Robust Face Tracking Using Particle Filter Based on Deformable Face Model and Skin Color

Fatemeh HourAli (1) – Mohammad Hossein Sedaaghi (2)
(1) Department of Electrical Engineering, Sahand University of Technology, Tabriz, Iran
F_Hourali@sut.ac.ir
(2) Department of Electrical Engineering, Sahand University of Technology, Tabriz, Iran
Sedaaghi@sut.ac.ir

Abstract: This paper presents an algorithm for real-time and robust human face tracking against pictures and other objects. Face detection is based on template matching, morphological operations and skin color segmentation and motion information. Tracing is performed using particle filter dependent upon skin color and PCA model for face. The face detector locates human faces from the face candidates by using motion information. Then the detected face is tracked using particle filter based on skin color and the probabilistic face model, which is updated using the information related to variability with respect to head rotation, illumination, facial expression, and occlusion. The proposed face tracking method is real-time and achieves high performance, robustness to illumination variations and geometric changes (such as viewpoint and scale changes). The experimental results show that the proposed algorithm is the best comparing with other methods specially when there is occlusion. Also the robustness of the suggested method, when the tracked face is close to an equicolor object, is proved.

Index Terms: Face detection, skin segmentation, probabilistic face model, particle filter, face tracking.

I. INTRODUCTION

Face tracking plays an important role in many applications such as video indexing, visual surveillance, human computer interaction or facial expression recognition [1]. In these applications it is necessary to detect the face, track it from frame to frame and analyze the track, for instance to understand its behaviour. In the simplest form, a tracker estimates the face trajectory by locating its position in every frame of the sequence. Suandi et al. present a technique to estimate human face pose from color video sequence using dynamic Bayesian network (DBN). As face and facial features trackers usually track eyes, pupils, mouth corners and skin region (face), their proposed method utilizes merely three of these features – pupils, mouth centre and skin region to compute the evidence for DBN inference [2]. An algorithm for human face tracking and facial feature based head tilt angle extraction based on a combination of color segmentation, connected component labelling and morphology is developed by Kabakli et al. [3]. Zheng and Bhandarkar in [4] proposed an algorithm, called “A boosted adaptive particle filter” for integrated face detection and face tracking. Their proposed algorithm is based on the synthesis of an adaptive particle filtering and Adaboost face detection algorithm. Verma et al. in [5] presented a new probabilistic method to detect and track face in a video sequence. They integrated the information of face probabilities provided by the detector and the temporal information provided by the tracker to produce a method superior to the available detection and tracking methods.

Liu and Wang in [6] introduced a new approach for combined face detection and tracking in video. The face detection algorithm is a fast template matching procedure using iterative dynamic programming. Vasant et al. in [7] proposed a detection algorithm that uses a statistical approach and was originally developed by Viola and Jones[8]. This algorithm used a feature set that is Haar-like and a cascade of boosted decision tree classifiers as a statistical model. Tracking was accomplished as continuous detection with the detected objects in two frames mapped using a greedy algorithm based on the distances between the centroids of bounding boxes. Yang and Liu in [9] proposed a hierarchical face detection method using the template matching algorithm and 2D PCA algorithm. The method includes two different classifiers. The first one is called rough classifier, which filters the most of the non-face. The second one is a core classifier which uses 2D PCA algorithm to detect the face based on the result from the first classifier.

In this paper an algorithm based on template matching, motion information and skin color segmentation is proposed for detecting the human face. Particle filter is based on a probabilistic model and skin color for tracking the human face against the picture and other objects. It has consistently provided performance which satisfies the following requirements: (1) the ability to automatically determine the initial position and size of the human face against picture and other objects and size and orientation of the detected face in the sequence of frames; (2) is insensitive to face orientation and scale changes; and (3) is also insensitive to lighting condition changes and when occlusion occurs locally in the face region. The remainder of the paper is organized as follows: Face detection method which is used for locating the human face against picture and other object is described in Section 2. The proposed Face tracking method is described in section 3. In Section 4, implementation and comparison with other algorithms is performed and finally, conclusions will be given in Section 5.

II. FACE DETECTION

The face detection algorithm used in this paper is based on skin segmentation, morphological operation and
A. Skin Segmentation

The first step of skin segmentation is to make a skin model. The input image has RGB format and is sensitive to lighting conditions because the brightness and color information are coupled together but it is not suitable for color segmentation under unknown lighting conditions. Therefore, color system transformation is required for skin’s color segmentation. We employ YCbCr color system (Y represents the luminance component while Cb and Cr represent the chrominance components of a color image). The color distribution of skin colors of different people was found to be clustered in a small area of the chromatic color space. The Y component is discarded as it is related to brightness, but Cb and Cr components are used because they contain the color information. A manually selected skin samples from color images were used to determine the color distribution of human skin in chromatic color space. The 2D Joint Gaussian model and the 2D histogram of Cb Cr values of sample skin images are shown in Fig.1. Then, skin likelihood value is computed for each pixel by computing Mahalanobis distance from mean. In the next step, skin likelihood values are normalized and shown as grayscale image where whiter areas have higher probability of being skin than the darker parts. After getting a grayscale image of skin likelihood, it is then necessary to threshold the image into a binary image. After an optimal threshold is set, all pixel values having likelihood values higher than the threshold are set to 1 and the rest of the pixels are set to 0 [10].

\[
F = \frac{1}{A} \sum_{i=1}^{a} \sum_{j=1}^{b} jB[i, j]
\]

\[
\overline{F} = \frac{1}{A} \sum_{i=1}^{a} \sum_{j=1}^{b} iB[i, j]
\]

(1)

The orientation angle is found by (2).

\[
\theta = \frac{1}{2} \tan \left( \frac{b}{a - c} \right)
\]

\[
a = \sum_{i=1}^{a} \sum_{j=1}^{b} x B[i, j], b = \sum_{i=1}^{a} \sum_{j=1}^{b} y B[i, j], c = \sum_{i=1}^{a} \sum_{j=1}^{b} x y B[i, j],
\]

\[
x' = x - \overline{x},
\]

\[
y' = y - \overline{y}
\]

(2)

where B is the image, and A is the area related to each region.

Figure1. (a) The 2D Joint Gaussian Model. (b) The 2D Histogram of Cr Cb values of sample skin images.

B. Morphological Operations

Morphological operations such as filling, erosion and dilation are applied in order to separate the skin areas which are loosely connected. Morphological closing is applied firstly to the binary image. Then, aggressive morphological erosion is applied using a proper structuring element. Then, morphological dilation is applied to re-grow the binary skin areas which are lost due to erosion in the previous step. Next, the dilated binary image is multiplied with the binary image from segmentation process to preserve the holes. This is due to later stage which will use number of holes to filter out some non-face regions. The resulting binary image needs to be labeled so that each clustered group of pixels can be identified at a single region and each region can further be analyzed to determine whether (or not) it is a face region. Under the assumption that a face region will contain at least one hole, the regions without holes are rejected by the next stage. All these steps are illustrated in Fig.2, for a sample image.

The center of the mass of each region is determined as follows:

Figure2. (a) original color image. (b) skin likelihood regions. (c) skin segmented regions. (d) image after binary closing and hole filling and connected regions labeling.

Template matching is the final stage of face detection where cross correlation between the face template and grayscale region is computed. The face template is an average frontal face of different faces taken from men and women without glasses and facial hair. The current template can dynamically be updated in size and content to adapt to temporal changes of the tracked face’s scale and orientation. The mass center of the segmented skin region is used to place the template directly in the center of the segmented image. The rotated template needs to be cropped properly and its size needs to be similar to that of the region. This process will completely fill the segmented area with the image of the template. Once the
template is placed inside the segmented image, it is necessary to see how well the template will fit inside the region. A way to determine this value is to find the two-dimensional correlation coefficient between the two matrices. It is computed as:

$$r = \frac{\sum \sum (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\sum \sum (A_{mn} - \bar{A})^2} \sqrt{\sum \sum (B_{mn} - \bar{B})^2}}$$  \hspace{1cm} (3)

where \(A\) and \(B\) are matrices of each region and the resized and rotated template which have the same size. \(\bar{A}\) and \(\bar{B}\) are average or mean of matrix elements.

### III. THE PROPOSED FACE TRACKING METHOD

We present details of the proposed algorithm for face tracking in this section. This algorithm is based on particle filter using the face model which can dynamically be updated in size and content to adapt to temporal changes of the face’s scale and orientation. Updating is performed using the information related to variability with respect to head rotation, illumination, facial expression and occlusion.

First the method for updating face model is described, next we describe our approach for drawing particles in the motion parameter space and predicting the most likely object location with the help of the learned face model and skin color information.

#### A. UPDATING FACE MODEL

A space built by Principle Component Analysis is used to model the face. This space can describe the range of appearances such as lighting variation, head rotation, facial expression and occlusion. This subspace will be updated by using a variant sequential Karhunen-Loeve algorithm [11] which in turns is based on the classic R-SVD method. The appearance of the face may change drastically due to temporal changes of the face’s scale, orientation and illumination condition. Therefore it is important to adapt the appearance model online, while tracking, to reflect these changes [11]. The appearance model is typically learned from a set of training images \(\{I_1,\ldots,I_n\}\), by taking computing the eigenvectors \(U\) of the sample covariance matrix \(\frac{1}{n-1} \sum_{i=1}^{n} (I_i - \bar{I})(I_i - \bar{I})^\top\) where \(\bar{I}\) is the sample mean of the training images. If we assume \(A = [I_1, I_2,\ldots, I_n]\) are the set of face images related to the previous frames and \(B = [I_{n+1}, I_{n+2},\ldots, I_{2n}]\) are the set of face images in the recent frames with changes in the scale and orientation of the head and the illumination conditions with respect to the previous frames, \(C\) is their concatenation and \(\bar{I}_A, \bar{I}_B, \bar{I}_C\) are their means. \(U\) and \(\Sigma\) can be calculated from SVD of \((A - \bar{I}_A)\), also \(U'\) and \(\Sigma'\) from SVD of \((C - \bar{I}_C)\). Then \(\bar{I}_C\) and \(\hat{B}\) and \(\bar{B}\) and \(R\) can be calculated using (4).

\[
T_c = \frac{n}{n+m}T_A + \frac{m}{n+m}T_B
\]

$$\hat{B} = [(I_{n+1} - \bar{B})(I_{n+2} - \bar{B})\cdots(I_{2n} - \bar{B})][(m/m) - (n/m)](\bar{B} - \bar{T}_A)$$  \hspace{1cm} (4)

\[
\hat{B} = \text{orth}(\hat{B} - \bar{U}U^\top \hat{B})
\]

\[
R = \left[ f \sum U^\top \hat{B} \right]
\]

In which \(f \in [0,1]\) is the forgetting factor. This factor is used to focus more on recently-acquired face images and less on earlier observations.

#### B. PARTICLE FILTERING FOR VISUAL TRACKING

The problem of tracking can be formulated as the Bayesian filtering

\[
p(x_t | z_{1:t}) \propto p(z_t | x_t) \int p(x_t | x_{t-1}) p(x_{t-1} | z_{1:t-1}) dx_{t-1}
\]

where \(x_t\) and \(z_t\) denote the hidden state of the object and observation vector at discrete time \(t\), respectively, whereas \(z_{1:t} = z_1,\ldots,z_t\) denotes all the observations up to current time step. With this recursion we can calculate the posterior, given a dynamic model \(p(x_t | x_{t-1})\) describing the state propagation and an observation model \(p(z_t | x_t)\) describing the likelihood that a state \(x_t\) causes the measurement \(z_t\), together with the following conditional independence assumptions we have: \(x_t \perp \perp z_1,\ldots,z_{t-1} \mid x_{t-1}, z_t\) and \(z_t \perp \perp x_{t-1} \mid x_t\).

Recently, sequential Monte Carlo methods, also known as particle filters, have become increasingly popular stochastic approaches for approximating posterior distributions \(p(x_t | z_{1:t})\). Particle filter operates by approximating the posterior distribution using a collection of weighted samples \(S_t = \{(x^{(1)}_t, \pi^{(1)}_t) | n = 1\ldots N \}_{t}\), where each sample \(x^{(n)}_t\) represents hypothesized state of the target and the weights are normalized such that \(\sum \pi^{(n)}_t = 1\).

The evolution of the sample set takes place by drawing new samples from a suitably chosen proposal distribution which may depend on the old state and the new measurements, i.e., \(x^{(n)}_{t+1} \sim \pi^{(n)}_{x^{(n)}_{t+1}} q(x^{(n)}_{t+1} | x^{(n)}_t, z_t)\) and then propagating each sample according to probabilistic motion model of the target. To give a particle representation of the posterior density the samples are set to

\[
\pi^{(n)}_{x^{(n)}_t} \propto \pi^{(n)}_{x^{(n)}_t} p(z_t | x^{(n)}_t) p(x^{(n)}_t | x^{(n)}_{t-1})
\]

\[
\int q(x^{(n)}_t | x^{(n)}_{t-1}, z_t)
\]

(6)
The particles should be re-sampled according to their weights to avoid degeneracy. Particle filters rely on importance sampling and in consequence their performance depends on the nature of the proposal distribution [12].

The location of face in an image frame can be represented by an affine image warp. This warp transforms the image coordinate system, centering the face within a canonical box such as the unit square.

In this work the state at time $t$ consists of the six parameters $X_t = (x_i,y_i,\theta_i,s_i,\alpha_i,x_i^{CbCr})$ where $x_i,y_i,\theta_i,s_i,\alpha_i,x_i^{CbCr}$ denote $x,y$ translation, rotation angle, scale, aspect ratio, and the $Cb$ and $Cr$ value of the $i$th sample in the $t$th frame.

To develop a tracker for generic applications, the dynamics between states in this space is modeled by Brownian motion. Each parameter in $X_t$ is modeled independently by a Gaussian distribution around its counterpart in $X_{t-1}$, and thus the motion between frames is itself an affine transformation. Specifically, the dynamic model can be represented as:

$$p(X_t | X_{t-1}) = N(x_i,x_0,\sigma_i^2)N(x_0,x_0,\sigma_0^2)$$

$$N(y_i,y_0,\sigma_i^2)N(y_0,y_0,\sigma_0^2)$$

$$N(s_i,s_0,\sigma_i^2)N(s_0,s_0,\sigma_0^2)$$

$$N(\alpha_i,\alpha_0,\sigma_i^2)N(\alpha_0,\alpha_0,\sigma_0^2)$$

$$N(x_i^{CbCr},x_0^{CbCr},\sigma_i^{CbCr})N(x_0^{CbCr},x_0^{CbCr},\sigma_0^{CbCr})$$

(7)

where $N(z | \mu,\sigma^2)$ denotes evaluation of the normal distribution function for data point $z$, using the mean $\mu$ and variance $\sigma^2$.

These parameters dictate the kind of motion of interest to a tracker and this generic dynamical model assumes the variance of each affine parameter does not change over time. Since our goal is to use a representation to describe the object that we are tracking, we model image observations using a probabilistic interpretation of principal component analysis and skin color model.

Many researches show that human skin colors of different people are clustered in a certain transformed 2-D color space. RGB representation expresses not only the color but the brightness as well. However, the brightness can be changed easily by a change in illumination conditions. Thus we use the YCbCr color space, the Y in YCbCr denotes the luminance component, and Cb and Cr represent the chrominance factors. As discussed in previous section, we construct skin color model in YCbCr color space. We define the following likelihood function as the skin color observation model. We represent the measurement weight $\pi(x_i^{CbCr})$ as the likelihood of skin color of state vector $x_i^{\gamma}$.

$$\pi(x_i^{CbCr}) = e^{(x_i^{CbCr} - m)^T\sigma^{-1}(x_i^{CbCr} - m)/2}$$

(8)

Given an image patch $I_i$ predicted by $X_i$, we assume it was generated from a subspace of the target object spanned by $U$ and centered at $\mu$. The probability of a sample being generated from this subspace is inversely proportional to the distance $d$ from the sample to the reference point (i.e., $\mu$) of the subspace, which can be decomposed into the distance-to-subspace, $d_u$, and the distance-within subspace from the projected sample to the subspace center, $d_w$. If we assume $x$ (image observation) is a high dimensional data sample and $z$ as the corresponding low dimensional latent variable. We can define a latent model with Gaussian noise $\varepsilon ~ N(0,\sigma^2 I_d)$ as (9).

$$x = Wz + \mu + \varepsilon$$

(9)

Where $W$ is an orthogonal matrix, and $z$ is a zero mean Gaussian $z \sim N(0,L)$ with $L$ being a diagonal matrix. From equation (9), we have

$$p(x | z) = N(Wz + \mu, \sigma^2 I_d)$$

(10)

In addition, we can compute the likelihood that observed sample $x$ is generated from the model, $p(x)$:

$$p(x) = \int p(x | z)p(z)dz = N(\mu,W LW^T + \sigma^2 I_d)$$

(11)

By taking the log of $p(x)$, we get

$$\log p(x) = \frac{1}{2}\{\log(2\pi)^d + \log |W LW^T + \sigma^2 I_d| + (x - \mu)^T(W LW^T + \sigma^2 I_d)^{-1}(x - \mu)\}$$

(12)

When parameters $W$, $L$ and $\sigma^2$ are fixed, $\log p(x)$ is determined by:

$$\log p(x) = (x - \mu)^T(W LW^T + \sigma^2 I_d)^{-1}(x - \mu)$$

(13)

According to the Sherman-Morrison Woodbury formula(14):

$$(W LW^T + \sigma^2 I_d)^{-1} = \frac{1}{\sigma^2}I_d - \frac{1}{\sigma^2}W^T(W^{-1} + \frac{1}{\sigma^2}I_d)^{-1}W^{-1}\frac{1}{\sigma^2}$$

(14)

We have:

$$W LW^T + \sigma^2 I_d)^{-1} = \frac{1}{\sigma^2}(I_d - WW^T) + W D^{-1}W^T$$

(15)

where $D$ is a diagonal matrix with $D = L + \sigma^2 I_d$.

In the probabilistic PCA model, $W$ and $D$ correspond to the eigenvectors and the diagonal matrix of eigenvalues of the sample covariance matrix. It is clear that $\log p(x)$ is determined by two terms: the Euclidean distance to the subspace$(d_u)$, $(x - \mu)^T(I_d - WW^T)(x - \mu)$ weighted by $\frac{1}{\sigma^2}$ and the Mahalanobis distance within the subspace$(d_w)$. $(x - \mu)^T D^{-1}W^T(x - \mu)$. Thus we have
\[
\pi(x_i) = N(x_i; \mu, \Sigma^{-1} U^T)
\]

where \( I \) is an identity matrix, \( \mu \) is the mean, and \( \varepsilon I \) term corresponds to the additive Gaussian noise in the observation process and \( \Sigma \) is the matrix of singular values corresponding to the columns of \( U \). Thus using (8) and (16) we have:

\[
\pi(x_i) = \pi(x_i^n) = N(x_i; \mu, UU^T + \varepsilon I)
\]

\[
N(x_i; \mu, U \Sigma^{-2} U^T) + e^{(x_i - \mu)^T \Sigma^{-2} (x_i - \mu)} / (2\pi)^{d/2} N(0, \Sigma^{-1})
\]

C. Error Function for Minimizing the Effect of Noisy Pixels

To minimize the effect of noisy pixels, we utilize a robust error norm:

\[
E = \frac{1}{N} \left( \exp\left(-\frac{x^2}{\sigma^2} + x \right) \right) + \frac{n}{N}
\]

where \( N \) is the number of the whole pixels inside the box containing the target and \( n \) is the number of skin color pixels inside this box. Also this error function is used to ignore the outliers (i.e., the pixels that are not likely to appear inside the target region given the current eigenbasis). This robust error norm is helpful when we use a rectangular region to enclose the target object (which inevitably contains some noisy background pixels).

IV. Experimental Results

In this section, we will evaluate performance of the proposed face tracking with the described algorithm on our collected database. The first step to detect a human face against a picture and other objects is to perform the skin color segmentation.

Figure 3. The first row is an example of the explained face tracking algorithm (indicated with a green box) comparing with other methods (extended mean shift algorithm (shown in pink ellipse) and kalman filter based algorithm (indicated with a blue box) under changes of pose and expressions and scale by implementing the explained method on the input frames 28, 155, 465 for different face distances and orientation, The second row, The same results in a different illumination condition for input frames 5, 136, 359. The third row. Results of applying the face tracking algorithm in which there is occlusion in the face and is captured under variable illumination for input frames 45, 139, 200.
Then using template matching and motion information, face detection is performed. In the next step the model of face is updated with respect to changes in head angle, illumination, facial expression and occlusion. Finally using the particle filter and alternatively face detection, robust and real time face tracking is performed. Experimental results are provided to demonstrate the efficiency of the proposed method.

In fact, this face detection and tracking method shows high performance, robustness to illumination variations and occlusion and geometric changes (such as viewpoint and scale changes) and at the same time entails a significantly reduced computational complexity. Also this method is comparing with other methods (face tracking method based on Kalman filter [13] and the extended mean shift based face tracking [14]) which is shown in figure 3. The results show that this algorithm is performing better comparing with other methods especially when there is occlusion in the tracked face and when there are vast changes in the head angle, illumination, facial expression. It is able to do so despite using a tighter target window around the tracked face. The extended mean-shift-based tracker performs poorly, experiencing significant drift off the traced face. It is because of the appearance model of the mean shift tracker which is based on histograms of pixel intensities and is not adapted over time. Therefore, this algorithm is not robust, specially when there are fast motions and also when partially occlusion occurs in the face region.

V. CONCLUSIONS

In this paper a method is proposed for robust and real time face detection and tracking against picture and other objects. If the purpose is face detection, a combinational skin model is produced then the skin regions are segmented, then using template matching and motion information, face detection is performed.

For face tracking the face’s model is updated with respect to changes in head angle, illumination, facial expression and occlusion. Then tracking is performed using the described particle filter. The proposed algorithm is shown to be efficient in tracking human face comparing with other methods specially when there is occlusion or when the tracked face is close to or on a object with the color near the skin color.

REFERENCES


