

Real-Time Face Tracking System Using Adaptive Face Detector and Kalman Filter

Jong-Ho Kim¹, Byoung-Doo Kang¹, Jae-Seong Eom¹, Chul-Soo Kim¹,
Sang-Ho Ahn², Bum-Joo Shin³, and Sang-Kyoon Kim¹

¹ Department of Computer Science, Inje University, Kimhae, 621-749, Korea
{lucky,dewey,jseom,charles,skkim}@cs.inje.ac.kr

² Department of Electronics and Intelligent Robotic Engineering, Inje University,
Kimhae, 621-749, Korea
elecash@inje.ac.kr

³ Department of Biosystem Engineering, Pusan National University, Miryang,
627-706, Korea
bjshin@pusan.ac.kr

Abstract. In this paper, we propose a real-time face tracking system using adaptive face detector and the Kalman filter. Basically, the features used for face detection are five types of simple Haar-like features. To only extract the more significant features from these features, we employ principal component analysis (PCA). The extracted features are used for a learning vector of the support vector machine (SVM), which classifies the faces and non-faces. The face detector locates faces from the face candidates separated from the background by using real-time updated skin color information. We trace the moving faces with the Kalman filter, which uses the static information of the detected faces and the dynamic information of changes between previous and current frames. In this experiment, the proposed system showed an average tracking rate of 97.3% and a frame rate of 23.5 frames per s, which can be adapted into a real-time tracking system.

1 Introduction

Recently, computer vision has been actively studied in a variety of fields.

In particular, face detection is being studied as the basic technology for face recognition and face tracking.

Using computer vision a face detection system detects a human face in real-time. Users only stare at a camera, and hence, it does not require physical contact from users and lessen the user's denial. Further, such a system can detect a face in a natural situation.

Human faces do not have fixed information as is the case with nonliving objects, instead, they possess various features such as facial expression, race, age, gender, hair style. Information is given by these features as well as mechanical features of light and of the camera, and therefore, there exist various face detection methods that have been studied.

A face detection method is divided into a local feature-based method[1] and a template based method[2]. The local feature-based method is based on the existence or nonexistence of unique facial features and the correlation of the positions of the eyes, a nose, and a mouth. This method shows a high recognition value when only one person is present in an image and the eyes, nose and mouth in the images are clear. However, it requires a considerable number of operations because it executes a scanning process using a multi scale window to detect faces of various-size in a given image.

The template-based method is divided into a shape-based method[2] and color-based method[3]. The shape based method studies face images, creates a standard template, applies a window or classifier to the input images, compares the images with the template, and detects the facial region. However, it cannot work efficiently when some part of the face is hidden by another face or shadow or the face is inclined to one side. The color-based method creates a skin color model using previously trained images and detects a face from these images by using the skin color. Color-based information has a fast processing time and detects a face image accurately with regularized value and fewer calculations. However, it is sensitive to the intensity and direction of light and cannot detect correctly when the background color and skin color are similar.

To solve these problems, we propose a real-time face tracking method using an adaptive face detector and the Kalman filter.

The face detector is constructed based on simple Haar-like features, PCA[4], and SVM[5]. The detector has acceptable detection speed and it is not affected to a large extent by the size of the training dataset. As such, it works well with a small quantity of training data. We can trace a moving face with the Kalman filter that uses static information from the face detector and dynamic information of changes between the previous and current frames.

The proposed system performs better face detection because it uses effective features selected from simple Haar-like features with PCA. The SVM classifier, which works well with the binary classification of faces and non-faces, also contributes toward better face detection. The Kalman filter, which has the best prediction ability, makes it possible for the system to trace faces efficiently using the face detector.

Furthermore, the face detector does not work for the entire image in a frame but it works only for face candidates that are separated by real-time updated skin-color information. This improves the processing time and decreases the mis-detection rate.

2 Overview of Face Tracking System

Fig. 1 shows the primary structure of our face tracking system.

In the classifier construction step, the proposed system extracts Haar-like features from the learning images, and subsequently extracts efficient features from them using PCA : these are then user as the SVM learning data and form an SVM face classifier. In the skin color detection step, the proposed system extracts face candidates using real-time updated skin color information. Lastly, it tracks the detected face by using the Kalman Filter.

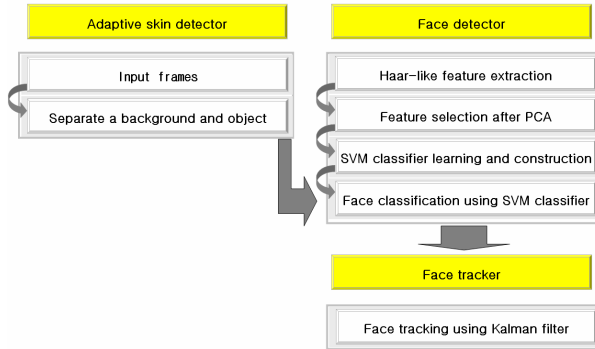


Fig. 1. Primary structure for face tracking

3 The Adaptive Skin Detection Algorithm

3.1 Training the Skin Color

The skin color training using 100 sets of faces and hands after manually segmenting the skin region. These images were captured using a PC camera and collected from the internet. Fig. 2 shows the skin color training images used in this study.



Fig. 2. skin color training images used in this study

3.2 Detection of Skin Pixels

In this step, we detect the skin color regions from the images by using trained skin color. Among these regions a region whose position is not changed in the next frame is not regarded as a face candidate. Only if we are able to compare the change in the position of a skin-color region in the current frame from that in the previous frame can we mathematically reconstruct skin color information, for example, as given by Eq.1.

$$S_{n+1} = (1-\alpha) \cdot S_n + \alpha \times S_m \quad (1)$$

- S_{n+1} is the new skin color for the next frame.
- S_n is the skin color region in the current frame.
- S_m is the in-motion pixels of the skin color.
- α is the weight for merging two frames.(0.05)

In this study, we use an α value of 0.05, which shows the best result. Moreover, we separate a skin color region and a background region by using the updated skin color. We eliminate noisy pixels using an Opening formula in the separated image. Subsequently, we eliminate the skin color region having fixed value from the face candidates by moving a 24×24 window by 12 pixels. We select the face candidates after combining adjacent windows.

The outputs of this step are those pixels that have with a higher probability of belonging to the skin color regions of the image.

Fig. 3(a) is an original picture. Fig. 3(b) is the picture in which the face candidate is detected by using trained skin color. Fig. 3(c) is the picture in which the skin color is updated and a small skin color region is removed by a 24×24 window. Fig. 3(d) is the final selected face candidates.

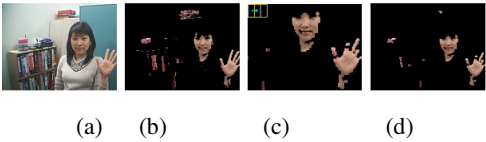


Fig. 3.

4 Construction of Face Detector

In this step, a face is detected by using a detector from the face candidates obtained in the previous step.

First, the detector detects Haar-like features from a face image.

Then, it selects features that can be used to judge whether a detected Haar-like feature is a face region or a non-face region by using PCA. The feature space is transformed into the principal component. The features selected from the principal component space are used as feature vectors of the SVM.

In the last step, an SVM classifier divides the face candidates into a face image and a non-face image using a trained pattern.

4.1 Feature Extraction

The face detector is based on the simple rectangular features that were presented by Viola and Jones[6]. It measures the differences between the regional averages at various scales, orientations, and aspect ratios. The rectangular features can be rapidly evaluated at any scale (see fig. 4). However, these features require very large training datasets. Therefore, after analyzing the principal components, we select only the useful features from each of the five rectangular features. These selected features are used as a feature vector for the SVM. The experiments demonstrate that they provide



Fig. 4. Haar-like features used in this study

useful information and improve the performance of accurate classification when small training datasets are used.

We used 12 principal components that explain features having a cumulative explanation rate greater than 90%. From these 12 principal components, we selected 288 useful features from the entire range of the 162,336 simple Haar-like features. Fig. 5 shows the 288 useful features selected by using PCA. Consequently, a training image is converted to 288 values corresponding to these useful features, and our SVM classifier uses this input vector having of 288 dimensions for training.



Fig. 5. The 288 useful features selected for this study

4.2 Training of Classifier

The training data were 1000 face images and 1000 non-face images randomly selected from the MIT CBCL face data set[7]. Each image was normalized to 24×24 pixels.

Fig. 6 gives an overview of the detection progress, which comprises simple feature extraction, feature analysis, and classifier construction. Firstly, from the simple Haar-like features, useful features are selected by using PCA. Training images, as shown in fig. 5, are converted into input vectors having 288 dimensions with the selected features. The SVM classifier uses these input vectors for training.

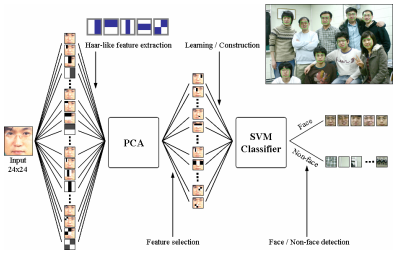


Fig. 6. Primary structure for face detection

5 Face Tracking with Kalman Filter

In this paper, we used the Kalman filter[8] to reduce the cost of operation and improve the tracking rate in the sequence of video images. The state vector of the Kalman filter uses static information from the face detector and dynamic information between frames.

For efficient multiple objects tracking, the Kamlan filter requires a setting of an appropriate trace model. We set the state vector as the center coordinates (x, y) of

the detected face and the quantity of change $(\Delta x, \Delta y)$ between previous and current frames.

The state vector of the Kalman filter in time t is defined as:

$$\mathbf{x}(t) = [x, y, \Delta x, \Delta y]^T, \quad (2)$$

The Kalman filter assumes that the system state vector, $\mathbf{x}(t)$, evolves according to time as:

$$\mathbf{x}(t+1) = \Phi(t)\mathbf{x}(t) + \mathbf{w}(t) \quad (3)$$

where $\mathbf{w}(t)$ is a zero mean Gaussian noise with covariance $\mathbf{Q}(t)$.

$$\mathbf{Q}(t) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

The measurement vector is given by:

$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{v}(t) \quad (5)$$

where $\mathbf{v}(t)$ is another zero mean Gaussian noise factor, with covariance $\mathbf{R}(t)$.

The covariance $\mathbf{R}(t)$ is defined as

$$\mathbf{R}(t) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (6)$$

We assumed that faces move with uniform speed and linear direction. The state transition matrix $\Phi(t)$ is defined as follows:

$$\Phi(t) = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

The input vector is a four-dimensional vector that uses the center coordinates of the multi object and changes of the x , y , Δt , Δt axis. Therefore, the measurement matrix $\mathbf{H}(t)$ is defined as:

$$\mathbf{H}(t) = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

The error is calculated using the margin of the measurement z_k and the prediction $\mathbf{H}_k \mathbf{x}_{k/k-1}$ of the previous step, the estimate is updated using the Kalman gain of the k state as a weight, and calculate the optimal estimate $\mathbf{x}_{k/k}$ finally.

6 Experimental Results

The proposed face tracking system was developed using Visual C++ on a 2.4GHz P4 PC with a Microsoft Windows operating system. To evaluate our system, we experimented with face detection and tracking on various video sequences. The video sequences were collected from various sources such as the Open-Video website[9], captured video from TV broadcasts, Boston University IVC Head Tracking Video Set[10], and PC cameras.

6.1 The Result of Skin Color Detection According to the Rate of Trained and Updated Skin Color

Fig. 7 shows a true positive rate and false negative rate depending on the merging weight value α of Eq.1.

True positive means that the skin color regions are correctly recognized as skin color regions, while false negative means that the background regions are incorrectly recognized as skin color regions.

When the value of α was near 0, the initial skin color value was not updated and some of the background regions incorrectly recognized as the skin color regions never shift to the background regions. On the contrary, when the value of α was near 1, the skin color value changed excessively to such an extent that mis-recognition rate increased. Experimentally, we used 0.05 as the value of α and this shows the best result.

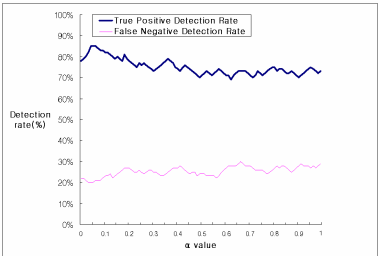


Fig. 7. Detection rate according to the Value of α

6.2 Tracking Experiment for Images Obtained Under Various Conditions

Figs. 8 (a) and (b) shows the face detection and tracking results from sequences under various conditions. As shown in fig. 8(a), although face detection fails due to non-inclusion of rotated faces in the training of the face detector, face tracking is possible.

Fig.8 (b) shows successful face tracking in which occlusion occurs locally in the face region.

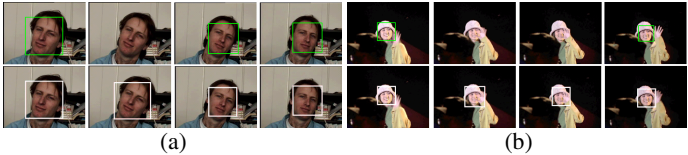


Fig. 8. (a) Results of face tracking in a heavily rotated face and (b) local occlusion sequence

Fig. 9 shows examples of detecting a non-face region as a face region when the face candidates are not assigned by a skin color. However, we detect only a face region completely such as in fig. 10 by limiting the face detection region to face candidates.



Fig. 9. Examples of recognizing a non-face image as a face image



Fig. 10. Examples of detecting a face image from face candidates

6.3 Comparison with Related Works

To compare our tracking system with related works, we selected a method from among the various tracking methods. This method[11] uses Viola-Jones’s basic features and Lienhart’s[12] extended Haar-like features in order to increase the detection rate of various facial poses. It creates a one-dimensional deformable face graph after subdividing the eyes and mouth in the detected face. When it fails to detect the face, it matches the face graph of the previous frame with that of the current frame by using DP(Dynamic Programming) to continuously trace the faces.

The steps used for comparison in our approach are the same as those used in method. The input sequence has up, down, left, and right face poses and frequent light changes. We experimented on 2000 frames from 10 sequences, and the results are listed in Table 1.

Table 1. The comparison of the obtained results using our system and those obtained using method[11]

Sequence	Method[11]				Our Approach			
	D.F.	D.R. (%)	T.F.	T.R. (%)	D.F.	D.R. (%)	T.F.	T.R. (%)
#1 (jam5.avi)	33	83.5	0	100	12	94.0	0	100
#2 (jary.avi)	38	81.0	0	100	40	80.0	8	96.0
#3 (jim2l.avi)	47	76.5	16	92.0	20	90.2	4	98.0
#4 (llrx.avi)	104	48.0	40	80.0	47	76.5	12	94.0
#5 (llm1.avi)	30	85.0	0	100	14	93.0	0	100
#6 (vam7.avi)	76	62.0	24	88.0	35	82.6	3	95.5
#7 (jam9.avi)	84	58.0	12	94.0	37	81.4	5	97.5
#8 (llm1r.avi)	35	82.5	5	97.5	50	72.0	12	94.0
#9 (llm4.avi)	90	55.0	20	90.0	24	87.8	5	97.5
#10 (mll6.avi)	31	84.5	0	100	8	96.0	0	100
Total	689	71.6	117	94.2	287	85.4	49	97.25

(D.F.: Detection fail, D.R.: Detection rate, T.F.: Tracking fail, T.R.: Tracking rate)

When we apply our method to an image(#8), the track rate is less than that of method[11]. For face candidates the face image is not detected accurately because the face is hidden by a shadow.

However, for the sequences of up, down, left, and right facial movements, such as #7 and #9, our system generally showed a higher tracking rate. In particular, our system

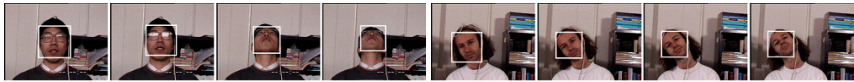


Fig. 11. Tracking result for sequences #4 and #6

Table 2. The comparison results for face detection from an entire image with those from face candidates

Sequence	Face detection from an entire image			Face detection from face candidates		
	D.R. (%)	T.R. (%)	F.P.S	D.R.(%)	T.R. (%)	F.P.S.
#1 (jam5.avi)	94.0	100	19	94.0	100	23
#2 (jary.avi)	83.6	94.4	18	80.0	96.0	24
#3 (jim2l.avi)	90.2	97.8	20	90.2	98.0	24
#4 (llrx.avi)	75.8	89.6	19	76.5	94.0	25
#5 (llm1.avi)	93.0	100	19	93.0	100	23
#6 (vam7.avi)	82.6	95.5	21	82.6	95.5	22
#7 (jam9.avi)	81.4	97.2	20	81.4	97.5	24
#8 (llm1r.avi)	75.0	93.2	21	72.0	94.0	25
#9 (llm4.avi)	87.8	94.6	20	87.8	97.5	24
#10 (mll6.avi)	96.0	100	19	96.0	100	25
Total	85.9	96.2	19.7	85.4	97.3	23.5

(D.R.: Detection rate, T.R.: Tracking rate, F.P.S. : Frames Per Second)

was much better than method[11] in which the front of the face does not appear during the given time and the face is extremely rotated (sequences #4 and #6) when the sequence involved a moving face(sequences #1 and #3) as shown in fig. 11.

Table 2 shows the results obtained from the detection of a face from an entire image and from face candidates by using skin color.

Face detection in the skin color regions show greater improvement in processing time, detection rate, and mis-detection rate then those for the entire image as shown in fig.4. In particular, the processing time is improved by more than 20% as compared to the case when the skin color region is not applied. This is because the system detects a face not from entire images but from face candidates, and it shows an improvement in the mis-detection rate of incorrect recognition of a background as a face because the system filters a possible face recognition region from the background.

In the experiment, a detector combining skin color with PCA and SVM showed an average detection rate of 85.4%. A tracking system showed an average tracking efficiency 97.3% by employing a detector that guarantees high detection rates and optimum face position through the prediction process of the Kalman filter. Moreover, it showed a processing time of 23.5 frames per s, which can be adapted into a real-time processing system.

7 Conclusion

In this paper we proposed a real-time face detection system to track a face from various input images that have illumination changes, complex backgrounds, various facial movements, and various poses.

Firstly, we designed an effective face detector using PCA and SVM. The useful features that discriminate between faces and backgrounds are extracted from simple Haar-like features with PCA. The feature vectors are used for learning patterns for SVM, which is appropriate for binary classification. The Kalman filter for tracking uses the face positions in each frame, which is the result of the face detector and changes in face position between frames, as the parameters for the state vector. The prediction process of the Kalman filter presents an optimal face position prediction for the next frame. Furthermore, the proposed system detected face candidates using real-time updating of changes in skin color information with position, improved the processing time, and decreased the incorrect detection rate by detecting a face from face candidates. Consequently, we implemented a tracking system combining an adaptive face detector with a high detection rate and the prediction ability of Kalman filter to obtain synergy.

In the experiment, we obtained an average tracking rate of 97.3% and 23.5 frames per s on sequences of 320×240 pixel images that are available in a real-time system.

However, it is difficult to track a face when the light is dim, the face is rotated quickly, and when some part of the face is continuously hidden.

Therefore, we are trying to study the detection of a face in various situations and how to tracking a face using several cameras when other objects hide some part of the face or when people rotate their face quickly.

References

1. Park, T., Park, S.K., Park, M.: An effective method for detecting facial features and face in human-robot interaction. *Information Sciences* 176, 3166–3189 (2006)
2. Brunelli, R., Poggio, T.: Face recognition: Features versus Templates. *IEEE Trans PAMI* 15, 1042–1052 (1993)
3. Hsu, R.L., Abdel-Mottaleb, M., Jain, A.K.: Face detection in color images. *IEEE transaction on Pattern Analysis and Machine Intelligence* 24(5), 696–706 (2002)
4. Johnson, R.A., Wichern, D.W.: *Applied Multivariate Statistical Analysis*, pp. 356–395. Prentice-Hall, Englewood Cliffs (2002)
5. Vapnik, V.: *The Nature of Statistical Learning Theory*, 2nd edn. Springer, Heidelberg (2001)
6. Viola, P., Jones, M.J.: Robust Real-Time Face Detection. *International Journal of Computer Vision* 57(2), 137–154 (2004)
7. MIT CBCL - Face Database <http://www.ai.mit.edu/projects/cbcl/>
8. Welch, G., Bishop, G.: An Introduction to the Kalman filter. University of North Carolina at Chapel Hill, Department of Computer Science, TR 95-041 (2004)
9. Open-Video <http://www.open-video.com>
10. Boston University IVC Head Tracking Video Set <http://www.cs.bu.edu/groups/ivc/>
11. Yao, Z., Li, H.: Tracking a Detected Face with Dynamic Programming. *Image and Vision Computing* 24(6), 573–580 (2006)
12. Lienhart, R., Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. *IEEE Int'l Conf. Image Processing* 1, 900–903 (2002)