FEATURE-BASED HUMAN FACES DETECTION IN COLORED COMPLEX BACKGROUND IMAGES

Chin-Shyurng Fahn and Yun-Zhi Lai

Department of Electrical Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan 106, Republic of China
E-mail: csfahn@mouse.ee.ntust.edu.tw

ABSTRACT

In this paper, we present an efficient faces detection system that can locate human faces in colored complex background images. The design philosophy of this faces detection system is based on the knowledge of color information and the geometrical relations of facial features. At the first stage, we extract possible eye blocks by means of edge detection. Then the number of possible eye blocks is reduced using the color information. At the next stage, we propose some rules for pairing these blocks to generate possible face candidates by the geometrical relations of facial features. Finally, we again employ the color information and other facial features to eliminate false candidates. The experimental results prove that our approach is feasible and very quickly.

Keywords: Face detection, feature-based location, color information, facial feature, complex background.

1. INTRODUCTION

In the recent years, automatic identification of human faces has drawn more and more attention from researchers with the progress of image processing and computer vision technology. It has been developed and applied to many areas for specific purposes, such as security systems and criminal identification systems. Many approaches to identifying human faces have been reported in literatures [1-4]. However, most of these researches have already known where human faces are. But, for an automatic human faces identification system, it is executed in two major phases: the first is to automatically locate face-like regions, and the next is to identify who the faces are. To develop such a human faces identification system, automatic location of human faces plays a crucial role. It is also required for other kinds of applications using human faces. If no face is detected well, the following steps always cannot work. Thus, the research of automatically locating human faces becomes a hot topic.

Several different techniques for automatic location of human faces in color or gray images have been proposed. In [5], they use color and shape information to locate possible face regions. They also employ a modified watershed method and min-max analysis to extract facial features. In [6], human faces are located by the aid of motion and color information. They use motion information as the first step to locate face regions, and then apply color information to extract facial features. In [7], the step of extracting face-like regions is based on color and texture information. First, they enhance images by using color information, then textural features are obtained from a space gray-level dependence matrix, and facial parts are detected with textural models. In [8], a neural network is applied to classify skin colors, and three shape descriptors of a lip feature are used to further verify. In [9], a hierarchical knowledge-based method consisting of three levels is adopted to locate unknown human faces. And in [10], a human head is detected from the scene in the form of multiresolution mosaic. Then the central part of a face is detected from the head region using the mosaic, and the precise position is determined based on the histograms of eye and nose regions. In [11], they employ a multilayer perceptron classifier to decide whether human faces exist. Additionally, there are some methods with geometrical relations of facial features, as presented in [12, 13]. Other methods using deformable templates in [14] or silhouettes of human faces in [15] also have been proposed.

Of the previous work, there is still a room for improvement. In [12], the proposed approach cannot detect multiple faces in an image. In [8], the face can be rotated by 10°~15° only, and in [9], the face cannot be tilted too much. If the orientation of a face is beyond the limit, the detection approaches will fail. Moreover, the shape of the faces that can be well detected is confined to be elliptic in [5, 6]. Compared to other literatures, the approaches in [9, 15] are tedious and spend much more execution time. In this paper, we propose a human faces detection system that can locate more than one face of an arbitrary shape in an image. The rotation angle of a face may be ranged from -90° to 90°, and the tilt angle from -25° to 25°. The design philosophy of such a system is based on the knowledge of color information and the geometrical relations of facial features, because the former is robust and the latter is invariant to achieve the detection in a complex background. The experimental results reveal that the efficiency of this detection system is very satisfactory.

On the whole, our developed system consists of two stages, including the generation and verification of face candidates. At the generation stage, we first process the image by edge detection and noise removal, followed by extracting possible eye regions. The basic ideas of the first step are that eye regions and skins generally have different colors and each eye region has an olive shape distinguished from the shapes of other regions. According to these two characteristics, an edge detection method can exactly extract eye regions. Thus, we obtain the fundamental features of detecting human faces. After this step, face candidates can be generated from mating any pair of the possible eye regions.

At the next stage, we have two principal methods to verify possible face regions. One employs color information, and another employs geometrical relations of facial features. Because skin colors are spread in a special color range, we apply this information to accomplish the primary verification. And the geometrical relations of facial features are almost unchangeable, for instance, the related positions of facial features and the ratios of the distances between them. This information is adopted to complete the final verification.

2. EYE CANDIDATES GENERATION

In our system, the first stage is to extract possible eye pair candidates. At this stage, we preprocess the original color image in the gray scale by using HSI color palettes. Then a block searching and pairing method is employed to generate possible eye candidates. This preprocessing stage is composed of three steps, including contrast enhancement, edge detection, and noise removal.

2.1 Preprocessing Stage

When we look at faces, it is observable that eyes have a special shape in faces and their colors are different from the skin color. At the preprocessing stage, we apply these two characteristics to extract possible eye regions from a face. In some literatures, they adopt threshold techniques to process images; however, it sometimes causes the loss of features since the illumination of an arbitrary image is different from each other. This makes it hard to determine proper thresholds for processing images. To solve this problem, we use edge detection to extract the features.

2.1.1 Contrast Enhancement

In some cases, the illumination of an entire image may be too dark or bright. If we encounter these situations, the step of edge detection may not clearly separate eye regions as we expect. In order to extract possible eye regions effectively, we perform the original image by contrast enhancement before edge detection. The contrast enhancement equation for the new gray level of a pixel is depicted below:

$$L = [(L_i - L_{\min})/(L_{\max} - L_{\min})] * 255,$$

where L_i is the original gray level of the pixel, $L_{\min}$ is the minimal gray level of the image, and $L_{\max}$ is the maximal gray level of the image.

After this process, images that have too dark or bright illumination can be improved. Then we continue the following process.

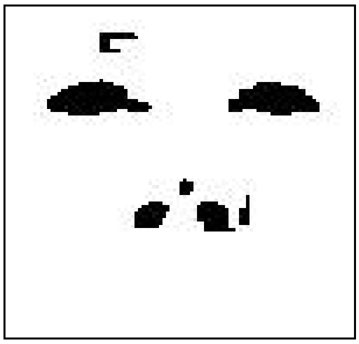
2.1.2 Edge Detection

Edge detection is an essential operation of detecting significant local changes in an image. The gradient is a useful function to measure such changes, and each image can be considered as an array of sampling the continuous function of its intensity. A way to avoid having the gradient calculated about an interpolated point between pixels is to use a 3*3 mask for calculating gradients. In our system, the Sobel operator is employed to measure the magnitude of the gradient. Following the above process, the edge-enhanced image is then converted into a binary image by setting the values of the pixels with larger gradients to zero and the others to one.

2.1.3 Noise Removal

Through the binarization process, the resulting image contains many individual components. Most of them are small needless noises. We want to remove the noises but keep other useful segments as many as possible. In order to achieve this purpose, we first apply a morphological dilation operation to expand each component, then two morphological erosion operations are followed to shrink the components, and a morphological dilation operation again expands them at last. The above morphological operations are all performed by a 3x3 square structuring element. In this manner, the first dilation operation is used to prevent the two subsequent erosion operations from deleting smaller components that comprise useful pixels. The last dilation operation is used to restore the size of each component. Figure 1(a) shows a binary image before the noise removal process, and Fig. 1(b) shows the processed result. From these figures, we can see that most neighboring pixels are merged into larger components and the pixels that constitute smaller components in face regions are eliminated.

(a)



(b)

Fig.1 Illustration of the morphological noise removal approach: (a) an original binary image; (b) the resulting image.

2.2 Possible Eye Blocks Labeling

Our labeling algorithm is similar to taking a walk in a maze. We scan the image from left to right and bottom to top. When we find a pixel associated with value zero, i.e., an edge pixel, we repeatedly scan all the edge pixels belonging to the eight neighbors of the currently visited edge pixel. We also set the visited pixels to be value one in order to prevent from revisiting. After a connected component has been completely checked, we create a new linked list to record its information and continuously find the next component until the whole image has been scanned. In this way, we have a linked list to record each connected component that comprises a possible eye block. The data structure of such a block is defined as:

```
struct EyeBlock {
    int x, y;
    int xmin, xmax, ymin, ymax;
    long size;
    struct EyeBlock *next;
};
```

The values x and y are used to record the center coordinates of the block. The values *xmin*, *xmax*, *ymin*, and *ymax* are used to record the bounds of the block. And the size is the number of the pixels that are within the block.

2.3 False Eye Blocks Elimination

After the labeling process described above, there are a lot of false eyes in the blocks yet. If we match all the blocks without screening them in advance, it will take much computation time to determine whether those candidates are real faces in the verification step. Since the processed images are colored, if the labeled blocks are real eyes, the color of pixels surrounding these blocks should be the skin color. In the labeling step, we have recorded the bounds of the blocks, so we can decide if those blocks are possible eyes by checking the pixels around the bounds. But, in some cases, such pixels may be not skin parts, like glasses, eyebrows or hair. To surmount this problem, a threshold technique with value 0.8 is applied. It means that if more than 80 percent of the pixels around the bound of a labeled block possess the skin color, then we regard the block as a possible eye block.

Through this elimination process, we can discard most of false eye blocks. For an arbitrary image, the remaining blocks are almost inside or around face regions. Thus, we can economize the execution time in the following steps.

3. FACE CANDIDATES GENERATION AND VERIFICATION

In this section, we will use possible eye blocks to generate face candidates and verify whether these candidates are real faces. We first position possible face regions by pairing the possible eye blocks. Then we adopt the color information to check the regions generated from the eye pair candidates for eliminating false face regions. Finally, the remaining possible face regions are further verified by the aid of mouth regions in the last step.

3.1 Eye Pair Candidates Generation Rules

In the previous section, we obtained possible eye blocks. Now we perform a matching operation on each possible eye block. To accomplish this, if we match the possible eye blocks with each other without constraints, it may waste execution time. Assuming that there are N possible eye blocks, we will have $N*(N-1)/2$ possible eye pairs to be verified. In order to prevent unnecessary computations, we give three simple match constraints as follows:

1) The sizes of a pair of eyes should be similar.

   If two blocks belong to a pair of eyes, they should have a similar size. Because the original image has been processed, the sizes of the pair of eyes may be different sometimes. Additionally, we do not know how large the faces in an image are. In our experiments, the tolerable ratio of the sizes of a pair of eyes is ranged from 0.8 to 1.2.

2) The ratios of both length and width of a pair of eyes should be similar.

   If two blocks contain a pair of eyes, they should have similar length and width. Besides this, the aspect ratios of these two blocks should be similar, too. Here, we compare three kinds of ratios. That is, the ratio of lengths, the ratio of widths, and the ratio of the aspect ratios of the two blocks. In our experiments, these three ratios are all given in a rough range from 0.8 to 1.2.

3) The distance between a pair of eyes should be within a range.

   If two blocks correspond to a pair of eyes, the distance between their centers should be within a range. In anthropometry, the distance between the centers of two pupils is about twice as much as the length of an eye. However, owing to individual differences and image preprocessing effects, the lengths of eyes may be some different from each other. In our system, the range of the distance is the multiple of the longer length of the two blocks, say from 1.5 to 2.5.

With the aid of matched eye pairs, we can determine possible face regions in the next step.

3.2 Face Candidates Classification Using Color Information

Because the objects that we detect are faces, the geometrical relations of facial features are invariant. For instance, eyes are always in the upper part of a face, and the mouth is always in the lower part of the face. And we observe that the color of skin is distributed over a range in the HSI color system, excluding some facial features, such as eyes and mouths. Therefore, we take this knowledge to segment a skin region from a pair of eyes and a mouth, and analyze the color information of this region to decide whether it is a possible face.

For the convenience of checking such a region, we can cut it out and then normalize it to a fixed size. But, in this step, the number of candidates is still too large. It is time-consuming to perform this. What follows is a fast process to acquire the skin region. Let the lower-left corner of an image be the origin, the horizontal line be in the X-direction, and the vertical line be in the Y-direction. A base line is made to pass through the centroid of a pair of eye blocks, and the Euclidean distance between the pair of eye blocks is denoted $Deye$. Consequently, we can use this base line and the value $Deye$ to derive another four boundary lines $(L_1 \sim L_4)$ to determine the region that we intend to check. The coordinates system and the four boundary lines are exemplified in Fig. 2.

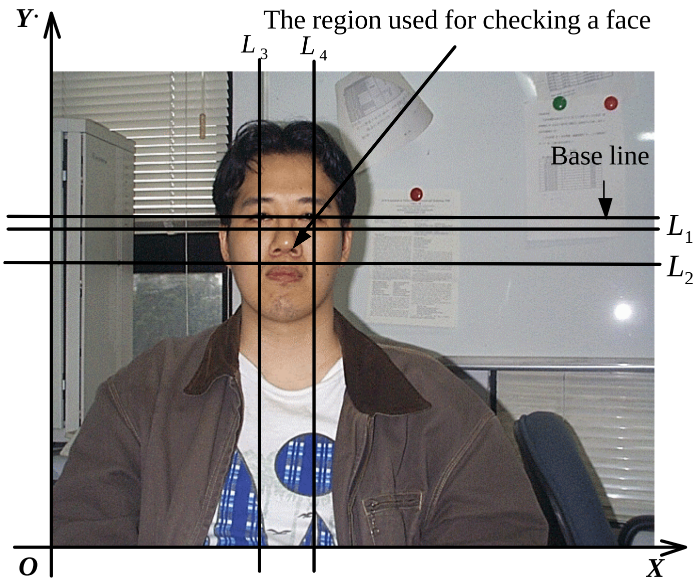


Fig.2 Illustration of the coordinates system of an image where the region deviated from a base line and surrounded by four boundary lines will be checked.

Suppose the coordinates of the centroids of left and right eyes are (lx, ly) and (rx, ry), respectively. Then $Deye = \sqrt{(rx-lx)^2+(ry-ly)^2}$, and the equation of the base line is expressed as follows:

$$ax + by + c = 0,$$

where $a = ry - ly,$

$b = rx - lx,$

and $c = ly \cdot rx - lx \cdot ry.$

The equations of the four boundary lines are stated below:

$$ax + by + c_1 = 0,$$

$$ax + by + c_2 = 0,$$

$$bx - ay + c_3 = 0,$$

and $bx - ay + c_4 = 0,$

where $c_1 = ((c^2/(a^2+b^2))^{1/2} - d_1) \cdot (a^2+b^2)^{1/2},$

$c_2 = ((c^2/(a^2+b^2))^{1/2} - d_2) \cdot (a^2+b^2)^{1/2},$

$c_3 = ((c^2/(a^2+b^2))^{1/2} - d_3) \cdot (a^2+b^2)^{1/2},$

and $c_4 = ((c^2/(a^2+b^2))^{1/2} - d_4) \cdot (a^2+b^2)^{1/2}.$

Note that the values $(d_1 \sim d_4)$ are determined by measuring the database of anthropometry retrieved from the web site: http://www.iosh.cla.gov.tw, which are the multiples of the value $Deye$. The database is statistically calculated to obtain the relative distances among facial features by surveying 1,200 different human faces. The values d_1, d_2, d_3, and d_4 are individually set to $0.16Deye$, $0.65Deye$, $0.25Deye$, and $1.25Deye$. After the four boundary lines are derived, we can calculate the intersection points of these lines to yield the bounds ($x_{\min}$, $x_{\max}$, $y_{\min}$, and $y_{\max}$) of a region. Accordingly, we check the colors of the pixels inside the region. The values of the four bounds are obtained in the following two cases:

1) For the face candidate rotated clockwise:

$$x_{\min} = \frac{bc_3 - ac_2}{a^2+b^2}, \quad y_{\min} = \frac{-bc_2 - ac_4}{a^2+b^2},$$

$$x_{\max} = \frac{bc_4 - ac_1}{a^2+b^2}, \text{ and } y_{\max} = \frac{-bc_1 - ac_3}{a^2+b^2}.$$

2) For the face candidate rotated counterclockwise:

$$x_{\min} = \frac{bc_3 - ac_1}{a^2+b^2}, \quad y_{\min} = \frac{-bc_2 - ac_3}{a^2+b^2},$$

$$x_{\max} = \frac{bc_4 - ac_2}{a^2+b^2}, \text{ and } y_{\max} = \frac{-bc_1 - ac_4}{a^2+b^2}.$$

In the most case, however, the base line and the X-axis are not in parallel. Hence, the four intersection points are not enough, when we want to check if the pixels are inside the region. In this case, we may resort to using the equations of the four boundary lines for each pixel to determine whether it is inside the region. It will require a lot of execution time. In order to reduce more computations, we calculate each range in the X-direction from $y_{\min}$ to $y_{\max}$ by the four line equations and store these ranges into an array. Since we have already known the intersection points of the lines, we only need $2n$ calculations for acquiring the ranges in the X-direction if the difference of $y_{\min}$ and $y_{\max}$ is $n-1$. Then we can employ the array to decide whether the pixels are inside the region, and avoid extra four calculations for each pixel.

The method of utilizing the color information to check possible face regions is similar to that to eliminate false eye blocks which we described in the previous section. We still apply the means of hue and saturation values to determine the color of each pixel. We also enlarge the range of the hue value, because the region to be checked may contain some facial features and cover objects whose colors are different from that of the skin, for instance, nostrils, lips, and glasses. The hue values of these objects may be out of the range of the hue value of the skin, so we change the method of determining the color of a pixel, and alter the ranges of hue and saturation values to handle the facial features and cover objects.

In this manner, we check the bounds of the region first. If the bounds are inside the image, then we continue the check on the color information of the pixels within the region. Hence, we can prevent excessive time spent in checking the candidates that are out of the image.

3.3 Candidates Verification and Elimination

After using the color information to eliminate false eye pair candidates, we almost obtain real faces. However, there still remain some false candidates that are paired by eyebrows or hair. In this subsection, we will apply mouths to perform the final verification. Through this verification, some candidates may be overlapped, and one of them is further selected and viewed as a human face.

3.3.1 Verification using the Mouth Feature

In order to verify the candidates, we can adopt other facial features, such as eyebrows, noses, or edges of cheeks. But, due to the light, background, and individual differences, it makes those features unstable. Nevertheless, the facial feature— the mouth is seldom affected by the aforementioned factors. The color of the mouth is always darker than that of the skin. Because the image has been processed by edge detection, we can find the edges in the mouth region of a face at all times.

As Fig. 3 shows, we define a region for checking the mouth feature. The values d_{up} and d_{down} are derived from measuring the database of anthropometry, which are the multiples of the value $Deye$. The values d_{up} and d_{down} are set to $0.67Deye$ and $1.31Deye$, respectively. The left and right bounds of the region are obtained from those of the extent of a pair of eye blocks.

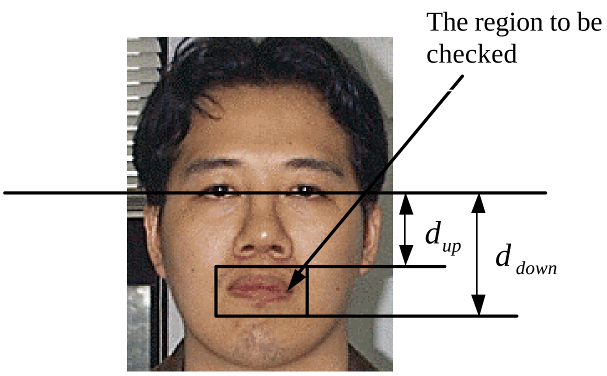


Fig.3 Illustration of the region for checking the mouth feature.

3.3.2 Overlaps Elimination

After the previous verification, some candidates possibly overlap in the same region of a human face. For example, these candidates may be composed of two eyebrows or an eyebrow and an eye. According to the constraint of our detection system, human faces should not be upside-down, so that real eyes should lie in lower positions than eyebrows do. Such a characteristic is employed to determine the real eyes by comparing the positions of the overlap candidates.

For the candidates rotated clockwise, we select the candidate that has a lower and more right position than the others do as an eye. For the candidates rotated counterclockwise, conversely, we select the candidate that has a lower and more left position as a real eye.

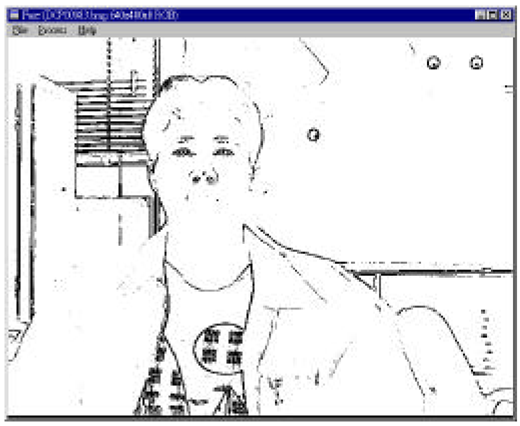
4. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we demonstrate the experimental results obtained from our developed detection system. The testing programs have been executed on the Microsoft Windows 98 of a personal computer with a Pentium II 300 CPU. The testing images are taken with a Fujifilm MX-2700 digital camera. The size of input images is 640x480 pixels. And the color depth of the images is 8-bits.

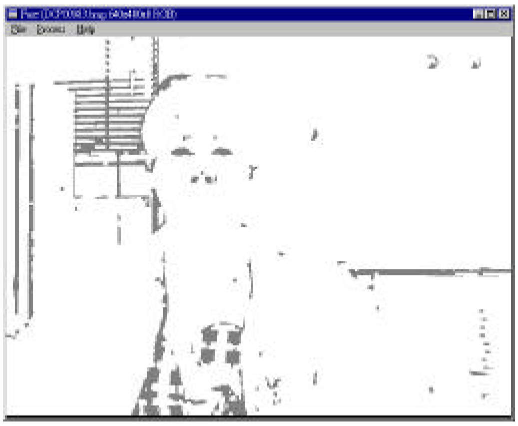
4.1 Experimental Results

In the following experiments, we use Figs.4(a) ~ (f) to illustrate the detection procedure of our system step by step. Figure 4(a) is the original image to be detected. Figure 4(b) is the result of edge detection after contrast enhancement, and Fig.4(c) is the result of noise removal. Notice that through the noise removal for this image, the number of blocks is reduced from 464 to 79. In Fig.4(d), we mark the blocks after the above process to eliminate false eye blocks by checking the color information around their boundaries. In this step, we further reduce the number of blocks in this image from 79 to 9. After the pairing process, we can see that two candidates remain in Fig.4(e). Figure 4(f) shows the final result after checking the color information of a face region.

(a)



(b)



(c)

(d)

(e)

(f)

Fig.4 Illustration of the faces detection procedure: (a) the original image; (b) the result of contrast enhancement followed by edge detection; (c) the result of noise removal; (d) the result from checking the colors around the boundaries of blocks; (e) the result from pairing the blocks; (f) the final result from checking the color information of possible face regions.

Some of other successful experimental results are illustrated in Figs.5~7. Figure 5 shows the detection result of the face rotated clockwise less than $45°$. Figure 6 shows the detection result of the face rotated counterclockwise more than $45°$. Figure 7(a) shows the detection result of the face with glasses, while Fig.7(b) shows the detection result of the image containing more than one face. Table 1 summarizes our experimental results. In this table, the detection rate is defined as the ratio of the number of faces correctly detected to the total number of faces in the images.

The execution time required to locate the faces depends on how complex the images are. For most of the testing images, the amounts of execution time are less than 0.5 second.

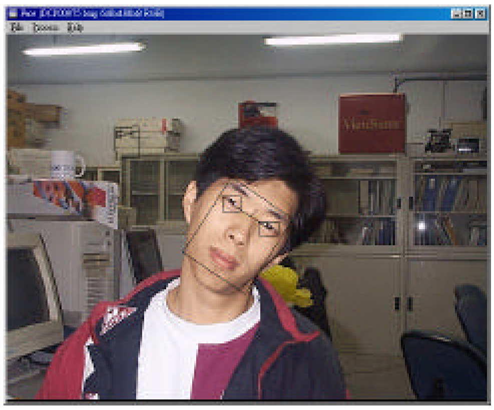


Fig.5 Detection result of the face rotated clockwise less than $45°$.

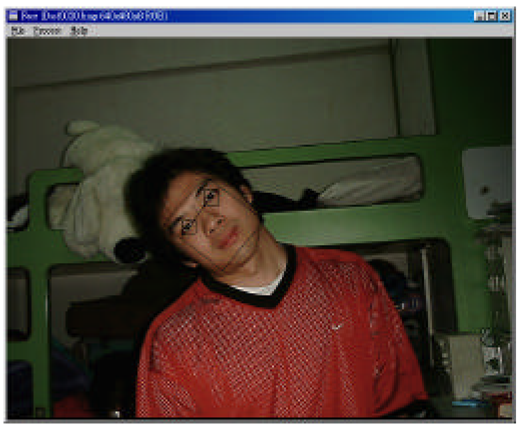


Fig.6 Detection result of the face rotated counterclockwise more than $45°$.

(a)

(b)

Fig.7 Detection results of: (a) the face with glasses; (b) the image containing more than one face.

Table 1 Summary of Experimental Results

Experiments \ Results	Detection rate	Average number of faces incorrectly located
One human face (70 images)	93%	0.14
More than one human face (20 images)	85%	0.20
Human faces with glasses (30 images)	43%	0.73

4.2 Discussion

There are some restrictions on our system, which result in the failure of detecting faces sometimes. The following gives the reasons why our system is out of order and also illustrates some examples that detect faces inaccurately.

1) If the focus of a face in an image is poor, it sometimes leads to wrong results in the step of extracting facial features. This is the motivation that we first apply an edge detection method to extract facial features. While an image is out of focus, the edge detection process often loses edges. Figure 8(a) shows a blurred image that is not in focus.

2) The faces cannot be detected correctly when they lie in a too dark or bright image. Such poor illumination conditions often make mistakes in the steps of extracting facial features and employing color information. Figure 8(b) shows a shaded image where the face of a person is too dark.

(a)

(b)

Fig.8 A poor image caused by: (a) out of focus; (b) ill illumination.

3) As Table 1 reveals, when a person wears glasses, it often causes the loss of faces detection. In this situation, three types of images are hard to detect faces. First, if eyes are covered with the frame of glasses, we always fail in locating faces. Second, if the frame of glasses is too close to eyes, we also have a low detection rate. Third, if the glasses reflect the light, we mostly lose the faces, too. It results from our preprocessing stage which is insufficient to deal with the above cases, except that the frame of glasses has a light color or is much larger than the sizes of eyes. Figure 9(a) shows the example that glasses reflect the light, while Fig.9(b) shows the example that eyes are covered by the frame.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a human faces detection system for color images by using the knowledge of color information and the geometrical relations of facial features. First, we extract possible eye blocks by edge detection.

(a)

(b)

Fig.9 A typical image of a person wearing glasses: (a) which reflects the light; (b) whose frame covers eyes.

Then both skin-like hue and saturation values are used to reduce the number of possible eye blocks. In this step, we eliminate most of false blocks, so we can save much computation time for the following steps. In the next step, we adopt three rules of pairing possible eye blocks to generate possible face candidates. Finally, in the verification step, we again employ skin-like hue and saturation values to check the color of a skin region and apply other facial features for further verification.

As Section 4 shows, although our detection system can locate human faces quickly and exactly in many cases, it still has some situations that we can not detect faces very well, for instance, persons wearing glasses. In the following, we list some suggestions for future work to improve our system.

1) Developing a better preprocessing method to extract blocks:

The eye blocks can be extracted more correctly by improving the preprocessing stage. So, we can handle more intractable cases, such as persons wearing glasses or faces being out of focus.

2) Using more constraints in the verification step to prevent from locating false faces:

The rate of locating false faces can be decreased, if we use more constraints in the verification step. For example, we expect to exactly detect faces no matter whether eyebrows exist or not.

3) Building a correlation table of hue and saturation values to correct the respective uses of hue and saturation values:

In our system, we prescribe two respective ranges of hue and saturation values constituting the color information to check the color of skin. However, in some cases, both hue and saturation values are individually within their ranges, but the corresponding color may not be a skin color. To improve this, we can build a table of the ranges of saturation values, each of which refers to a range of hue values, respectively. Therefore, the validity of the color information may be increased.

6. REFERENCES

[1] R. Brunelli and T. Poggio, "Face recognition: features versus templates," IEEE Trans. Pattern Anal. Machine Intelli., vol. 15, no. 10, pp. 1042-1052, 1993.

[2] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in Proc. Comput. Vision Pattern Recogni., Lahaina, Maui, Hawaii, 1991, pp. 586-591.

[3] P. J. Phillips et al., "The FERET evaluation methodology for face-recognition algorithms," in Proc. Comput. Vision Pattern Recogni., San Juan, Puerto Rico, 1997, pp. 137-143.

[4] F. Goudail et al., "Face recognition system using local autocorrelations and multiscale integration," IEEE Trans. Pattern Anal. Machine Intelli., vol. 18, no. 10, pp. 1024-1028, 1996.

[5] K. Sobottka and I. Pitas, "Looking for faces and facial features in color images," Pattern Recogni. Image Anal., vol. 7, no. 1, pp. 124-137, 1997.

[6] C. H. Lee, J. S. Kim, and K. H. Park, "Automatic human face location in a complex background using motion and color information," Pattern Recogni., vol. 29, no. 11, pp. 1877-1889, 1996.

[7] Y. Dai and Y. Nakano, "Face-texture model based on SGLD and its application in face detection in a color scene," Pattern Recogni., vol. 29, no. 6, pp. 1007-1017, 1996.

[8] C. Chen and S. P. Chiang, "Detection of human faces in colour images," IEE Proc. Vis. Image Signal Process., vol. 144, no. 6, pp. 384-388, 1997.

[9] G. Yang and T. S. Huang, "Human face detection in a complex background," Pattern Recogni., vol. 27, no. 1, pp. 53-63, 1994.

[10] M. Kosugi, "Human-face search and location in a scene by multi-pyramid architecture for personal identification," Syst. Comput. in Japan, vol. 26, no. 6, pp. 27-38, 1995.

[11] G. Burel and D. Carel, "Detection and localization of faces on digital images," Pattern Recogni. Letters, vol. 15, no. 10, pp. 963-967,1994.

[12] S. H. Jeng et al., "Facial feature detection using geometrical face model: an efficient approach," Pattern Recogni., vol. 31, no. 3, pp. 273-282, 1998.

[13] G. Chow and X. Li, "Towards a system for automatic facial feature detection," Pattern Recogni., vol. 26, no. 12, pp. 1739-1755, 1993.

[14] L. C. De Silva, K. Aizawa, and M. Hatori, "Detection and tracking of facial features by using edge pixel counting and deformable circular template matching," IEICE Trans. Inf. & Syst., vol. E78-D, no. 9, pp. 1195-1207, 1995.

[15] A. Samal and P. A. Iyengar, "Human face detection using silhouettes," Int. Jour. Pattern Recogni. Artificial Intelli., vol. 9, no. 6, pp. 845-867, 1995.