Robust Face Tracking via Collaboration of Generic and Specific Models

Peng Wang, *Member, IEEE*, and Qiang Ji, *Senior Member, IEEE*

***Abstract*—Significant appearance changes of objects under different orientations could cause loss of tracking, "drifting." In this paper, we present a collaborative tracking framework to robustly track faces under large pose and expression changes and to learn their appearance models online. The collaborative tracking framework probabilistically combines measurements from an offline-trained generic face model with measurements from online-learned specific face appearance models in a dnamic Bayesian nework. In this framework, generic face models provide the knowledge of the whole face class, while specific face models provide information on individual faces being tracked. Their combination, therefore, provides robust measurements for multiview face tracking. We introduce a mixture of probabilistic principal component analysis (MPPCA) model to represent the appearance of a specific face under multiple views, and we also present an online EM algorithm to incrementally update the MPPCA model using tracking results. Experimental results demonstrate that the collaborative tracking and online learning methods can handle large pose changes and are robust to distractions from the background.**

***Index Terms*—Collaborative tracking, generic face model, mixture of probabilistic principal component analysis (MPPCA), multiview face tracking, online learning.**

I. Introduction

MULTIVIEW face tracking aims to continuously detect faces that could undergo pose changes, based on their temporal coherence in videos. In real-world environments, the face undergoes view and expression changes, in additional to position and scale changes. Thus, a problem with multiview face tracking is that an imperfect measurement model could fail the tracking. Under a state-space model, a robust multiview face model is desirable to handle face appearance variations under different poses. However, such a model generally is not available prior to tracking, because there is no prior information about the specific face to be tracked. Without a robust multiview face model, the background distraction will cause inaccurate measurements, and the error accumulated over frames will eventually lead to the loss of tracking, i.e., "drifting."

In this paper, a probabilistic framework is developed to robustly track multiview faces by combining multiple measurements, and to learn face appearance models online. Since two categories of face models—*specific* and *generic* face models—will be intensively used in the later discussion, they are explained before presenting details of the algorithm. A *specific face model* is an appearance model of the individual face being tracked. For example, in a template-matching tracking method, a face template is a simple specific model to represent an individual face. A robust multiview appearance model of a specific face is desirable for multiview face tracking to handle face appearance variations. To construct such a model offline before tracking needs human intervention, which is not realistic in some applications, so our work focuses on learning specific face appearance models online. Another type of face model, *generic face model*, represents the whole face class including many persons' faces, not just a single person's face. The generic face model can be trained offline with faces of different people, and non-faces, as well. Recently developed probabilistic face detectors, such as in [1] and [2], can be directly applied as a generic face model. We believe that such a generic face model contains rich information of the face class and can be helpful to specific multiview face tracking and learning.

This paper presents a "collaborative tracking" algorithm, which probabilistically combines measurements from an offline trained generic face model with measurements from specific face models for robust multiview face tracking. In the collaborative tracking, the offline-trained generic face model collaborates with a specific multiview face model, which will be updated online using tracking results. For the online learning of specific face model, a mixture of probabilistic principal component analysis (MPPCA) model is used to represent the specific multiview face appearance. Furthermore, an online EM learning method is developed to automatically update specific face models using probabilistic tracking results.

Compared to the previous work [3]–[6], the novelty and contribution of our presented methods are twofold. First, we present a probabilistic framework to combine two types of measurements that are acquired from an offline learned generic face model and from an on-line learned specific face model, whereas currently existing methods mainly combine measurements from the online observations. Since the generic model used in our method contains rich information of the whole face class, it can improve the tracking robustness by discriminating faces of different poses and eliminating distractions from background. Second, we present an online EM algorithm, which utilizes probabilistic tracking results, to adapt the face appearance model to the specific faces being tracked. The two aspects are naturally integrated in a principled probabilistic framework, i.e., the proposed "collaborative tracking."

The rest of this paper is organized as follows. Related work is reviewed in Section II. Section III presents our collaborative

Manuscript received February 5, 2007; revised March 4, 2008. This work was supported in part by the Rensselaer Polytechnic Institute from the Task Support Working Group (TSWG) of the United States under Grant N41756-03- C-4028. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zoltan Kato.

P. Wang is with Siemens Corporate Research, Princeton, NJ 08540 USA.

Q. Ji is with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180 USA.

Digital Object Identifier 10.1109/TIP.2008.924287

multiview face tracking framework. Section IV introduces an online EM algorithm for learning specific appearance models from tracking results. Experimental results are shown in Section V. Section VI concludes this paper.

II. Related Work

In this section, some work related to multiview face tracing is reviewed. Most visual tracking algorithms can be unified in a "state-space" model [7], [8], on which our method is solidly based. In this model, the visual tracking is to infer unknown states of objects from visual observations. There are three fundamental elements in the state-space model: the "state," the "measurement model," and the "inference strategy." The "state" in the state-space model describes the status of an object. The states can be position, scale, shape, kinetic motion or any other properties concerning an object [7]–[9]. The state undergoes dynamic change, which is described by a "transition model" (or named "system model"). Some tracking methods, such as Kalman filtering [10], generally assume linear transition of states with Gaussian noise, while sampling-based methods can handle nonlinear dynamic systems [11], [12]. In tracking, the measurement model relates unknown states to visible observations or features, such as image intensity, color histogram, object shape and subspace representation [7] and [13]–[15]. Given the measurement model and the system model, states are estimated from observations, based on some inference strategies. One of the strategies, temporal Bayesian estimation, provides a powerful formalization of inference with sequential data. Monte Carlo sampling methods realize the Bayesian estimation by sampling from a density function. An example of sampling based visual tracking methods is "particle filtering" [12] (called CONDENSATION in [8]).

A. *Robust Measurements in Visual Tracking*

Among all the elements in the "state-space" model, the measurement model has the largest impact on tracking performance as it directly relates states to observation. It is also a key to solve the "drifting" problem. Much work has already been done to obtain measurement models that can handle object appearance changes, and are robust to background distraction. Roughly speaking, current efforts fall into three directions: building measurement models from exemplars, fusing multiple measurements, and online updating appearance models.

A robust appearance model can be constructed from offline-collected exemplars. Toyama *et al.* present an exemplar-based probabilistic tracking algorithm in [16]. The measurement model of an object is represented by a parametric mixture model whose centers and weights are learned from exemplars. In the Eigentracking by Black and Jepson [14], objects are represented by their projections in the subspace, which is learned offline from training samples. Under the assumption of "subspace consistency," an object is tracked by minimizing matching errors in eigenspace. Although such methods can handle small appearance changes, the multiview face appearance usually has a large variation, which are difficult to be effectively represented in a single PCA subspace or parametric models.

The fusion of multiple measurements is claimed to be more robust to "drifting" than a single measurement [3], [5], [17]. Leichter *et al.* combines multiple tracking algorithms with a probabilistic framework [3], in which tracking algorithms using different features and the same state space are directly combined together by a probabilistic factorization. In Wu's work [17], color and shape cues act as two modalities in a tracking algorithm, and provide priors for each other in sequential importance sampling (SIS). Such "co-inference" reduces clutters caused by each modality; therefore, it improves tracking performance. The above methods only use measurements from specific object models, and have not utilized offline trained generic models.

Some other methods, such as [4], [6],[13], [15], and [18], update measurement models during tracking so that tracking algorithms quickly adapt to both object and environmental changes. To overcome the limitations of static templates, a template updating strategy with drift correction has been proposed in [13]. This method updates a template only when current tracking result is consistent with an initial template. Jepson *et al.* use a mixture of Gaussian to model object appearance, and update model parameters online [15]. The mixture model has three components for long-term stable appearance, 2-frame change and outliers, respectively. Some methods have been developed to update the appearance subspace online in order to represent tracking objects [4], [6], [18]. In [4] and [18], algorithms are presented to incrementally update subspace online. The method used by Lee *et al.* can incrementally update both the bases and means of subspace [6]. All of the above methods update the specific models without using generic models; however, online learned object models are usually susceptible to tracking errors, and could cause the tracker to drift over time.

B. *Multiview Face Modeling in Tracking*

A characteristic of multiview face tracking is that the face undergos view changes as well as position and scale changes. Since it is difficult to handle all the views with a single face model, multiview face tracking usually needs multiview face models and the corresponding pose state in the "state-space" model. Sherrah *et al.* track the 3-D pose and position simultaneously by modeling the combination of the pose and position states using the CONDENSATION algorithm [19]. The method does not consider the scale change due to the limitation of state dimension. To simplify the state-space model, a switch mechanism has been applied to track multiview faces with multiple view templates [20]. In this method, the switching algorithm selects the best template out of all view templates from current tracking results for tracking at next frame. However, this method needs to build multiview face templates for each view before tracking, which is not realistic in real world applications.

There are also efforts to utilize generic face detector in face tracking. A simple way is to use face detection results as initialization for tracking [21]. More elegantly, the output of face detector can be integrated as one of measurements in tracking [5], [22]. In [22], probabilistic outputs from two face detectors, one for frontal faces and another one for profile faces, are used as measurements in particle filtering. Their method only uses generic face model. Wang *et al.* factorize the posterior probability of face tracking into two parts corresponding to generic and specific face tracking, respectively, and further assume that generic face tracking and specific face tracking are independent, so that the two posterior probabilities can be multiplied for multiview face tracking [5]. However, the independence assumption is not always valid. In their method, a specific face model is used for all the views and is simply updated using the average of previous tracking results.

Lee *et al.* present an online learning algorithm to update appearance manifolds for face tracking and recognition [6]. In their method, each pose of a face is assumed to be a single Gaussian distribution in a subspace. To track multiview face, each specific face begins with a generic face subspace, which is learned offline from training samples. During tracking, the means and bases of the pose subspace are incrementally updated. They synthesize faces from the tracked faces and offline collected sample faces to update specific face models. However, their method only uses some offline collected training samples instead of an offline trained generic face model, and, moreover, the generic face model learned offline ceases to exist during online learning.

In this paper, we present a novel solution, which is significantly different from [6] in the following aspects: it presents a probabilistic framework to combine measurements from a generic face model and measurements from a specific face model; the generic model directly utilizes existing probabilistic face detectors, without making any parametric assumption as in [6] the generic face model continuously works as a modality during tracking; the specific face models of multiple views are simultaneously updated with the use of probabilistic tracking results.

III. Collaborative Tracking

This section presents a collaborative tracking algorithm. The key of collaborative tracking is the combination of two types of measurement models: specific and generic face models. For this purpose, a probabilistic formalization based on dynamic Bayesian networks (DBN) is introduced for multiview face tracking. By factorizing the measurement model in the DBN, we incorporate both generic and specific face models in the tracking. We will introduce the generic face model used in the tracking, and leave the discussion of online learning specific face models at Section IV.

Notations are explained before presenting algorithmic details. The observed data is denoted as Z, and Z_t is the observation at the time-step t while $Z_{1:t}$ is the observation history from the beginning to the time-step t. X is the unknown state. Since a specific model is built for each person's face, we use X^i to represent the state of the ith person's face.

A. Probabilistic Framework for Multiview Face Tracking

The state-space model is a commonly used formalization for probabilistic visual tracking algorithms. The model can be concisely represented by a DBN, which naturally handles probabilistic inferences using sequential data [23]. In DBN, two Markov property assumptions are made for tracking, i.e., $P(X_t|X_{1:t-1}) = P(X_t|X_{t-1})$ and $P(Z_t|X_{1:t}) = P(Z_t|X_t)$. With these assumptions, the posterior probabilities of unknown states can be estimated from a measurement model $P(Z_t|X_t)$ and a propagated prior $P(X_t|Z_{1:t-1})$ based on Bayes' rule. Thus, the scheme to track a specific face with label i is shown in (1)

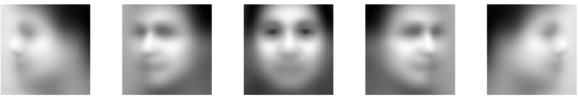


Fig. 1. Mean faces of different poses. From left to right is the pose of full left, half left, frontal, half right, and full right.

$$\begin{aligned} &P\left(X_t^i|Z_{1:t}\right)\\ &\quad= \frac{1}{C} P\left(Z_t|X_t^i\right) P\left(X_t^i|Z_{1:t-1}\right)\\ &\quad\propto P\left(Z_t|X_t^i\right) \sum_{X_{t-1}^i} P\left(X_t^i|X_{t-1}^i\right)\\ &\qquad\times P\left(X_{t-1}^i|Z_{1:t-1}\right) \end{aligned} \tag{1}$$

where C is the normalization constant. $P(X_t^i|X_{t-1}^i)$ is the state transition model.

The states in tracking usually include position, size, velocity and any other properties concerning faces. For multiview face tracking, the face pose is a very important factor. In this paper, a face pose is defined as a set of out-of-plane rotation angles, as shown in Fig. 1. Since faces under different poses show significantly different appearances, multimodal face appearance models are desirable to effectively handle appearance changes caused by face poses. For this purpose, a pose state β is introduced in the multiview face tracking. The states X^i used for specific multiview face tracking are factorized into two subsets, i.e., $X^i = \{\alpha^i, \beta^i\}$. The state set α^i represents position, size and/or velocity of ith face. β^i is the pose state, which takes a discrete value. $\beta^i = m$ indicates the existence of the pose m ($m = 1, \ldots, 5$ in our method).

A basic assumption of our method is that the two sets of states are independent to each other, i.e., $P(\beta^i|\alpha^i) = P(\beta^i)$. This assumption is reasonable since different poses can appear at any position and size. The total state transition is the product of two transitions, i.e., $P(X_t^i|X_{t-1}^i) = P(\alpha_t^i|\alpha_{t-1}^i)P(\beta_t^i|\beta_{t-1}^i)$, so the dimensionality of the transition matrix is reduced. The graphical model for the multiview tracking is shown in Fig. 2, and the tracking scheme for this model is simplified as (2)

$$\begin{aligned} &P\left(\alpha_t^i, \beta_t^i|Z_{1:t}\right)\\ &\quad\propto P\left(Z_t|\alpha_t^i, \beta_t^i\right)\\ &\qquad\times \sum_{\alpha_{t-1}^i, \beta_{t-1}^i} P\left(\alpha_t^i|\alpha_{t-1}^i\right) P\left(\beta_t^i|\beta_{t-1}^i\right)\\ &\qquad\times P\left(\alpha_{t-1}^i, \beta_{t-1}^i|Z_{1:t-1}\right). \end{aligned} \tag{2}$$

In the following paragraphs, the superscript i in (2) is dropped for simplicity. The superscript will be used only when it is necessary to discriminate different face trackers in tracking multiple faces.

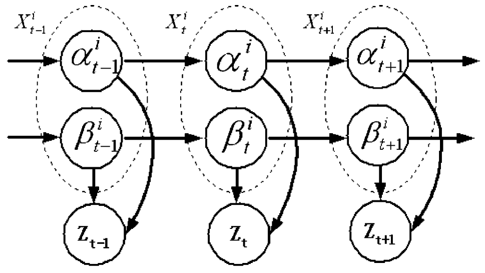


Fig. 2. Dynamic Bayesian network for multiview face tracking with decoupled states α and β.

B. Collaborative Tracking

One difficulty of directly applying (2) to multiview face tracking is that it requires the measurement model $P(Z_t|\alpha^i_t, \beta^i_t)$ for a specific person. Such a model is rarely available before tracking because usually there is no prior information about an unknown face under different poses. Some methods use the face detected at the first frame as the face model in tracking [21], but cannot handle later face appearance variations due to pose changes. Another solution is to update the face model $P(Z_t|\alpha_t, \beta_t)$ online using tracking results [6]. However, it remains a difficult task to select an appropriate pose model for updating without the help of generic face models, because initial specific face models cannot discriminate well among different poses. Also, the initial face models are usually rough, so it is very likely that the tracking can fail before all the models are learned.

Our solution to this problem is the presented "collaborative tracking" framework, which utilizes both a generic face model that is trained offline with collected face samples, and a specific appearance model that is updated online for the face being tracked. The key of the collaborative tracking is the factorization of the measurement model $P(Z_t|\alpha_t, \beta_t)$, as in (3), under the previous assumption $P(\beta_t|\alpha_t) = P(\beta_t)$

$$P(Z_t|\alpha_t, \beta_t) = \frac{P(\beta_t|Z_t, \alpha_t)P(Z_t|\alpha_t)}{P(\beta_t)} \tag{3}$$

where $P(\beta_t|Z_t, \alpha_t)$ is the posterior probability of pose β_t given observation Z_t and state α_t. In our method, the posterior is obtained from a probabilistic multiview face classifier, i.e., a "generic" face model; the input to the generic model is the observation Z_t and the state set α_t. Generally, the generic face model normalizes the current observation into an image patch using given position and size α_t. The output of the generic face model is the probability of the input image patch being a face at a pose β_t (see Section III-C for more details).

Another component in (3) is $P(Z_t|\alpha_t)$, which is a specific face model. In fact, $P(Z_t|\alpha_t)$ and $P(Z_t|\alpha_t, \beta_t)$ have an inherent relationship, as indicated in (4), when assuming $P(\beta_t|\alpha_t) = P(\beta_t)$

$$P(Z_t|\alpha_t) = \sum_{\beta_t} P(Z_t|\alpha_t, \beta_t)P(\beta_t) \tag{4}$$

Both $P(Z_t|\alpha_t)$ and $P(Z_t|\alpha_t, \beta_t)$ are called specific appearance models because they aim to model a specific person's face. For clarity, we call $P(Z_t|\alpha_t, \beta_t)$ a *specific single-view face model* for a pose β_t, and call $P(Z_t|\alpha_t)$ a *specific multiview face model* because $P(Z_t|\alpha_t)$ integrates all the view models together; thus, it does not discriminate different poses.

Without online learning, (3) and (4) are mathematically equivalent; therefore, neither can solve the difficulty of multiview face tracking. To address this problem, specific face models are incrementally updated during tracking (see Section IV for detail). The specific face models $P(Z_t|\alpha_t)$ in (3) and $P(Z_t|\alpha_t, \beta_t)$ in (4) are the models learned from previous frames, so the collaborative tracking can be rewritten as

$$P^-(Z_t|\alpha_t) = \sum_{\beta_t} P^-(Z_t|\alpha_t, \beta_t)P(\beta_t) \tag{5}$$

$$P(Z_t|\alpha_t, \beta_t) = \frac{P(\beta_t|Z_t, \alpha_t)P^-(Z_t|\alpha_t)}{P(\beta_t)} \tag{6}$$

where $P^-(Z_t|\alpha_t)$ and $P^-(Z_t|\alpha_t, \beta_t)$ denote the specific face models that are learned before frame t.

In summary, the collaborative multiview face tracking consists of two steps. In the first step shown in (5), a specific multiview face model $P^-(Z_t|\alpha_t)$ is constructed from specific single-view face models $P^-(Z_t|\alpha_t, \beta_t)$ that are learned from previous frames. In the second step, the constructed specific multiview face model $P^-(Z_t|\alpha_t)$ is combined with the offline learned generic model $P(\beta_t|Z_t, \alpha_t)$, as (6), to produce the likelihood measurement $P(Z_t|\alpha_t, \beta_t)$, which is then used in (2) to compute $P(\alpha_t, \beta_t|Z_{1:t})$. Compared to directly updating $P(Z_t|\alpha_t, \beta_t)$ without applying generic face models, the benefit of the two-step tracking algorithm is that we can depend on the generic face model $P(\beta_t|Z_t, \alpha_t)$ for tracking before the specific face models $P(Z_t|\alpha_t, \beta_t)$ are learned online. Unlike the existing methods that tend to update all the face models online, we believe it is important to keep the generic model intact so that it will not be "polluted" by the errors in the image data used for online updating. Hence, even after the specific face models are learned online, the generic face models can still be applied in its original form in collaborative tracking to improve the tracking robustness.

In a DBN model, the prior $P(\beta_t)$ at time t can be assumed to be the prior probability obtained from previous time steps, i.e., $P(\beta_t = l) = P(\beta_t = l|Z_{1:t-1})$, so (5) can be rewritten as

$$P^-(Z_t|\alpha_t) = \sum_{l} P^-(Z_t|\alpha_t, \beta_t = l)P(\beta_t = l|Z_{1:t-1}) \tag{7}$$

where $P(\beta_t|Z_{1:t-1}) = \sum_{\alpha_t} P(\alpha_t, \beta_t|Z_{1:t-1})$. Following the similar principle, $P(\beta_t)$ in (6) is also replaced by $P(\beta_t|Z_{1:t-1})$. The superscript "-" in (5)–(7) will be dropped in the following paragraphs for simplicity, but please note that the specific multiview face model used for current tracking is constructed from previously learned specific single-view face models.

The collaborative tracking scheme significantly differs from pervious work on measurement fusion. It not only uses image observations on a specific face, but also utilizes the prior knowledge of the whole face class, which is contained in the offline-trained generic face model. The offline-trained generic face model can handle face appearance variations, and eliminate the single Gaussian assumption made in [6]. Also,

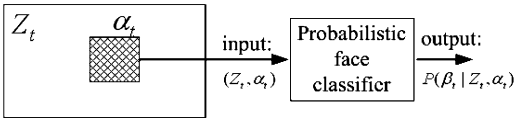

Z_t
α_t
input:
(Z_t, α_t)
Probabilistic face classifier
output:
$P(\beta_t \mid Z_t, \alpha_t)$

Fig. 3. Generic face model for tracking.

the training samples for the generic face model include both faces and non-face data, so it can distinguish the background and faces, thus reducing the distraction from the background. Furthermore, the training samples of the generic model include faces of different poses, so it is able to discriminate different poses. None of the above benefits can be achieved by currently existing tracking methods. The collaborative tracking algorithm presented in this paper also differs from the work in [5] in that this method directly factorizes the measurement model instead of the posterior probability; thus, it can be directly incorporated in a sampling-based tracking algorithm.

C. Generic Face Model for Tracking

In collaborative tracking, the generic multiview face model is trained offline, and outputs probabilistic scores for each pose given the image observation Z_t and the state α_t, as shown in Fig. 3. Our method uses a probabilistic face classifier as a generic face model. To achieve good accuracy and reasonable speed, the real AdaBoost algorithm [24] is used to train the generic multiview face model [2]. More details on the AdaBoost algorithms can be found in [25] and [24], and the AdaBoost based face classifiers are discussed in other sources [1], [2], [26]. To train a generic multiview face model, exemplar faces of various poses are collected, and non-faces are collected from thousands of background images. The training details are explained in Section V. Please note that the accuracy requirement of the generic face model for tracking is not as high as in the face detection. Based on our experiments, a misclassification error rate of 10% is acceptable for the generic face models used in tracking.

In real AdaBoost, the final classifier is a combination of many weak classifiers. Assuming Z^α is the observation Z associated with the state α, the classification result of AdaBoost is $h_\beta(Z^\alpha) = \sum_k h_{\beta,k}(Z^\alpha)$, where $h_{\beta,k}$ is the kth weak classifier for pose β. It has been shown in [24], that the posterior probability can be approximated by (8)

$$P(\beta|Z,\alpha) = \frac{e^{h_\beta(Z^\alpha)}}{e^{-h_\beta(Z^\alpha)} + e^{h_\beta(Z^\alpha)}}. \tag{8}$$

In our method, all multiview faces are roughly divided into five poses. They are full left, half left, frontal, half right, and full right, as shown in Fig. 1. Five independent face classifiers are trained for the five poses. The probabilistic scores from these pose models are then normalized to provide a posterior probability of face pose, for the used in the collaborative tracking.

D. Algorithm Summary

Here, we summarize the collaborative tracking algorithm to help readers understand the whole algorithm. The flowchart of our method is illustrated in Fig. 4. A face tracker is initialized when a face is detected from images. At the beginning of tracking, each specific single-view face model $P(Z_t|\alpha_t, \beta_t)$ for a pose β_t begins with an initial model, from which a specific multiview model $P(Z_t|\alpha_t)$ is built, as (7). Then the specific face model collaborates with a generic face model for multiview face tracking, as (6) and (2). From the posterior probability obtained during collaborative tracking, face location and pose are estimated, as explained in Section V-B. During tracking, $P(Z_t|\alpha_t, \beta_t)$ is updated using probabilistic tracking results via the online learning algorithm, which is described in Section IV.

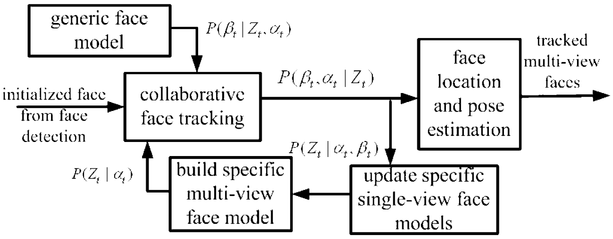

generic face model
$P(\beta_t \mid Z_t, \alpha_t)$
initialized face from face detection
collaborative face tracking
$P(\beta_t, \alpha_t \mid Z_t)$
face location and pose estimation
tracked multi-view faces
$P(Z_t \mid \alpha_t, \beta_t)$
$P(Z_t \mid \alpha_t)$
build specific multi-view face model
update specific single-view face models

Fig. 4. Illustration of the proposed multiview face tracking, online face appearance model learning, and pose estimation.

IV. Online Specific Face Model Learning

In this section, an online learning algorithm is presented to update specific face models during the collaborative tracking. Our method first applies the probabilistic principal component analysis (PPCA) [27] to model the face appearance at each single view. Thus, according to (7), a specific multiview face model can be represented as a mixture of PPCA models, with each specific single-view face model as a component in the mixture model. The advantage of using the mixture of Gaussian model is that the well-studied online EM learning algorithms can be directly applied to update model parameters [28]. In our method, only specific models are updated, and the generic model continuously contributes to face tracking.

A. Subspace Feature and Mixture of PPCA Model

In our method, the image intensity is used to represent the face appearance. However, it is difficult to directly use the image intensity in a Gaussian model due to its high dimensionality. For example, for a face normalized to the size of 20 × 20, the dimension of the intensity feature vector will be 400, which means that the covariance matrix has 400 × 400 elements. Even considering the symmetry of the covariance matrix, it is still a challenging task to learn so many parameters using a limited amount of data. An assumption to simplify the high-dimensional model is that all the pixels in a face image are independent, and they have identical variance, so the Gaussian model is simplified as $N(Z;\mu,\sigma) = (1)/(\sqrt{2\pi}\sigma)\exp\{(-\|Z - \mu\|^2)/(2\sigma^2)\}$. This assumption dramatically reduces the number of parameters to be estimated, but also causes loss of information for face modeling.

The probabilistic principal component analysis (PPCA) has been proposed to provide a flexible and probabilistic formalization for the feature dimension reduction without making extreme assumptions [27]. In PPCA, a d-dimensional vector Z is assumed to be related with a q-dimensional latent variable s, based on the linear model as (9)

$$Z = Ws + \mu + \epsilon. \tag{9}$$

Usually, $q \ll d$, and W is a d by q matrix where μ is the mean, and ϵ represents a random noise. Conventionally, it is assumed that $s \sim N(0, I)$, and $\epsilon \sim N(0, \Psi)$, where Ψ is a diagonal matrix, and I is the identity matrix. The factor analysis in (9) is found to be related with PCA when assuming $\Psi = \sigma^2 I$. Under such an assumption, the columns of W are the principal eigenvectors of the sample covariance matrix in PCA with an arbitrary rotation [27]. The probabilistic distribution in PPCA is derived as follows:

$$P(Z|s) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left\{-\frac{1}{2\sigma^2}\|Z - Ws - \mu\|^2\right\}. \tag{10}$$

With a Gaussian prior over the latent variable s, i.e., $P(s) = (1)/((2\pi)^{q/2}) \exp\{-(1)/(2)s^T s\}$, we obtain the distribution of Z in the following form:

$$\begin{aligned} P(Z) &= \int P(Z|s)P(s)ds \\ &= \frac{1}{(2\pi)^{d/2}|C|^{\frac{1}{2}}} \\ &\quad \times \exp\left\{-\frac{1}{2}(Z - \mu)^T C^{-1}(Z - \mu)\right\} \end{aligned} \tag{11}$$

where $C = \sigma^2 I + WW^T$. Similarly, we can obtain $P(s|Z) = N(s; M^{-1}W^T(Z - \mu), \sigma^2 M^{-1})$ where $M = W^T W + \sigma^2 I$. Since W is only a d by q matrix, and $q \ll d$, the number of parameters to be estimated is largely reduced. Such parameters of a PPCA model can be estimated using a batch EM method [27].

We assume that each view can be represented with a PPCA model, i.e., $P(Z_t|\alpha_t, \beta_t = i) = P_i(Z_t^\alpha) = N(Z_t^\alpha; \mu_i, C_i)$, where Z_t^α is the observation Z_t associated with α_t. Thus, the specific multiview face model $P(Z_t|\alpha_t)$ is a mixture of PPCA (MPPCA), according to (7). $P(Z_t|\alpha_t)$ is denoted as $P(Z_t^\alpha)$ for convenience, as in (12)

$$P(Z_t^\alpha) = \sum_i \pi_i P_i\left(Z_t^\alpha\right) = \sum_i \pi_i N\left(Z_t^\alpha; \mu_i, C_i\right) \tag{12}$$

where π_i is the weight of ith component $P_i(Z_t^\alpha)$.

An EM algorithm has been proposed to learn MPPCA parameters in a batch mode [29]. The EM algorithm updates weights, means, and sample covariance matrices of MPPCA in a way similar with those of mixture of Gaussian models. One difference between traditional mixture of Gaussian models and MPPCA is the way in which the covariance matrix C_i is updated. In the traditional mixture of Gaussian model, the observation covariance matrix C_i is the same as the data sample covariance matrix $S_i = E\{(Z - \mu_i)^T(Z - \mu_i)\}$, while in MPPCA, due to linear latent variable model, C_i has a constrained form as $C_i = (\sigma_i^2 I + W_i W_i^T)$. Using MPPCA to model face appearance can reduce feature dimension; therefore, the specific face models can be efficiently learned within limited video length. The PPCA model also provides a probabilistic framework to interpret traditional PCA such that it can be used as one of the measurements in the collaborative tracking.

B. Online EM Learning of Specific Face Models

Our method begins with an initial MPPCA face model, which is learned offline from collected face samples, and then update the face model online from tracking results. The online updating strategy aims at progressively updating the specific appearance models $P(Z_t|\alpha_t, \beta_t)$ using collaborative tracking results. Our method is based on the online EM algorithm, which has a concise form to incrementally update parameters of a mixture of Gaussian models [28]. The online EM algorithm for mixture of Gaussian distributions can be easily generalized to update parameters in MPPCA with some minor modifications.

As in the standard EM algorithm, our online EM algorithm for MPPCA also includes two steps, i.e., E-step and M-step. In the E-step of the EM algorithm, the expectation is calculated for each component in the mixture model. By comparing (12) with (7), it shows that $\pi_i^t = P(\beta_{t-1} = i|Z_{1:t-1})$. Our method applies the posterior probability $P(\beta_{t-1} = i|Z_{1:t-1})$ from collaborative tracking as the initial weights π_i^t in the mixture model, and then uses the tracked faces and associated posterior probabilities to update model parameters. The benefit of using probabilistic tracking results is that we do not deterministically specify only a single-view face model to update, but provide probabilities of each single-view face model; hence, all the specific single-view face models can be automatically and simultaneously updated with tracked faces. In the M-step of the online EM algorithm, the weights π_i, means μ_i and sample covariance matrices S_i are incrementally updated. In MPPCA, an additional step is needed to update C_i based on the updated sample covariance matrix S_i. Thus, the online algorithm for MPPCA includes the standard online EM algorithm, followed by an additional step updating W_i, M_i, σ_i, and C_i of MPPCA, as in (15).

The online EM algorithm is summarized as follows.

E-Step

$$\begin{aligned} \pi_i^t &= P(\beta_{t-1} = i|Z_{1:t-1}) \\ \rho_i^{t+1} &= \frac{\pi_i^t N\left(\hat{Z}_t; \mu_i^t, C_i^t\right)}{\sum_i \pi_i^t N\left(\hat{Z}_t; \mu_i^t, C_i^t\right)}. \end{aligned} \tag{13}$$

M-Step

$$\begin{aligned} \mu_i^{t+1} &= \mu_i^t + \frac{\rho_i^{t+1}}{t\pi_i^t}\left[\hat{Z}_t - \mu_i^t\right] \\ S_i^{t+1} &= S_i^t + \frac{\rho_i^{t+1}}{t\pi_i^t}\left[\left(\hat{Z}_t - \mu_i^t\right)^T\left(\hat{Z}_t - \mu_i^t\right) - S_i^t\right] \\ \pi_i^{t+1} &= \pi_i^t + \frac{1}{t}\left[\rho_i^{t+1} - \pi_i^t\right] \end{aligned} \tag{14}$$

and

$$\begin{aligned} W_i^{t+1} &= S_i^{t+1} W_i \left(\sigma_i^2 I + \left(M_i^t\right)^{-1}\left(W_i^t\right)^T S_i^{t+1} W_i^t\right)^{-1} \\ \left(\sigma_i^{t+1}\right)^2 &= \frac{1}{d} tr\left(S_i^{t+1} - S_i^{t+1} W_i^t \left(M_i^t\right)^{-1}\left(W_i^{t+1}\right)^T\right) \\ M_i^{t+1} &= \left(\sigma_i^{t+1}\right)^2 I + \left(W_i^{t+1}\right)^T W_i^{t+1} \end{aligned} \tag{15}$$

where $\hat{Z}_t$ is the image intensity of tracked face, i.e., the image observation associated with estimated face location $\hat{\alpha}_t$. The details of estimating $\hat{\alpha}_t$ are presented in Section V-B. At each time step, the parameters of specific single-view face models $P(Z_t|\alpha_t, \beta_t)$ are updated using tracking result $\hat{Z}_t$, and the updated specific single-view models will be further used to construct the specific multiview face model $P(Z_t|\alpha_t)$ for collaborative tracking at the next frame.

An underlying difference between our online learning algorithm and traditional EM is the introduction of the pose tracking result $P(\beta_{t-1}|Z_{1:t-1})$. In the traditional online EM, the parameter learning depends on the percentage of input faces from different poses, and their orders in learning, while our method is actually a "semi-supervised" learning algorithm, since the generic face model, which provides pose information, is obtained from supervised learning. Such a generic model can guide the selection of appropriate pose model for learning, thus improving learning and tracking robustness.

V. Experiments

This section first explains implementation details of our method, and then present experiments to demonstrate the advantages of our collaborative tracking and online learning algorithms.

A. Implementation

Implementing the collaborative tracking and online learning algorithms involves three parts: training the generic face model, implementing the state-space model for tracking, and implementing online EM learning. The first step is to train a probabilistic multiview face classifier offline as a generic model in tracking. The generic face model is obtained based on the same principle that is used to train a probabilistic face detector (the details of training generic face models are explained in [2]). At a each face view, positive training samples include the faces at the corresponding view, and negative training samples include both background images and faces at other views; therefore, the trained generic face model can separate faces from the background, and can also discriminate different face poses. The probabilistic outputs of face detectors at multiple views are the normalized to provide measurements as a generic face model. Compared to face detection in static images, the accuracy requirements of the generic face model in tracking are modest, so it does not need as many layers in the cascade structure as in a face detector.

The Monte Carlo sampling method is implemented to realize the state-space model for tracking. For the ith face, the particles are denoted as $\{\alpha_t^i(k), \beta_t^i(k), w_t^i(k)\}$. $w_t^i(k)$ is the likelihood score associated with the state $\{\alpha_t^i(k), \beta_t^i(k)\}$. In the state-space model, the transition probabilities of α^i are assumed to be a Gaussian distribution $P(\alpha_t^i|\alpha_{t-1}^i) = N(\alpha_t^i; \alpha_{t-1}^i, \Sigma_0)$. The pose transition probability is represented with a transition matrix, i.e., $P(\beta_t^i = m|\beta_{t-1}^i = n) = P_{m,n}$. The parameters Σ_0 and $P_{m,n}$ are set empirically.

Each specific single-view face model $P(Z_t|\alpha_t, \beta_t)$ begins with an initial PPCA model, which is trained offline with collected faces at corresponding pose β_t in a batch mode [29]. The dimension of the latent variable is set as $q = 5$ for each PPCA. During tracking, the tracked face is cropped from a frame, preprocessed, and then used to incrementally update specific appearance models via online EM. To prevent the online learned model from being polluted by low-quality face images, such as occluded or turn-away faces, numeric outputs from the generic face model are used to preselect faces before using them for learning. Outputs from an AdaBoost-based face detector are large for faces of good quality, and small for occluded or out-of-view faces, as well as non-faces [2]. A detected face is used for online learning only when its output from the generic face model is above a preset threshold. Also, since the initial specific face models may not accurately model the appearance of a specific face, only the generic face model is used to detect faces at the first several frames to avoid tracking failures caused by inaccurate specific face model. After a certain number of faces (e.g., a typical number is set as 10 in this method) have been acquired to update the specific face models online, we then incorporate specific face models in the collaborative tracking. Furthermore, when more tracked faces are acquired for model learning, the learning rate $1/t$ in (14) tends to zero; therefore, the online model learning will slow down. In our implementation, we set the learning rate no smaller than a constant, e.g., 0.1, to allow for continuous modeling updating, and to incrementally correct potential learning errors during the entire tracking. To further tune the relative strengths of offline and online models, an additional measurement is needed to automatically validate the online-learned specific face models. Such work is, however, beyond the scope of this paper, but it will be our future research.

The most time-consuming part of our algorithm is the generic face model, since it performs face detection of multiple views. However, its computation is less expensive compared to face detection because of the modest accuracy requirements. The online EM learning is efficient except for the calculation of covariance matrix norms in the MPPCA model. To solve this problem, the covariance matrix norms are only updated once for every several frames. The speed of the whole algorithm is about seven frames per second using a Pentium IV 2.6 G machine.

B. Face Location and Pose Tracking

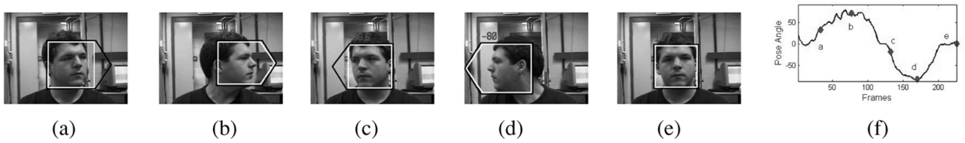
Face location and pose can be estimated from the posterior probabilities $P(\alpha_t, \beta_t|Z_{1:t})$. The face position and size $\hat{\alpha}_t$ can be estimated as $\hat{\alpha}_t = \sum_{\alpha_t} \alpha_t P(\alpha_t|Z_{1:t}) = \sum_{\alpha_t} \sum_{\beta_t} \alpha_t P(\alpha_t, \beta_t|Z_{1:t})$. We can also estimate face pose from $P(\alpha_t, \beta_t|Z_{1:t})$. The posterior probability of pose β_t, i.e., $P(\beta_t|Z_{1:t})$ is estimated as

$$P(\beta_t|Z_{1:t}) = \sum_{\alpha_t} P(\alpha_t, \beta_t|Z_{1:t}). \tag{16}$$

Then the pose with maximum posterior probability is selected as the current pose $\hat{\beta}_t$, as in (17)

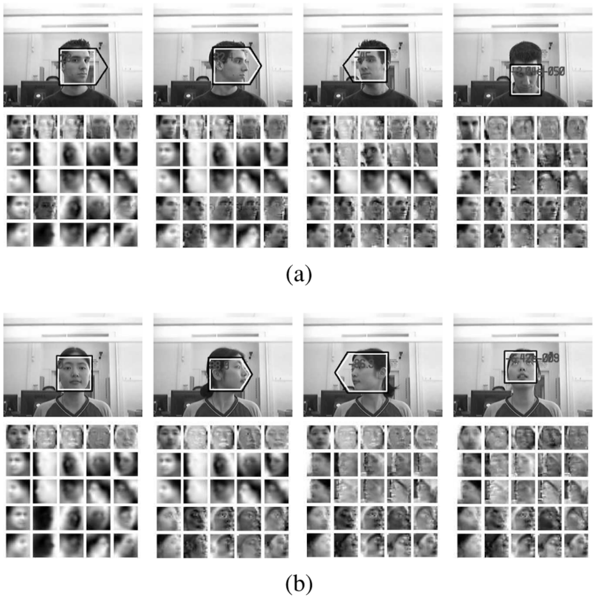
$$\hat{\beta}_t = \arg_{\beta_t} \max P(\beta_t|Z_{1:t}). \tag{17}$$

Face angle can be roughly estimated from different pose models. Assuming that kth pose is associated with a typical pose angle



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

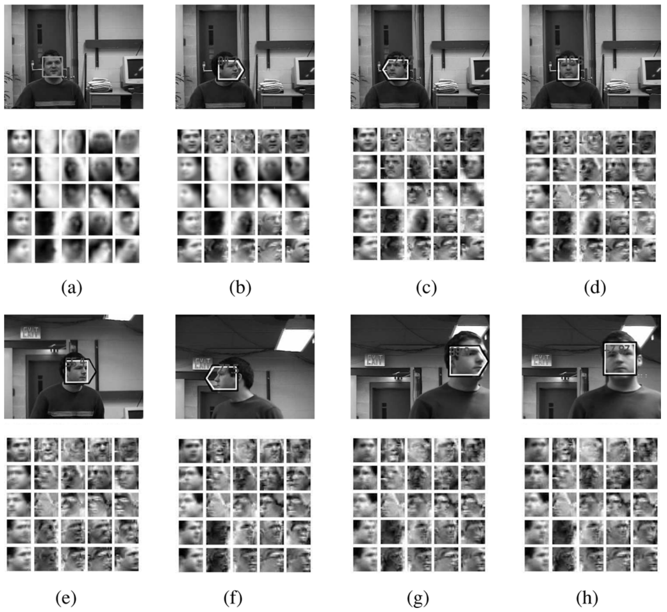
Fig. 5. Collaborative multiview face tracking and pose estimation results: (a)–(e) tracking results in a video sequence; (f) pose angle estimation results.



(a)

(b)

Fig. 6. Multiview face tracking and learning results of two specific faces. (a), (b) At each graph, the first row shows tracking results, and the second row shows the learned mean faces and first four transformation vectors of MPPCA for each of the five poses.



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

(e) (f) (g) (h)

Fig. 7. Tracking dynamic pose changes. (a)–(h) Tracking results at some frames. At each graph, the first row shows tracking result, and the second row shows the learned mean faces and first four transformation vectors of MPPCA for each of the five poses.

$\theta_k,\ k = 1,\ldots,M$, the pose angle can be estimated from the interpolation of multiview face tracking results, as in (18)

$$\bar{\theta}_t = E[\theta_t] = \sum_k \theta_k P(\beta_t = k | Z_{1:t}). \tag{18}$$

An experiment of collaborative tracking and pose estimation is illustrated in Fig. 5. In the experiment, the pose angles associated with different poses are $-90°, -45°, 0°, 45°, 90°$, respectively. The rectangle, half arrow and full arrow represent frontal, half profile and full profile faces, respectively. As shown in Fig. 5, the face begins from the frontal pose, and changes gradually to different views. The curve at Fig. 5(f) shows estimated face angles, and clearly indicates the pose motion.

C. Online Face Learning During Tracking

The method is applied to several representative sequences to demonstrate how the collaborative tracking method handles dynamic face appearances and improves tracking robustness. In the first experiment, multiview faces of two persons are tracked, and their specific face models are learned online, as shown in Fig. 6. In each graph of this figure, the first row shows the tracking results. The columns of the second row show the learned mean faces, and the first four column vectors in the transformation matrix W_i for each of five poses. All the specific single-view face models are initialized with face appearance models that are learned offline in a batch mode using collected samples. Such initial specific face models contain large variance caused by interpersonal differences, and do not accurately model each individual's faces. During collaborative tracking, tracked faces are used to update specific face models through EM learning algorithms. The specific face models gradually converge to individual persons' faces. The experiment demonstrates that the collaborative tracking method can simultaneously track a face, estimate its pose, and update specific face models.

To validate the capability of the collaborative tracking in learning specific face models under difficult conditions, the method is also tested on a sequence in which faces undergo large face location and pose variations, as well as illumination changes, as shown in Fig. 7. All the environmental conditions and face dynamics contribute to the difficulties of face tracking. Our collaborative tracking method can successfully track multiview faces, and simultaneously learn specific face appearance models. It is worth emphasizing that without the generic face model, the online face learning is vulnerable to tracking errors, especially during initial stages of tracking, because the initialized specific face models do not have discriminative capability of separating faces of different poses from the background. As shown in Fig. 7, at the first several frames, the principal components in MPPCA are all blurred as they are initialized with

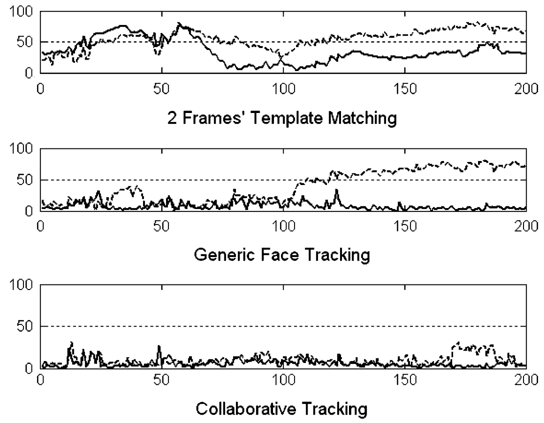
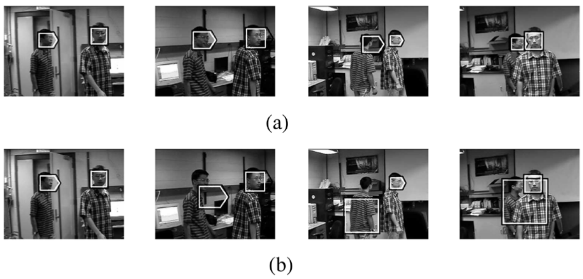


Fig. 8. Comparing tracking results of different methods under perturbation. The horizontal axis is the index of frame. The vertical axis is the pixel distance of tracked face to ground truth. The solid line: $\sigma_p = 0.02$ L. The doted line: $\sigma_p = 0.04$ L. L is the actual size of face.

models learned from samples of different faces. During collaborative tracking, the online model learning at the early stage in tracking is regularized by the generic model, which can select appropriate views for model updating. Also, since the generic face models are trained using both the faces and non-faces, it is able to prevent tracking from drifting before the appropriate specific face models are established. The learned specific face models in turn contribute to the collaborative tracking. In summary, the collaborative tracking strategy optimally utilizes both the offline-collected information of the generic face and the online-obtained information of the specific faces being tracked, and it, hence, improves the tracking robustness.

We further evaluate the robustness of the collaborative tracking by comparing it with a 2-frame template matching method, and also with a generic face tracking method. In the 2-frame template matching method, we search the face scale and position through template matching [30]. During tracking, the face template is updated with previously tracked faces, and this method is only based on appearance of specific faces. The generic face tracking method, on the other hand, has the same formalization as the collaborative tracking, except that the specific face models in generic face tracking are ignored by assuming that their outputs are uniform, i.e., $P(Z_t|\beta_t, \alpha_t) \propto P(\beta_t|Z_t, \alpha_t)$; therefore, the generic face tracking is based only on the offline-learned generic face model. The comparison is performed under simulated perturbation. To simulate the environmental distraction and inaccurate measurements, a random Gaussian noise $N(0, \sigma_p)$ is imposed on locations and sizes of tracked face. In the 2-frame template matching method, the position and size of tracked faces are perturbed at each frame. In generic face tracking and collaborative tracking, the same noise is added to the state of each particle. σ_p is set to be proportional to the actual face size L at each frame, and will be changed to simulate different levels of perturbation. The pixel distance between tracked faces and the ground truth of different methods is measured, as shown in Fig. 8, where the sequence of Fig. 6(a) is used. It shows that template matching method loses tracking very easily under perturbation, as expected. The generic face tracking method shows more robustness than the template matching method for small perturbation, but it also fails under a larger perturbation. Collaborative tracking, which combines multiple measurements, is more robust against large perturbations.


(a)

(b)

Fig. 9. Tracking results of different measurement models: (a) collaborative tracking results at some frames; (b) 2-frames' template matching tracking results at corresponding frames; (c) generic face tracking results at corresponding frames.

Besides the simulation experiment, we also validate the robustness of collaborative tracking under real world conditions. Fig. 9 shows actual tracking results of a sequence using different tracking algorithms. The faces in this sequence undergo large scale changes, along with significant illumination and pose changes. With collaborative tracking, faces are successfully tracked, whereas the template matching and generic tracking methods fail at certain frames. Fig. 9(b) shows that only using appearance templates, tracking easily fails when there is an inaccurate measurement of face scale and position. Although offline face model can handle some degrees of environmental and face pose variation, it can still fail in certain situations where false detections are introduced by the background, as shown in Fig. 9(c). Collaborative method combines two types of measurements, therefore being able to yield more robust tracking results.

Our method can also be applied to tracking multiple faces. Our method first applies face detection to search faces in the video. Once a new face is detected, a collaborative face tracker is created to track the face. For the ith face being tracked, a specific appearance model $P(Z_t|\alpha_t^i, \beta_t^i)$ is built only for this individual, while multiple face trackers share the same generic face model $P(\beta_t|Z_t, \alpha_t)$. Please note that the superscript i refers to a face tracker, not a state variable in a tracker. The collaborative tracking scheme is then applied to each individual face independently, following (1), (5), and (6). Fig. 10 shows tracking results for a sequence containing two persons' faces. Without using specific face models, the generic face tracking method is affected by the background and finally loses tracking, as shown in the third column in Fig. 10(b), while the collaborative tracking results show robustness against both environmental distraction and pose variation. This experiment, along with the experiments shown in Figs. 8 and 9, demonstrate the advantages of collaborative tracking.

Occlusion between multiple faces is a practical issue that could fail our face tracking. When two faces come close together and occlude, their respective measurements will affect each other, therefore polluting each face's tracking. Although this paper does not particularly address the occlusion problem,

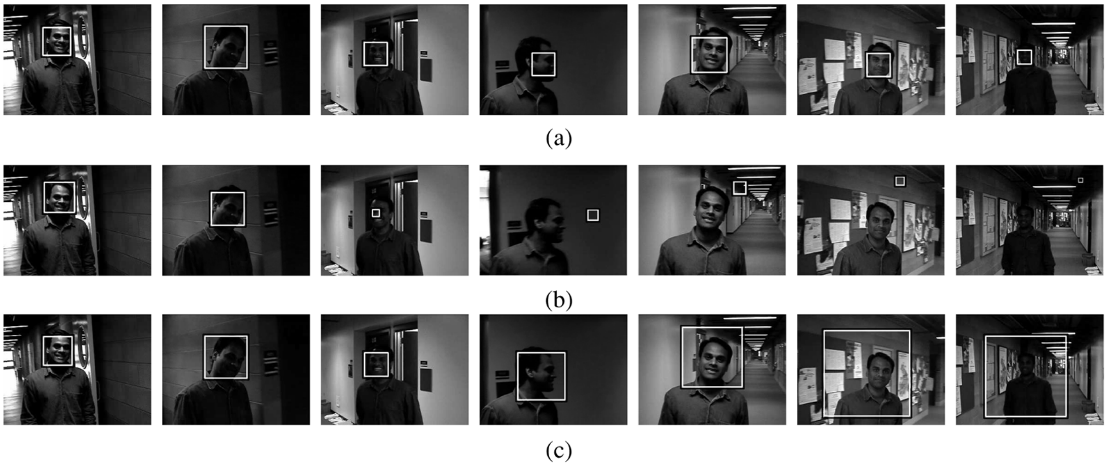


Fig. 10. Multipeople multiview face tracking: (a) the first row shows collaborative tracking results; (b) the second row shows tracking results using only generic face models.

our framework can combine existing methods to handle occlusions by including the face identity in tracking. Since specific face models will be retained for each individual face, the face identity can be included as a state variable, in addition to the face location and pose states, in the collaborative tracking framework. When faces are close to each other, one face could be partially or entirely occluded. Such occlusion can be handled via a template switching scheme [20], which selects an appropriate face appearance model based on occlusions of faces. The switching scheme in [20] is also based on DBNs and can be naturally incorporated into our collaborative tracking framework. Extending our method for a comprehensive solution to the occlusion problem will be our future research.

VI. Conclusion

In this paper, a probabilistic framework is presented to track multiview faces and learn their appearance model online. Without knowing the prior information for a specific face, our method can robustly track the face under pose and environmental changes and automatically build appearance models for the specific face under multiple poses during tracking. A collaborative tracking method is developed to combine the information obtained from both an offline trained generic face model and an online learned specific face models, thus handling large face pose and environmental changes. The probabilistic tracking results are also used to update specific face appearance models by applying an online EM algorithm to the MPPCA models. With collaborative tracking and online learning, our proposed method can robustly track oblique pose changes, and simultaneously estimate pose. Our future work will be the application of online learned face models for the face recognition in videos. We are also interested in applying this method to track other types of objects under large pose changes.

References

[1] S. Li and Z. Zhang, "Floatboost learning and statistical face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1112–1123, Sep. 2004.

[2] P. Wang and Q. Ji, "Multi-view face and eye detection using discriminant features," *Comput., Vis., Image Understand.*, vol. 105, no. 2, pp. 99–111, 2007.

[3] I. Leichter, M. Lindenbaum, and E. Rivlin, "A probabilistic framework for combining tracking algorithm," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2004, vol. 2, pp. 445–451.

[4] D. Ross, J. Lim, and M.-H. Yang, "Adaptive probabilistic visual tracking with incremental subspace update," in *Proc. Eur. Conf. Computer Vision*, 2004, vol. 2, pp. 470–482.

[5] P. Wang and Q. Ji, "Multi-view face tracking with factorial and switching hmm," in *Proc. IEEE Workshop on Application of Computer Vision*, 2005, vol. 1, pp. 401–406.

[6] K.-C. Lee and D. Kriegman, "Online learning of probabilistic appearance manifolds for video-based recognition and tracking," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2005, pp. 130–136.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 564–577, May 2003.

[8] M. Isard and A. Blake, "CONDENSATION: Conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[9] V. Pavlovic, J. Rehg, C. Tat-Jen, and K. Murphy, "A dynamic bayesian network approach to figure tracking using learned dynamic models," in *Proc. IEEE Int. Conf. Computer Vision*, 1999, vol. 1, pp. 20–27.

[10] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME—J. Basic Eng. D*, vol. 82, pp. 35–45, 1960.

[11] J. S. Liu and R. Chen, "Sequential monte carlo methods for dynamic systems," *J. Amer. Statist. Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.

[12] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[13] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.

[14] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," in *Proc. Eur. Conf. Computer Vision*, 1996, vol. 1, pp. 329–342.

[15] A. Jepson, D. J. Fleet, and T. F. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1296–1311, Oct. 2003.

[16] K. Toyama and A. Blake, "Probabilistic tracking in a metric space," in *Proc. IEEE Int. Conf. Computer Vision*, 2001, vol. 2, pp. 50–57.

[17] Y. Wu and T. S. Huang, "Robust visual tracking by integrating multiple cues based on co-inference learning," *Int. J. Comput. Vis.*, vol. 58, no. 1, pp. 55–71, Jun. 2004.

[18] J. Ho, K. Lee, M. Yang, and D. Kriegman, "Visual tracking using learned linear subspace," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2004, vol. 1, pp. 782–789.

[19] J. Sherrah and S. Gong, "Fusion of perceptual cues for robust tracking of head pose and position," *Pattern Recognit.*, pp. 1565–1572, 2001.

[20] Y. Wu, T. Yu, and G. Hua, "Tracking appearances with occlusions," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Jun. 2003, vol. 1, pp. 789–795.

[21] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated person tracking using stereo, color, and pattern detection," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Jun. 1998, pp. 601–608.

[22] R. C. Verma, C. Schmid, and K. Mikolajcayk, "Face detection and tracking in a video by propagating detection probabilities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1216–1228, Oct. 2003.

[23] K. Murphy, "Dynamic Bayesian networks: representation, inference and learning," Ph.D. dissertation, Univ. California, Berkeley, 2002.

[24] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Statist.*, vol. 28, no. 2, pp. 337–374, 2000.

[25] R. E. Schapire, "A brief introduction to boosting," in *Proc.16th Int. Joint Conf. Artificial Intelligence*, 1999, pp. 246–252.

[26] P. Viola and M. Jones, "Robust real-time object detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[27] M. Tipping and C. Bishop, "Probabilistic principal component analysis," *J. Roy. Statist. Soc. B*, no. 3, pp. 611–622, 1999.

[28] D. Titterington, "Recursive parameter esitmation using incomplete data," *J. Roy. Statist. Soc. B*, vol. 46, no. 2, pp. 257–267, 1984.

[29] M. Tipping and C. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, no. 2, pp. 443–482, 1999.

[30] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Conf. Artificial Intelligence*, 1981, pp. 674–679.

Peng Wang (M'03) received the B.S. and M.S. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, in December 2005.

He is currently a Research Scientist with Siemens Corporate Research, Princeton, NJ. Before joining in Siemens, he was a Postdoctoral Research Fellow at the Section of Biomedical Image Analysis, Department of Radiology, University of Pennsylvania, Philadelphia. His research interests are in computer vision, pattern recognition, statistical machine learning, and biomedical image analysis.

Qiang Ji (SM'04) received the Ph.D. degree in electrical engineering from the University of Washington, Seattle.

He is currently an Associate Professor with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute (RPI), Troy, NY. Prior to joining RPI in 2001, he was an Assistant Professor with Department of Computer Science, University of Nevada, Reno. He also held research and visiting positions with the Beckman Institute, University of Illinois at Urbana-Champaign, Urbana; the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA; and the U.S. Air Force Research Laboratory. He currently serves as the director of the Intelligent Systems Laboratory (ISL), RPI. His research interests are in computer vision, pattern recognition, and probabilistic graphical models. He has published over 100 papers in peer-reviewed journals and conferences.

Prof. Ji's research has been supported by major governmental agencies including NSF, NIH, DARPA, ONR, ARO, and AFOSR, as well as by major companies including Honda and Boeing. He is an editor for several computer vision and pattern recognition related journals and he has served as a program committee member, area chair, and program chair in numerous international conferences/workshops.