PCA-Base Real-Time Face Detection and Tracking

Do-Joon Jung[1], Chang-Woo Lee[1], Yeon-Chul Lee[1],
Sang-Yong Bak[2], Jong-Bae Kim[1], Hyun Kang[1] and Hang-Joon Kim[1]

[1] Department of Computer Engineering, Kyungpook National University
702-701, 1370, Sangyuk-dong, Buk-gu, Daegu, Korea
Tel. +82-53-940-8694, Fax. +82-53-957-4846
[2] LG Eletronics Inc., 730-702, 299, Gongdan-dong, Kumi, Korea
Tel. +82-54-460-5994, Fax. +82-54-460-5104
e-mail : {djjung, cwlee, yclee, kjblove, hkang, kimhj}@ailab.knu.ac.kr
baksang@lge.com

Abstract: This paper proposes a real-time face detection and tracking a method in complex backgrounds. The proposed method is based on the principal component analysis (PCA) technique. For the detection of a face, first, we use a skin color model and motion information. And then using the PCA technique the detected regions are verified to determine which region is indeed the face. The tracking of a face is based on the Euclidian distance in eigenspace between the previously tracked face and the newly detected faces. Camera control for the face tracking is done in such a way that the detected face region is kept on the center of the screen by controlling the pan/tilt platform. The proposed method is extensible to other systems such as teleconferencing system, intruder inspection system, and so on.

1. INTRODUCTION

Video signal processing has many applications such as teleconferencing system considering with the visual aspects of communication, lip reading system for handicapped people. In many systems mentioned above, the detection and tracking of human face is an indispensable component. In the literature, several systems for tracking the face region in real-time are reported [1-3]. In general, there are two kinds of grouping of tracking methods according to their views. Some people group tracking methods as recognition-based tracking and motion-based tracking and the others group them as edge-based tracking and region-based tracking [4].

Recognition-based tracking is really based on the object recognition technique and the performance of the tracking system is limited by the efficiency of the recognition method. Motion-based tracking relies on the motion detection technique, which can be divided into the optical flow method and the motion-energy method.

Edge-based methods track the edges in an image sequence, which are usually boundaries of objects of interest. However, these methods suffer from the changes in color or illumination since boundaries of objects to be tracked have to show a strong edge variation in color or illumination. Moreover, it is difficult to provide reliable results in a case where the background of an image has strong edges. Most of the current work related to this type of method stems from the efforts of Kass et al. on snakes [5]. Since the video scenes acquired from a real-time camera generally contain many kinds of noises, many systems are hard to obtain reliable results of the tracking of faces. Many of the recent researches on face tracking are in trouble with the presence of background noise and apt to track an unverified face, for example, arms and hands.

In this paper, we propose a PCA-base real time face detection and tracking method that detects and tracks a human face with an active camera as shown in Fig. 1. This proposed method consists of two main steps: face detection and face tracking. Using two consecutive frames, first, the candiate face regions are verified to determine which region is indeed the face using PCA. Last the verified face is tracked using the eigen-technique.

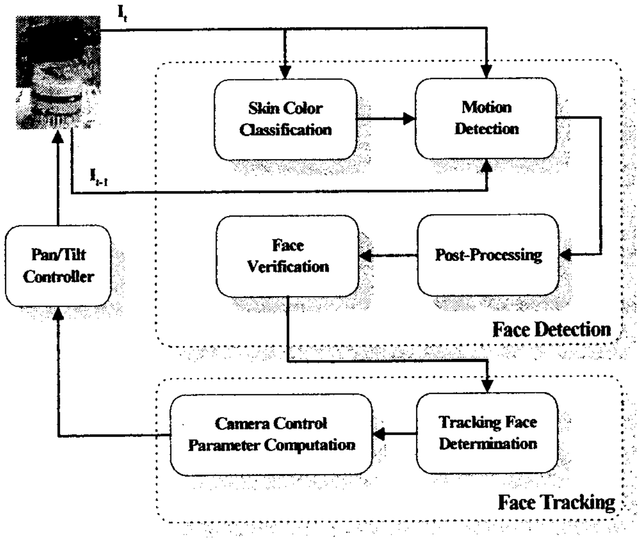

Fig. 1. The architecture of the proposed method

2. FACE DETECTION

In this section, the techniques used to detect faces in the proposed method are introduced. For improving the accuracy of the face detection, we combine several published techniques such as a skin color model [1, 6] and PCA [7, 8].

2.1 Skin Color Classification

Detecting pixels with the skin color provides a reliable method for detecting and tracking faces. Since a RGB representation obtained by most video cameras includes not only color but also brightness, this color space is not necessarily the best color representation for detecting pixels with skin color [1, 6]. The brightness may be removed by dividing the three components of a color pixel by the intensity. The color distribution of human faces is clustered in a small area of the chromatic color space and can be approximated by a 2D-Gaussian distribution. Therefore, the skin color model is approximated by a 2D-Gaussian model where the mean and variance are as follows.

$$m = (\bar{r}, \bar{g}) \quad where \;\; \bar{r} = \frac{1}{N}\sum_{i=1}^{N} r_i \;\; and \;\; \bar{g} = \frac{1}{N}\sum_{i=1}^{N} g_i \quad (1)$$

$$\Sigma = \begin{bmatrix} \sigma_{rr} & \sigma_{r} \\ \sigma_{r} & \sigma \end{bmatrix}. \quad (2)$$

Table 1 shows the mean and variance of the skin color model obtained from 200 skin color images.

Table 1. The mean and variance of the skin color

Parameters	Values
μ_r	117.588
μ_g	79.064
σ_{rr}	24.132
σ_{gr}	-10.085
σ_{rg}	-10.085
σ_{gg}	8.748

Once the skin color model is created, a straightforward way to locate a face is to match it with an input image to find the face color clusters. Each pixel of the original image is converted into the chromatic color space and then compared with the distribution of the skin color model.

2.2 Motion Detection

Although skin color is the most widely used feature, skin color alone is not suitable for the face detection in the case when skin colors appear in the background areas as well as in the human skin areas. This drawback can be effectively removed by using motion information. To be precise, after the skin classification, only those skin color regions are considered, which contain motion. As a result, the combined skin color model with motion information results in a binary image that indicates the foreground (face regions) and background (non-face region). The binary image is defined as

$$M_t(x,y) = \begin{cases} 1, & I_t(x,y) \in S_t \;\; \& \;\; |I_t(x,y) - I_{t-1}(x,y)| > \theta_t \\ 0, & otherwise \end{cases}, \quad (3)$$

where $I_t(x,y)$ and $I_{t-1}(x,y)$ are the intensities of the current frame and previous frame at pixel (x,y), respectively. S_t is a set of the skin color pixels of the current frame and θ_t is a threshold value calculated using an adaptive thresholding technique [9]. As a post-processing we simplify the M_t image using morphological operations and connected component analysis.

2.3 Face Verification Using PCA

In a sequence, tracking of the face of interest is difficult because there are many moving objects. Moreover, a process is needed to verify that the moving object is a face or not. For the face verification problem, we use the weight vectors of candidate regions in eigenspace. For the dimensionality reduction of the feature space, we project an N-dimensional candidate face image to the lower-dimensional feature space, called eigenspace or facespace [7, 8]. In eigenspace, each feature component accounts for a different amount of the variation among the face images.

To be brief on the eigenspace, let a set of images be $I_1, I_2, I_3, \ldots, I_M$, which is the N-dimensional column vector of each image and used for constructing the facespace. The average of the training set is defined by $A = 1/M \sum_{i=1}^{M} I_i$. A new set of vectors with zero mean at each dimension is computed as $\Phi_i = I_i - A$. To produce the M orthogonal vectors that optimally describe the distribution of face images, the covariance matrix is originally computed as

$$C = \frac{1}{M}\sum_{i=1}^{M} \Phi_i \Phi_i^T = YY^T, \quad (4)$$

for $Y = [\Phi_1 \Phi_2 \ldots \Phi_M]$. Since the matrix C, however, is $N \times N$ dimension, determining the N-dimensional eigenvectors and N eigenvalues is an intractable task. Therefore, for the computational feasibility, instead of finding the eigenvectors for C, we calculate M eigenvectors v_k and eigenvalues λ_k of $[Y^T Y]$, so that u_k, a basis set is computed as

$$u_k = \frac{Y \times v_k}{\sqrt{\lambda_k}}, \quad (5)$$

for k=1, ..., M. Of the M eigenvectors, the M' significant eigenvectors are chosen as those with the largest corresponding eigenvalues. For M training face images, the feature vectors $W_i = [w_1, w_2, \ldots, w_{M'}]$ are calculated as

$$w_k = u_k^T \Phi_i, \quad k = 1, \ldots, M. \quad (6)$$

To verify the candidate face region is indeed the face image, the candidate face regions are also projected into the trained eigenspace using equation (6). The projected regions are verified using the minimum distances of the detected regions with the face cluster and the non-face cluster according to equation (7).

$$\min(\| W_k^{candidate} - W_{face} \|, \| W_k^{candidate} - W_{nonface} \|), \quad (7)$$

where $W_k^{candidate}$ is the kth candidate face region in trained eigenspace, and W_{face}, $W_{nonface}$ are the center coordinate of the face cluster and non-face cluster in trained eigenspace respectively, and $\| * \|$ denotes the Euclidean distance in eigenspace.

3. FACE TRACKING

Among the newly detected faces, the face to be tracked in the next image sequence is determined by using a distance measure in the eigenspace. For tracking of the face, the Euclidian distance between the feature vectors of the face tracked and those of the K newly detected faces is calculated as

$$obj = \arg\min_k \|W_{old} - W_k\|, \quad k = 1, \ldots, K \ . \quad (8)$$

After the determination of the face region, the distance between the center of the detected face region and the center of the screen is calculated as

$$dist_t(face, screen) = Face_t(x, y) - Screen\left(\frac{height}{2}, \frac{width}{2}\right), \quad (9)$$

where $Face_t(x,y)$ is the center of the detected face region at t time and $Screen(height/2, width/2)$ is the center of the screen. Using this distance vector, the orientation and pan/tilt duration of the camera is controlled. The camera control is done in such a way that the detected face region is kept on the center of the screen by controlling an active camera mounted on the pan/tilt platform. In Table 2, the parameters used in controlling the active camera are shown. The procedure of the pan/tilt duration and the orientation of the camera is shown as following pseudo code.

Table 2. Camera control parameters

Duration		Orientation(Pan/Tilt)
None	0 sec (0 degree)	Up, Down, Left, Right, Up-Left, Up-Right Down-Left, Down-Right
Close	0.5 sec (3.1 degree)	
Far	1 sec (6.2 degree)	

Pseudo code for calculation of the pan/tilt duration and the orientation

```
Procedure Duration (x, y)
  Begin
    Sig_d = None;
    distance = sqrt(x^2 + y^2);
    IF distance > θ_close then
    Sig_d = Close;
    ELSEIF distance > θ_far then
    Sig_d = far;
    Return (Sig_d);
  End Duration;
Procedure Orientation (x, y)
  Begin
    Sig_o = None;
    IF x > θ_x then
    Add "RIGHT" to Sig_o;
    ELSEIF x < -θ_x then
    Add "LEFT" to Sig_o;
    IF y > θ_y then
    Add "UP" to Sig_o;
    ELSEIF x < -θ_y then
    Add "DOWN"to Sig_o;
    Return (Sig_o);
  End Orientation;
```

4. EXPERIMENTAL RESULTS

The experimental environment was the laboratory room where possible noises were existed and the lighting condition was changing. The camera used in this experiment is a Mitsubish CCD-300 color video camera mounted on the SPT-2410 pan/tilt platform and the speeds of panning and tilting is 6.2°/sec and 4.2°/sec, respectively. And the proposed method was tested on PentiumII-233Mhz PC running Windows 98 OS. Fig. 2 shows the setup and the interface of the proposed method.

(a) (b)

Fig. 2. The setup of the proposed method (a) the setup (b) the interface

The proposed method was tested on 20 different test sequences and the training set consists of 13 individuals at 5 different head orientations. Fig. 3 shows the part of he training images which is used in construction of the eigenspace.

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

Fig. 3. The part of the training images (a) front views (b) left views (c) right views (d) up views (e) down views

The analysis of a set of images captured during the experiment revealed that the correct rate of face verification was 94.3% in an average. In Table 3, the correct verification rates that the face region is verified as the face and the non-face region is verified as the non-face are shown.

$$\textit{Face verification rate} = \frac{\textit{Number of correctly verified faces}}{\textit{Number of images verified as true face}} \quad (10)$$

$$\textit{Non-face verification rate} = \frac{\textit{Number of correctly verified images as non\_faces}}{\textit{Number of images verified as non\_face}} \quad (11)$$

In Fig. 4, we show the results of the proposed method in which two input frames are in (a), the camera positions are in (b) and the detected faces are in (c).

(a) (b) (c)

Fig. 4. The results of the proposed method (a) video inputs (b) camera positions (c) the detected faces

Table 3. Verification rates

Face	Non-Face	Total
96.5 %	92%	94.3%

5. CONCLUSIONS

A PCA-base real time face detection and tracking method was proposed in this paper. The proposed method was operated in real-time and performed in two main steps: face detection and face tracking. In the input video sequences, first, we detected the face regions using multi-cues such as color, motion information and PCA. The tracking of a face is done in such a way that the detected face region is kept on the center of the screen through controlling an active camera mounted on the pan/tilt platform. The robustness of the proposed method in possibly noisy environment was shown in the experimental result. For the future work, we will extend the proposed method to the facial animation system through the facial feature extraction in the detected face region.

REFERENCES

[1] Jie, Y.; Waibel, A., "A real-time face tracker", *Applications of Computer Vision, 1996. WACV '96.*, pp. 142-147, 1996

[2] Spors, S.; Rabenstein, R., "A real-time face tracker for color video", *Acoustics, Speech, and Signal Processing, 2001. Proceedings.* Vol. 3, pp. 1493 -1496, 2001

[3] Schwerdt, K.; Crowley. J.L., "Robust Face Tracking Using Color", *Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000*, pp. 90 -95, 2000

[4] K.Shearer; K.D.Wong; S.Venkatesh, "Combining multiple tracking algorithms for improved general performance", *Pattern Recognition*, Vol. 34, pp. 1257-1269, 2001

[5] M.Kass; A.Witkin; D.Terzopoulos, "Snakes: active contour models", *Int. J. Computer Vision*, Vol. 1, pp. 321-331, 1996

[6] S. H. Park; E. Y. Kim; S. W. Hwang; Y. C. Lee; H. J. Kim, "Face detection for security system on the internet", *Proceeding of IEEE Consumer Electronics, 2001*, pp. 276-277, 2001

[7] M. Turk; A. Pentland, "Face Recognition Using Eigenfaces", *IEEE Proceedings CVPR '91, 1991*, pp. 586 -591, 1991

[8] Guan, A.X.; Szu, H.H., "A local face statistics recognition methodology beyond ICA and/or PCA", *International Joint Conference on Neural Network*, pp.1016-1021, 1999

[9] N. Habili; A. Moini; N. Burgess, "Automatic Thresholding for Change Detection in Digital Video", *SPIE Proceedings on Visual Communications and image Processing*, Vol. 4067, pp. 133-142, 2000