Real-Time Eye Detection using Face Circle Fitting and Dark-Pixel Filtering*

Daw-Tung Lin and Chian-Ming Yang
Department of Computer Science and Information Engineering
Chung-Hua University
No. 707, Sec. 2, Wu-Fu Rd., Hsinchu, Taiwan, 30012
dalton@chu.edu.tw

Abstract

In this paper, we develop a real time system for eye detection and tracking in video sequences. The HSI color model is used to extract skin-color pixels. The same type of pixels are gathered by a region-growing algorithm and the maximum area of this region is selected as the face candidate. Furthermore, a Face-Circle Fitting (FCF) method is proposed to confirm the exacted face region. Once the face is detected, a Dark-Pixel Filter (DPF) is used to perform the eye detection and tracking. The proposed system has been implemented on personal computer with a PC-camera and can perform eye detection and tracking in real-time. The correct eye identification rate is as high as 92% during the actual operation.

1 Introduction

Due to the need for more efficient and user-friendly human computer interaction, studies of human face and motion detection have expanded rapidly in recent years [1]. The related techniques of face processing in computer science include facial expression recognition, face recognition, etc. However, these techniques are not straightforward, and they are based on a key issue of face detection. An effective and accurate face detection approach not only can identify the face location, but it can also obtain very useful information for further applications. Biometric recognition plays an important role in the advancement of human computer interaction since it provides a natural and efficient interface with computers. Face and eye detection, as well as face recognition are among the first issues to be investigated. The most well-known methods include color-based approaches [2, 3, 4, 5, 6], neural network models [7, 8, 9], genetic algorithm methods [10, 11], principal component analysis approaches [12, 13, 14], and morphology-based processing [15].

The color-based approaches use statistical skin-color models to extract skin-color pixels from an image, and subsequently label each sub-region as a face if it contains a large area of skin color pixels. It can cope with different views of faces, but it is sensitive to skin color and the face shape (some researches [16, 17] use an elliptical shape to solve the shape problem). Therefore, it is a good approach to be a preprocessing tool in face detection system to roughly extract the face regions. The neural network approach detects faces by sub-sampling different

* This work was supported in part by the National Science Council, Taiwan, R.O.C. (NSC 93-2622-E216-001-CC3, and Chung-Hua University Grant CHU-93-TR-004 and CHU 93-2622-E216-001-CC3.

regions of the images to compare with assemble a standard sized sub-image and then passes it through a neural network. This algorithm performs very well for front-parallel faces but performance deteriorates when it is extended to different views of a face. It is not possible extend the algorithm to detect faces in different views with re-training. The genetic algorithm approach performs well to search for the destination in a wide searching space. The face detection problem can be considered as a data searching problem in a 2D searching space. The algorithm is used to code the location of the searching window as the genes. Then, it generates several generations and selects the area with highest fitness score. The algorithm can converge quickly to the results; although it can not cope with multiple faces, and the fitness function is difficult to determine. There have been several studies which have applied principal components analysis (PCA) to the problem of face recognition by projecting the facial image onto the "face space region" and using these projections to represent the facial images. Thus, they analyzed face image in different domain of face space.

In this paper, we used the HSI color model to extract skin-color pixels and then use the region growing algorithm [18] to group these pixels. After that, more than one face-like regions may be obtained, and we only select the maximum region as the face. We propose a Face Circle Fitting (FCF) method to identify the face region. This method is capable of finding groups of skin-color pixels within an approximately circular shape, and eyes are detected in the upper area of the face circle. We also propose a Dark-Pixel Filter (DPF) to extract dark pixels, and group them by the region growing algorithm as eye candidates. Then we use the geometric relation to find the exact position of the eyes.

2 Face Detection

In this section, we present in detail each component of the proposed face detection technique, and Fig. 2 shows the flowchart of the proposed face detection technique. For the sake of on-line performance in video sequence, we discard the odd field scan-lines and compute only the even field scan-lines in each acquired image frame. We then perform averaging sub-sampling operation (2:1) for each scan-line. As a result, each 320×240 image frame acquired from the camera is reduced to a 160×120 miniature.

2.1 Skin-Color Blocks Extraction

We adopted HSI color model to extract skin-color pixels and a detailed presentation of the method for converting from RGB representation to HSI model can be found in [18]. When the H value of a pixel falls in the range of [H_{L0}, H_{Hi}], we consider this pixel to be a skin-color pixel. H_{L0} and H_{Hi} are adapted and updated every five minutes during the operation according to the proposed scheme. Initially, H_{L0} and H_{Hi} are determined by acquiring histogram of the H value of the previous time interval of the face region. Assume the H histogram is $h_H(i), 0 \le i \le 60$ of the face region representing the frequency of the H value i in the face region and $h_H(\max)$ is the maximum frequency. Then H_{L0} can be determined by finding the first H value whose frequency is greater than $h_H(\max)/10$ from 0 to 30, and H_{Hi} can be

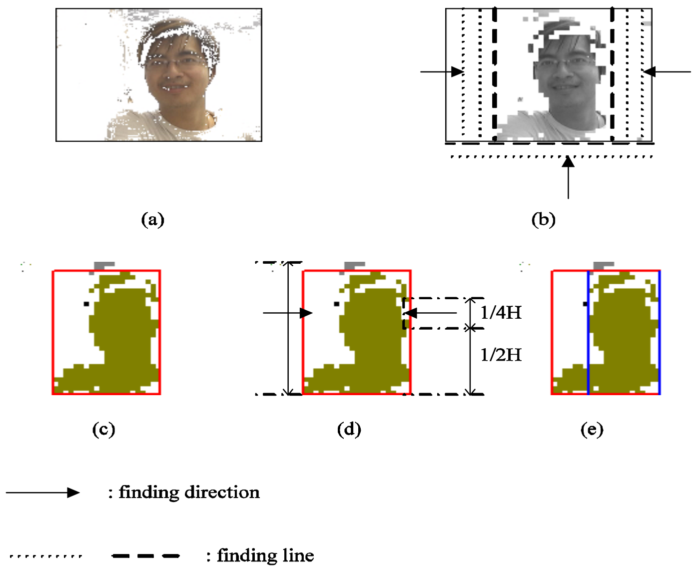


Figure 1: The process of adjusting left bound and right bound: (a) skin-color pixels images, (b) boundary searching, (c) the result of (b), (d) the left and right boundary adjustment, (e) the result of (d).

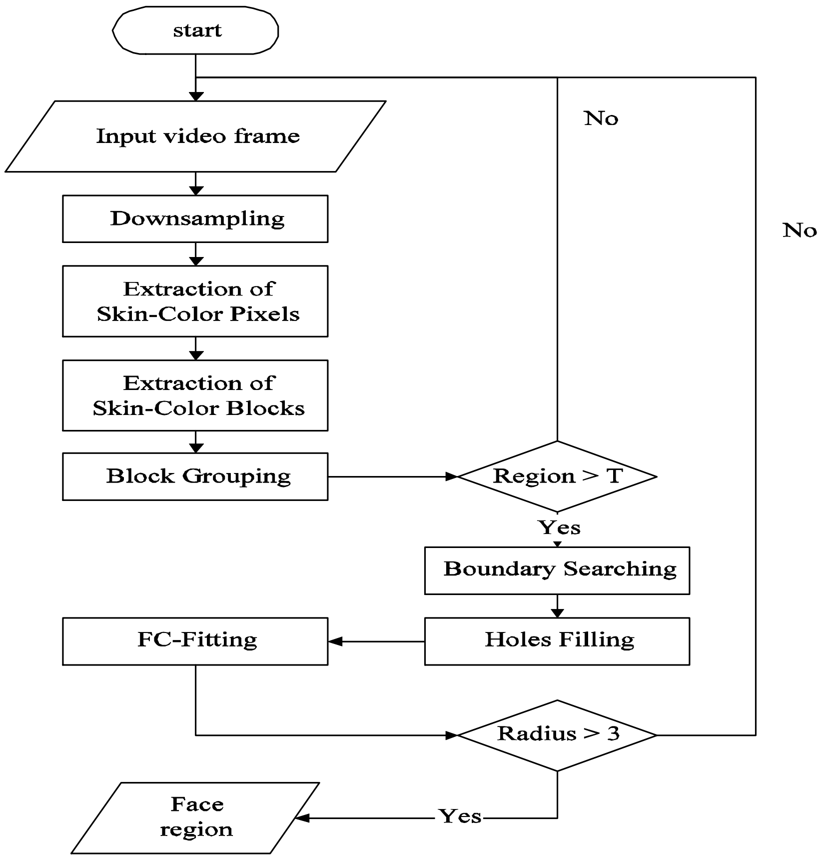


Figure 2: Flowchart of face detection.

determined from 60 to 31 by the same way. To cope with color noise, we divide an image into blocks of 4×4 pixels. When more than half of the pixels of a block are within the skin-color range, we assume this block to be a skin-color block. The output of this process is a block image (image size 30×40). After the skin-color block searching procedure, there may be more than one face-like region in the block image, so we use a region growing algorithm to find the maximum region as the face region. Then the boundaries $(f_{top}, f_{botton}, f_{left}, f_{right})$ of a face region can be defined. Figure 1 illustrates the concept of boundary adjustment. For accuracy in the next process (Face-Circle Fitting), we will fill holes which may be the areas of eyes or eye brows. The method of filling holes is to search every non-skin-color pixel in the region of the upper half of the face region and determine whether this pixel is a pixel of a hole. If it is a pixel in a hole, we will change it into skin-color pixel. The results of skin-color pixels grouping and hole filling are shown in Fig. 3.

2.2 Face Circle Fitting

However, there is a significant drawback in using the region growing algorithm for pixels grouping. This is because the noise introduced by skin-color pixel extraction procedure and non-face skin-color objects may form connected regions. To overcome this drawback, we propose a Face Circle Fitting (FCF) method, which is capable of finding groups of skin-color pixels of a circular shape, in which a circle center is approximately in the center of a face. We describe this method in detail below.

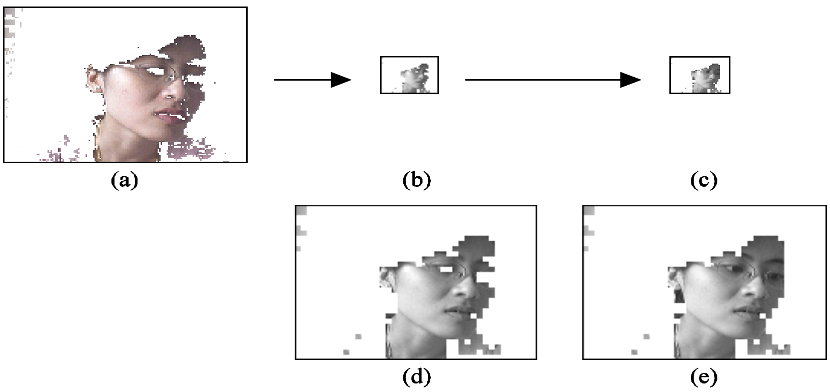


Figure 3: The process of filling holes: (a) skin-color pixels image, (b) block grouping, (c) the result of hole filling, (d) enlargement of image (b), (e) enlargement of image (c).

The circular model is initialized and placed near the estimated center of the face and then is adapted step-by-step to fit the face. The initialization of the circular model will influence the accuracy of face circle fitting and the speed of the adaptation process. The procedures are described as follows:

Step 1 Determine the horizontal position of the center of the circular model C_{hc}.

Step 2 Adjust the top of the face (f_{top}).

Step 3 Determine the radius of the circular model: $r = (f_{bottom} - f_{top})/2$

Step 4 Define the vertical position of the center of the circular model: $C_{vc} = f_{top} + r$

In Step 1, by performing the vertical projections on skin-color pixels in the face region, we obtain $p_v[i]$; $f_{left} \le i \le f_{right}$ and $p'_v[i] = \sum_{j=i-5}^{i+5} p_v[j]$, $f_{left} \le i \le f_{right}$, where $P_V[i]$ stands for vertical projections, $P'_v[i]$ denotes the summation of $P_v[j], i-5 \le j \le i+5$ and $P'_v[k]$ is the maximum value among $P'_v[i]$, then the value of C_{hc} is k.

In Step 2, by performing the horizontal projections on skin-color pixels in the range $[C_{hc}-5, C_{hc}+5]$, we can get $p_h[i]$, $f_{top} \le i \le f_{botton}$. If k is the first position which satisfies $P_h[k] \ge T$ and $P_h[k+1] \ge T$ (T value is determined empirically.), then f_{top} is updated by k.

After the circular model is initialized, it will be adapted step-by-step. The parameters Area1 and Area2 are defined as the upper area and the lower area of the circular model, respectively (see Fig. 4). Area3[i], $0 \le i \le 3$ is defined as the specific areas in the upper side, lower side, left side and right side of the circular model., respectively. The adaptation process may encounter the following cases:

Case 1 C_{top} is always fixed in the position of f_{top}. Here, the radius r of the circular model is determined by the area inside the circle: Area = min(Area1,Area2), r = $\sqrt{Area \times 2 / 0.85 / \pi}$, $C_{vc} = f_{top} + r$ = ftop+r, C_{hc} is no change.

Case 2 Set stop = true. Then the vertical center of the circular model is fixed.

Case 3 Compute $Area3[i]$. Assume Area[k] is the minimum among $Area3[i]$, then Area = $Area[k]$ and AreaNo= k. If $Area < r \times (r \setminus 2) \times 0.5$ (rectangle area$\times 0.5$), the center is fixed and subtract 1 from r, else r is fixed and the center is determined by $C_{vc} = C_{vc} + 1$, $C_{vc} = C_{vc} - 1$, $C_{hc} = C_{hc} + 1$, $C_{hc} = C_{hc} + 1$ for AreaNo= 0, 1, 2, 3, respectively.

Figure 5 illustrates the flowchart of the adaptation process. Examples of the adaptation process are shown in Fig. 6. According to the normal position of eyes in a face, it is reasonable to assume that the likely location of eyes to be in the region of the upper half of the face circle. We define $e_{left} = (C_{hc} - r) \times 4$, $e_{right} = (C_{hc} + r) \times 4$, $e_{top} = (C_{vc} - r) \times 4$, and $e_{bottom} = C_{hc} \times 4$.

3 Eye Detection with Dark Area Extraction

Since eyes are similar to dark areas, we can use this feature to find candidate eye locations. In order to detect these little dark areas, we designed a Dark Pixel Filter (DPF) shown in Fig. 8 to extract pixels which belong to eye pixels. The rule of the dark-pixel filter is described as follows, and the flowchart of eyes detection is illustrated in Fig. 7:

Step 1 If ($I_{(x,y)}$ < dark) goto Step 2, otherwise goto Step 5.

Step 2 If ($I_{(x,y+t_1)} - I_{(x,y)}$) > edge; $UpB = y - t_1$ goto Step 3, otherwise goto Step 5.

Step 3 If ($I_{(x,y+t_2)} - I_{(x,y)}$) > edge; $DoB = y - t_2$ goto Step 4, otherwise goto Step 5.

Step 4 If $DoB - UpB \le r$, then $I_{(x,y)}$ is a candidate eye pixel.

Step 5 Examine the next pixel.

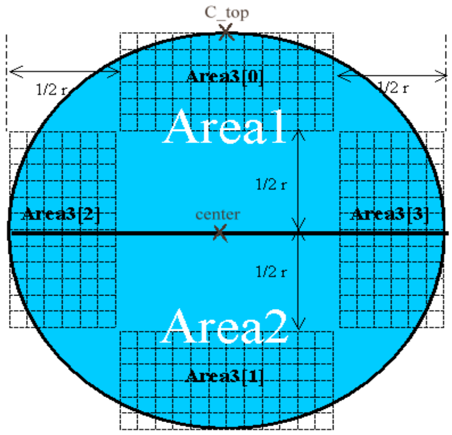


Figure 4: Circular model.

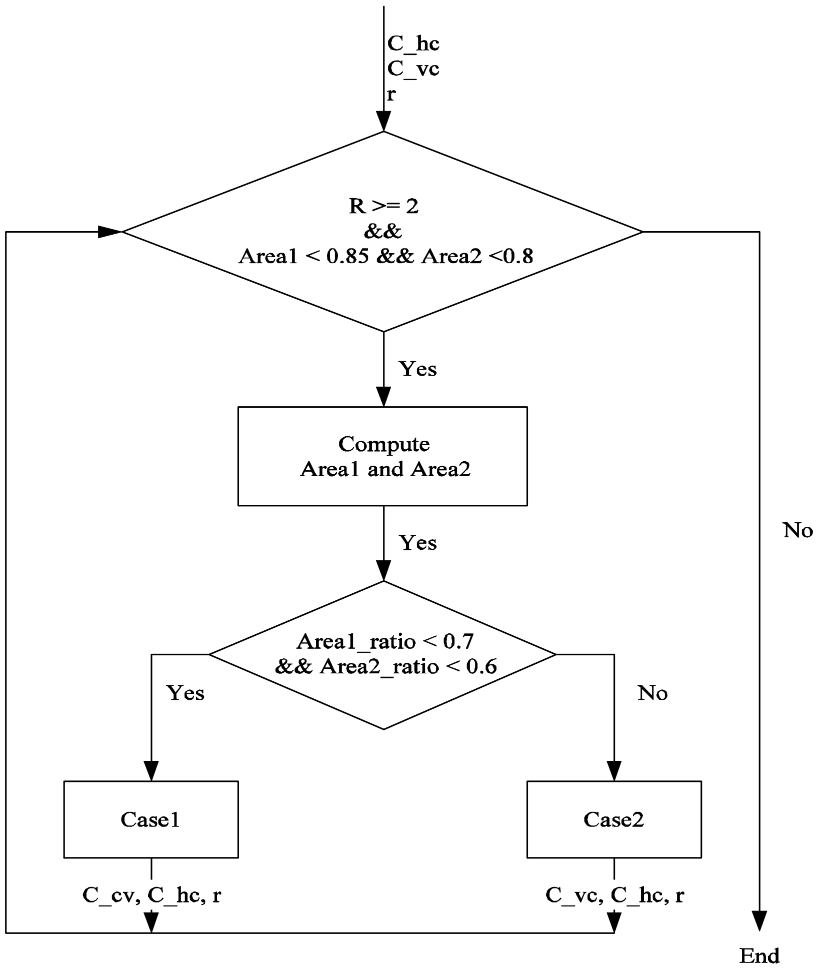
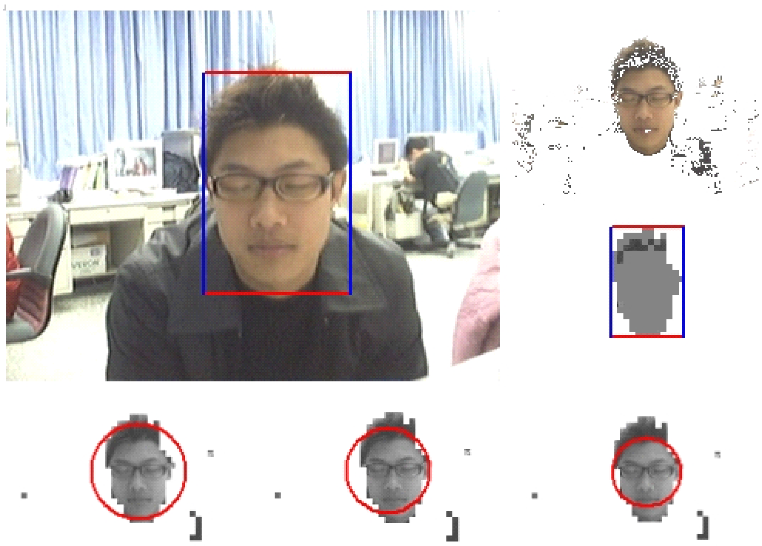
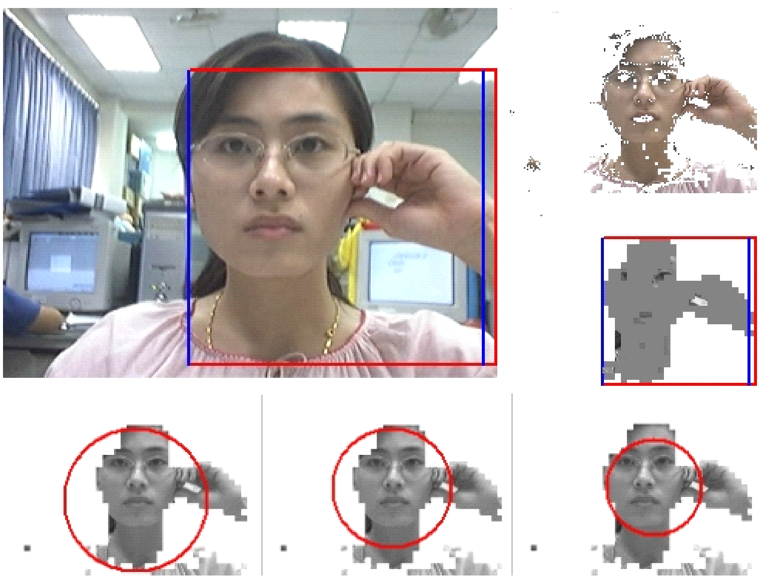


Figure 5: The flowchart of the adaptation process.

$1 \le t_1, t-2 \le r, I_{(x,y)}$ denote the grey level intensity of the current processing pixel (x, y), and t is the searching index. Dark represents the grey level threshold for distinguishing eye pixel from skin pixel, edge is a grey level threshold for identifying the edge of eye, UpB and DoB are the variables recording the upper-gap and bottom-gap of current processing pixel, respectively. We use rectangle shapes to mark groups of dark pixels and select these rectangles as eye candidates, and then apply the center of these rectangles to represent the eyes centers. Examples can be seen in Fig. 9(c).



Example1



Example2

Figure 6: Examples of the adaptation process.

3.1 Verification Process

Among those eye candidates obtained from the previous process, some pairs of them are real eye pairs, but some are just dark areas. In this step, we only check whether the face has one real eye pair, but we do not verify those eye pairs to locate the possible eye position. Therefore, we must remove some unreasonably large areas by the following conditions: if the height of $blob \leq 3 \times r$, and if the width of $blob \leq 3 \times r$,(r is the radius of circle). Figure 9(d) shows the results of removing unreasonable area.

After observing over thirty different human face images, we found that eye areas and eyebrow areas are usually larger than others parts of the eye region. Accordingly, we select only four large areas ($area$ size$\geq 2 \times r$) and check each pair among them until one pair meets the condition of $2 \times r \leq d \leq 6 \times r$ and $\theta \leq 45°$ Figure 10 shows the relations of the parameter d, and θ.

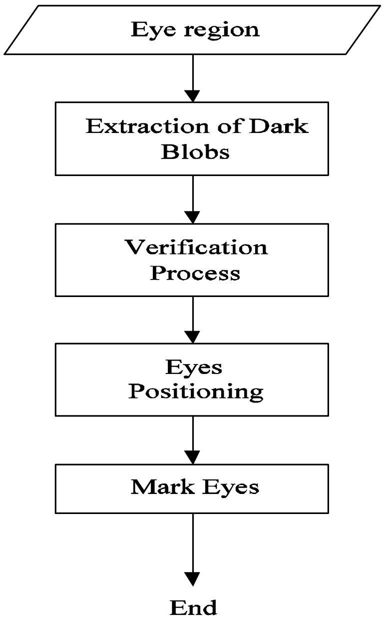


Figure 7: Flowchart of the eye detection.

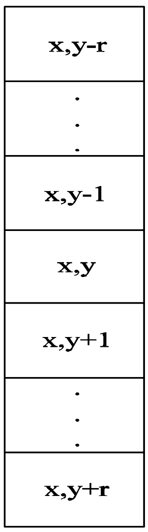


Figure 8: Dark-pixel filter.

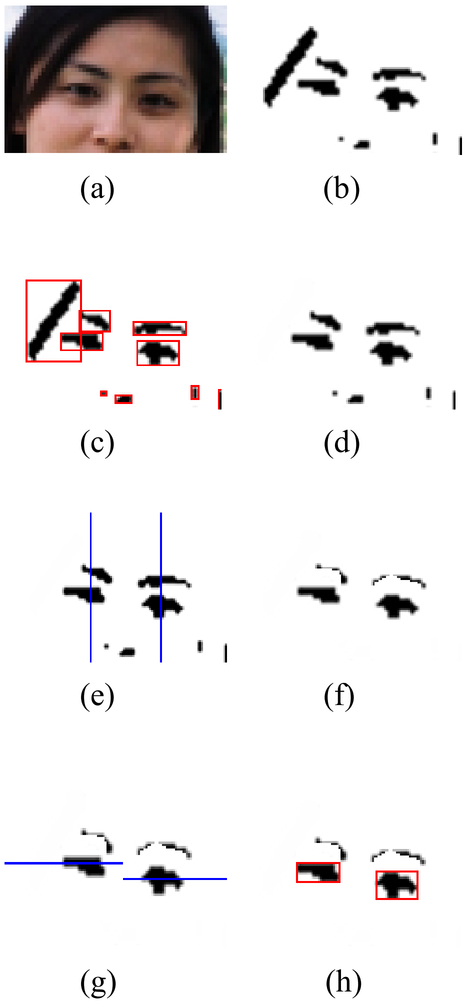


Figure 9: Eye positioning, (a) source image, (b) eye pixel extraction, (c) eye pixel grouping, (d) unreasonable pixel removal, (e) horizontal position locating, (f) nose pixel and eyebrow pixel removal, (g) vertical position location, (h) eye marking.

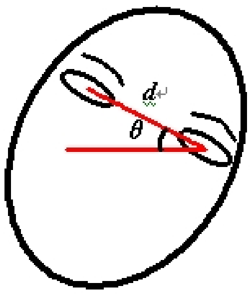


Figure 10: The face model.

3.2 Eyes Positioning

After this verification process, we can use the distribution of dark pixels to locate the eye position. By performing vertical projections on dark pixels, we get $P_v(i)$, $e_{left} \le i \le r_{right}$ and $P_h'(i) = \sum_{j=i-r}^{i+r} P_h(j+r)$, $e_{left} \le i \le e_{right}$. Assume that $P_v'(m)$ is the maximum among $P_v'(i)$, $e_{left} \le i \le e_{right}$, and m will be the horizontal position of one eye. Consequently, we set

$P_v'(k)=0, m-r\times 1.5 \le k \le m+r\times 1.5$ Again, we search for the maximum $P_v'(n)$ from $P_v'(i)$, $e_{left} \le i \le e_{right}$, then n is the horizontal position of another eye. After the horizontal position of the eye is located, the vertical position of the eyes will be located by the same method. Since eyebrow pixels may cause the locating process to fail and nose pixels may cause the eyebrow removal process to remove some eye pixels, we must first remove nose pixels and then remove eyebrow pixels. The method of removing nose pixels is to find a horizontal line which just lies in the bottom of one eye from the bottom, and remove the pixels which lie below the line. The position of this line can be determined by performing horizontal projections in the range of $[m-r, m+r]$ or $[n-r, n+r]$. Assume that $P_h(i)$ is the horizontal projection of the position i, then $P_h'(i)=\sum_{j=i}^{i-r} P_h(j)$. When $P_h'(k)$ exceeds a certain threshold, one line is located in the position k. A second line is then located in the same fashion. The projection ranges of the left line and the right line are $\lfloor e_{left}, (m+n)/2 \rfloor$ and $\lfloor (m+n)/2, e_{right} \rfloor$, respectively. Since the eye is in the bottom of the eyebrow, a pixel of the eyebrow can be determined by searching to determine whether there is a pixel below this pixel.

Once the nose pixels and eyebrow pixels are removed, we perform horizontal projection in the range of $[m-r, m+r]$ or $[n-r, n+r]$ to locate the vertical eye position. Assuming that $P_h(i)$ is the horizontal projection of the position i, $e_{top} \le i \le e_{bottom}$, then we can compute $P_h'(i)=\sum_{j=i-r}^{i-r} P_h(j)$. If $P_h'(k)$ is the maximum among $P_h'(i)$, then the eye is located in the position k. Finally, we can group eye pixels and mark them with a rectangle shape.

4 Experimental Results

The system is implemented on a PC with Pentium IV 1.6 GHz CPU with a Microsoft Windows 2000 operating system. We used a Logitech QuickCam Pro3000 camera to capture video sequence, and the format of input video is 320×240 true color. The face and eye detection can achieve the speed of 13-25 frames per second. Eyes can be detected accurately, even when the eyes are closed. We have tested seven people with different aspect, rotation, tilt angle under normal indoor lighting conditions. Table 1 shows the average hit ratios of eye detection with different aspect angles, rotation, and tilt angles. As we can see, the proposed system achieves 92% correct eye identification when the variation is less than 15 degrees. The average hit ratio is 86% if the angle is between 15 degree and 30 degree.

We also tested the HHI face database and compared the proposed algorithm with Hsu's algorithm [19]. The HHI database contains 206 images, each with size 640×480 pixels. The HHI face database includes several racial groups and lighting conditions (overhead lights and side lights). Furthermore, these images contain frontal, near-frontal, half-profile, and profile face views of different sizes.

Table 2 shows a performance comparison of detection hit ratio and processing speed of the proposed system and the method of Hsu et al on the HHI database. Although the proposed method does not perform better than that of Hsu, our processing speed is much faster and can be implemented in real-time applications. Furthermore, Hsu et al introduced a lighting compensation technique to correct the color bias and obtained a better identification results for various lighting conditions. In our experiments, lighting conditions (such as side lights) did cause some face detection failures. Figure 12 shows six of the eye detection results of fourteen people in HHI database using the proposed system.

Table 3 illustrates the performance references of our method and others approaches. Although the test images of these approaches were all different, the statistical results in general can imply that our method is comparable to that of others. In addition, the computation speed of the proposed method is very low and can be implemented to real-time applications.

Table 1: Experimental results of the proposed detection system.

Head Pose	$\pm 15^{\circ}$	$\pm 15^{\circ} \sim \pm 30^{\circ}$
Aspect	92%	85%
Rotation	91%	87%
Tilt	92%	86%
Average Ratio	92%	86%

Table 2: Comparison of detection hit ratio and processing speed of the proposed system and the method of Hus et. al. on the HHI database.

	Near-Frontal	Half-Profile	Speed (sec)
No. of images	120	75	
Hit ratio (ours)	74%	69%	0.06 ± 0.02
Hit ratio (Hsu et. al.)	90%	75%	22.97 ± 17.35

Table 3: A list of performance references.

Method	Detection Rate	Speed	Data
Ours	85%~92%	13~25 frames/sec	Seven persons
Talmi and Liu [20]	77%~100%	5 frames/sec	ORL database [21]
Feng and Yuen [22]	92.5%	Off-line	MIT AI Lab
Huang and Wechsler [23]	70%~85%	N/A	FERET database [24]
Han etc. [25]	98.8%	0.2 frame/sec	AR database [26]
D'Orazio [27]	96%	3 frames/sec	Six persons

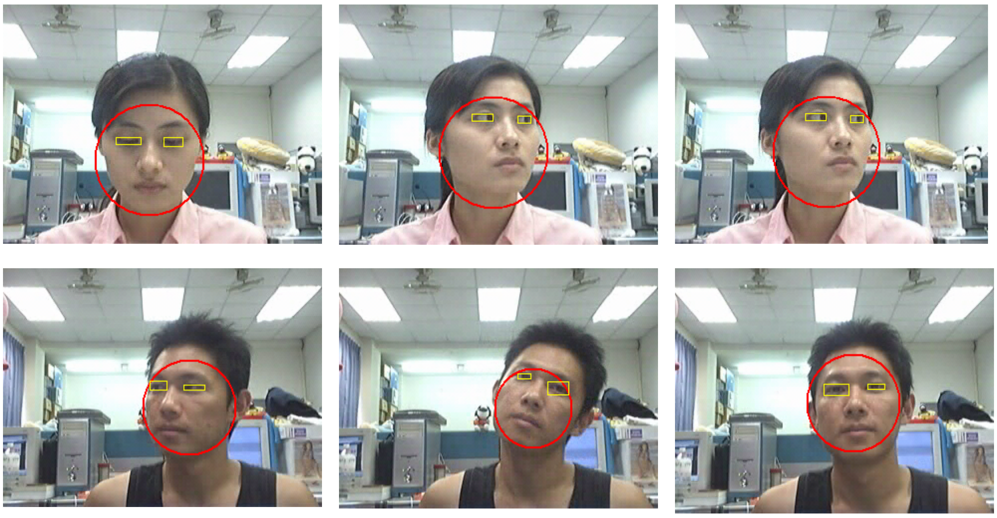


Figure 11: The eye detection results of two persons.

5 Conclusion

We have implemented a real time face and eye detection system, wherein the color-based approach is first used to extract the similar skin color regions. This processing can filter out the background very quickly and effectively. We then propose a FCF method using an adaptive circle to include the exacted region of the face. After the face is detected, we use a DPF method to detect the eye candidates. The concept of the DPF is to treat the eye as a small dark area. Based on the geometric features of these dark pixels, we can extract the eye positions.

Figure 13: The eye detection results of six subjects from HHI database by using the proposed system.

References

[1] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. Pattern Recognition, 36:585-601, 2003.

[2] J.Yang and A.Waibel. Skin-color modeling and adaptation. IEEE Signal Processing

Society Workshop,2:624-633, 2000.
[3] J. C. Terrillon, M. David, and S. Akamatsu. Automatic detection of human faces in natural scene images by use of a skin color model and invariant moments. In Proc. IEEE 3rd Int. Conference on Automatic Face and Gesture Recognition, pages 112-117, 1998.
[4] M. H. Yang and N. Ahuja. Detecting human faces in color images. In International Conference on Image Processing, volume 1, pages 127-130, 1998.
[5] S. Kawato and N. Tetsutani. Circle-frequency filter and its application. In Int. Workshop on Advanced Image Technology, pages 217-222, Feb. 2001.
[6] R. S. Feris, T. D. Campos, and R. C. Junior. Detection and tracking of facial features in video sequences. Lecture Notes in Artificial Intelligence, 1793:197-206, 2000.
[7] Z. Bian, H. Peng, and C. Zhang. Human eyes detection using hybrid neural method. In Fouth International Conference on Signal Processing Proceedings, volume 2, pages 1088-1091, 1998.
[8] M. Fukumi, Y. Mitsukura, and N. Akamatsu. A design of face detection system by lip detection neural network and skin distinction neural network. In IEEE International Conference on Systems, Man, and Cybernetics, volume 4, pages 2789-2793, Oct. 2000.
[9] G. T. Park and Z. Bien. Neural network-based fuzzy observer with application to facial analysis. Pattern Recognition Letters, 21:93-105, 2000.
[10] K. W. Wong and K. M. Lam. A reliable approach for human face detection using genetic algorithm. In TENCON 99.Proceedings of the IEEE Region 10 Conference, volume 4, pages 499-502, 1999.
[11] C. H. Lin and J. L. Wu. Automatic facial feature extraction by genetic algorithms. IEEE Transactions on Image Processing, 8(6), 1999.
[12] J. W. Kwon, S. H. Hong, S. J. Lee, and S. B. Jung. Face detection and recognition using PCA. In TENCON 99.Proceedings of the IEEE Region 10 Conference, volume 1, pages 84{87, 1999.
[13] T. Kondo and H. Yan. Automatic human face detection and recognition under non-uniform illumination. Pattern Recognition, 32:1707-1718, 1999.
[14] C. H. Su, Y. S. Chen, Y. P. Hung, and C. S. Chen. A real-time eye-tracking system for auto-stereoscopic displays. In 15th IPPR Conference on Computer Vision, Graphics and Image Processing, pages 170{177, 2002.
[15] C. C. Han, H. Y. Mark Liao, G. J. Yu, and L. H. Chen. Fast Face Detection via Morphology-based Pre-processing. Pattern Recognition, 33 (10):1701-1812, 2000.
[16] H. S. Kim, W. S. Kang, J. I. Shin, and S. H. Park. Face detection using template matching and ellipse fitting. IEICE Trans. Inf. and Syst., pages 28-30, May 2002.
[17] V. Vezhnevets. Method for localization of human faces in color-based face detectors and trackers. In the Third International Conference on Digital Information Processing And Control In Extreme Situations, volume 11, pages 2008-2011, Nov. 2000.
[18] R. C. Gonzalea and R. E. Woods. Digital Image Processing, 2nd. Ed., Addision-Wesley, 1992.

[19] R. L. Hsu, A. K. Jain, and M. Abdel-Mottaleb. Face detection in color image. IEEE Transactions on patterm analysis and machine intelligence, 24(5):96-706, May 2002.
[20] K. Talmi and J. Liu. Eye and gaze tracking for visually controlled interactive stereoscopic displays. Signal Processing: Image Communication 14:799-810, 1999.
[21] Olivetti database, The ORL Database of Faces. Http://www.cam-orl.co.uk/facedatabase.html
[22] G.C. Feng and P.C. Yuen. Multi-cues eye detection on gray intensity image. Pattern Recognition. 34: 1033-1046, 2001.
[23] J. Huang and H. Wechsler. Eye detection using optimal wavelet packets and radial basis functions. International Journal of Pattern Recognition and Artifical Intelligence, 13(7): 1-18, 1999.
[24] P. J. Phillips, H. Wechsler, J. Huang and P. Rauss. The FERET database and evaluation procedure for face recognition, Image and Visual Computing, 16(5): 295-306, 1998.
[25] H. Han, T. Kawaguchi and R. Nagata. Eye detection based on grayscale morphology, in the Proceedings of IEEE TENCON'02, pages 498-502, 2002.
[26] A. M. Martinea and R. Benaventa. The AR face database, CVC Technical Reports, No. 24, June 1998.
[27] T. D'Orazio, M. Leo, G. Cicirelli and A. Distante. An algorithm for real time eye detection in face images, in the Proceedings of the 17th International Conference on Pattern Recognition, 1051-4651, 2004