Proceedings of the 2006 IEEE/RSJ
International Conference on Intelligent Robots and Systems
October 9 - 15, 2006, Beijing, China

An Efficient Face Normalization Algorithm Based on Eyes Detection

Ganhua Li, Xuanping Cai and Xianshuai Li
Joint Center for Intelligent Sensing and Systems
National University of Defense Technology
Changsha, China
liganhua@nudt.edu.cn xpcai@nudt.edu.cn

Yunhui Liu
Dept. of Automation and Computer Aided Engineering
Chinese University of Hong Kong
Hong Kong, China
yhliu@acae.cuhk.edu.hk

***Abstract* - This paper presents an effective and efficient face normalization method based on eyes location. The face is rapidly detected based on boosted cascade of simple Haar-like features firstly. Then, the algorithm detects the position of pupils in the face image using the geometric relation between the face and the eyes. Finally, the algorithm normalizes the orientation, the scale and the grayscale of the face image. The experimental results demonstrated that this algorithm can detect and normalize the face image efficiently and accurately. The algorithm can be used in face recognition because the normalized faces can improve the recognition rate.**

Index Terms* - *Face detection, Face recognition, Eye location, Face normalization

I. INTRODUCTION

Face detection and recognition are being intensively investigated in recent years in computer vision and artificial intelligence for potential applications in person identification systems, surveillance and human-computer interactions, etc [1]. Practical applications of face recognition technologies in our daily life mainly involve embedded systems. Due to limited computation power in embedded systems, the face recognition algorithm must be so designed that it can produce the real-time efficiency without sacrificing the recognition rate.

Detecting the face in an image is the first and essential step in face recognition, so the efficiency and accuracy of face detection are crucial to performance of a face recognition system. Face detection [14] is a process of localizing and extracting the face region from the background. The detected face varies in rotation, brightness, size, and etc. in different images even for the same person. Those features are independent of face features and will affect the recognition rate significantly. One method to solve the problem is to normalize the detected faces. The objective of the face normalization is to reduce the effect of the useless, interferential and redundant information, such as background, hair, cloth and etc so as to enhance the recognition. And it normalizes the different images of the same person's face to the similar condition, which is about similar face contour, size, rotation and the brightness distribution. As a result, the normalized face images appear more consistent for the same person.

In order to normalize the face, the basic feature points are commonly selected by using the center points of the eyes. Many eye location algorithms have been presented before [2, 3, 4]. However, they are either computationally expensive or unable to precisely localize the double eyeballs. Li has proposed a fast location algorithm [15] based on geometric complexity, which mainly uses the horizontal grey histogram analysis to obtain the eyes' positions with less computational expense. But this method is suitable for face images that face the camera rightly. Cao proposed an algorithm for face orientation adjustment based on the eye location [16], which is mainly based on the pattern matching of the pupils in the face image. However, this method is computationally expensive.

In this paper, we propose an efficient algorithm for normalizing geometry and brightness of faces so as to improve the recognition rate and real-time efficiency. In this algorithm, we employ the Harr-like method proposed by Viola [5, 6] to detect the face region based on location of eyes. Using eyes for face detection has the advantages of being easy to detect and robust to image noises. Then, the face normalization is performed by two steps. First, the geometric normalization is carried out by a rotation based on the eyeball's location and an affine translation defined by locations of the eyeballs and the center of the mouth. Then, the brightness is normalized by the grey information. Since the algorithm is mainly based on the histogram analysis, it can normalize the face efficiently. The experiments carried out demonstrated that the proposed algorithm can efficiently and effectively normalize the face images. We have also conducted comparisons between the recognition rates of faces with and without face normalization using the EHMM (Embedded Hidden Markov Model) method. The experiments showed that after the normalization the face recognition rate was improved from 97% to 99% based on the ORL face image database.

II. ALGORITHM OUTLINE

For a face recognition system, it needs to detect the face region from the input images firstly and then compare it with the face features in the database. There are many factors affecting the result significantly, such as brightness, size, rotation and lot's other useless information. The face normalization is to reduce affects of factors that are independent of face features.

In the face normalization, we need to normalize the geometric information and brightness of the face. The geometric normalization is to rotate the face in the specified orientation and scale it to the same face, and the brightness normalization is to give a uniform distribution of brightness

1-4244-0259-X/06/$20.00 ©2006 IEEE

on the face. The rotation of the face is determined by the locations of two eyes and the center point of the mouth. The scale factor is calculated by the distance between the eyeballs. Using the eyes has the advantages that it is easy to detect their positions and the detection is robust to noises in images. Then, the brightness distribution is normalized to the same mean and variance based on the algorithm proposed by Liang [12].

The algorithm is outlined by the flowchart in Fig. 1. The details in each step will be explained in the coming sections.

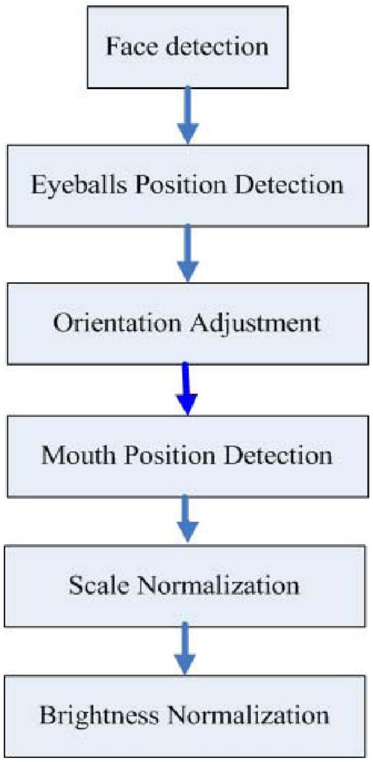


Fig.1 The flowchart of the normalization algorism.

III. Face Detection Based on Harr-Like Feature

Many algorithms have been developed for face detection [7, 8, 9]. The major methods include skin color detection method [10], pattern matching method [11], mosaic image method, statistical feature method and etc. In our application, we will use the face detection method developed by Viola based on an adaboost and cascade algorithm [4], which is considered as an important contribution for face detection. This algorithm is efficient and hence more practical for embedded applications. Since the emphases of our algorithm are to normalize the face detected, the face detection method is not presented in detail in this paper.

IV. Eyeballs' Localization

The eyeballs' positions are located mainly based on the brightness histogram using pattern matching. In order to reduce the computation cost, we use the brightness histogram and the bright feature to detect the eyeballs' position. In this algorithm, the grey image is processed directly. For simplicity, we define the left-top point of the image as the origin point of the image coordinate frame (Fig. 2).

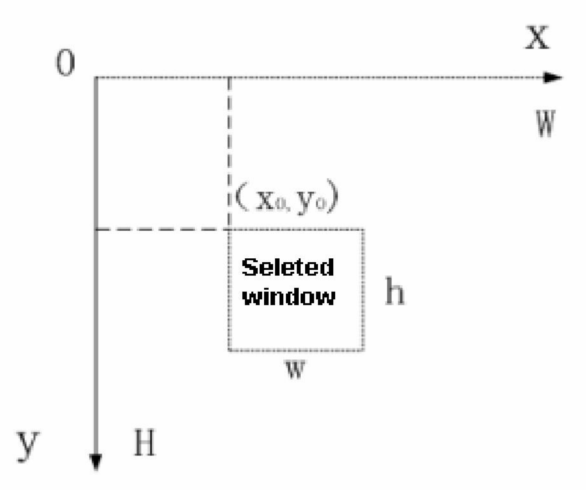


Fig. 2 The window selection for eyeball location.

It is well-known that the five sense organs' positions in human face follow the static rule. The human face obeys the "Three Parts-Five Eyes", which is a well-known Chinese old saying. "Three Parts" means that the face can be equally divided into 3 parts from the bottom of the hair to the mandible. "Five Eyes" means that the horizontal distance between the two eyeballs is about 5 times of the eye's width. In addition, there are other two important facts: (1) the eyes are in the middle of the head in the vertical direction, and (2) the horizontal length of an eye, the distance between two eyes and the width of the nose are similar in length. Based on those facts, the eye's region can be roughly estimated by selecting a candidate window. In order to cover all the eye regions, the rectangle window should be big enough. Denote height and width of the face image by h and w, respectively. We determine the parameters of the window based on extensive experimentation. As Fig. 2 shows, the parameters selected for the window are as follows: the coordinates of the left-top vertex are $\left(\frac{w}{7},\frac{h}{5}\right)$; the coordinates of the right-bottom vertex are $\left(\frac{10w}{21},\frac{2h}{5}\right)$, the height is $\frac{h}{5}$ and the width is $\frac{w}{3}$.

In the selected window, it is easily found that the brightness of the eyeballs and the eyebrow is darker than other regions. By this characteristic, we can analyze the histogram of the grey window region. Our algorithm firstly sorts all the pixels in the windows by the grey value. Secondly we choose the 5% parts with smallest grey value as the eyeball candidate regions, as image (b) of Fig. 3 shows.

$$T = h_{(x,y)}(5\% \times N) \tag{1}$$

$$f(x) = \begin{cases} 0 & x < T \\ 255 & x \geq T \end{cases} \quad (2)$$

The segment threshold T will be obtained by equation (1), where $h_{(x,y)}(i)$ is grey value of pixel with coordinate (x, y), and its sorted index number is i. N is the total number of the pixels in window. We can segment the eyeballs and eyebrow from the face. The image can be translated to binary image by setting accepted pixels with grey value 0 and setting refused pixels with value 255 as equation (2).

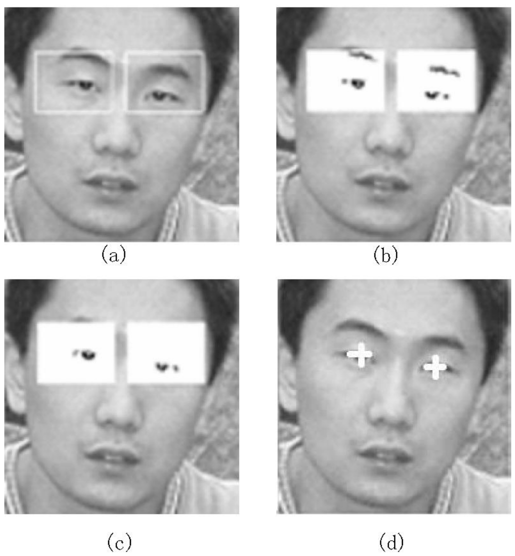


Fig. 3 The main steps result of eyeballs localization. (a) shows the selected windows by white rectangles. (b) shows the selected binary windows. (c) shows the result of eyeballs detection. (d) shows the eyeball centres detection result which was marked by the white cross line.

Project the binary window image to horizontal and vertical histogram. We can establish the histogram as equation (3) and (4). Commonly, the horizontal histogram has two minimum extremes. The lower one corresponds to the eye and the upper one corresponds to the brow. Then the brow can be removed from the window by selecting the lower extremes, as shown as the image (c) of the Fig. 4. The horizontal position of the eyeballs can be also obtained by the lower extremes. Then the vertical position of the eyeball obtained as same as the method of horizontal position.

$$pv(y) = \sum_{x=1}^{N} I(x, y) \quad (3)$$

$$pv(x) = \sum_{y=1}^{N} I(x, y) \quad (4)$$

V. Face Normalization

This section presents 4 major steps in the face normalization process, which is to normalize face images of the same person to those in similar orientation, brightness and size. Face normalization is an important process for improving the face recognition rate.

A. Orientation Adjustment

After the eyeballs localization, two center points of the eyeballs were detected. A line segment L can be obtained from the two points. Then, the face coordinate can be obtained by the line segment. The middle point of the line segment can obtain a basic centre point of the face coordinate, and the horizontal ordinate of the face coordinate can be set by line segment. We can get the acute angle θ between the line and the horizontal line. This step can adjust the pose of the face. Because the input image was a rectangle image, after the image is rotated of angle θ, the points out of the image will be removed and the points hollowed will be filled by the pixels before rotation. The images at the third line of Fig. 4 are the orientation adjustment results.

B. Mouth Centre Localization

The face image is adjusted now. In order to get a correspondent height of the face, we can establish the mouth window to obtain the center position of the mouth similar with the eyeballs localization. The top-left point's coordinate of the rectangle window is $\left(\frac{w}{4}, \frac{7h}{10}\right)$. The width of the window is $\frac{7w}{15}$ and the height is $\frac{h}{5}$. We establish the horizontal projection histogram for the image in the mouth window. The mouth horizontal position can be selected from the minimal value position of histogram. And the vertical position is as same as the vertical position of midpoint of the two eyeballs. The images at the third line of Fig. 4 show the mouth centre location results.

C. Scale Normalization

In order to let the face image be normalized more similar with the corresponding image in the face database, here we extract the rectangle region of the face image more precisely based on the basic point of the eyeballs' center points and the mouth's center point. The distance between the eyeballs is W_I, and the vertical distance between the eyeball and the mouth is H_I.The left-top point coordinate (X_L, Y_L) of the rectangle selected can be obtained by the equation (5) and (6). The width of the rectangle is $\frac{16}{9}W_I$, and the height is $\frac{5}{2}H_I$.

$$X_L = \begin{cases} X_c - \frac{8}{9}W_I & \text{if } (X_c - \frac{8}{9}W_I) > 0 \\ 0 & \text{if } (X_c - \frac{8}{9}W_I) < 0 \end{cases} \quad (5)$$

$$Y_L = \begin{cases} Y_c - \frac{3}{4}H_I & \text{if } (Y_c - \frac{3}{4}H_I) > 0 \\ 0 & \text{if } (Y_c - \frac{3}{4}H_I) < 0 \end{cases} \tag{6}$$

After the main face region is extracted, we can remove the useless information of the input face image, such as the most hair region, the background, and the cloth etc. At last we scale the image to the same size. We use 100×100 pixels in our system. The images at the fourth line of the Fig.4 show the selection and scale results.

D. Brightness Normalization

The face images captured at different times or positions often have different brightness. These commonly affect the recognition significantly. In order to reduce the affection of the brightness, we use a brightness normalization method to let the image have the same mean and the same variance value. If the width and the height of the image are *w* and *h*, the face image can be considered as a matrix $M[w][h]$.

The equation (7) and (8) are the mean and the variance of the input image. From the equation (9), we translate all input images to the same mean μ_0 and variance σ_0 based on the $\overline{\mu}$ and $\overline{\sigma}$, which was proposed by Liang [12].

$$\overline{\mu} = \frac{1}{w \bullet h} \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} M[i][j] \tag{7}$$

$$\overline{\sigma}^2 = \frac{1}{w \bullet h} \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} \left(M[i][j] - \overline{\mu}\right)^2 \tag{8}$$

$$M[i][j] = \frac{\sigma_0}{\overline{\sigma}} (M[i][j] - \overline{\mu}) + \mu_0 \tag{9}$$

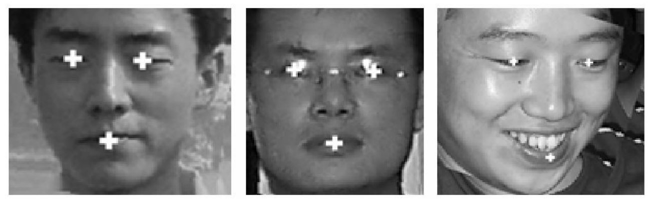
The images at the fourth line of the Fig.4 show the brightness normalization results. From the experimental results, we can find that the illumination's affection of the images can be reduced to a more likely condition with each other.

VI. Experimental Results

To verify the proposed method, we have implemented it and conducted experiments. In the experiments, all the face images were captured by a digital camera with 1,000,000 pixels. We first normalized the image by the method presented in this paper. Then, to demonstrate that the normalization algorithm can really improve the face recognition rate, we carried out comparison experiments of face recognition with and without face normalization. The algorithm for face recognition in the comparison experiments is based on EHMM （Embedded Hidden Markov Model）.

(a) The input image.

(b) The result of eye balls detection.



(c) The result of orientation adjustment and mouth

(d) The result of scale and brightness normalization.

Fig. 4 The three main normalization steps of three kind special face images. These input images are face detection results from different images with different conditions.

Fig.4 shows the results, obtained at the major steps of the algorithm, of face images of three people. The three input images were properly selected so that the eyeballs are not easy to detect. In the first image, it is hard to see the eyeballs of the person, and hence their localization is difficult. However, the result demonstrated that the algorithm can detect their positions with an acceptable accuracy. As shown in the left image in Fig. 4b, the detected position of the right eyeball slightly shifts to the left. In the second image (middle column in Fig. 14), the person wears glasses. It is shown that the algorithm can correctly localize the eyeball even with the glasses (the middle column of Fig. 4b). In the third image, the person does not face to the camera. The normalized images of the three images are shown in Fig. 4(d). As shown in the images, the normalized faces have similar sizes, orientation and brightness distribution.

Fig.5 The result of face detection. The red rectangles indicate the faces regions.

Fig.6 The results of face detection (upper), eyeballs localization (middle) and face normalization (lower).

In order to show the efficiency of the proposed method, we also investigated the computational time. Here we consider possible applications of this method in cellular phones. Most cameras used in cellular phones usually capture images with about 30,000 pixels. Fig. 5 is the input image with 640*480 pixels. It took 0.299529 second to detect three face regions by the Harr-like feature face detection algorithm. Fig.5 shows the face detection result. It took 0.004 second to 0.008(s) to normalize the face images in a window of 100*90 pixels. Fig.6 shows result obtained at the major steps. Assume that all the face images stored in database are of 100*90 pixels. It took 0.14657 second to recognize the face images from the database. The total computation time of each face is about 0.45(s). The experiments were conducted on an Intel Pentium Ⅳ processor running at 2.0 GHz with 512 RAM. Therefore, it is possible to use the normalization algorithm in embedded system.

In order to prove the improvement of our normalization method for face recognition, we have also done the recognition experiment of the original images of ORL face database. Fig. 7 shows some of the experiments results. The upper row images of (a) and (b) of Fig.7 are the original images from ORL face database, and the lower row images are normalization results. It's easily found that the normalization results are obviously more standard and more like each other than original groups, which can reduce the difficulty of recognition. We also compared face recognition of the original images with face recognition of the images normalized by our method. We used 200 images of 20 persons. The face images were trained and recognized based on DCT translation and EHMM (Embedded Hidden Markov Model). We used 5 images of a person to train and used the other 5 images to be recognized. The correct recognition rate is 97% with the original images. And it arrived to 99% with the image normalized by our method. It can prove that our method can improve the recognition rate of faces.

VII. CONCLUSIONS

This paper presented a fast algorithm for normalizing face images detected by Harr-like feature algorithm. The algorism can normalize the input images to images with similar brightness, size and orientation. We have implemented this algorithm and verified its performance by experiments, which demonstrated that this method can effectively improve the face recognition rates. The experiments also showed that the proposed normalization algorithm is efficient for real-time applications.

ACKNOWLEDGMENT

This work is partially supported by the NFSC (No. 60334010 and 60475029).

REFERENCES

[1] H. Wechsler et al., "Face recognition from theory to applications," *Proceeding of the NATO ASI*, Germany, Springer-Vrelag, 1998.

[2] J. II Choi, C. W. La, et al. "Face and eye location algorithms for visual user interface," *1977 IEEE 1st Workshop on Multimedia Signal Processing*, pp239-244. H. Simpson, Dumb Robots, 3rd ed., Springfield: UOS Press, pp. 6-9, 2004.

[3] K. M. Lam, and H. Yan, "Location and extracting the eye in human face images," *Pattern Recognition*, The Journal of The Pattern Recognition Society Volume 29, pp. 771-779, Number 5, May 1996.

[4] J. Huang and H. Wechsler, "Visual routines for eye location using learning and evolution," *IEEE Transaction on Evolutionary Computation*, Vo.4, Issue: 1, pp. 73-82, Apr 2000.

[5] P. Viola, and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE CVPR*, 2001.

[6] R. Lienhart, and J. Maydt, "An extended set of Haar-like features for rapid object detection," Intel Labs, Intel Corporation.

[7] K. M. Lam, and H. Yan, "An analytic-to-holistic approach for face recognition based on a single frontal view," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, No.7, pp. 673-686, 1998.

[8] X. Li, N. Roeder, "Face contour extraction form front-view images," *Pattern Recognition*, Vol.28, No. 8, pp. 1167-1179, 1995.

[9] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Interlligence*, Vol.20, No.1, pp. 23-38, 1998.

[10] G. Z. Yang and T. S. Huang, "Human face detection in complex background," *Pattern Recognition*, 27(1), pp. 53-63, 1994.

[11]T. Sakai, M. Nagao, and S. Fujibayashi, "Line extraction and pattern detection in a photograph. *Pattern Recognition*, Vol. 1, pp. 233-248, 1969.

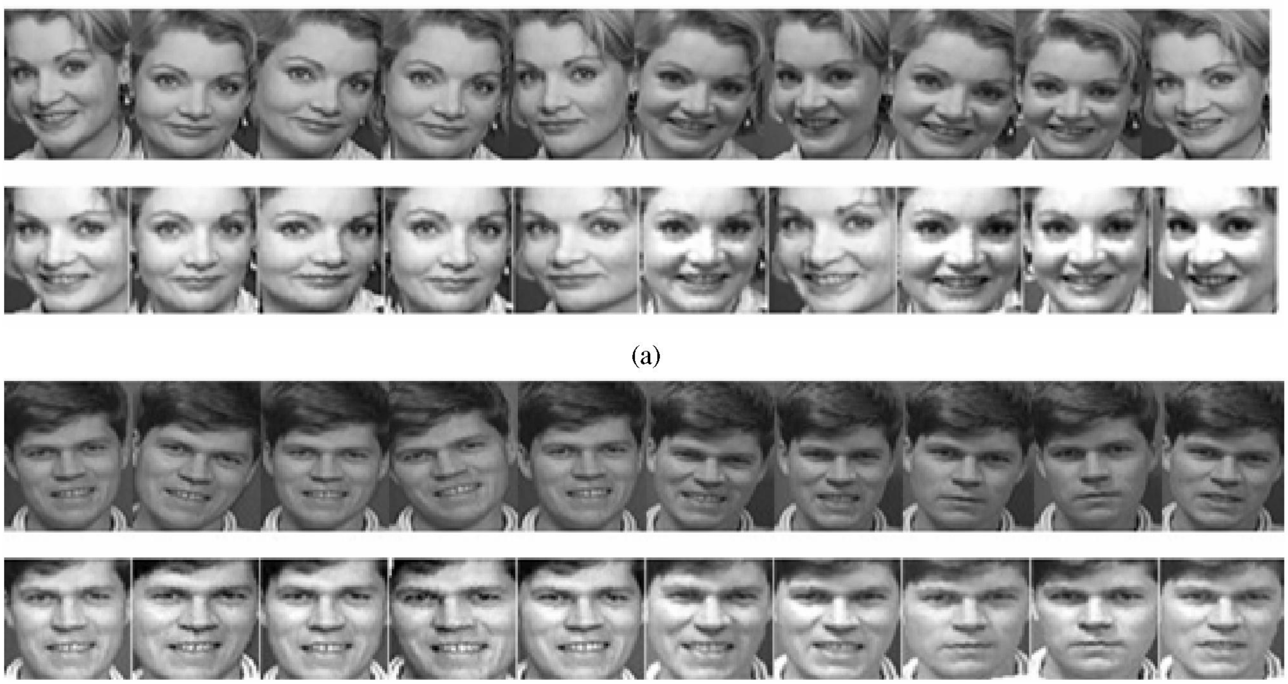
[12]L. L. Hong, H. Z. Ai, K. Z. He and B. Zang. "Face detection based on the matching of multiple related templates," *Journal of Software*, Vol. 12, No. 1, Dec 2001.

[13]V. Nefian Ara and H. Hayes III Monson, "Face recognition using an embedded HMM," *IEEE International Conference Audio Video Biometric based Person Authentication*, pp. 19-24, March 1999.

[14] M. H. Yang, D. Kriegman and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans Pattern Analysis and Machine Intelligence*, 4(1), pp. 34-58, 2002.

[15] W. J. Li, J. Xu and S. J. Wang, "A fast eye location algorithm based on the geometric algorithm based on the geometric complexity," *Proceeding of the 5th World Congress on Intelligence Control and Automation*, Hangzhou, P. R. China, pp. 4105-4107, June 15-19, 2004.

[16] W. M. Cao, H. Feng, G. Xiao, Y. B. Gu and S. J. Wang, "Study of An algorithm for face pose adjustment based on eye location," *Proceeding of the 5th World Congress on Intelligence Control and Automation*, Hangzhou, P. R. China, pp. 4190-4194, June 15-19, 2004.



(a)

(b)

Fig.7 The upper images of the (a) and (b) are the input image, and the lower images are the normalization results.