Adaptive Probabilistic Visual Tracking with Incremental Subspace Update

David Ross[1], Jongwoo Lim[2], and Ming-Hsuan Yang[3]

[1] University of Toronto, Toronto, ON M5S 3G4, Canada
[2] University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
[3] Honda Research Institute, Mountain View, CA 94041, USA
dross@cs.toronto.edu jlim1@uiuc.edu myang@honda-ri.com

Abstract. Visual tracking, in essence, deals with non-stationary data streams that change over time. While most existing algorithms are able to track objects well in controlled environments, they usually fail if there is a significant change in object appearance or surrounding illumination. The reason being that these visual tracking algorithms operate on the premise that the models of the objects being tracked are invariant to internal appearance change or external variation such as lighting or viewpoint. Consequently most tracking algorithms do not update the models once they are built or learned at the outset. In this paper, we present an adaptive probabilistic tracking algorithm that updates the models using an incremental update of eigenbasis. To track objects in two views, we use an effective probabilistic method for sampling affine motion parameters with priors and predicting its location with a maximum *a posteriori* estimate. Borne out by experiments, we demonstrate the proposed method is able to track objects well under large lighting, pose and scale variation with close to real-time performance.

1 Introduction

Visual tracking essentially deals with non-stationary data, both the object and the background, that change over time. Most existing algorithms are able to track objects, either previously viewed or not, in a short span of time and in a well controlled environment. However these algorithms usually fail to observe the object motion or have significant drifts after some period of time, either due to the drastic change of the object appearance or large lighting variation in the surroundings. Although such situations can be ameliorated with recourse to view-based appearance models [1] [2], adaptive color-based trackers [3] [4], contour-based trackers [5] [4], particle filters [5], 3D model based methods [6], optimization methods [1] [7], and background modeling [8], most algorithms typically operate on the premise that the target object models do not change drastically over time. Consequently these algorithms build or learn models of the objects first and then use them for tracking, without adapting the models to account for changes of the appearance of the object, e.g., large variation of pose or facial expression, or the surroundings, e.g., lighting variation. Such

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 470–482, 2004.

an approach, in our view, is prone to performance instability and needs to be addressed for building a robust tracker.

In this paper, we present an efficient and adaptive algorithm that incrementally updates the model of the object being tracked. Instead of using a simple contour to enclose an image region or color pixels to represent moving "stuff" [9] for target tracking, we use an eigenbasis to represent the "thing" being tracked. In addition to providing a compact representation of the model based on the reconstruction principle, the eigenbasis approach also renders a probabilistic interpretation and facilitates efficient computation. Furthermore, it has been shown that for a Lambertian object the set of all images taken under all lighting conditions forms a convex polyhedral cone in the image space [10], and this polyhedral cone can be approximated well by a low-dimensional linear subspace using an eigenbasis [11] [12].

Given an observed image, we track the object based on a maximum *a posteriori* estimate of location - the image region near the predicted position that can be best approximated by the current eigenbasis. We compute this estimate using samples of affine parameters, describing the frame to frame motion of the target, drawn from their prior distributions, and combining them with the likelihood of the observed image region under our model.

The remaining part of this paper is organized as follows. We first review the most relevant tracking work in the next section. The details of our tracking algorithm is presented in Section 3, followed by numerous experiments to demonstrate its robustness under large pose and lighting variation. We conclude this paper with remarks on possible extensions for future work.

2 Context and Previous Work

There is an abundance of visual tracking work in the literature, from a simple two-view template matching approach [13] to a 3D model-based algorithm [6]. These algorithms differ mainly in the representation scheme – ranging from color pixels, blobs, texture, features, image patches, templates, active contours, snakes, wavelets, eigenspace, to 3D geometric models – and in the prediction approach, such as correlation, sum of square distance, particle filter, Kalman filter, EM algorithm, Bayesian inference, statistical models, mixture models, and optimization formulations. A thorough discussion of this topic is beyond the scope of this paper, thus in this section we review only the most relevant object tracking work and focus on the algorithms that operate directly on gray scale images.

In [1] Black and Jepson advocated a view-based eigenbasis representation for object tracking and formulated the two-view matching process as an optimization problem. Black et al. later extended the eigentracking algorithm to a mixture model to account for changes in object appearance [14]. The major advantages of using an eigenbasis representation are that it allows the tracker to have the notion of the "thing" being tracked, and the tracking algorithm operates on the subspace constancy assumption as opposed to the brightness constancy assumption of optical flow estimation. One disadvantage of the abovementioned algorithms is the use of viewed-based representation. In other words, one needs to learn the eigenbasis of an object at each viewpoint before tracking, and these subspaces do not adapt over time. Furthermore, these algorithms need to solve a complex optimization problem [1], perhaps by using an iterative EM algorithm [14].

Brand proposed a method to track and build 3D models from video [15] [16] in which a set of feature points are manually selected and tracked, thereby obtaining 3D models based on the structure from motion principle. An incremental update of SVD, akin to [17], was introduced and improved to handle missing data. However, this method performs well only when the object is close to the camera with limited object motion.

Birchfield used a simple ellipse to enclose the region of interest, and integrated color and gradient information to track human heads [3]. This algorithm was further extended by Wu and Huang [4] in which they formulated the problem using a graphical model with a sequential Monte Carlo sampling method for estimating the state variables and thereby the object location. Their sampling method is largely based on the Condensation algorithm [5]. Although these methods perform well in constrained environments, the representation scheme is rather primitive, e.g., color and contour, in order to reduce the size of state space, and the tracker simply treats the target region as moving "stuff," paying little attention to what lies inside the contour. In other words, such trackers do not adapt to appearance change of the object and are likely to fail under large illumination change. Comaniciu and Meer presented the mean-shift algorithm for estimating how the mode of a density function changes over time [18], and then applied it to object tracking [19] using the histogram of color pixels. Due to the use of a simple pixel-based representation, it is not clear whether this algorithm will perform well under large illumination change.

A few attempts have been proposed to track objects under large illumination change [20] [6] [14]. These algorithms follow the same line of work of [21] and use a low dimensional linear subspace to approximate the space of all possible images of the object under different lighting conditions. Though they have demonstrated good empirical results, one needs to construct the basis images from a set of images acquired at fixed pose under different lighting conditions before tracking.

Most recently, some attention has been paid to the development of tracking methods that adapt the object models to changes in the object appearance or surroundings [22] [23] [2]. In [22], De La Torre et al. developed a tracking algorithm based on [1] in which they built a specific eigenbasis for the person being tracked by performing singular value decomposition (SVD) on a subset of training images which were most similar to the incoming test frames. Skin color was used to segment a face from the background and the affine motion parameters were estimated using a Kalman filter. Jepson et al. proposed the $\mathcal{WSL}$ model for learning adaptive appearance models in the context of object tracking [23]. They used the response of wavelet filters for object representation, and a mixture model to handle possible tracking scenarios. The weights in the mixture model are estimated using the EM algorithm and the affine motion parameters are computed using the stable features from wavelet filters. While their adaptive model is able to handle appearance and lighting change, the authors pointed out that it is possible for their model to learn the stable structure of the background if the background moves consistently with the foreground object over a period of time. Consequently, their model may drift from the target object and lose track of it.

Our approach bears some resemblance to the classic particle filter algorithms [5], but with a more informative representation through the use of an eigenbasis. On the other hand, our approach is significantly different from the eigentracking approach [1]. First, we constantly update the eigenbasis using an computationally efficient algorithm. Consequently, our tracker is able to follow objects under large lighting and pose variation without recourse *a priori* to the illumination cone algorithm [24] or view-based approaches [1] . Second, we use a sampling technique to predict the object location without solving computationally expensive complex optimization problems. Our current implementation in MATLAB runs about 8 frames per second on a standard computer, and can certainly be improved to operate in real time. Third, our sampling technique can be extended, akin to the Condensation algorithm [5], to predict the location based on sequential observations, incorporating multiple likely hypotheses. Finally, the adaptive eigenbasis approach facilitates object recognition, thereby solving tracking and recognition problem simultaneously.

3 Adaptive Probabilistic Tracking

We detail the proposed algorithm and contrast the differences between this work and prior art in this section. Possible extensions of our algorithm are also discussed in due context.

3.1 Probabilistic Model

The tracking algorithm we propose is cast as an inference problem in a probabilistic Markov model, similar to the Hidden Markov Model and Kalman Filter [25]. At each time step t we observe an image region F_t of the sequence, and the location of the target object, L_t, is treated as an unobserved state variable. The motion of the object from one frame to the next is modeled by a distribution $p(L_t|L_{t-1})$, indicating the probability of the object appearing at L_t, given that it was just at L_{t-1}. This distribution encodes our beliefs about where the object might be at time t, prior to observing the current image region. Given F_t, we model the likelihood that the object is located at L_t with the distribution $p(F_t|L_t)$. Using Bayes' rule to incorporate our observation with our prior belief, we conclude that the most probable *a posteriori* object location is at the maximum l_t^* of $p(L_t|F_t, L_{t-1}) \propto p(F_t|L_t)p(L_t|L_{t-1})$. A graphical depiction of this model is illustrated in Figure 1.

We represent L_t, the location of the object at time t, using the four parameters of a similarity transformation (x_t and y_t for translation in x and y, r_t for rotation, and s_t for scaling). This transformation warps the image, placing the target window - the object being tracking - in a rectangle centered at coordinates (0,0), with the appropriate width and height. This warping operates as a function of an image region F and the object location L, i.e., $w(F, L)$.

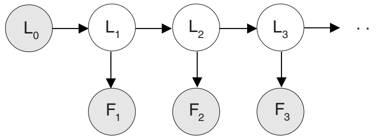

Fig. 1. Graphical model of the proposed tracking algorithm.

Our initial prior over locations assumes that each parameter is independently distributed, according to a normal distribution, around a predetermined location L_0. Specifically

$$p(L_1|L_0) = N(x_1; x_0, \sigma_x^2)N(y_1; y_0, \sigma_y^2)N(r_1; r_0, \sigma_r^2)N(s_1; s_0, \sigma_s^2) \tag{1}$$

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

Since our aim is to use an eigenbasis to model object appearance, we employ a probabilistic principal components distribution [26] (also known sensible PCA [27]) to model our image observation process. Given a location L_t, we assume the observed image region was generated by sampling an appearance of the object from the eigenbasis, and inserting it at L_t. Following Roweis [27], the probability of observing a datum z given the eigenbasis B and mean μ is $N(z; \mu, BB^T + \varepsilon I)$, where the εI term corresponds to the covariance of additive Gaussian noise present in the observation process. In the limit as $\varepsilon \to 0$, $N(z; \mu, BB^T + \varepsilon I)$ is proportional to the negative exponential of the squared distance between z and the linear subspace B, $\|(z - \mu) - BB^T(z - \mu)\|^2$.

3.2 Predicting Object Location

According to our probabilistic model, since L_t is never directly observed, full Bayesian inference would require us to compute the distribution $P(L_t| F_t, F_{t-1}, \ldots, F_t, L_0)$ at each time step. Unfortunately this distribution is infeasible to compute in closed form. Instead, we will approximate it using a normal distribution of the same form as our prior in Equation 1 around the maximum l_t^* of $p(L_t|F_t, l_{t-1}^*)$.

We can efficiently and effectively compute an approximation to l_t^* using a simple sampling method. Specifically, we begin by drawing a number of sample locations from our prior $p(L_t|l_{t-1}^*)$. For each sample l_s we compute its posterior probability $p_s = p(l_s|F_t, l_{t-1}^*)$. The posterior p_s is simply the likelihood of l_s under our probabilistic PCA distribution, times the probability with which l_s was sampled, disregarding the normalization factor which is constant across all samples. Finally we select the sample with the largest posterior to be our approximate l_t^*, i.e.,

$$l_t^* = \arg\max_{l_s} p(l_s|F_t, l_{t-1}^*) \tag{2}$$

This method has the nice property that a single parameter, namely the number of samples, can be used to control the trade-off between speed and tracking accuracy.

To allow for incremental updates to our object model, we specifically do not assume that the probability distribution of observations remains fixed over time. Rather, we use recent observations to update this distribution, albeit in a non-Bayesian fashion. Given an initial eigenbasis B_{t-1}, and a new appearance $w_{t-1} = w(F_{t-1}, l^*_{t-1})$ we compute a new basis B_t using the sequential Karhunen-Loeve (K-L)algorithm [17] (see next section). The new basis is used when calculating $p(F_t|L_t)$. It is also possible to perform an on-line update of the mean of the probabilistic PCA model.

The proposed sampling method is flexible and can be applied to localize targets in the first frame, though manual initialization or sophisticated object detection algorithms are applicable. By specifying a broad prior (perhaps uniform) over the entire image, and drawing enough samples, our tracker could locate the target by the maximum response using the current distribution and the initial eigenbasis.

Finally the parametric sampling method in the current implementation can be extended, akin to particle filters such as the Condensation algorithm [5], to integrate temporal information by a recursive Bayesian formulation, and allow multiple hypotheses by using a non-parametric distribution.

3.3 Incremental Update of Eigenbasis

Since visual tracking deals with time-varying data, and we use an eigenbasis for object representation, it is imperative to continually update the eigenbasis from the time-varying covariance matrix. This problem has been studied in the signal processing community, where several computationally efficient techniques have been proposed in the form of recursive algorithms [28]. In this paper, we use a variant of the efficient sequential Karhunen-Loeve algorithm to update the eigenbasis [17], which in turns is based on the classic R-SVD method [29].

Let $X = U\Sigma V^T$ be the SVD of a data $M \times P$ matrix X where each column vector is an observation (e.g., image). The R-SVD algorithm provides an efficient way to carry out the SVD of a larger matrix $X^* = (X|E)$, where E is a $M \times K$ matrix consisting of K additional observations (e.g., incoming images) as follows.

– Use an orthonormaliztion process (e.g., Gram-Schmidt algorithm) on $(U|E)$ to obtain an orthonormal matrix $U^{'} = (U|\tilde{E})$.
– Form the matrix $V' = \begin{pmatrix} V & 0 \\ 0 & I_K \end{pmatrix}$, where I_K is a K dimensional identity matrix.
– Let $\Sigma^{'} = U^{'T} X^* V^{'} = \begin{pmatrix} U^T \\ \tilde{E}^T \end{pmatrix} (X|E) \begin{pmatrix} V & 0 \\ 0 & I_K \end{pmatrix} = \begin{pmatrix} U^T XV & U^T E \\ \tilde{E}^T XV & \tilde{E}^T E \end{pmatrix} = \begin{pmatrix} \Sigma & U^T E \\ 0 & \tilde{E}^T E \end{pmatrix}$ since $\Sigma = U^T XV$ and $\tilde{E}^T XV = 0$. Notice that the K rightmost columns of $\Sigma^{'}$ are the new vectors (images), represented in the updated orthonormal basis spanned by the columns of $U^{'}$.
– Compute the SVD of $\Sigma^{'} = \tilde{U}\tilde{\Sigma}\tilde{V}^T$ and the SVD of X^* is

$$X^* = U^{'}(\tilde{U}\tilde{\Sigma}\tilde{V}^T)V^{'T} = (U^{'}\tilde{U})\tilde{\Sigma}(\tilde{V}^T V^{'T}) \quad (3)$$

By exploiting the orthonormal properties and block structure, the SVD computation of X^* can be efficiently carried by using the smaller matrices, $U^{'}$, $V^{'}$,

Σ' and the SVD of smaller matrix Σ'. The computational complexity analysis and details of the R-SVD algorithm are described in [29].

Based on the R-SVD method, the sequential Karhunen-Loeve algorithm further exploits the low dimensional subspace approximation and only retains a small number of eigenvectors as new data arrive. See [17] for details of an update strategy and the computational complexity analysis.

3.4 Proposed Tracking Algorithm

Our tracking algorithm is flexible in the sense that it can be carried out with or without an initial eigenbasis of the object. For the case where training images of the object are available and well cropped, an eigenbasis can be constructed which the proposed tracker can use at the onset of tracking. However situations arise where we do not have training images at our disposal. In such cases, the tracker can gradually construct and update an eigenbasis from incoming images if the object is localized in the first frame. An application of this technique is demonstrated experimentally in the following section.

Putting the inference, sampling and subspace update modules together, we obtain an adaptive probabilistic tracking algorithm as follows:

1. **(Optional) Construct an initial eigenbasis:** From a set of training images of the object, or similar objects, learn an initial eigenbasis. This includes the following steps: a) histogram-equalize all of the training images b) subtract the mean from the data c) compute the desired number principle components.
2. **Choose the initial location:** Given the tracking sequence, locate the object in the first frame. This can be done manually, or by using an automatic object detector. Alternatively, we can draw more samples to detect objects in the first frame if an eigenbasis of the object is available.
3. **Search possible locations:** Draw a number of samples from the prior distribution over locations. For each location, obtain the image region, histogram equalize it, subtract the mean learned during the training phase, and compute it's probability under the current eigenbasis.
4. **Predict the most likely location:** Select the location with the maximum *a posteriori* probability. Update the prior to be centered at this location.
5. **Update the eigenbasis:** Using the image window selected in 3, update the eigenbasis using the sequential Karhunen-Loeve algorithm. Note that this need not be performed for every frame. Instead, it is possible and possibly preferable to store the tracked image regions for a number of previous frames and perform a batch update.
6. Go to Step 3.

For most vision applications, it suffices to use a small number of eigenvectors (those with the largest eigenvalues). To achieve this, one may simply discard the unwanted eigenvectors from the initial basis (Step 1c), and from the updated basis after each invocation of sequential K-L (Step 5).

As a result of keeping only the top eigenvectors at each stage of the sequential SVD (equation 3), the samples seen at an earlier time are gradually forgotten.

None of the tracking experiments shown in the next section are improved by using an explicit forgetting factor, but it is likely that an explicit forgetting factor helps in certain situations.

In this work, the mean of the subspace is not updated. Any changes in the mean can be compensated for, without loss of modeling accuracy, by simply including additional eigenvector in the basis [30]. In fact, some studies have shown that uncentered PCA (in which the mean is not subtracted) performs as well as centered PCA [30]. Empirical results (see next section) show that our method with incremental update performs well under large variation in lighting, pose and deformations. Our future work will focus on developing a method to update the subspace with running means.

4 Experiments

We conducted numerous experiments to test whether the proposed tracking algorithm performs well in terms of following the object position and updating the appearance model. All the image sequences consist of 320×240 pixel grayscale videos, recorded at 30 frames/second and 256 gray-levels per pixel. As a baseline, we compared our algorithm with three other trackers. The first is an eigentracker using a fixed basis, which was was implemented by removing the sequential K-L updates from our tracker. The second is a simple template tracker that, at each time step, searches for the window most like the appearance of the object in the first frame of the sequence. The third one is a two-view tracker, which searches at each time step for the window most resembling the appearance in the preceding frame based on sum of square errors.

For all the experiments, the parameters of the prior distribution of affine parameters were, $\sigma_x = \sigma_y = 5$, $\sigma_r = \sigma_s = 0.1$. Typically we used between 100 and 500 samples, but good results can be obtained using as few as 64. For the experiments with face objects, we built an initial eigenbasis using a set of 2901 well-cropped 19×19 face images from the MIT CBCL data set[1]. We restricted this eigenbasis to the top 50 eigenvectors. For efficiency, an eigenbasis is updated every 5 frames using the sequential Karhunen-Loeve algorithm.

Figure 2 shows the empirical results where the tracked object is enclosed by a rectangle by our algorithm and the top 10 eigenvectors at each time instance are shown. The object location in the first frame is manually initialized and and the eigenbasis is built from the MIT face data set. Notice that in the first few frames, the eigenvectors resemble the generic object patterns (upper left panel). Since the eigenbasis is constantly updated with the incoming frames of the same object, the eigenbasis gradually shifts to capture the details of that target as shown by the eigenvectors in the following two panels (e.g., the eye glasses are visible in the first eigenvectors of the upper middle and upper right panels) though a few eigenvectors still contain high frequency noise. As this target changes its pose, the updated subspace also accounts for the variation in appearance (see the first three eigenvectors in upper middle panel). Finally with

[1] CBCL Face Database #1, MIT Center For Biological and Computation Learning, available at `http://www.ai.mit.edu/projects/cbcl`

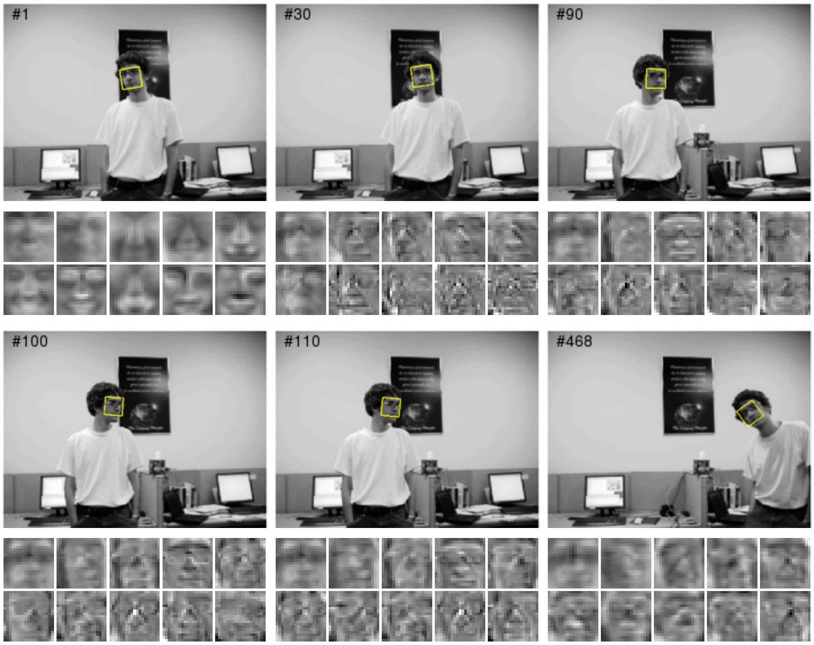

Fig. 2. Tracking an object undergoing a large pose variation (See video at `http://www.cs.toronto.edu/~dross/ivt/`).

enough samples, the subspace has been updated that capture more facial details of this individual (lower three panels). Notice that the proposed algorithm tracks the target undergoing large pose variation. On the other hand, the eigentracker with a fixed eigenbasis fails as this person changes pose. Though the eigentracker can be further improved with view-based subspaces, it is difficult to collect the training images for all possible pose variations. We note that all the other three trackers mentioned above fail to track the target object after a short period of time, especially when there is a large pose variation. The fixed template tracker fails during the first out-of-plane rotation of the subject's face. In this video, the two-view tracker quickly drifts away from the subject's face, instead tracking the side of his head.

Our algorithm is able to track and learn the eigenbasis of the target object under large appearance and lighting variation, as demonstrated in Figure 3. The target object is first manually located, and the initial eigenbasis is again learned from the MIT data set. Notice that the eigenbasis adapts from a generic eigenbasis to one that captures the details of this individual as shown evidently in the first few eigenvectors of all the panels. The third eigenvector of the upper middle panel captures the facial expression variation. Notice also that there is drastic lighting change that changes the facial appearance in the second, third, and fourth panels. Our tracker is still able to follow the target object in these situations. On the other hand, both the eigentracker and template tracker fail to

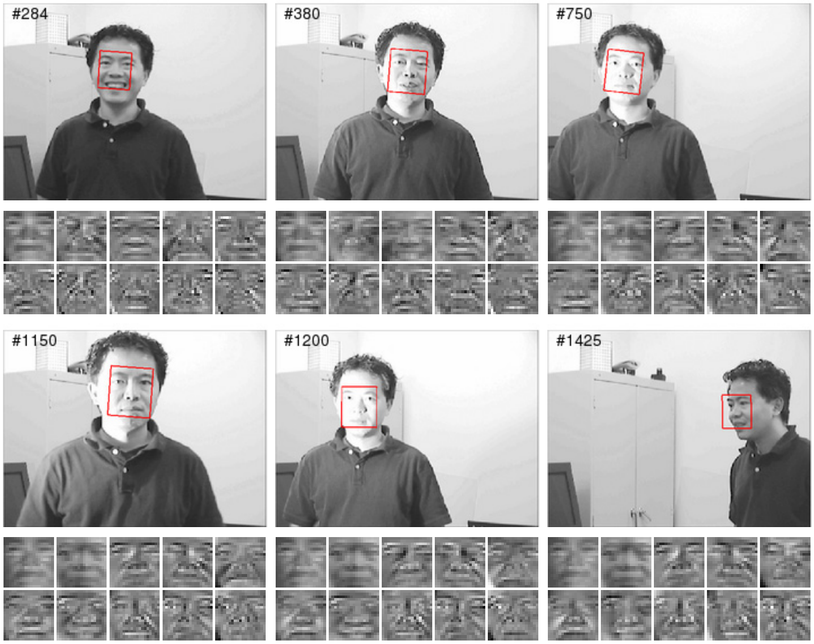

Fig. 3. Tracking an object undergoing a large appearance, pose, and lighting variation (See video at `http://www.cs.toronto.edu/~dross/ivt/`).

track the object when there is large lighting variation (starting from the middle panel). The two-view tracker suffers during the scale change in this sequence, drifting to track a small subregion of the face as the subject approaches the camera. Finally, the proposed algorithm is able to track the target object under a combination of lighting, expression and pose variation (lower right panel).

Figure 4 shows an example where we do not have an eigenbasis of the target object to begin with. In such situations, our algorithm is still able to gradually learn an eigenbasis and use that for tracking. The upper left panel shows the first few eigenvectors at the 6th frame of the sequence, immediately following the first eigenbasis update. Note that the first eigenvector captures some details of the target object with only a few frames, while the remaining few eigenvectors encode high frequency noise. As more frames become available, our method clearly learns an eigenbasis of the object, as shown in the first few eigenvectors of the top middle panel. As the object moves, our algorithm uses the learned eigenbasis for tracking and continues to update the appearance model. Notice that the tracker is able to follow the object well when it undergoes large pose and scale variation (See video at `http://www.cs.toronto.edu/~dross/ivt/`).

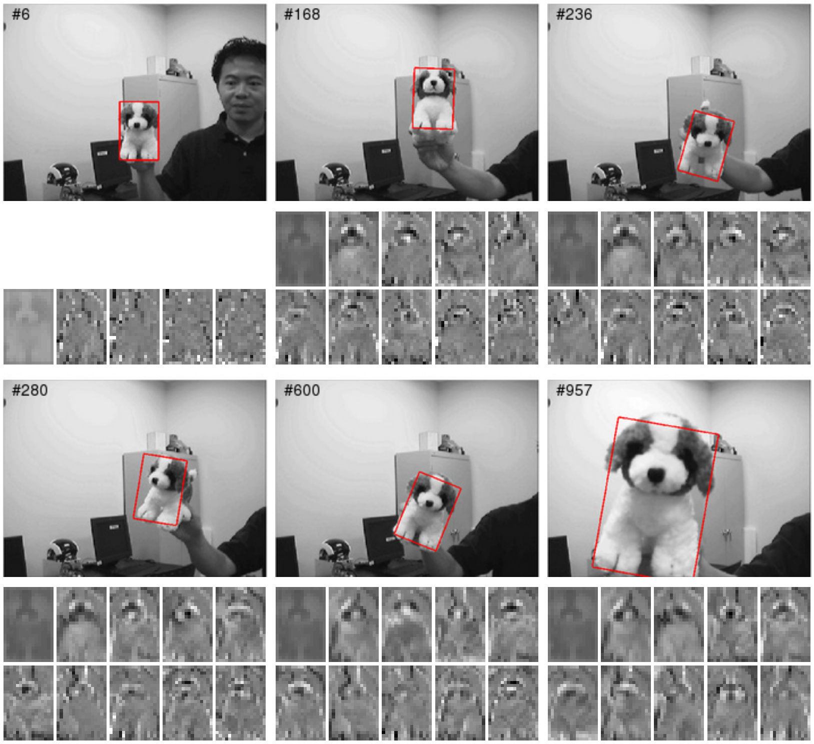

Fig. 4. Learning an eigenbasis while tracking the target object.

5 Concluding Remarks and Future Work

We have presented an adaptive probabilistic visual tracking algorithm that constantly updates its appearance model to account for the intrinsic (e.g. facial expression and pose) and extrinsic variation (e.g. lighting). To track an object in two views, we use an effective probabilistic method for sampling affine motion parameters with priors and predict its location with maximum a posteriori estimate. Through experiments, we demonstrated that the proposed method is able to track objects well in real-time under large lighting, pose and scale variation.

Though the proposed algorithm works well under large appearance, pose, lighting, and scale variation, our tracker sometimes drifts off by a few pixels before recovering in later frames. In addition, the current tracker does not handle occlusion well. This problem can be ameliorated by computing a mask before carrying out the eigen-decomposition [1] or using a method to deal with missing data [15], and will be addressed in our future work. A more selective adaption mechanism, e.g. [31], is also required to ensure the integrity of the newly arrived samples before updating the eigenbasis. Meanwhile, we plan to investigate better sampling strategy and extend our work to integrate temporal information. Furthermore, we plan to extend the current work to use eigenbases for constructing illumination cone for fixed pose, and for object recognition.

Acknowledgements. This work was carried out while the first two authors visited at Honda Research Institute in the summer of 2003. We appreciate the valuable comments and suggestions of the anonymous reviewers. We thank the MIT Center For Biological and Computation Learning for providing data used in our experiments (CBCL Face Database #1 - `http://www.ai.mit.edu/projects/cbcl`).

References

1. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using view-based representation. In Buxton, B., Cipolla, R., eds.: Proceedings of the Fourth European Conference on Computer Vision. LNCS 1064, Springer Verlag (1996) 329–342
2. Morency, L.P., Rahimi, A., Darrell, T.: Adaptive view-based appearance models. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2003) 803–810
3. Birchfield, S.: Elliptical head tracking using intensity gradient and color histograms. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1998) 232–37
4. Wu, Y., Huang, T.: A co-inference approach for robust visual tracking. In: Proceedings of the Eighth IEEE International Conference on Computer Vision. Volume 2. (2001) 26–33
5. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In Buxton, B., Cipolla, R., eds.: Proceedings of the Fourth European Conference on Computer Vision. LNCS 1064, Springer Verlag (1996) 343–356
6. La Cascia, M., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 322–336
7. Black, M.J., Fleet, D.J., Yacoob, Y.: Robustly estimating changes in image appearance. Computer Vision and Image Understanding **78** (2000) 8–31
8. Harville, M.: A framework for high-level feedback to adaptive, per-pixel mixture of Gaussian background models. In Heyden, A., Sparr, G., Nielsen, M., Johansen, P., eds.: Proceedings of the Seventh European Conference on Computer Vision. LNCS 2352, Springer Verlag (2002) 531–542
9. Adelson, E.H., Bergen, J.R.: The plenoptic function and the elements of early vision. In Landy, M., Movshon, J.A., eds.: Computational Models of Visual Processing. MIT Press (1991) 1–20
10. Belhumeur, P.N., Kriegman, D.J.: What is the set of images of an object under all possible illumination conditions. International Journal of Computer Vision **28** (1998) 1–16
11. Hallinan, P.: A low-dimensional representation of human faces for arbitrary lighting conditions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1994) 995–999
12. Basri, R., Jacobs, D.: Lambertian reflectance and linear subspaces. In: Proceedings of the Eighth IEEE International Conference on Computer Vision. Volume 2. (2001) 383–390
13. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of International Joint Conference on Artificial Intelligence. (1981) 674–679

14. Black, M.J., Fleet, D.J., Yacoob, Y.: A framework for modeling appearance change in image sequence. In: Proceedings of the Sixth IEEE International Conference on Computer Vision. (1998) 660–667
15. Brand, M.: Morphable 3D models from video. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2001) 315–322
16. Brand, M.: Incremental singular value decomposition of uncertain data with missing values. In Heyden, A., Sparr, G., Nielsen, M., Johansen, P., eds.: Proceedings of the Seventh European Conference on Computer Vision. LNCS 2350, Springer Verlag (2002) 707–720
17. Levy, A., Lindenbaum, M.: Sequential Karhunen-Loeve basis extraction and its application to images. IEEE Transactions on Image Processing **9** (2000) 1371–1374
18. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 603–619
19. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 564–577
20. Hager, G., Belhumeur, P.: Real-time tracking of image regions with changes in geometry and illumination. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1996) 403–410
21. Shashua, A.: Geometry and Photometry in 3D Visual Recognition. PhD thesis, Massachusetts Institute of Technology (1992)
22. De la Torre, F., Gong, S., McKenna, S.J.: View-based adaptive affine tracking. In Burkhardt, H., Neumann, B., eds.: Proceedings of the Fifth European Conference on Computer Vision. LNCS 1406, Springer Verlag (1998) 828–842
23. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2001) 415–422
24. Hager, G.D., Belhumeur, P.N.: Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1025–1039
25. Jordan, M.I., ed.: Learning in Graphical Models. MIT Press (1999)
26. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society, Series B **61** (1999) 611–622
27. Roweis, S.: EM algorithms for PCA and SPCA. In Jordan, M.I., Kearns, M.J., Solla, S.A., eds.: Advances in Neural Information Processing Systems 10, MIT Press (1997) 626–632
28. Champagne, B., Liu, Q.G.: Plane rotation-based EVD updating schemes for efficient subspace tracking. IEEE Transactions on Signal Processing **46** (1998) 1886–1900
29. Golub, G.H., Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press (1996)
30. Jolliffe, I.T.: Principal Component Analysis. Springer-Verlag (2002)
31. Vermaak, J., Pérez, P., Gangnet, M., Blake, A.: A framework for high-level feedback to adaptive, per-pixel mixture of Gaussian background models. In Heyden, A., Sparr, G., Nielsen, M., Johansen, P., eds.: Proceedings of the Seventh European Conference on Computer Vision. LNCS 2350, Springer Verlag (2002) 645–660