Fast face detection algorithm based on improved skin-color model and adaptive threshold

Zhenxue Chen, Faliang Chang, Chengyun Liu
(Shandong University,
School of Control Science & Engineering, Jinan 250061, China)

Abstract: Automatic face recognition is one of the most challenging tasks in fields of computer vision and pattern recognition, and face detection is the first critical step in full automatic face recognition system. The skin-color feature is an effective feature, but this feature is interfered easily. This paper proposes a method of face detection from a picture based on an improved skin-color model. Firstly, an improved "reference white" method is used to remove the interference of non-skin- color region; and then design Color-Classifier based on statistic large number of skin-color pixels and detect each pixel in color picture is skin-color or non-skin-color through the Color-Classifier; finally, detect face on the candidate regions and remove the non-face regions, and then locate the face regions. Experimental results show that the algorithm can effectively detect face with skin-color interference under complex background.
Keywords: Face detection; skin-color model; skin-color classifier; reference white; non-face regions

1. Introduction

Human face detection is to search human faces from an input image, and output the description including the position and size information of the faces. As a special case of pattern recognition, face detection has been obtained great attention [1]-[3]. Many scholars are researching on face detection and recognition system, and the various approaches are proposed for face detection: Stan Z. Li proposed an algorithm for learning a boosted classifier for achieving the minimum error rate [4]. Christophe Garcia presented a novel face detection approach based on a convolutional neural architecture [5]. Christopher A. presented a face detection method using spectral histograms and support vector machines (SVMs) [6]. Rein-Lien Hsu proposed a face detection algorithm for color images in the presence of varying lighting conditions [7]. Wen-Kwang proposed a novel face detection method based on the MAFIA algorithm [8]. Chiunhsiun Lin developed an efficient face detection scheme that can detect multiple faces in color images with complex environments and different illumination levels [9]. Songyan Ma presented a color based on the Adaboost face detection method [10].

In fact, skin-color feature is a basic and effective feature of human faces, and is widely used in face detection. In the past decades, many researches rebuilt skin model in different color spaces, and proposed many face detection algorithms based on skin-color [9]-[12]. But skin-color feature is susceptible to interference caused by non-face regions that are similar to skin color. Many algorithms can apply only to simple background cases, but the false detection rate is higher in complex background. This paper proposes an improved algorithm based on skin-color to detect human face, and this algorithm is effective to locate face regions from complex backgrounds.

2. Design of skin-color classifier

The skin-color classifier is shown as Fig. 1. According to the skin-color classifier, we can decide that each pixel in color image is skin-color or non-skin-color. There are two parts: establishment of skin-color model and segmentation of skin-color region.

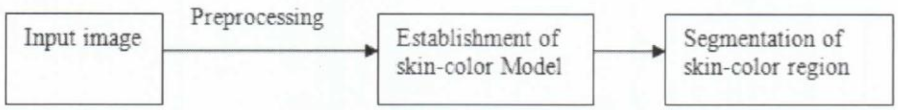


Fig. 1 Skin-color classifier

2.1 Preprocessing

The input images are often affected by the changes of light, and this will affect the detection result. So before the detection, the image should be given light compensation. This paper presents a light compensation technique called "Reference white" [13]. Firstly, this method sorts all pixels' values of the input image from small to large, and then selects the top 5 percent pixels. When the number of these pixels is sufficiently large (>100), these pixels are regarded as the "reference white". Then R, G and B components of the input color image are adjusted by the "reference white".

In fact, "reference white" makes the detection results even worse sometimes. So, in this paper, an improved "reference white" method is given. The detection results show that this method can avoid the skin-color-similar regions effectively. The algorithm is carried out as follows:

Step 1: Judge whether the image needs "reference white" treatment or not. Firstly, the input colorful image is transformed to gray style, the conversion formula is:

$$Gray = 0.3 * R + 0.59 * G + 0.11 * B$$

where R, G and B are red, green and blue components of the input image, respectively, and $Gray$ is the gray-scale image.

Then find out the maximum and minimum value of the gray-scale image, let $\max\_Gray$ stores the maximum value and $\min\_Gray$ stores the minimum value. If $\max\_Gray = 255$ and $\min\_Gray = 0$, the image does not need "reference white" treatment; else, go to Step 2.

Step 2: Add up each gray-level of R, G and B components of the color image, respectively, and put them in array $level_x[256]$, where $x = R, G, B$, respectively.

Step 3: Calculate the probability of each gray-level of RGB components, and put them in array $p_x\_level[256]$. The formula for i gray_level of x component is:

$$p_x\_level[i] = \frac{level_x[i]}{N}, \quad i = 0, 1, \cdots, 255 \tag{1}$$

where N is the total pixel numbers of the input image.

Step 4: Compute the cumulative probability of R, G and B components, respectively. The cumulative probability from 0 to i of x component is calculated as followed:

$$p_x\_total[i] = \sum_{j=0}^{i} p_x\_level[j] \tag{2}$$

Step 5: Change i from 0 to 255: if $p_x\_total[i]$ is greater than 5% and $p_x\_total[i-1]$ is less than 5%, consider the value of i as "reference black ($refB_x$)", that is to say, $refB_x = i$; if $p_x\_total[i]$ is greater than 95% and $p_x\_total[i-1]$ is less than 95%, consider the value of i as "reference white ($refW_x$)", that is to say, $refW_x = i$.

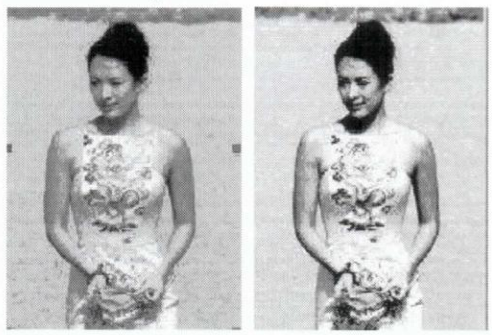
Step 6: Do "reference white" treatment to all pixels of the input image. Let the given pixels:

$f=\{f_i\}_{i=1}^{N}$, the treatment for f_i is:

$$g_{ix}=\begin{cases}0 & f_i<refB_x\\ \dfrac{255*\left[\ln f_i-\ln refB_x\right]}{\ln refW_x-\ln f_i} & refB_x<f_i<refW_x\\ 255 & f_i>refW_x\end{cases} \tag{3}$$

g_{ix} refers to the *i*th pixel of x component.

The processing result is shown in Fig.2.



(a) (b)

Fig.2 “reference white” processing result

(a) Original image; (b) “reference white” processing image

2.2 Establishment of the skin color model

The color models commonly used are: RGB, YCbCr, HSV, YIQ, etc. In RGB space, there are three primary colors: (r,g,b), in which the luminance information makes the detection vulnerable to the light changes. But the luminance component is separated in YCbCr model, and can be got by linear transformation of RGB model. So, the YCbCr model is chosen. The color space conversion of the input image is as follows:

$$\begin{cases}Y=0.299R+0.587G+0.114B\\ Cb=-0.169R-0.331G+0.500Y\\ Cr=0.500R-0.419G-0.081B\end{cases} \tag{4}$$

As the Ref. [9] shows that the skin colors of people of different races and different ages are gathered in a small range, so a suitable skin color model can established in the color space. Through the analysis of 3150370 skin color pixels, color clustering in the YCbCr color space shown in Fig.3 is in a smaller area.

Different skin color have the same two-dimensional Gaussian model $G=(m,C)$, where m is the mean and C is the covariance matrix [13].

$$m=E(X)=\frac{1}{n}\sum_{i=1}^{n}x_i \tag{5}$$

$$C=E\left[(X-m)(X-m)^T\right]=\frac{1}{n}\sum_{i=1}^{n}(x_i-m)(x_i-m)^T \tag{6}$$

where $x_i=(Cb,Cr)^T$ is color values of pixel i in the training samples, n is the number of color pixels.

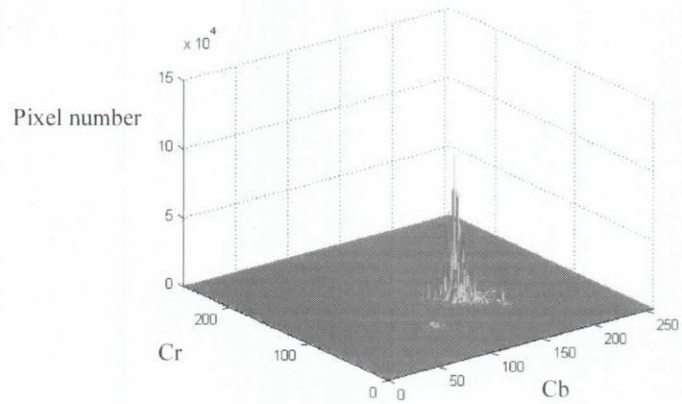


Fig.3 the distribution of skin pixels in CrCb space

After a statistic of 3150370 skin pixels from 200 color face images, we can get:

$$m = [117.4361, 156.5599], \quad C = \begin{bmatrix} 160.1301 & 12.1430 \\ 12.1430 & 299.4574 \end{bmatrix}$$

In this skin color model, the probability of each pixel of input color image can be got, and the output of the image is the skin likelihood image. The probability is calculated as Eq. 7.

$$p(Cb, Cr) = \exp\left[-0.5(x-m)^T C^{-1}(x-m)\right] \tag{7}$$

where $x = (Cb, Cr)^T$.

2.3 skin-color segmentation

After the skin likelihood image is obtained, image segmentation will be completed according to a certain threshold. There, threshold selection is a key technique. If the threshold selection is too large, you may miss a lot of color areas; if the threshold selection is too small, it will contain a large number of non-skin regions [13]. In the paper, the adaptive threshold method is used and described as follow:

Setp1: Set the changing regions of the threshold at $255 * (0.6 \sim 0.1)$, define iteration times as M. So, we can define two arrays and two variables: the first array can be used to save the changing of skin color dots, as $Change[M]$; the second array can be used to save the skin image after binarization, as $Array\_bin$; the two variables can be used to save the counts of skin pixels, as $Sum1$ and $Sum2$, and initialize to $Sum1 = Sum2 = 0$;

Setp2: Set threshold $Temp\_threshold = 255 \times (0.6 - 0.1 \times index)$, $index = (0, 1, \cdots, M-1)$, and then make skin color image binarization. And the binarization results are saved in array $Array\_bin$;

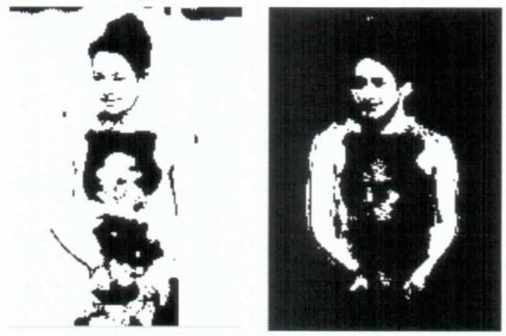
Setp3: Sum the all counts which pixels are 255 in array $Array\_bin$ and save the counts to $Sum1$;

Setp4: Sum the counts of change of skin color pixel as threshold is changed at Step3, $Change[index] = Sum1 - Sum2$, and set $Sum2 = Sum1$;

Setp5: Make $index = index + 1$, if $index < M$, go to Step2, else, go to Step6;

Setp6: Compute $Best\_index = \arg\min_{index=0,1,\cdots,M-1} Change[index]$, so, the adaptive threshold can be gotten from $Adap\_threshold = 255 \times (0.6 - 0.1 \times Best\_index)$.

In this paper, according to testing results of training sets. $M = 10$. The result of adaptive threshold segmentation is shown in Fig. 4.



(a) (b)

Fig. 4 Results of adaptive threshold segmentation

(a) The result of directly skin color segmentation of the original color image; (b) the result of skin color segmentation after the treatment of the "reference white"

Fig. 4 shows that most of the skin regional interference has been filtered out, after "reference white" skin-color processing.

3. Face Detection

3.1 Morphological processing

The color images obtained include noise and interferences, so they should be dealt form morphology. The common morphological processing includes erosion, expansion, opening operation and closing operation. In this paper, the opening operation and closing operation are used to process the previous results obtained. Opening operation is mainly used to remove small areas which are smaller than the structuring element; and closing operation can be used to fill small holes which are smaller than the structuring element.

3.2 Face region segmentation

The skin color region usually contains the neck and its below region, and the face should be segmented from these regions. The specific methods are: Firstly, the edge image *A* can be obtained by canny edge detection processing; Then, to take negation operation to *A* and obtain image *B*; Finally, to do "and" operation between image B and skin-color image, and obtain image *C*. The image *C* is skin-color image of face region [14].

However, experiment results show that there is a problem, that is to say, with many faces in the image, some color areas contain the neck and its below region, while the other color regions is not included. So, they should be treated, respectively. A two-step method of face segmentation method is used in this article:

Step 1: To compute the length-to-width ratio on the detected skin-color region. If the length-to-width ratio is less than threshold *T*, we can conclude that the skin-color region contains only face; otherwise, we can conclude that it includes the neck region. After calculation of 200 human face images, we found that the appropriate value of *T* takes about 1.6.

Step 2: To the region which have been identified include only the skin color region of face, no treatment will be done; otherwise to the region with the neck skin areas, face segmentation processing as the Ref. [14] will be done.

3.3 Remove non-face regions

In this paper, the non-face regions will be removed based on the following four criteria mainly:

1) According to the size of the region share

Share of the regional area is the ratio of total number of color pixels in the region accounted for that number in the entire image. Generally believed that the regions with the ratio which is more than 3 / 4 (according to experience and statistics) or less than 1 / 50 (according to experience and statistics) belong to non-face region, should be removed.

2) According to the length-to-width ratio of skin area

Typically, the length-to-width ratio of human face is about 1 (according to experience and statistics). By experiments of 200 human face images, the regions with the ratio which is less than 0.8 can be removed, while the regions with the ratio which is greater than 1.6 can also be removed in the case of having removed the neck and the below region [23].

3) According to ellipse area

The ellipse area criterion is $Se = \frac{4N}{\pi l_1 l_2}$, where N is the total pixels number of color region, and l_1 and l_2 is the axis length of the smallest external ellipse of the region.

According to the ellipse area criterion, the probability of each skin color region belongs to face region can be calculated. If the Se value of the tested region is greater than threshold 0.7, the region is considered containing the human face [24].

4) According to hair region

For a color image, Y=0.30R+0.59G+0.11B is defined. When Y <40, according to testing results of training sets. the region may be hair. The detection result is shown in Fig.5.

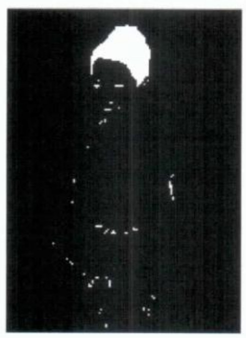


Fig.5 Hair region detection result

The non-face region can be removed, when the method is below: For the detected skin color rectangle region, the number of possible hair pixels is recorded as *Hair_count1*. Because there may remain hair region above the detected skin color rectangle region, so it is still necessary to detect the area in which the length equals to 1/3 of the rectangle region and width equals to the rectangle region. The number of possible hair pixels of this area is recorded as *Hair_count2*. The sum is *Hair_count*:

$$Hair\_count = Hair\_count1 + Hair\_count2$$

Then the ratio of the hair region to the candidate face region is calculated as:

$$Rate = Hair\_count / Area$$

where *Area* is the area of the candidate face region.

An experiment of 200 human face images shows that the ratio of the hair region to human face region is more than 1/50. So the region that isn't satisfied this condition can be removed as non-face regions.

3.4 Face Location

After removing the non-face region, the region remained can be treated as face region. The region can be calibrated as a rectangle, whose center is the centroid of the face region and its width and height are as the bounding rectangle of the face region [25].

The centroid of the region can be gotten by Eq. 8.

$$\bar{x}=\frac{1}{A}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} i\times f(i,j)\,,\ \bar{y}=\frac{1}{A}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} j\times f(i,j) \tag{8}$$

where $A=\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} f(i,j)$.

According to the centroid position and the width and height of the rectangle, human face is located in original image and is shown in Fig.6.

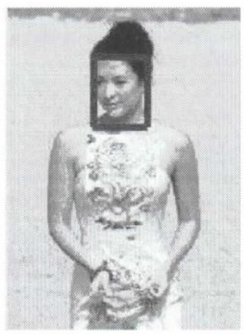


Fig.6 Result of face detection

4. Experimental results

In order to verify the validity of the algorithm, 269 color images have been collected randomly for our experiments, which have 137 images (include 427 faces) in simple scene and 132 (include 716 faces) images in complex scene. Moreover, we also use some parts of the 'FERET face database' in which contain 700 color images corresponding to 100 people's faces. Face images in this database have mainly different facial poses, such as frontal view faces and deflection view faces. Using *OpenCV* functions, the algorithm is realized in *VC++*, the detection results are shown as Fig.7 and Fig.8.

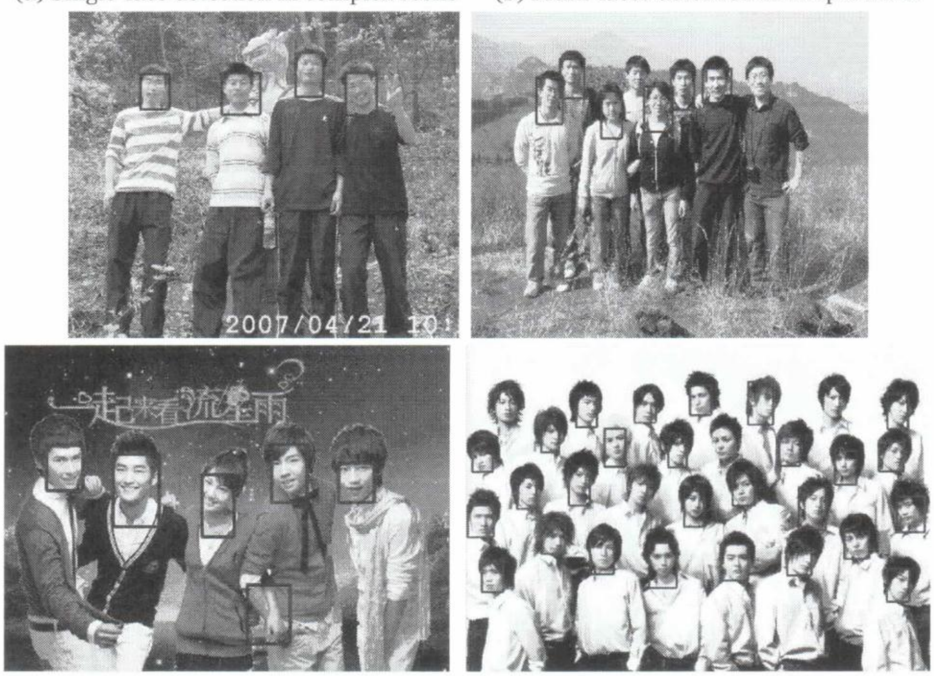
The detection rates are listed in Table 1, which shows that the detection performances of improved skin-color algorithm. We can draw a conclusion that the method of this paper maintains better detection performance. The algorithm has fewer restrictions on face, and wearing glasses or not, front view or side view, rotation or not, all these situations have little effect on detection results.

Table 1: Face detection data table

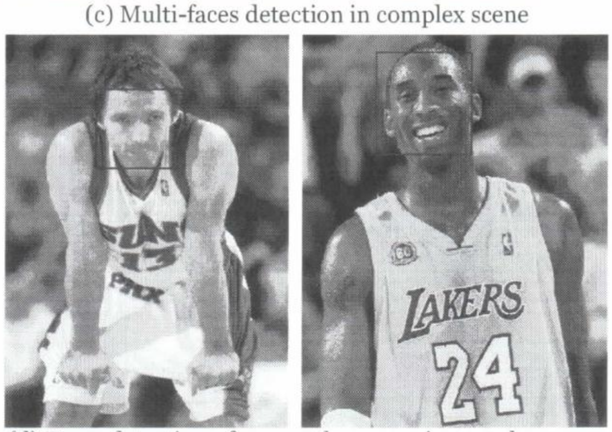
Testing scene	Numbers of face	Correct number	Missed number	False detection number	Detection rate (%)
Simple scene	427	415	12	20	97.2
Complex scene	716	655	61	94	91.5
FERET database	700	676	24	15	96.8

To demonstrate the effectiveness of the modified "reference white", an experiment on face detection in simple scene with/without "reference white" is listed in Table2. It is obvious that method with "reference white" has higher detection rate.

(a) Single-face detection in complex scene   (b) Multi-faces detection in simple scene

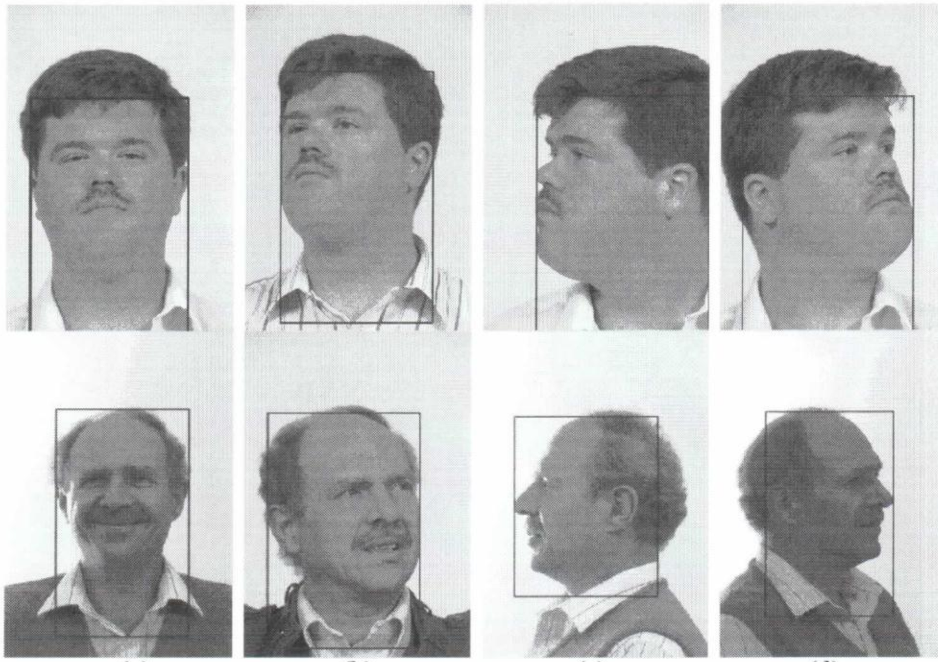


(c) Multi-faces detection in complex scene



(d) Faces detection of some other races in complex scene

Fig.7 Results of face detection in different situations



(a) (b) (c) (d)

Fig. 8 Face images with different poses by using adaptive thresholds
(a) Neutral frontal pose; (b) deflection 30°; (c) left deflection 90°; (d) right deflection 90°

Table 2: Experiment with/without "reference white" in simple scene

With/without "reference white"	Numbers of face	Correct number	Missed number	False detection number	Detection rate (%)
Without "reference white"	427	391	36	45	91.6
With "reference white"	427	415	12	20	97.2

Currently, our detection algorithm is running on a Pentium IV-2.26 GHz PC. The experimental time data in complex scene are listed in Table 3, which shows that the speed performances of several methods differ greatly from a time consuming standpoint. It is obvious that the improved skin-color algorithm maintains better time consuming performance for face detection, therefore the method of this paper is a real-time method. The most significant point is the discrepancy of detection rate is not bigger.

Table 3: Elapsed time (unit: ms) and Identification rate (%) using different methods for test

Detection method	Genetic Algorithm [18]	Artificial Neural Network [9]	Template Matching [14][19]	Support Vector Machine [6] [16]	Adaboost [10]	Local Binary Patterns [13] [22]	YCbCr [26]	Proposed method
Computational cost	602	912	831	822	663	774	511	287
Detection rate	87.5	92.2	89.7	86.7	91.3	94.1	94.8	91.5

To show the robustness of the proposed method, we performed a noise sensitivity test. The simple scene test is created by adding Gaussian noise with different SNR values. Table 4 shows the results. As the rate of noise increases, the rate of correct detection decreases. However, the proposed system shows an average 93.0% of correct detection on the SNR −3dB noise added test images. This shows the robustness of the proposed method to the noise.

Table 4: Results of sensitivity analysis

SNR (dB)	3	0	-3	-5	-10
Total rate (%)	97.2	95.8	93.0	86.9	68.1
Correct number	415	409	397	371	291

5. Conclusions and future works

The face detection has significant promotion in research areas such as face recognition, facial expressions recognition, video monitoring, status authentication, etc. The face detection method based on skin-color is small elapsed time and not affected by facial expressions and image rotation. The method is limited when the color of the background or human clothes is similar to skin-color. For the same reason, if there is too much skin uncovered, it affects the detection result also. So, this paper proposes a face detection algorithm based on improved skin-color model, which makes up these limitations: (1) An improved "reference white" method is used to remove the interference caused by changes of Illumination; (2) In "remove non-face regions" section, this paper proposes an effective method to remove the interference caused by the case where there are too much skin uncovered.

For the false detection in complex scene, the future research is how to reduce false detection rate. A possible method is to fuse the skin-color method and template-matching method, and combine the location of eyes and mouth to detect human face.

Acknowledgment

This work has been supported by National Natural Science Foundation of China (61203261, 61273277 and 60975025), Natural Science Foundation of Shandong Province of China (ZR2012FQ003) and State Key Laboratory of Robotics and System (HIT) (SKLRS-2010-MS13).

References

[1] X. Z. Zhang, Y. S. Gao. Face recognition across pose: A review. *Pattern Recognition*, 2009, 42(11): 2876-2896.

[2] Kakumanu P, Makrogiannis S. Bourbakis N. A survey of skin-color modeling and detection methods. *Pattern Recognition*, 2009, 40 (3): 1106-1122.

[3] M. H.Yang, David Kriegman and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2002, 24 (1): 34-58.

[4] Stan Z. Li, ZhenQiu Zhang. Float Boost Learning and Statistical Face Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004, 26 (9): 1112-1123.

[5] Christophe Garcia, Manolis Delakis. Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2004, 26 (11): 1408-1423.

[6] Christopher A. Waring and Xiuwen Liu. Face Detection Using Spectral Histograms and SVMs. *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, 2005, 35 (3): 467-476.

[7] Rein-Lien Hsu, Mohamed Abdel-Mottaleb and Anil K. Jain. Face detection in color Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*,2002, 24 (5): 696-706.

[8] Wen-Kwang Tsao, Anthony J. T. Lee and Ying-HoLiu. et. al. A data mining approach to face

detection. *Pattern Recognition*, 2010, 43(3): 1039-1049.

[9] Chiunhsiun Lin. Face detection in complicated backgrounds and different illumination conditions by using YCbCr color space and neural network. *Pattern Recognition Letters*, 2007, 28(16): 2190-2200.

[10] S. Y. Ma, T. C. Du. Improved Adaboost Face Detection. *2010 International Conference on Measuring Technology and Mechatronics Automation*, 2010, 2: 434-437.

[11] Chiang C. C., Huang C. J. A robust method for detecting arbitrarily tilted human faces in color images. *Pattern Recognition Letters*, 2009, 30(6): 2518-2536.

[12] Kyoung-Mi Lee, Component-based face detection and verification. *Pattern Recognition Letters*, 2008, 29 (3): 200-214.

[13] Timo A., Abdenour H. and Matti P. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2006, 28(12): 2037-2041.

[14] Wang Jizeng, Yang Hongmei. Face Detection Based on Template Matching and 2DPCA Algorithm. *2008 Congress on Image and Signal Processing*. 2008, 4: 575 – 579.

[15] Jae Hee Park, Hae Chul Choi and Seong Dae Kim. Bayesian face detection in an image sequence using face probability gradient ascent. *2005 IEEE International Conference on Image Processing*,2005, 2: 346-349.

[16] Hyungkeun Jee, Kyunghee Lee and Sungbum Pan. Eye and Face Detection using SVM. *2004 Conference Intelligent Sensors, Sensor Networks and Information Processing*, 2004: 577-580.

[17] Huang, J. Gutta, S. and Wechsler, H. Detection of human faces using decision trees. *1996 Proceedings of the Second International Conference on Automatic Face and Gesture Recognition*, 1996: 248-252.

[18] Z. M. Liu, J. L. Zhou and G. Q. Xiong, et. al. An efficient approach for face detection based on genetic algorithms. *IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino Expositions*, 2002, 2: 1127-1131.

[19] A. P. Chen, L. Pan and Y. B. Tong. et. al Face Detection Technology Based on Skin Color Segmentation and Template Matching. *2010 Second International Workshop on Education Technology and Computer Science*, 2010, 2: 708-711.

[20] C. Huang, H. Z. Ai and Y. Li, et. al. High-Performance Rotation Invariant Multiview Face Detection. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 2007, 29 (4): 671-686.

[21] Timo O., Matti P. and Topi M. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2002, 24 (7): 971-987.

[22] Timo A., Abdenour H. and Matti P. Face Recognition with Local Binary Patterns. *The Eighth European Conference on Computer Vision*, 2004, 3021: 469-481.

[23] D.N. Vizireanu. Generalizations of binary morphological shape decomposition. *Journal of Electronic Imaging*, 2007, 16 (1): 01302, p. 1-6.

[24] R.M. Udrea, N. Vizireanu. Iterative generalization of morphological skeleton. *Journal of Electronic Imaging*, 2007, 16 (1): 010501, p. 1-3.

[25] Chiunhsiun Lin. Face detection in complicated backgrounds and different illumination conditions by using YCbCr color space and neural network. *Pattern Recognition Letters*, 2007, 28(16): 2190-2200.

[26] Rein-Lien Hsu, Mohamed Abdel-Mottaleb and Anil K. Jain. Face detection in color Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*,2002, 24 (5): 696-706.