Video-Based Face Tracking and Recognition on Updating Twin GMMs

Li Jiangwei and Wang Yunhong

Intelligence Recognition and Image Processing Laboratory,
Beihang University, Beijing, P.R. China
{jwli,yhwang}@buaa.edu.cn

Abstract. Online learning is a very desirable capability for video-based algorithms. In this paper, we propose a novel framework to solve the problems of video-based face tracking and recognition by online updating twin GMMs. At first, considering differences between the tasks of face tracking and face recognition, the twin GMMs are initialized with different rules for tracking and recognition purposes, respectively. Then, given training sequences for learning, both of them are updated with some online incremental learning algorithm, so the tracking performance is improved and the class-specific GMMs are obtained. Lastly, Bayesian inference is incorporated into the recognition framework to accumulate the temporal information in video. Experiments have demonstrated that the algorithm can achieve better performance than some well-known methods.

Keywords: Face Tracking, Face Recognition, Online Updating, Bayesian Inference, GMM.

1 Introduction

Recently, more and more research interesting has been transferred from image-based face detection and recognition [1-2] to video-based face tracking and recognition [3-5]. Compared to image-based face technologies, multiple frames and temporal continuity contained in video facilitate face tracking and recognition. However, large variations of image resolution and pose, poor video quality and partial occlusion are the main problems video-based face technologies have encountered. To deal with these difficulties, many researchers have presented their solutions [3-5], which adopted various strategies to fully use temporal and spatial information in video.

The capability of online learning is very favorable for video-based algorithms. The models can be updated when new sample comes, so the memory is saved without preserving the sample, and the model turns to fit current and future patterns with the time elapsing. Among all video-based face technologies, only a few algorithms [3,4] introduced the updating mechanism into their framework.

In this paper, we propose a new framework for video-based face tracking and recognition based on online updating Gaussian Mixture Models (GMMs). At any instance, we use the new sample to update two GMMs, called as "twin GMMs". As

S.-W. Lee and S.Z. Li (Eds.): ICB 2007, LNCS 4642, pp. 848–857, 2007.

shown in Fig. 1, in the training stage, according to different requirements of face tracking and recognition, we design twin initial models, namely tracking model and recognition model. Then, use the tracking model to locate the face of the incoming frame. At each instance, the detected face is learned to update both twin models with different updating rules. By learning all frames in the sequence, the recognition model gradually evolves to the class-specific model, and the tracking model becomes more powerful by merging the learned samples into its framework. In the testing stage, given the testing video, the recognition score is calculated by accumulating the current likelihood and previous posteriors using Bayesian inference.

For most traditional methods, they train gallery models in batch modes and then use it to perform recognition. Contrastively, with online sequential updating, our learning mechanism saves memory loads and is more adaptive for real-time applications. Moreover, the recognition approach based on Bayesian inference effectively captures temporal information. Experimental results show that our algorithm can effectively track and recognize faces in video even with large variations.

Note the differences between our algorithm and some existing updating methods [3, 4]. In our paper, we emphasize the necessity of designing different models to deal with different tasks. Compared to [3, 4], we note the distinctions between face tracking and recognition, so twin models with various initialization and learning strategies for each task are proposed. It is fairly a delicate learning mechanism. Furthermore, with our learning mechanism, the evolved class-specific models for distinct individuals have different forms. These advantages make the algorithm flexible.

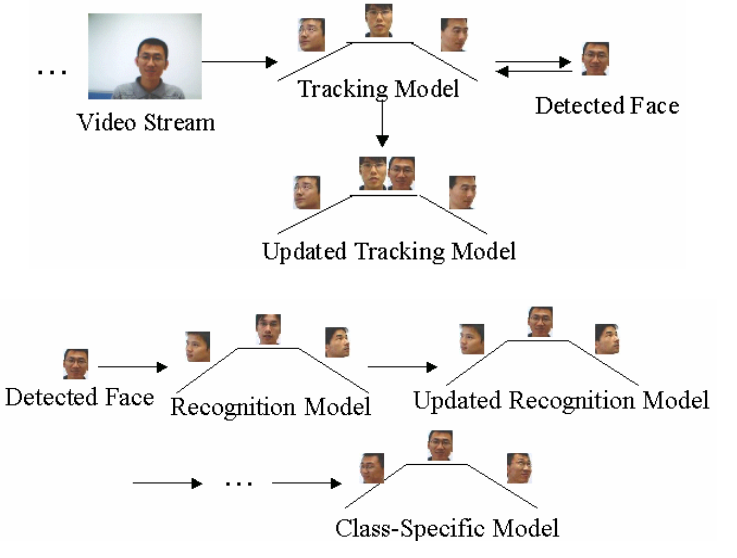


Fig. 1. Online updating twin models

2 The Framework of Updating

GMM is a special form of HMM and have been well studied for many years. The GMM assumes the probability that the observed data belongs to this model takes the following form:

$$G(\vec{x}) = p(\vec{x} \mid \lambda_l) = \sum_{m=1}^{l} \alpha_m N(\vec{x}, \mu_m, \theta_m) \tag{1}$$

where $N(\vec{x}, \mu_m, \theta_m)$ denotes the multi-dimensional normal distribution with the mean μ_m and the covariance matrix θ_m, and α_m is the weight of the corresponding component, satisfying:

$$\alpha_m \geq 0, m = 1, \cdots, l, \text{ and } \sum_{m=1}^{l} \alpha_m = 1 \tag{2}$$

In the following, we begin with the initialization of the twin GMMs. Then will show how to learn the incoming sample to update the twin models in the training stage.

2.1 Initialization

Motivated by the distinctions between face tracking and face recognition, it is necessary to initialize the face tracking model and the face recognition model in different ways. For face tracking, considering there exists numerous variations in face pattern, the initial tracking model must train on a large scale of data samples. For face recognition, to fasten the evolution from the initial model to a class-specific model, the recognition model can learn on much less samples. In addition, to ensure proper convergence, the recognition model is initialized with enough components to spread over the face space. The dimension of all training data is reduced to d by PCA to prevent "the curse of dimensionality".

The method begins with the initialization of the twin GMMs. Denote $G_T(\vec{x}) = p(\vec{x} \mid \lambda_{l_1})$ as the initial tracking model with l_1 components and $G_R(\vec{x}) = p(\vec{x} \mid \lambda_{l_2})$ the initial recognition model with l_2 components. Further assume there exists a training face set with p (p>5000) samples. The initialization proceeds as the following:

- ***Initialization of the tracking model***

For $G_T(\vec{x})$, considering the diversities of face patterns in feature space, the whole set with p samples is used to train the model using some unsupervised learning method [6], which is characterized as being capable of selecting number of components and not requiring careful initialization. In $G_T(\vec{x})$, each Gaussian component can be treated as a pose manifold. Since it is random for a face in video being a certain pose, we discard the learned weight coefficients for all components and fix them as $1/l_1$, namely all components in $G_T(\vec{x})$ are equally weighted. So the initial parameters of

$G_T(\vec{x})$ are $\{l_1, \frac{1}{l_1}, \mu_{(m,0)}, \theta_{(m,0)}\}$, where l_1 is the number of components, and $\frac{1}{l_1}$, $\mu_{(m,0)}$ and $\theta_{(m,0)}$ are initial weight, mean and covariance of each Gaussian components.

- ***Initialization of the recognition model:***

For $G_R(\vec{x})$, we randomly select l_2 points from the set as the mean vectors, and initialize the weight $\alpha_{(m,0)} = 1/l_2$. To weaken the influence of the training data on class-specific models, only $q(q << p)$ samples are selected from the set to compute the initial covariance matrix:

$$\theta_{(m,0)} = \frac{1}{10d} trace(\frac{1}{q}\sum_{i=1}^{q}(\vec{x}_i - m)(\vec{x}_i - m)^T)I \tag{3}$$

where $m = \frac{1}{q}\sum_{i=1}^{q}\vec{x}_i$ is the mean of the selected samples and I is the d-dimensional identity matrix. So the initial parameters of $G_R(\vec{x})$ are $\{l_2, \alpha_{(m,0)}, \mu_{(m,0)}, \theta_{(m,0)}\}$, where l_2 is the number of components, and $\alpha_{(m,0)}$, $\mu_{(m,0)}$ and $\theta_{(m,0)}$ are initial weight, mean and covariance of each Gaussian components.

Generally, as face variations of an individual are much less than variations of all individuals, the number of Gaussian components in the initial tracking model should be more than that in the initial recognition model, i.e., $l_1 > l_2$. This is beneficial for the fast evolution of $G_R(\vec{x})$ from initial recognition model to class-specific models as well.

2.2 Updating Process

In the training stage, with the initial tracking model $G_T(\vec{x})$, we can use the model to continuously track the face. Denote $\{I_0, \cdots, I_t, \cdots I_N\}_i$ the *i*th incoming video sequence. The updating process can be expressed as:

$$G_T(\vec{x}) \oplus \{I_0, \cdots I_t, \cdots, I_N\}_i \rightarrow G_T(\vec{x}),\ G_R(\vec{x}) \oplus \{I_0, \cdots I_t, \cdots, I_N\}_i \rightarrow G_i(\vec{x}) \tag{4}$$

where $\oplus$ is the operator of incremental updating, and $G_i(\vec{x})$ is the class-specific model for the *i*th sequence.

In video, the dynamics between frames can facilitate the tracking process. It is formulated as a Gaussian function:

$$\mathrm{K}(s_t, s_{t-1}) = \exp\{-(s_t - s_{t-1})C_{t-1}^{-1}(s_t - s_{t-1})\} \tag{5}$$

where s_t is the current state variable, including face position and pose information. From the current frame I_t, according to Eq.(5), draw y image patches around the previous position of the detected face as face candidates and normalize them into

d-dimensional vectors $\{F_{(1,t)},\cdots,F_{(r,t)},\cdots,F_{(y,t)}\}$. The face is detected with maximum likelihood rule:

$$F_t^* = \arg\max_i G_T(F_{(i,t)}) \tag{6}$$

After we obtain the face, we use it to update both models. Note that the twin models should be updated in different ways:

- ***Updating of the tracking model:***

Give the model $G_T(\vec{x})$ with the parameter $\{l,\frac{1}{l},\mu_{(m,t-1)},\theta_{(m,t-1)}\}$ at time $t-1$, where m is the *mth* Gaussian component. For the new sample F_t^*, we first find its ownership:

$$o_{(m,t)}(F_t^*) = N(F_t^*,\mu_{(m,t-1)},\theta_{(m,t-1)}) \ , \ \ m^* = \arg\max_m o_{(m,t)}(F_t^*) \tag{7}$$

In Eq.(7), the probability of F_t^* in the m^*th component is largest. All weights keep invariant, and only update the parameters of the m^* component with the rate λ_T:

$$\varsigma = F_t^* - \mu_{(m^*,t-1)}, \ \ \mu_{(m^*,t)} = \mu_{(m^*,t-1)} + \lambda_T o_{(m^*,t)}(F_t^*)\varsigma$$

$$\text{and } \theta_{(m^*,t)} = \theta_{(m^*,t-1)} + \lambda_T o_{(m^*,t)}(F_t^*)(\varsigma\varsigma^T - \theta_{(m^*,t-1)}) \tag{8}$$

To facilitate face tracking in current video and simultaneously avoid over-fitting, we keep the weights of all components invariant and only update the mean and the covariance of the component with highest ownership score. These will prevent the model converging to a class-specific model so as to still keep good tracking performance when the following video comes.

- ***Updating of the recognition model:***

For $G_R(\vec{x})$, we use some existing technique for updating. There are several incremental learning methods for GMM [7,8], while only [8] can update the model one by one. So the method in [8] is used.

Assume the parameter is $\{l_{t-1},\alpha_{(m,t-1)},\mu_{(m,t-1)},\theta_{(m,t-1)}\}$ at time $t-1$. As the new data F_t^* comes, for each component, we first calculate its ownership confidence score:

$$o_{(m,t)}(F_t^*) = \alpha_{(m,t-1)} N(F_t^*,\mu_{(m,t-1)},\theta_{(m,t-1)}) / G_R(\vec{x}) \tag{9}$$

Then use the score to update the corresponding weight:

$$\alpha_{(m,t)} = \alpha_{(m,t-1)} + \lambda_R(\frac{o_{(m,t)}(F_t^*)}{1-l_{t-1}C} - \alpha_{(m,t-1)}) - \lambda_R \frac{C}{1-l_{t-1}C} \tag{10}$$

In Eq.(10), λ_R determines the updating rate, and $C = \lambda N/2$ is a constant, where $N = d + d(d+1)/2$ is the number of parameters specifying each mixture component.

Check all $\alpha_{(m,t)}$. If $\alpha_{(m,t)} < 0$, it means too few data belong to the component m, so cancel this component, set $l_t = l_{t-1} - 1$ and renormalize $\alpha_{(m,t)}$. The remaining parameters are updated as:

$$\varsigma = F_t^* - \mu_{(m,t-1)},\ \mu_{(m,t)} = \mu_{(m,t-1)} + \lambda_R \frac{o_{(m,t)}(F_t^*)}{\alpha_{(m,t-1)}}\varsigma,$$

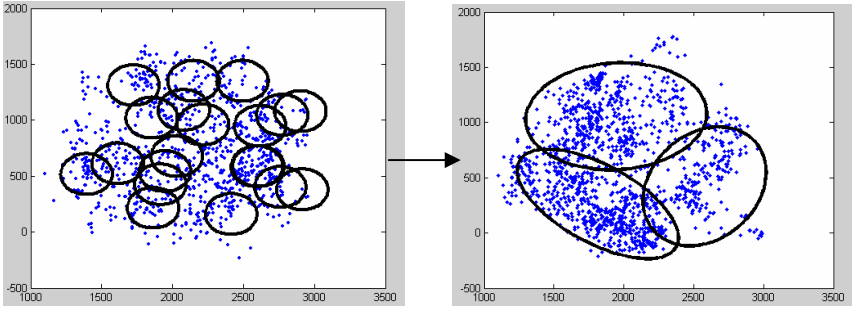
$$\text{and}\quad \theta_{(m,t)} = \theta_{(m,t-1)} + \lambda_R \frac{o_{(m,t)}(F_t^*)}{\alpha_{(m,t-1)}}(\varsigma\varsigma^T - \theta_{(m,t-1)}) \tag{11}$$

Then use the new parameter $\{l_t, \alpha_{(m,t)}, \mu_{(m,t)}, \theta_{(m,t)}\}$ for next updating.

2.3 Updating Results

Except for above updating rules, note the following additions: (1) For the face recognition model, to learn more intra-personal patterns and tolerate face location error, at any instance, the model is updated with more generated virtual samples by locating the face with errors and mirror operation. (2) Two models should be updated with different updating rate. Generally, the updating rate of the face recognition model is much faster than that of the face tracking model, i.e., $\lambda_T << \lambda_R$. Given limited samples, this will accelerate the evolution from the face recognition model to class-specific models.

With this framework, both the twin models are updated gradually. As in learning, the detection capability of the tracking model is enhanced. The parameters are adjusted to fit the current video data, and this will benefit the following tracking. The recognition model evolves to the class-specific model. With its own initial and updating rules, the recognition model can converge to proper face models. Fig.2 shows an example of evolution result given an image sequence. The left image is the initial recognition model and the right is the evolved class-specific model. The result looks encouraging for good fitness of video data. Compared to those existing methods [3,4], the component number in a class-specific model is not pre-fixed and totally determined by the learned video data. So our updating mechanism is flexible to model the real video image data distribution.



(a) Initial recognition model (b) Class-specific model

Fig. 2. An online updating example

3 Recognition Framework

Repeatedly applying the algorithm to J sequences $\{1,\cdots,i,\cdots,J\}$, we can obtain J class-specific GMMs $\{G_1(\vec{x}),\cdots,G_i(\vec{x}),\cdots,G_J(\vec{x})\}$. In the testing stage, we incorporate temporal continuity into the recognition framework. By assuming constant identity, using time recursion and Bayes rule, when the tracked face F_t^* comes, its identity i^* is determined with the evolution of posterior probability:

$$\begin{aligned} i^* &= \arg\max_i p(i \mid F_t^*, F_{0:t-1}^*) \\ &= \eta \arg\max_i p(F_t^* \mid i) \cdot p(i \mid F_{t-1}^*, F_{0:t-2}^*) \\ &= \eta \arg\max_i G_i(F_t^*) \cdot p(i \mid F_{t-1}^*, F_{0:t-2}^*) \end{aligned} \tag{11}$$

In the above framework, we do not use the priors of personal pose transitions to improve recognition performance as in [4]. We argue that these statistical priors are too reliant on the dynamics of head moving of the training video to be used as a reliable cue when recognize the testing video. Also, we can use the sample F_t^* to update both the tracking model and the i^* th class-specific model based on the confidence measure of the recognition result.

4 Experimental Results

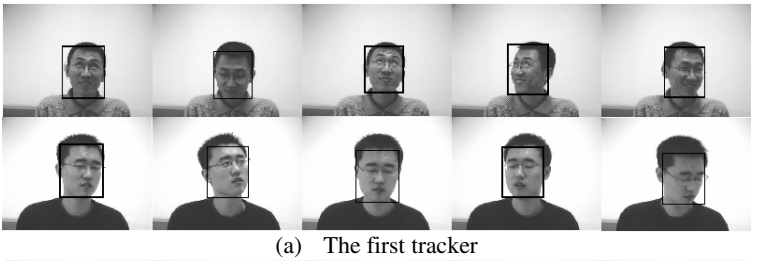
In this section, we will evaluate both tracking and recognition performance with the described algorithm on our collected database. The database is composed of 56 video sequences of 25 people. Each person has at least two sequences, one for gallery and one for probe. The remaining 6 sequences are used to train both the twin initial models. Some examples of the database are shown in Fig. 3.

When train initial twin models, we manually crop the face and generate the mirror image from the remaining 6 sequences, then normalize them into 23×28 pixels. In addition, we enlarge the training set by appending some standard face databases into it. In total, we have 6,500 face images in the set. The parameters for twin models are: $l_1 = 30$, $l_2 = 20$, $d = 18$, $\lambda_T = 0.0001$ and $\lambda_R = 0.001$. With initial models, updating algorithm and Bayesian inference, the faces can be effectively tracked and correctly identified.

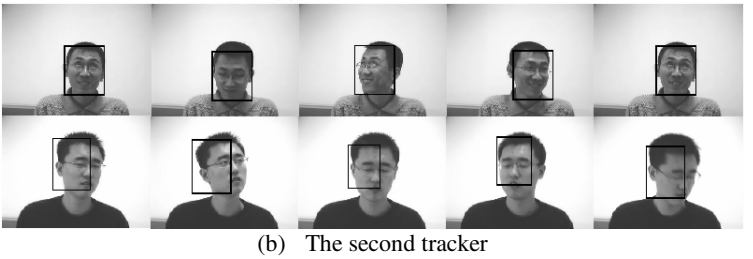
Fig. 4 shows some tracking results on two video sequences. Though the background in the video is rather clear, as faces can move freely in and out of plane, precisely locating all faces is difficult as well. Here two trackers are compared. The differences between them are: the first tracker updates itself with our rule, i.e., keeps the component weights invariant and only updates the mean and the covariance of the relative component which the current sample belongs to, while the second tracker updates itself with the rule in [8], i.e., all parameters of each component are updated

with the learning of samples. As shown in Fig.4, both of two trackers can properly detect the faces of the first video. When tracks the second sequence, only the first tracker still keeps good performance. This is because with the learning rule of the first tracker, it can fit the current video data by adjusting the parameters of the relative component, and simultaneously, it avoids over-fitting by keeping all component weights invariant and only adjusting the relative component. Compared to the first tracker, after learning one sequence, the second tracker tunes its parameters to completely cater for the learned sequence, so fails to be robust to the next sequence. This experiment indicates the importance of designing different rules for the initialization and updating of the tracking model.

Fig. 3. Some examples in the database



(a) The first tracker



(b) The second tracker

Fig. 4. Some tracking results on the database

The next experiment is to verify the recognition performance of our algorithm. The testing database contains 7852 frames from 25 sequences. The faces are detected using our proposed tracker. Five frame-based algorithms are compared: our algorithm BGMM (class-specific GMMs with Bayesian inference), GMM (class-specific GMMs without Bayesian inference), PCA, LDA and NN (Nearest neighbor). We take the optimal number of eigenvectors for PCA and LDA, namely 643 and 24, respectively. The experimental results are shown in Fig. 5, where the recognition rates for all methods are listed on the top. From the experimental results, we can note that:

(1) Compared to GMM, PCA, LDA and NN, our algorithm achieves best recognition rate of 94.0%. It indicates the success of our algorithm. The initial recognition model can converge to proper class-specific model after the training sequence is learned. Furthermore, the application of Bayesian inference is important as well. It can accumulate historical recognition information.
(2) The recognition rate of GMM is slightly lower than that of PCA and LDA. It is mainly due to more parameters need to be estimated in GMM. However, on one hand, GMM can fit any complex video data distribution so have great potential to improve performance. On the other hand, the use of class-specific model based on GMM is reasonable as its component number is completely determined by the training data distribution. So GMM is used instead of subspace [4] for updating.

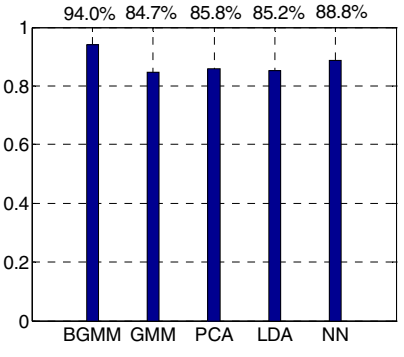


Fig. 5. Recognition results on the database

5 Conclusion

This paper has presented a method for both video-based face tracking and recognition based on GMM updating. We have noted the distinctions between the tasks of face tracking and face recognition, so designed two initial models and update them with different rules, respectively. By learning video samples online, the tracking model will be gradually enhanced and the class-specific models are obtained.

Note the samples in recognition can be used for updating as well. In the future, we will focus on this issue. With the current learning mechanism, the component number in GMM will decrease with time elapse. However, if necessary, some of them may have to be split to better fit the data. Another future issue is to develop the corresponding solution for this problem.

Acknowledgments. This work was supported by Program of New Century Excellent Talents in University, National Natural Science Foundation of China (No. 60575003, 60332010), Joint Project supported by National Science Foundation of China and Royal Society of UK (60710059), and Hi-Tech Research and Development Program of China (2006AA01Z133).

References

[1] Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.J.: Face Recognition: A Literature Survey. Technical Reports of Computer Vision Laboratory of University of Maryland (2000)
[2] Li, S., Jain, A. (eds.): Handbook of Face Recognition. Springer, Heidelberg (2004)
[3] Liu, X., Chen, T.: Video-Based Face Recognition Using Adaptive Hidden Markov Models. In: Proceedings of Computer Vision and Pattern Recognition (2003)
[4] Lee, K.C., Kriegman, D.: Online Learning of Probabilistic Appearance Manifolds for Video-based Recognition and Tracking. In: Proceedings of Computer Vision and Pattern Recognition (2005)
[5] Aggarwal, G., Chowdhury, A., Chellappa, R.: A System Identification Approach for Video-based Face Recognition. In: Proceedings of International Conference on Pattern Recognition (2004)
[6] Figueiredo, M., Jain, A.K.: Unsupervised Learning of Finite Mixture Models. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(3) (2002)
[7] Wu, J., Hua, X., Zhang, H., Zhang, B.: An Online-Optimized Incremental Learning Framework for Video Semantic Classification. ACM Multimedia (2004)
[8] Zivkovic, Z., Heijden, F.: Recursive Unsupervised learning of Finite Mixture Models. IEEE Transaction on Pattern Analysis and Machine Intelligence 26(5) (2004)