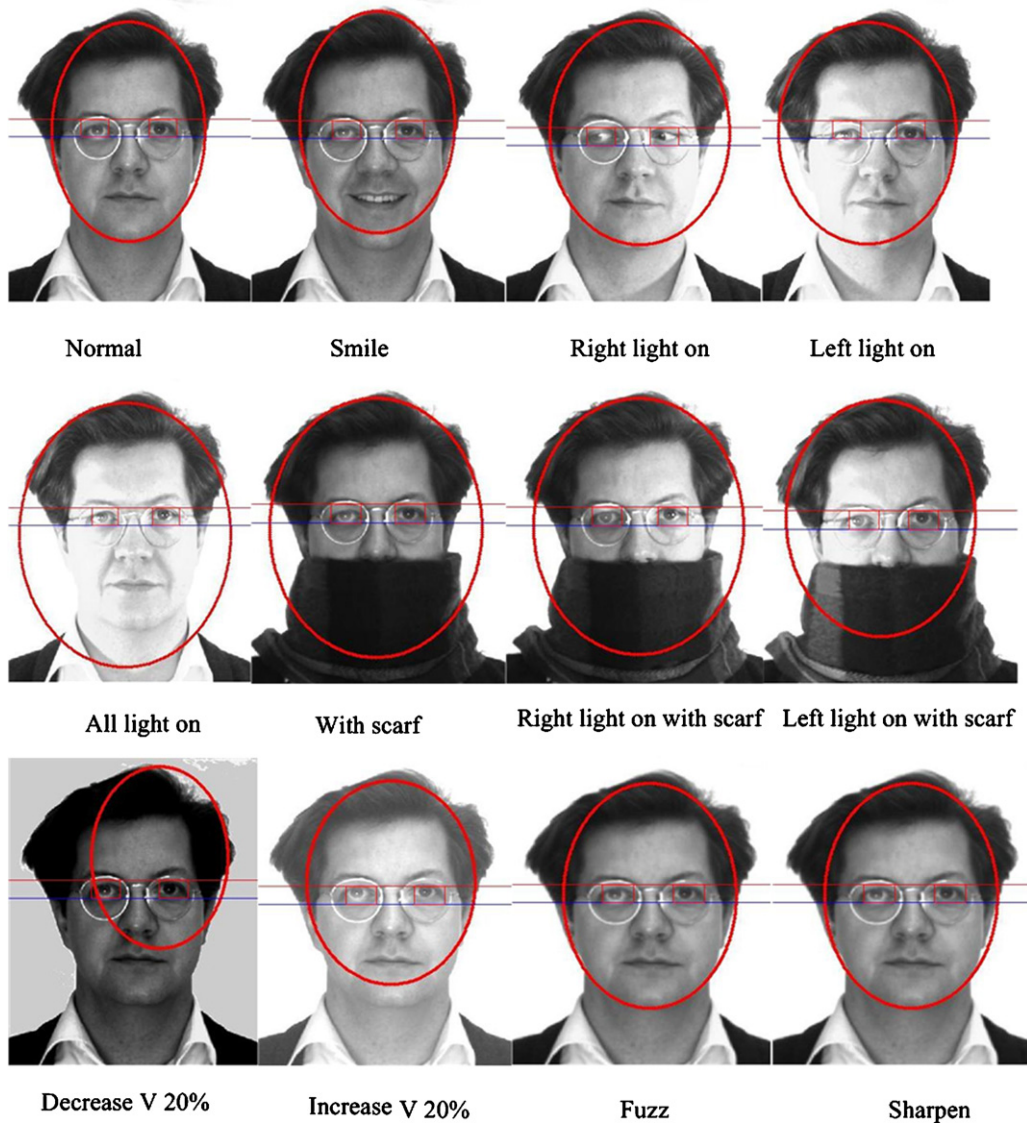


Fig. 14. Some results in AR database and custom condition.

Table 1
Recognition rate in AR database

Sections	1, 11 (normal, wearing scarf)	2 (smile)	5,6,7(left, right, or all side light on)	12,13 (left, right light on with scarf)	14 (increase or decrease V component of HSV up to 20%, fuzz or sharpen in 100 received simples)
Hit rate (%)	98.2	89.2	96	96	100
Rejected reason	Violent repercussion of glasses	Closure eyes	Violent repercussion of glasses	Violent repercussion of glasses	

Table 2
Comparison between segmentation texture and existing methods

Methods	Hit rate (%)	Need edge detect?	Need training?	Approved tilted angle of head direction	Computational load
[2]	94.98	Yes	No	Not mentioned	Not mentioned
[1]	87–97 under various threshold	Yes	Yes	Not mentioned	Not mentioned
Segmentation texture	89.2–98.2	No	No	–90–90°	$O(N \log 2(N))$

(see Fig. 12(a)), in other words:

$$\phi_{13} = 0, \quad t_1 > 10 \tag{23}$$

Moreover, in edge region, in Fig. 12(b):

$$\phi_{13} \neq 0, \quad t_1 < 37 \tag{24}$$

In the same way, Fig. 13 shows the relationships between t_1 , t_3 and t_4 in SG_{eye}^2 and SG_{edge}^2 . Following Eqs. can be concluded:

In eye region,

$$\phi_{13} = 0, \quad t_1 > 17 \tag{25}$$

In edge region,

$$\phi_{13} \neq 0, \quad t_1 < 30 \tag{26}$$

So, according to Eqs. (22)–(26), we can draw a conclusion that:

$$t_1 = 25, \quad t_2 = 1000, \quad t_3 = 13, \quad t_4 = 0$$

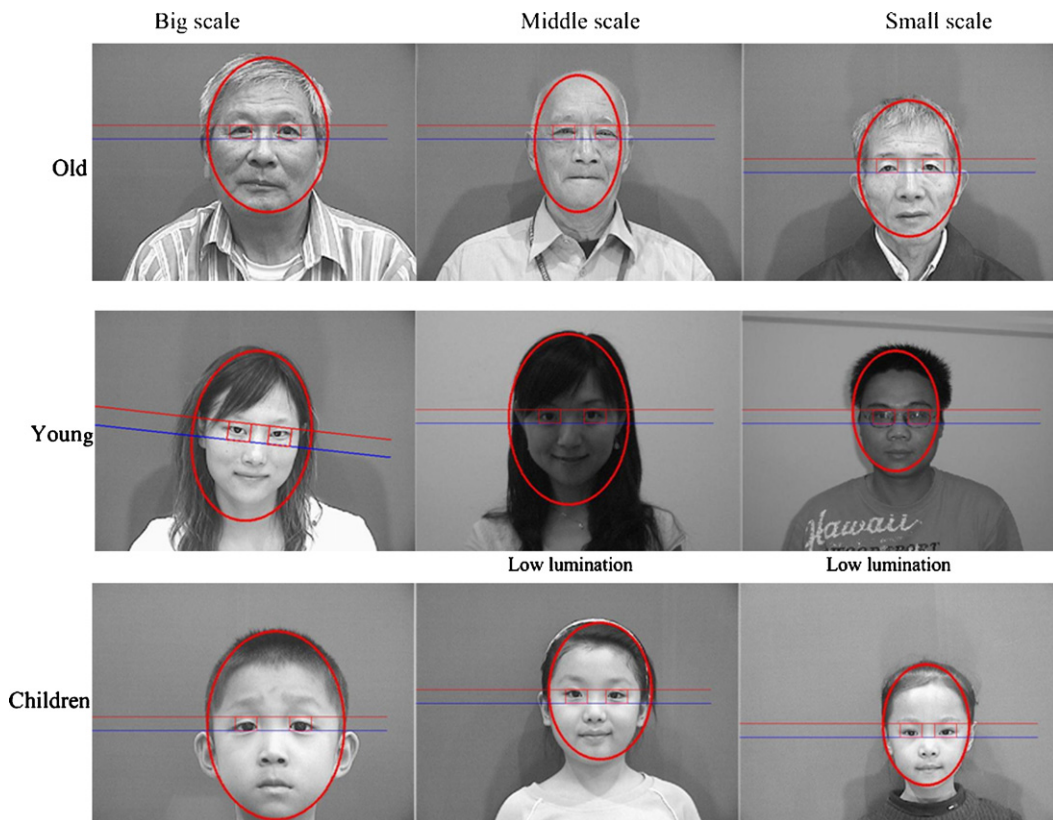


Fig. 15. Some results in private database.

5.2. Experiments

An eye location will be recognized in a rectangle. In order to get a more precise pupil position, some reliable algorithms in [2,7–9] can be employed in these rectangles. In AR database [14], 540 color images that classified into sections 1, 2, 5, 6, 7, 11, 12, 13 are utilized to test our method. Furthermore, for validating robustness further, in received samples, 100 samples will be increased or decreased their V component of HSV up to 20%, fuzzed by 3×3 median filtering or sharpen by edge enhancement to build up sample section 14. The algorithm is programmed by VC++, compute platform is CM 1.6G CPU, and the average runtime is about 1.2s per image. These recognition rates are listed in Table 1. The results of these samples are showed in Fig. 14.

In Table 1, violent repercussion of glasses will mainly take the responsibility for failure recognition because these white regions with glare will disturb texture distribution of eyes. But, common glasses repercussion and edges of glasses would not result in rejection. The light sources from different direction, such as sections 5, 6, 7, 12, and 13, are unable to make failures. In sections 2 and 3, eyes can be located in samples with general expression if eyes do not close. Generally, if the height of eye is less than 8 pixels, it will bring wrong recognition. Table 2 shows the comparison between segmentation texture and existing methods for dealing with AR database.

In private database, there are 204 samples with various ages and lamination, such as old people, children and youngling. These samples are shot at different scales. The minimum size of height of eye is 10 pixels; the maximum size is 32 pixels. In private database, the recognition rate is 98.3%. Some recognition results are showed in Fig. 15. Even under low lamination, the recognition can be achieved too (see Fig. 15, the samples labeled by “low lamination”).

6. Conclusion

The present paper focuses on image processing method for detecting eyes in facial image based on segmentation texture. Its main contribution consists in proposing, analyzing, validating a new addressing eyes method, which only depends on texture information. Any edge detecting, skin model and pre-process is not required, as result of it, the method shows high robustness. In a tilted face, eyes can be located when $-90 \leq \theta_h \leq 90$. Moreover, in different scale images (eye high is 10–40 pixels), modifying parameter is not required.

Experiments are performed on AR database, and in different sections the recognition rate is 89.2–98.2%. For further proving robustness of algorithm, 100 received samples are modified color parameter, fuzzed or sharpened. Experiments denote the recognition rate is 100% when dealing with these disturbances. In addition, in private database, the global accuracy of 98.3% proves the method is promising.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under grant No. 60304010 and the Start-up Grant for Young Teachers from the Shanghai Jiao Tong University. All of the authors are grateful to the owners of these original images used in this paper.

References

- [1] M. Bianchini, L. Sarti, An eye detection system based on neural autoassociators, artificial neural networks in pattern recognition, in: Second IAPR Workshop, ANNPR 2006, Proceedings, 2006, pp. 244–252.
- [2] M. Dobes, J. Martineka, D. Skoupila, Z. Dobesovab, J. Pospis, Human eye localization using the modified Hough transform, *Optik* 117 (2006) 468–473.
- [3] K.-H. Lin, K.-M. Lan, W.-C. Siu, Locating the eye in human face images using fractal dimensions, *IEE Proc. Vis. Image Signal Process.* 148 (December (6)) (2001) 413–421.
- [4] H. Koshimizu, M. Tominaga, T. Fujiwara, et al., On KANSEI facial processing for computerized facial caricaturing system PICASSO, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 1999, pp. 294–299.
- [5] R.N. Shet, K.H. Lai, E.A. Edirisinghe, P.W.H. Chung, Use of neural networks in automatic caricature generation: an approach based on drawing style capture, in: IEE International Conference on Visual Information Engineering, VIE 2005, 2005, pp. 23–29.
- [6] Fujiwara, Takayuki, Watanabe, Takashi, Development of caricaturing robot and its prospect through the prototype robot exhibition in EXPO 2005, *Proc. SPIE-Int. Soc. Opt. Eng.*, vol. 6051, Optomechatronic Machine Vision (2005) 605111.
- [7] M. Rizon, T. Kawaguchi, Automatic eye detection using intensity and edge information, *Proc. IEEE TENCON 2* (2000) 415–420.
- [8] L. Zhang, P. Lenders, Knowledge-based eye detection for human face recognition, in: Proceedings of the 4th Int. Conf. on Knowledge-based Intelligent Electronic System, KES, vol. 1, 2000, pp. 117–120.
- [9] S.A. Suandi, S. Enokida, T. Ejima, An extended template matching technique for tracking eyes and mouth in real-time, in: Proceedings of Visualization, Imaging and Image Processing, 2003, pp. 586–591.
- [10] H. Kim, Lee, H.J., Kee, S.C. Seok, A fast eye localization method for face recognition, 13th IEEE International Workshop on Robot and Human Interactive Communication (2004) 241–245.
- [11] G. Du, Eye location method based on symmetry analysis and high-order fractal feature, *IEE Proc. Vis. Image Signal Process.* 153 (1) (2006) 11–16.
- [12] E.J. Koh, P.K. Rhee, Image context-driven eye location using the hybrid network of k-means and RBF, *Lecture Notes in Computer Science*, v 4222 LNCS-II, in: Advances in Natural Computation—Second International Conference, ICNC 2006, Proceedings, 2006, pp. 540–549.
- [13] B.B. Mandelbrot, Fractal analysis and synthesis of fracture surface roughness and related forms of complexity and disorder, *Int. J. Fract.* 138 (1–4) (2006) 13–17.
- [14] A.M. Martinez, R. Benavente, The AR Face Database, CVC Technical Report #24, June 1998.
- [15] G.C. Feng, P.C. Yuen, Multi-cues eye detection on gray intensity image, *Pattern Recognit.* 34 (2001) 1033–1046.
- [16] L. Tao, H.K. Kwan, Automatic localization of human eyes in complex background, in: IEEE International Symposium on Circuits and Systems, vol. 5, 2002, pp. V/669–V/672.
- [17] S.C.H. Yan, X.F. He, Y.X. Hu, H.J. Zhang, Bayesian shape localization for face recognition using global and local textures, *IEEE Trans. Circuits Syst. Video Technol.* 14 (1) (2004) 102–113.
- [18] L. Machala, J. Pospisil, Proposal and verification of two methods for evaluation of the human iris video-camera images, *Optik* 112 (8) (2001) 335–340.

- [19] Z.M. Liu, W. Zhong, X. He, J.L. Zhou, A method of precise eye location from image with complex background, *Proc. SPIE-Int. Soc. Opt. Eng.* 5286 (1) (2003) 133–138.
- [20] K. Singh, M. Ma, D.W. Park, A content-based image retrieval using FFT and cosine similarity coefficient, in: *Proceedings of the Fifth IASTED International Conference on Signal and Image Processing*, vol. 5, 2003, pp. 315–319.
- [21] M. Morikawa, A. Katsumata, K. Kobayashi, Pixel-and-column pipeline architecture for FFT-based image processor, in: *Proceedings-IEEE International Symposium on Circuits and Systems*, vol. 3, 2002, pp. III/687–III/690.
- [22] R.W. Cox, R.Q. Tong, Two- and three-dimensional image rotation using the FFT, *IEEE Trans. Image Process.* 8 (9) (1999) 1297–1299.

Biographies

Fei Ni received his MS degree in engineering from the School of Mechanical and Electronic Engineering at WuHan University of Technology, China in 2004. He is currently working on the PhD in engineering science at Shanghai Jiao Tong University, China. His research interests include image processing algorithm and signal processing.

Zhuang Fu received the PhD degrees in mechanical engineering from Harbin Institute of Technology in Harbin, China in 2000. From 2001 to 2002 he was a post-doctoral researcher at the Robotics Institute of the School of Mechanical Engineering at Shanghai Jiao Tong University in Shanghai, China, where he is now an associate professor. His research interests include image processing algorithm, mobile robot and electronical system design.