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

Daw-Tung Lin and Chen-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
Real-time eye tracking is a challenging problem. In this paper, 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. Furthermore, 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
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 [1, 2], neural network approaches [3, 4], genetic algorithm approaches [5], and principle component analysis approaches [6]. In this paper, we used the HSI color model to extract skin-color pixels [7]. When the $H$ value of a pixel falls in the range of $[H_{Lo}, H_{Hi}]$, we consider this pixel to be a skin-color pixel. $H_{Lo}$ and $H_{Hi}$ are adapted and updated periodically during the operation. $H_{Lo}$ and $H_{Hi}$ are determined by acquiring histogram of the $H$ value of the previous time interval of the face region. $H_{Lo}$ can be determined by finding the first $H$ value whose frequency is greater than $h_H(\text{max})/10$, where $h_H(\text{max})$ is the maximum frequency in the range of $H$ from 0 to 30. $H_{Hi}$ can be determined in the range from 31 to 60 by the same method. 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 $(b_{top}, b_{bottom}, b_{left}, b_{right})$ of a face region can be defined. We further need to fill holes which may be the areas of eyes or eye brows. However, 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. The procedures are described as follows:

2 Face Circle Fitting
To detect human face in a color image, we adopted HSI color model to extract skin-color pixels [7]. When the $H$ value of a pixel falls in the range of $[H_{Lo}, H_{Hi}]$, we consider this pixel to be a skin-color pixel. $H_{Lo}$ and $H_{Hi}$ are adapted and updated periodically during the operation. $H_{Lo}$ and $H_{Hi}$ are determined by acquiring histogram of the $H$ value of the previous time interval of the face region. $H_{Lo}$ can be determined by finding the first $H$ value whose frequency is greater than $h_H(\text{max})/10$, where $h_H(\text{max})$ is the maximum frequency in the range of $H$ from 0 to 30. $H_{Hi}$ can be determined in the range from 31 to 60 by the same method. 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 $(f_{top}, f_{bottom}, f_{left}, f_{right})$ of a face region can be defined. We further need to fill holes which may be the areas of eyes or eye brows. However, 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. 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: $C_{vc} = f_{top} + r$.

The parameters $Area1$ and $Area2$ are defined as the upper area and the lower area of the circular model, respectively (see Fig. 2). $Area3[i], 0 \leq i \leq 3$ is defined as the specific areas in the upper side, lower side, left side and right side of the circular model.
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: $Area = \min(Area1, Area2)$, $r = \sqrt{Area \times 2/0.85/\pi}$, $C_{vc} = f_{top} + r$, $C_{hc}$ is no change.

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

Case 3 Compute $Area3[t]$. Assume $Area[k]$ is the minimum among $Area3[i]$, then $Area = Area[k]$ and $AreaNo = k$. If $Area < r \times (r \times 2) \times 0.5$ (rectangle area x0.5), the center is fixed and subtract 1 from $r$, else $r$ is fixed and the center is determined by $C_{vc} = C_{vc} + r$, $C_{vc} = C_{vc} - 1$, $C_{hc} = C_{hc} + 1$, and $C_{hc} = C_{hc} + 1$ for $AreaNo = 0, 1, 2, 3$, respectively.

3 Eye Detection with Dark Area Extraction

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 $e_{bottom} = C_{hc} \times 4$. We designed a Dark Pixel Filter (DPF) to extract pixels which belong to eye pixels as follows.

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

Step 2 If $(I_{(x,y-t1)} - I_{(x,y)}) > edge$, $UpB = y - t1$ goto Step 3, otherwise goto Step 5.

Step 3 If $(I_{(x,y+t2)} - I_{(x,y)}) > edge$, $DoB = y + t2$ goto Step 4, otherwise goto Step 5.

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

Step 5 Examine the next pixel.

$I_{(x,y)}$ denotes 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 as eye candidates. Examples can be seen in Fig. 3(c). Among those eye candidates obtained from the previous process, some pairs of them are just dark areas. Therefore, we must remove some unreasonably large areas by heuristic rules. By performing vertical projections on dark
Figure 3: 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.

pixels, we get $P_v(i), c_{left} \leq i \leq c_{right}$. Compute $P_v(i) = \sum_{j=i-r}^{i+r} P_h(j+r), c_{left} \leq i \leq c_{right}$ and let $P_v(m)$ be the maximum among $P_v(i)$, we will obtain $m$ as the horizontal position of one eye. Similarly, the horizontal position $n$ of another eye is located. The vertical position of the eyes will be identified by the same method. Since eyebrow and nose pixels may cause the locating process to fail, we must remove them. 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 $[n - r, n + r]$. Since the eye is right beneath the eyebrow, a pixel of the eyebrow can be determined by searching to determine whether there is a pixel below this pixel. Finally, we can group eye pixels and mark them with a rectangle shape.

4 Experimental Results

We used a Logitech QuickCam Pro3000 camera to capture users video sequence, and the format of input video is 320 × 240 true color. The system is implemented on a PC with Pentium IV 1.6 Ghz CPU with a Microsoft Windows 2000 operating system. The face and eye detection can achieve the speed of 13-25 frames per second. We have tested 630 frames of seven people with different aspect, rotation, tilt angle under normal indoor lighting conditions. As we can see in Table 1, 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. Figure 4 shows the detection results of four people.

<table>
<thead>
<tr>
<th>Head pose</th>
<th>Near-Frontal</th>
<th>Half-Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of images</td>
<td>120</td>
<td>75</td>
</tr>
<tr>
<td>Hit ratio (proposed)</td>
<td>74%</td>
<td>69%</td>
</tr>
<tr>
<td>Hit ratio (Hsu et al)</td>
<td>90%</td>
<td>75%</td>
</tr>
<tr>
<td>Speed (proposed)</td>
<td>0.06±0.02 (sec/per frame)</td>
<td></td>
</tr>
<tr>
<td>Speed (Hsu et al)</td>
<td>22.97±17.35 (sec/per frame)</td>
<td></td>
</tr>
</tbody>
</table>

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

we have tested the HHI face database and compared the proposed algorithm with Hsu’s algorithm [8]. 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). 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 5 shows eight of the eye detection results of HHI database using the proposed system.

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.

References


