Online Learning of Probabilistic Appearance Manifolds for Video-based Recognition and Tracking

Kuang-Chih Lee
Computer Science
University of Illinois, Urbana-Champaign
Urbana, IL 61801
klee10@uiuc.edu

David Kriegman
Computer Science & Engineering
University of California, San Diego
La Jolla, CA 92093
kriegman@cs.ucsd.edu

Abstract

This paper presents an online learning algorithm to construct from video sequences an image-based representation that is useful for recognition and tracking. For a class of objects (e.g., human faces), a generic representation of the appearances of the class is learned off-line. From video of an instance of this class (e.g., a particular person), an appearance model is incrementally learned on-line using the prior generic model and successive frames from the video. More specifically, both the generic and individual appearances are represented as an appearance manifold that is approximated by a collection of sub-manifolds (named pose manifolds) and the connectivity between them. In turn, each sub-manifold is approximated by a low-dimensional linear subspace while the connectivity is modeled by transition probabilities between pairs of sub-manifolds. We demonstrate that our online learning algorithm constructs an effective representation for face tracking, and its use in video-based face recognition compares favorably to the representation constructed with a batch technique.

1 Introduction

Thanks to low cost cameras and powerful personal computers, it is now possible to apply machine learning directly on video streams and build interesting real-time applications such as video-based face recognition.

However, algorithms proposed in recent papers are only able to perform the recognition task in real-time, while the training process usually runs off-line in a batch mode. All of training data must be stored, and the time complexity is at least proportional to the size of the dataset. Hence, batch training algorithms are not practical for huge datasets such as lengthy video sequences, nor can they be applied in tasks that require real-time training (e.g., video tracking algorithms using a generative model of tracked object constructed from incoming video).

In contrast, online learning algorithms typically discard data as soon as it is processed and require only a small amount of memory to store say model parameters or the algorithm's state. For processing of video streams, this is very desirable.

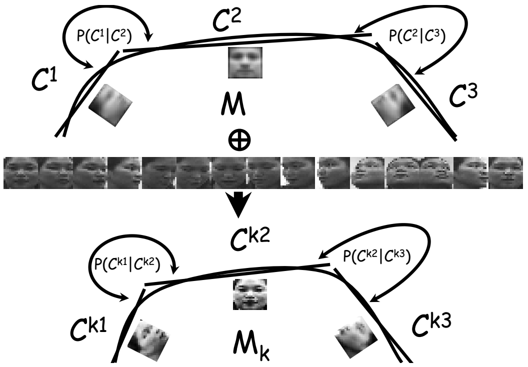


Figure 1. Learning an appearance manifold. A general-to-specific online approach is presented to evolve a generic appearance manifold $\mathcal{M}$ to a specific manifold $\mathcal{M}_k$ from a video sequence. The complex and nonlinear appearance manifold $\mathcal{M}$ can be approximated as the union of several simpler sub-manifolds and the connectivity between them; here, each sub-manifold $\mathcal{C}^i$ is represented by a PCA plane. The connectivity between the pose sub-manifolds describes the probability $P(C^i|C^j)$ of moving from one sub-manifold C^j to another sub-manifold C^i at any time instance.

In this paper, we present an online learning algorithm for constructing a probabilistic appearance manifold. A probabilistic appearance manifold [9], illustrated in Figure 1, is modeled as a collection of submanifolds (called pose manifolds) in the image space and the connectivity between them. In previous work [9], this representation was learned by a batch training process from short video clips. First, the K-means algorithm was applied to partition frames from the training video into clusters. Images assigned to the same cluster usually arise from neighboring poses. Principal component analysis (PCA [12]) was then applied to each cluster to yield a low dimensional linear subspace approximation. The connectivity among these linear subspaces was represented by a transition matrix whose elements capture the likelihood that successive frames will be make a transition between a pair of pose subspaces.

The online learning algorithm introduced in this paper constructs the appearance manifold in a different manner. We start with a prior generic appearance manifold that has been constructed from multiple pre-training video sequences of different instances of the class. In addition, we have a video of a particular instance of the class. At each time only one frame is available in the video sequence, and the appearance manifold is updated using this frame. Over the sequence, the generic appearance model evolves to an object-specific appearance manifold. The online learning algorithm consists of two steps. The first is a pose estimation problem, where our goal is to identify the best submanifold to which the current image of the specific object belongs with the highest posteriori probability. The second step is to incrementally update the appearance manifold. The result from the first step is applied to find a set of pre-training images that are expected to appear similar to the specific object in other poses. Then all of the subspaces in the appearance manifold are updated to minimize the reconstruction error.

We have tested our algorithm for online learning of probabilistic manifolds on video sequences of human faces with significant 2-D and 3-D head rotation. The learned appearance manifolds are shown to be effective for video-based face recognition performance. In addition, our learned representation has been applied within subspace-based video tracking, and it outperforms the existing Eigen-tracking algorithm [1].

2 Related Work

Although numerous appearance-based tracking and recognition algorithms have been proposed, online learning algorithms are only studied and applied in a small portion of these algorithms [2, 7, 8, 13].

Brand [2] and Ross et al. [13] applied an incremental SVD algorithm to tracking where appearance was represented by a single subspace. Since the appearance manifold arising from large pose variation is highly nonlinear, a single subspace is likely to be inadequate. Therefore such trackers experience difficulties when the tracked target exhibits drastic changes in appearance.

In Ho et al. work [7], a small neighborhood of the appearance manifold was represented as a single subspace that is constructed online using the most recent images in a video. This method can robustly track an object despite large pose changes; however, since the learned appearance model does not represent the whole appearance manifold, its application to recognition is limited.

Jepson et al. [8] proposed an elaborate mixture model and an online EM algorithm for tracking. Such a mixture model captures stable properties of the image appearance and assigns different weights to image pixels in motion estimation. Our work bears some resemblance to their method in the sense that our method also utilizes a mixture structure and an online update procedure. However, our algorithm admits clear and concise geometric and Markovian interpretations in terms of the appearance manifolds and the pose transition probability in the image space. The advantage of applying the complex structures in our framework is that these structures provide better guidelines to the online update process, yielding a more accurate appearance model and leading to improved recognition and tracking performance.

3 Mathematical Framework

Let $\mathcal{M}$ denote the generic appearance manifold, which consists of a collection of m simpler disjoint sub-manifolds, $\mathcal{M} = \mathcal{C}^1 \cup \cdots \cup \mathcal{C}^m$, with $\mathcal{C}^i$ denoting a sub-manifold. Each $\mathcal{C}^i$ is assumed to be amenable to linear approximation by a low-dimensional linear subspace computed using principal component analysis (i.e., a PCA plane). In the following derivation, consider each sub-manifold $\mathcal{C}^i$ to be a *pose subspace* since it captures the appearance of the object in nearby poses.

In our framework, $\mathcal{M}$ can be easily trained by a batch process, where we manually assign a set of face images with similar pose into m clusters, and then apply PCA to each cluster to yield the low-dimensional pose subspace $\mathcal{C}^i$. The transition matrix is initialized with a uniform distribution. In addition, for each person in the pre-training video dataset, we compute the average face image in each pose. For each pose subspace $\mathcal{C}^i$, we gather a set of pre-training exemplars $\{\mathbf{x_1^i}, \cdots, \mathbf{x_Q^i}\}$ from Q people, where $\mathbf{x_q^i}$ represents the average face image of person q seen in pose i.

Now, let $\{F_1, \cdots, F_l\}$ denotes l frames of a video sequence of person k, and let I_t be a region containing a face cropped from F_t. We assume that each image I_t in the training video sequence is a fair sample drawn from the appearance manifold $\mathcal{M}_k$. Now consider the problem of doing online updating of the appearance manifold $\mathcal{M}$ upon seeing a new face image I_t at time t in order to evolve $\mathcal{M}$ to $\mathcal{M}_k$. We can subdivide this into the following two sub-problems. The first is the pose estimation problem. The probabilistic estimate of the pose manifold $\mathcal{C}_t^{i*}$ given the current face image I_t and the previous estimated result $\mathcal{C}_{t-1}^j$ can be written as:

$$\mathcal{C}_t^{i*} = \arg\max_i p(\mathcal{C}_t^i | I_t, \mathcal{C}_{t-1}^j). \quad (1)$$

The detailed algorithm is described in Sec. 3.1.

The second problem is how to update the appearance manifold $\mathcal{M}$ in order to minimize the following L^2 reconstruction error:

$$Error^2(\mathcal{M}, \{I_1, \cdots, I_{t-1}, I_t\}) \quad (2)$$

where $\{I_1, \cdots, I_{t-1}\}$ denotes the previous face images in the video. However, in online learning, we do not retain $\{I_1, \cdots, I_{t-1}\}$. The available information about $\mathcal{M}$ are the parameters characterizing each component $\mathcal{C}^i$, such as the centers and eigenbasis computed from $\{I_1, \cdots, I_{t-1}\}$ during previous updates. Another piece of information that we have is the current observation I_t. Sections 3.2 and 3.3 show how to incrementally update each pose subspace and reach a good approximation of $\mathcal{M}$. Finally, the incremental update of the pose transition matrix is described in Sec. 3.4.

3.1 Pose Estimation

In a Bayesian framework, we wish to choose the pose subspace $\mathcal{C}_t^i$ which maximizes the posterior probability $p(\mathcal{C}_t^i|I_t, \mathcal{C}_{t-1}^j)$ in Equation 1. We further assume that I_t and $\mathcal{C}_{t-1}^j$ are independent given $\mathcal{C}_t^i$, and the transition probability $p(\mathcal{C}^i|\mathcal{C}^j)$ is time invariant. Using Bayes' rule and these assumptions, we have the following:

$$\begin{aligned} p(\mathcal{C}_t^i|I_t, \mathcal{C}_{t-1}^j) &= \alpha p(I_t|\mathcal{C}_t^i, \mathcal{C}_{t-1}^j) p(\mathcal{C}_t^i|\mathcal{C}_{t-1}^j) \\ &= \alpha p(I_t|\mathcal{C}_t^i) p(\mathcal{C}^i|\mathcal{C}^j) \end{aligned} \tag{3}$$

where α is a normalization term to ensure a proper probability distribution.

$\mathcal{C}^i$ is approximated by an affine subspace $\mathcal{L} = \{\mathbf{c}, \Phi, \Lambda, P\}$ constructed using PCA, where $\mathbf{c}$ is the center of the subspace, Φ is the eigenvector matrix, and Λ is the corresponding diagonal matrix of eigenvalues, i.e., $\Lambda_{jj} = \lambda_j$. P records the number of image samples used to construct pose subspace $\mathcal{L}$. Note that we omit the superscript i for each pose subspace $\mathcal{L}$ and its parameters to simplify the notation in the following. The linear projection from I_t to $\mathcal{L}$ can be written as $\mathbf{y} = (y_1, \ldots, y_M) = (\Phi)^T(I_t - \mathbf{c})$. Now the likelihood probability can be expressed as the product of two Gaussian densities [11]

$$\begin{aligned} & p(I_t|\mathcal{C}^i) = p(I_t|\mathcal{L}) \\ & = \left[\frac{\exp\left(-\frac{1}{2}(\sum_{r=1}^{M}\frac{y_r^2}{\lambda_r})\right)}{(2\pi)^{\frac{M}{2}}\prod_{r=1}^{M}\lambda_r^{\frac{1}{2}}}\right]\left[\frac{\exp\left(-\frac{1}{2\rho}d^2(I_t, \mathcal{L})\right)}{(2\pi\rho)^{\frac{N-M}{2}}}\right], \end{aligned} \tag{4}$$

where N denotes the dimension of the image space, M denotes the dimension of the pose subspace $\mathcal{L}$, and $d^2(I_t, \mathcal{L})$ denotes the L^2 distance between an image I_t and pose subspace $\mathcal{L}$. The only parameter ρ in Equation 4 can be chosen as $\frac{1}{N-M}\sum_{i=M+1}^{N}\lambda_i$ [11], or simply $\frac{1}{2}\lambda_{M+1}$ [3].

The two Gaussian densities in Equation 4 have important geometric interpretations (See Figure 2). The first Gaussian distribution can be interpreted as the likelihood of the inplane Mahalanobis distance which acts to bound the pose subspace $\mathcal{C}^i$ and decrease the false positive rate during pose estimation. The second Gaussian distribution can be interpreted as the likelihood of the out-of-plane L^2 distance from a point to a subspace, and it works like a detection process to decide which pose subspace the current image I_t belongs to.

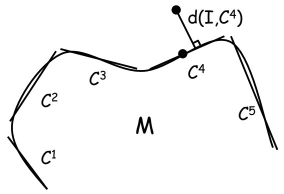


Figure 2. Geometric interpretation of the probabilistic distance measure between an image to a pose subspace.

The prior probability $p(\mathcal{C}^i|\mathcal{C}^j)$ stands for the transition probability between pose subspaces, and it captures the temporal dynamics of the face motion in the training video sequence. The transition probability $p(\mathcal{C}^i|\mathcal{C}^j)$ encodes the temporal coherence of human motion as a face cannot move suddenly from $\mathcal{C}^i$ to $\mathcal{C}^j$ if these two poses are not connected or have a low probability (e.g., one cannot move from the leftmost pose to rightmost pose without going through some intermediate pose.).

3.2 Image Approximation in Other Poses

After we find out which pose subspace $\mathcal{C}^{i*}$ the current image I_t belongs to, the next step is to update the appearance manifold $\mathcal{M}$. Rather than just updating $\mathcal{C}^{i*}$, we update all of the pose subspaces, even those with differing pose than the training image. The idea is the following: If the person in the current training image looks like a combination of some people from the pre-training set, say Frank and Joe in pose i, then that person probably looks like the same combination of Frank and Joe in all other poses as well. More concretely, we first find a set of K nearest neighbors $\{\mathbf{z_1^{i*}}, \cdots, \mathbf{z_K^{i*}}\}$ of I_t from the set of the pre-training exemplar face images $\{\mathbf{x_1^{i*}}, \cdots, \mathbf{x_Q^{i*}}\}$ for the pose subspace $\mathcal{C}^{i*}$. Then we use these K nearest neighbors to linearly approximate I_t by determining a collection of coefficients w_r that minimizes the objective function

$$\min \|I_t - \sum_{r=1}^{K} w_r z_r^{i*}\|_{L^2}^2, \tag{5}$$

with the constraint that $\sum_{r=1}^{K} w_r = 1$. Let $\{\mathbf{z_1^j}, \cdots, \mathbf{z_K^j}\}$ correspond to the average face image of another pose subspace $\mathcal{C}^j$, where $\mathbf{z_r^j}$ and $\mathbf{z_r^{i*}}$ contains the same person's face. The coefficients w_r computed in Equation 5 are applied to the corresponding image set $\{\mathbf{z_1^j}, \cdots, \mathbf{z_K^j}\}$ to synthesize an image I_t^j for pose j by the following equation:

$$I_t^j = \sum_{r=1}^{K} w_r z_r^j. \tag{6}$$

The results is a set of synthetic face images for all other poses. We then update each pose subspace $\mathcal{C}^j$ incrementally using the corresponding real or synthetic image; the update algorithm is detailed in the next subsection.

3.3 Incremental Subspace Update

After we synthesize images for each pose subspace $\mathcal{C}^i$, the next step is to update the current eigenspace model of $\mathcal{C}^i$ with the new image sample. Numerous algorithms have been developed to incrementally update eigenbasis as new observations become available [2, 4, 5, 10]. However, only the algorithm developed by Hall et al. [4, 5] updates the mean vector and eigenbases without storing the covariances or the previous training examples. In this section, we summarize Hall's algorithm [4] used in our framework to incrementally update the subspace parameters for a fixed subspace dimension.

Assume that we are trying to incrementally update the current subspace $\mathcal{L}$ specified by $\{\mathbf{c}, \Phi, \Lambda, P\}$, to a new subspace $\mathcal{L}'$ specified by $\{\mathbf{c}', \Phi', \Lambda', P+1\}$ with the new available observation $\mathbf{x}$ in order to minimize the reconstruction error in Equation 2. The parameters $\mathbf{c}$, Φ, and Λ denote the center, eigenvectors, and eigenvalues of the subspace $\mathcal{L}$ respectively. P denotes the number of images which are used to construct the subspace $\mathcal{L}$. The minimization problem is actually equivalent to solving the eigenproblem

$$S'\Phi' = \Phi'\Lambda', \tag{7}$$

where S' is the new covariance matrix, and Φ' and Λ' are the corresponding new eigenbasis and eigenvalues.

The projection of the new observation to the current subspace $\mathcal{L}$ and the orthogonal residue vector are given by:

$$\begin{aligned} \mathbf{g} &= \Phi^T\bar{\mathbf{x}}, & (8) \\ \mathbf{h} &= \bar{\mathbf{x}} - \Phi\mathbf{g}, & (9) \end{aligned}$$

where $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{c}$.

Now the new mean $\mathbf{c}'$ and new covariance matrix S' can be easily verified as:

$$\begin{aligned} \mathbf{c}' &= \frac{1}{P+1}(P\mathbf{c} + \mathbf{x}) & (10) \\ S' &= \frac{P}{P+1}S + \frac{P}{(P+1)^2}\bar{\mathbf{x}}\bar{\mathbf{x}}^T & (11) \end{aligned}$$

We then expand the old subspace by increasing its dimension by one to cover the new observation $\mathbf{x}$. This is done by simply adding the unit residue vector $\hat{\mathbf{h}}$

$$\hat{\mathbf{h}} = \begin{cases} \frac{\mathbf{h}}{\|\mathbf{h}\|_2} & \text{if } \|\mathbf{h}\|_2 \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

to form a new orthonormal basis $[\Phi, \hat{\mathbf{h}}]$. The key idea is to figure out a rotation matrix R to translate the new orthonormal basis to the eigenbasis Φ' which expands the new subspace C', i.e.,

$$\Phi' = [\Phi, \hat{\mathbf{h}}]R. \tag{13}$$

After substituting Equations 13 and 11 into Equation 7 and multiplying $[\Phi, \hat{\mathbf{h}}]^T$ in both sides, we obtain

$$[\Phi, \hat{\mathbf{h}}]^T(\frac{P}{P+1}S + \frac{P}{(P+1)^2}\bar{\mathbf{x}}\bar{\mathbf{x}}^T)[\Phi, \hat{\mathbf{h}}]R = R\Lambda'. \tag{14}$$

The covariance matrix S can be approximated as

$$S \approx \Phi\Lambda\Phi^T \tag{15}$$

By substitution of Equations 15 and 8 and some simple matrix algebra into Equation 14, Equation 14 can be further simplified as

$$\left(\frac{P}{P+1}\left[\begin{array}{cc} \Lambda & 0 \\ 0 & 0 \end{array}\right] + \frac{P}{(P+1)^2}\left[\begin{array}{cc} \mathbf{g}\mathbf{g}^T & \gamma\mathbf{g} \\ \gamma\mathbf{g} & \gamma^2 \end{array}\right]\right)R = R\Lambda', \tag{16}$$

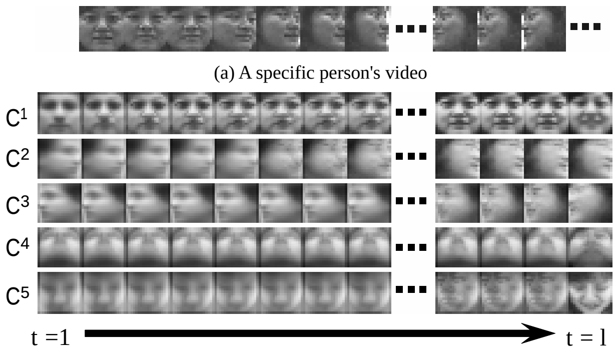
where γ is equal to $\hat{\mathbf{h}}^T\bar{\mathbf{x}}$. Now we have finished updating all the parameters in the new pose subspace. The new mean $\mathbf{c}'$ is updated using Equation 10. The new eigenvalues Λ' are directly the solution of the eigen-problem in Equation 16, and the new eigenvectors Φ' can be computed by Eq. 13 once R is available. Only the first M new eigenvalues and eigenvectors are finally retained. The size of the square rotation matrix R is $(M+1) \times (M+1)$, where M is the dimension of the pose subspace. Since M is a small value, Equation 16 can be evaluated efficiently in real-time[1]. In addition, Eq. 16 will converge to a stable solution as the number of samples P goes to infinity.

3.4 Model Construction

The online construction of the appearance manifold is summarized as follows. For each incoming face imageI_t from the online training video of the k-th person's head movement, we first determine which pose subspace it belongs to by evaluating Equations 1, 3, and 4. Once this pose subspace $\mathcal{C}^{i*}$ is found, we then synthesize all the possible face images in other poses by Equations 5 and 6. With these images, we evaluate all the equations from Equation 7 to 16 in order to update the parameters representing all pose subspaces [2]. Finally, the transition matrix can be updated incrementally by counting the actual transitions between different pose manifolds observed in the training sequence:

[1] For the numerical stability, we can keep a slightly larger number rather than exactly using M during the incremental subspace updating. Then after online updating, only the first M eigenvalues and eigenvectors are used to evaluate Eq. 4 during pose estimation.

[2] In our implementation, we reduce the effect of the synthetic images to its pose subspace over time because it is only an approximation. This can be done simply by adding another large counter to P increasing with time. Also we can simply stop updating the synthetic image after some time period.



(a) A specific person's video

(b) Changes of centers in pose subspaces over time

Figure 3. Online learning process. Part (a) shows frames from an input video. Part (b) shows the evolution of the appearance manifold where each row is the evolution of one pose subspace and individual images are the PCA centers at a particular time instant. The first column at time $t = 1$ shows the PCA centers for the initial (generic) pose manifold. Moving to the right, the centers are updated using the particular input frame shown in part (a) until at time $t = l$, the final appearance manifold shown in the rightmost column looks more like the person shown in (a) than the generic person shown in the first column.

$$p(\mathcal{C}^{ki}|\mathcal{C}^{kj})' = \frac{T_{ij}p(\mathcal{C}^{ki}|\mathcal{C}^{kj}) + \delta(I_t \in \mathcal{C}^{ki})\delta(I_{t-1} \in \mathcal{C}^{kj})}{T_{ij} + \delta(I_t \in \mathcal{C}^{ki})\delta(I_{t-1} \in \mathcal{C}^{kj})} \tag{17}$$

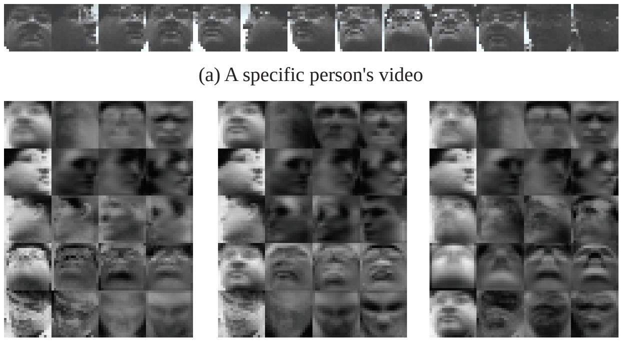
where $\delta(I_t \in \mathcal{C}^{ki}) = 1$ if I_t has the smallest probabilistic distance to pose subspace $\mathcal{C}^{ki}$, and otherwise it is 0. T_{ij} is the accumulated count of the actual transitions between $\mathcal{C}^{ki}$ and $\mathcal{C}^{kj}$, and will be increased by one if the actual transition happens in the current frame I_t, i.e., $\delta(I_t \in \mathcal{C}^{ki})\delta(I_{t-1} \in \mathcal{C}^{kj}) = 1$. The online learning process is illustrated in Figure 3.

3.5 Experimental Result of Learning Process

In this section, we experimentally demonstrate the effectiveness of our online updating procedure. Our method is compared to two other online update strategies. The input video sequence is shown in Figure 4(a), and the resulting appearance manifolds for the three methods are depicted in Figure 4(b) The first column of Figure 4(b) shows the result for our method.

The first alternative method is to estimate the closest pose subspace to the incoming image, and only update that subspace. The results for this are shown in the central column of Figure 4(b). We observe that without updating other subspaces using synthetic images, the center of the fourth pose subspace (Row 4) changes significantly from the initial center for the generic prior, and converges to an image with a different pose.

In the second alternative method, we only use the generic prior appearance manifold to estimate the pose for each incoming image. The linear subspace for each pose is only constructed from the training images of the specific individual, and the prior model is not used at all. This amounts to doing pose estimation followed by incremental subspace learning for each pose. We can observe in Figure 4 that the third pose subspace (Row 3) is constructed from a mixed set of images coming from frontal, left and upper poses, and the center and eigenbasis of the third pose subspace (Row 3) looks blurry. In addition, the parameters of the fourth pose subspace (Row 4) are not updated at all as no input image is classified as coming from pose 4. Finally, the center of the fifth subspace (Row 5) converges to the wrong pose as well.



(a) A specific person's video

(b) Learning results from three online update strategies

Figure 4. Comparison of different online update strategies. With an input video of a specific person shown in (a), (b) compares the pose manifolds learned using three different on-line strategies. Five pose manifolds are constructed, and each row depicts the the PCA center and three eigenbasis images of a pose manifold. The left columns shows the result from our method, the center column shows the result of online updating without synthetic images in other poses, while the right column shows the result of online updating without using the generic prior.

4 Visual Tracking

Our online learning algorithm can be naturally extended to a tracking algorithm. Let $\mathcal{M}$ denote the generic appearance manifold composed of a collection of pose subspace $\mathcal{C}^i$, and let $\{F_1, \cdots, F_l\}$ denote a video sequence of l frames. In our tracking work, we estimate the object location in each frame F_t at time t. We treat the object location as a rectangular region specified by a set $\mathbf{u}$ of five parameters: center location (in image coordinates), width, height, and orientation. Let $f(\mathbf{u}, F_t)$ denote the cropping function which returns a subimage I cropped from the rectangular region, specified by $\mathbf{u}$, from the current image frame F_t. Our tracking algorithm can be succinctly formulated as the following optimization problem:

$$\mathbf{u}_t^* = \arg\max_{\mathbf{u}} \; p(f(\mathbf{u}, F_t)|C_{t-1}^j), \tag{18}$$

where $p(I|\mathcal{C}_{t-1}^i)$ is defined in Equation 4, and it denotes the likelihood between the cropped image I and pose subspace $\mathcal{C}_{t-1}^j$. $\mathbf{u}_t^*$ is the tracking result for frame t. Our tracker evaluates Equation 18 by sampling a collection of sub-images specified by different $\mathbf{u}$ based on a Gaussian distribution centered at $\mathbf{u}^*_{t-1}$[3]. Once the tracking result $\mathbf{u}^*_t$ is obtained, the cropped image I_t can be used to perform pose estimation as well as to incrementally update the subspace as in Section 3. The detailed algorithm is summarized in Figure 5.

Online Pose Subspace Update Tracking Algorithm:

Input Parameters: (Ω, S)

$\Omega = \{\omega_x, \omega_y, \omega_w, \omega_h, \omega_\theta\}$: the set of five parameters for sampling windows on the screen.

S: the number of windows sampled for each frame.

Output: ($I^*, \mathbf{u}^*$)

I^*: image of the tracked object.

$\mathbf{u}^*$: the screen position of I^*.

Model Parameters: ($m, n, L, T, \mathbf{u}^*$)

m: the number of pose subspaces $\mathcal{C}^1, \ldots, \mathcal{C}^m$ of the appearance manifold $\mathcal{M}$.

M: the (common) dimension of the linear subspaces $\mathcal{C}^i$.

$\mathcal{C}^i$: i-th pose subspace, represented by a local mean and a set of orthonormal basis images.

$\mathbf{T}$: a m-by-m probability transition matrix for the tracked object where each entry is an estimated transition probability $p(\mathcal{C}^i|\mathcal{C}^j)$.

$\mathbf{u}^* = (x, y, w, h, \theta)$: the location of the object in image, represented by a rectangular box in the image centered at (x, y) and of size (w, h) with orientation θ.

Initialization:

The tracker is initialized either manually or by an object detector in the first frame. Let I^* be the initial cropped image from the first frame. Using I^*, the initial $\mathcal{C}^i$ is determined by the maximum probabilistic likelihood distance between I^* and each pose subspaces $\mathcal{C}^i$.

Begin

1. **Sample Windows**: Draw S samples of windows $\{W_1, ..., W_r, ..., W_S\}$ in current image frame specified by $\{u_1, ..., u_r, ..., u_S\}$ at various locations of different orientations and sizes according to a 5-dimensional Gaussian distribution centered at $\mathbf{u}^*$ with diagonal covariance specified by Ω.

2. **Tracking**: Rectify each window W_r to a 19-by-19 image and rasterize it to form a vector I_r in $\mathbb{R}^{361}$. Compute the probabilistic likelihood between each I_r and the pose subspace $\mathcal{C}^{i*}$ found in the previous frame by evaluating Equation 4. Choose I^* with $\mathbf{u}^*$ that gives the minimal L^2 distance to $\mathcal{C}^{i*}$ as the tracking output.

3. **Model Update**: Compute Equation 3 to find the pose subspace $\mathcal{C}^{i*}$ which the current tracking result I^* belongs to. Follow the procedure describing in Subsection 3.2 and 3.3 and evaluate Equations 5 - 17 to incrementally update each pose subspace $\mathcal{C}^i$ of the appearance manifold $\mathcal{M}$. Loop back to Step 1 till the last frame.

End

Figure 5. Summary of the proposed tracking algorithm.

5 Experiments and Results

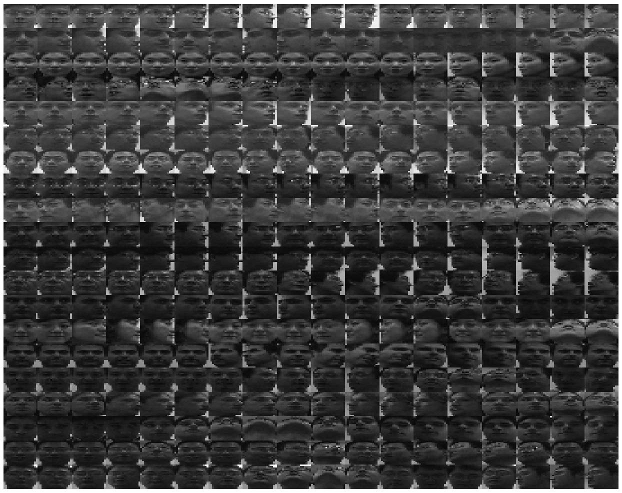


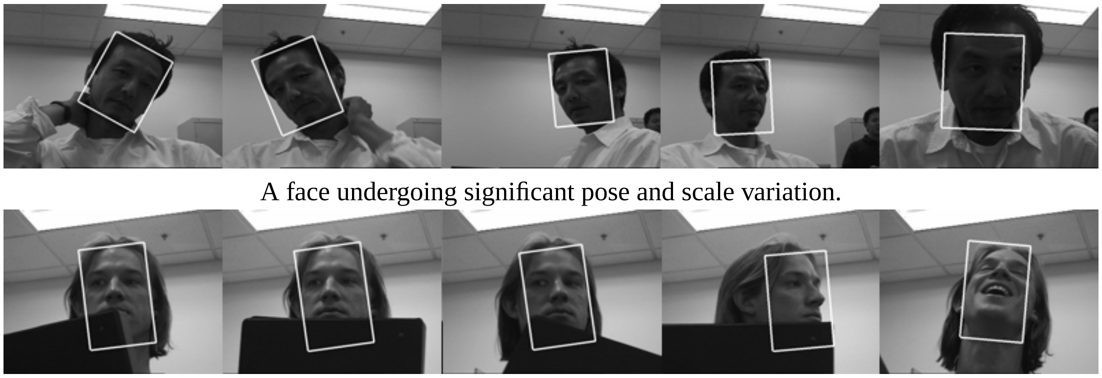
Figure 6. Samples of the videos used in the experiments. All sequences contain significant pose variation.

In this section, we report on the results of applying our online learning approach to video-based face recognition and tracking. Comparisons with well-known existing face recognition and tracking algorithms are presented as well.

5.1 Data Preparation and Pre-training

Since there is no standard video database for evaluating face recognition and tracking algorithms, we collected a set of 82 video sequences of 35 different persons for all of our experiments. Each video sequence was recorded indoors at 15 frames per second over a duration of at least 20 seconds. Each individual appeared in at least two video sequences with the head moving with different combinations of 2-D (in-plane) and 3-D (out-of-plane) rotation, with expression changes, and with differing speed. The pre-training process requires a set of cropped face images taken from frames in the video sequences. These cropped image regions were obtained using a simple face tracker (a variant of the Eigen-Tracking algorithm of [1]) and then manually cleaned up. The images are down-sampled to a standard size of 19×19 pixels. Some examples of cropped face images are shown

[3] More sophisticated sampling techniques for non-Gaussian distributions (e.g., the CONDENSATION algorithm [6] for incorporating dynamic changes in probability distributions) can also be applied.



A face undergoing significant pose and scale variation.

A face partially occluded by a black folder.

Figure 7. Qualitative tracking results for two different video sequences. Each row displays a set of five key frames from a video sequence.

in Figure 6. All of the video sequences will be made available for downloading at *http://vision.ucsd.edu/kriegman-grp/research/*. The main part of the pre-training process is to compute the local linear approximation of the face appearance manifold for a "generic human" as well as the connectivity between these local approximations We picked 15 (out of 35) video sequences to construct the generic face appearance manifold $\mathcal{M}$. We manually assigned every cropped face images into five different pose clusters. Then a 10-D pose subspace was computed from the images in each cluster using PCA. As described in the previous section, the connectivity between different subspaces can be modeled by a transition matrix T, where each matrix entry is initialized with constant probability.

5.2 Face Recognition

COMPARISON OF RECOGNITION METHODS

Method	Accuracy (%)	
	Videos w/o occlusion	Videos with occlusion
Prob. Manifold Recog. w/ Online Learning	95.6	94.0
Prob. Manifold Recog. w/ Off-line Learning	97.2	92.9
Eigenfaces	69.3	53.7
Fisherfaces	74.5	65.4
Nearest Neighbor	81.6	76.3

Table 1. Comparison of Recognition Accuracy for Different Recognition Methods

In the face recognition experiment, we used the remainder of the video sequences (i.e., 20 of 35) to train the person-specific appearance manifold starting with the generic appearance manifold using the online learning algorithm described in Section 3. Once the appearance manifold for each person was constructed, we performed recognition using the algorithm presented in [9] with the other 32 sequences of these 20 people. These test sequences included many difficult situations that occur in "real-world video streams," such as large pose variation, large scale change, partial occlusion, and short term departure of the face from the field of view.

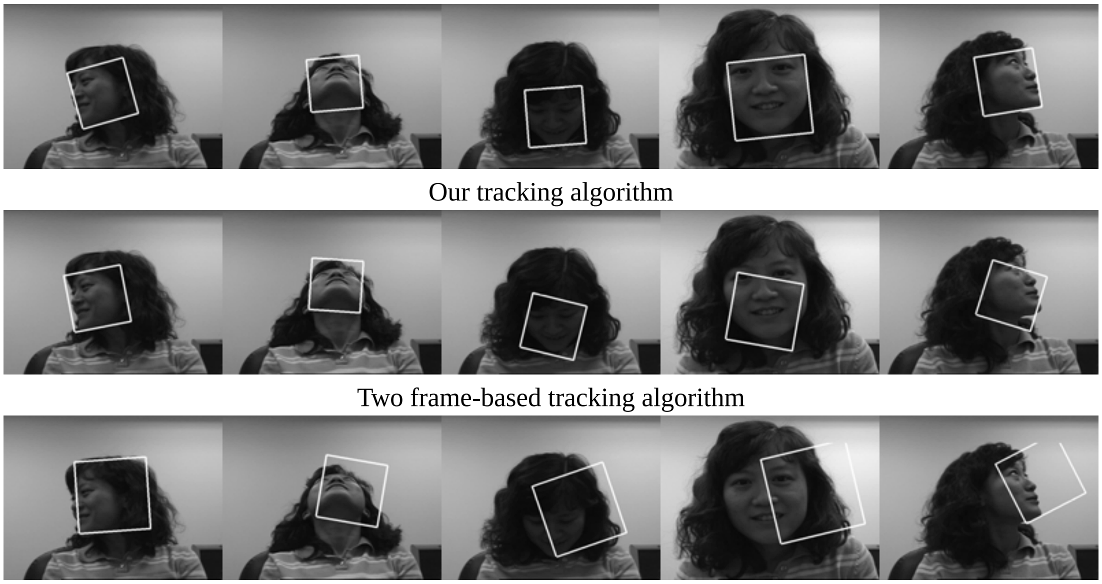
Table 1 shows the result of the probabilistic manifold face recognition using our online training method, off-line training method described in [9] as well as three other standard face recognition algorithms. The error rates were computed by taking the ratio of the number of correctly recognized frames across all test videos and the total number of frames used in the experiment. The results show that the probabilistic manifold face recognition algorithms with online and off-line training are comparable with each other, and they outperform other standard face recognition algorithms.

5.3 Visual Tracking

In this section, we present qualitative studies of the effectiveness of our tracking algorithm that was summarized in Figure 5. Figure 7 illustrates the tracking results for five key frames from two different video sequences. The results demonstrate that despite significant pose variation, our tracker delivered precise tracking results under difficult conditions, such as partial occlusion and large scale changes.

Next, we qualitatively compare our tracking results with two other trackers: the two-frame-based tracker and the Eigen-based tracker [1]. The two-frame-based tracker is the simplest appearance-based tracking algorithm because the appearance model is simply the tracking result from the most recent frame. The Eigen-based tracker tested in this comparative study employed a single generic appearance manifold $\mathcal{M}$ to track all people, and this could be implemented by our algorithm but without the online updating procedure at Step 3 of Fig. 5. Our implementation was slightly different from the original Eigen-based tracker [1] in that we used a collection of subspaces (an appearance manifold $\mathcal{M}$) instead of a single global subspace.

Figure 8 shows the tracking results for five key frames



Our tracking algorithm

Two frame-based tracking algorithm

Eigen-tracking algorithm

Figure 8. Qualitative comparison among our tracker, two-frame-based tracker, and Eigen-based tracker. For each frame, the tracking result is indicated by a white rectangle. Five key frames, the 46nd, 71th, 105th, 150th, and 235th, of 261 frames in the test sequence are shown.

in three tracking algorithms. As the figure suggests, our tracker outperforms the other two standard trackers. Although both of these do not lose the target, one can easily notice significant misalignments in some of the key frames in Figure 8. The misalignment issues might make these tracking results impractical for many image-based recognition applications.

6 Summary and Conclusions

We have proposed an online learning algorithm to construct an appearance manifold from a generic prior and a video of an object instance. We have demonstrated that our online learning algorithm is effective for video-based face recognition and face tracking. One obvious limitation is that our algorithm requires a generic prior model. Our tracking algorithm therefore cannot track an object without knowing and having a model of its class. How to learn an appearance manifold online without a prior model is an important direction in future work.

Acknowledgments

This work was funded under NSF CCR 00-86094, NSF IIS 00-85980, Honda Research Institute, and U.C. MICRO Program.

References

[1] M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int. J. Computer Vision*, 26(1):63–84, 1998.

[2] M. Brand. Incremental singular value decomposition of uncertain data with missing values. In *European Conf. on Computer Vision*, volume 2350, pages 707–720, 2002.

[3] T. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Active shape models - Their training and application. *Computer Vision and Image Understanding*, 61:38–59, 1995.

[4] P. M. Hall, D. R. Marshall, and R. R. Martin. Incremental eigenanalysis for classification. In *The British Machine Vision Conference*, pages 286–295, 1998.

[5] P. M. Hall, D. R. Marshall, and R. R. Martin. Merging and splitting eigenspace models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1042–1049, 2000.

[6] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. In *Int. J. Computer Vision*, 1998.

[7] M.-H. Y. J. Ho, K.-C. Lee and D. Kriegman. Visual tracking using learned subspaces. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 782–789, 2004.

[8] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 25, pages 1296–1311, 2003.

[9] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 313–320, 2003.

[10] A. Levy and M. Lindenbaum. Sequential karhunen-loeve basis extraction and its application to images. *IEEE Transactions on Image Processing*, 9(8):1371–1374, 2003.

[11] B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[12] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. IEEE Conf. on Comp. Vision and Patt. Recog.*, pages 84–91, 1994.

[13] D. Ross, J. Lim, and M.-H. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *European Conf. on Computer Vision*, pages 470–482, 2004.