Adaptive Discriminative Generative Model for Object Tracking

Ruei-Sung Lin[1] **Ming-Hsuan Yang**[2] **Stephen E. Levinson**[3]

Abstract. This paper presents an adaptive visual learning algorithm for object tracking. We formulate a novel discriminative generative framework that generalizes the conventional Fisher Linear Discriminant algorithm with a generative model and renders a proper probabilistic interpretation. Within the context of object tracking, we aim to find a discriminative generative model that best separates the target class from the background. We present a computationally efficient algorithm to constantly update this discriminative model as time progresses. While most tracking algorithms operate on the premise that the object appearance or environment lighting condition does not significantly change as time progresses, our method adapts the discriminative generative model to reflect appearance variation of the target and background, thereby facilitating the tracking task in different situations. Numerous experiments show that our method is able to learn a discriminative generative model for tracking target objects undergoing large pose and lighting changes.

1 INTRODUCTION

Tracking moving objects is an important and essential component of visual perception, and has been an active research topic in computer vision community for decades [8]. Object tracking can be formulated as a continuous state estimation problem where the unobservable states encode the locations or motion parameters of the target objects, and the task is to infer the unobservable states from the observed images over time. At each time step, the tracker first predicts a few possible locations (i.e., hypotheses) of the target in the next frame based on its prior and current knowledge. The prior knowledge includes its previous observations and estimated state transitions. Among these possible locations, the tracker then determines the most likely new location of the object based on the new observation. An attractive and effective prediction mechanism is based Monte Carlo sampling in which the state dynamics (i.e., transition) can be learned with a Kalman filter or simply modeled as a Gaussian distribution. Such a formulation indicates that the performance of a tracker is largely based on a good observation model for validating all hypotheses. Indeed, learning a robust observation model has been the focus of most recent object tracking research within this framework, and is also the focus of this paper.

Most of the existing approaches utilize static observation models and construct them before tracking processes start. To account for all possible variation in a static observation model, it is imperative to collect a large set of training examples with the hope that it covers all possible representative variations of the object's appearance. However, it is well known that the appearance of an object varies significantly under different illumination, viewing angle, and self deformation. It is a daunting, if not impossible, task to collect a training set covering all possible cases. An alternative approach is to develop an adaptive method that contains a number of trackers that track different features or parts of the target object [3]. Therefore, even though each tracker may fail under certain circumstances, it is unlikely all of them fail at the same time. The tracking method then adaptively selects the trackers that are robust at current situation to perform the validation process. Although this approach improves the flexibility and robustness of a tracking method, each tracker has a static observation model and has to be trained beforehand and severely restricts its application domains. There are many cases, e.g., robotics applications, where the tracker is expected to track a previously unseen target once it is detected. To the best of our knowledge, considerably less attention is paid to adaptive model to account for appearance variation of a target object (e.g., pose, deformation) or environment changes (e.g., lighting conditions and viewing angles) as tracking task progresses.

One formulation is to learn a model for determining the probability of the observed image region of a predicted location being generated from the class of the target or the class of background. That is, we can formulate a binary classification problem and develop a discriminative model to distinguish observations from the target class and the background class. While conventional discriminative classifiers simply predict the class label of each test sample, a good model within the abovementioned tracking framework needs to select the most likely sample that belongs to target object class from a set of samples (or hypotheses). In other words, an observation model needs a classifier with proper probabilistic interpretation.

In this paper, we propose an object tracking algorithm that constantly updates its observation model for validating hypotheses. Our method takes a discriminative generative formulation, which not only facilitates the validation process but also allows our algorithm to be easily incorporated to other probabilistic visual tracking framework. We estimate a discriminative generative model to best separate the target object class and the background class. This is formulated as an optimization problem and we show that it is a direct generalization of the conventional Fisher Linear Discriminant algorithm with proper probabilistic interpretation. In this regard, our method can be regarded as a hybrid approach that combines a generative model with discriminative analysis. Our experimental results shows that our algorithm can reliably track moving objects whose appearance changes under different poses, illumination, and self deformation.

The rest of this paper is organized as follows. Section 2 provides a brief review of the probabilistic framework for object tracking. Section 3 explains our discriminative generative model, which is the focus of this paper. We first present our model in batch learning mode. Next, we describe our generative model that is based on probabilis-

[1] University of Illinois at Urbana-Champaign, USA, email: rlin1@uiuc.edu
[2] Honda Research Institute, USA, email: myang@honda-ri.com
[3] University of Illinois at Urbana-Champaign, USA, email: sel@ifp.uiuc.edu

tic principle component analysis, and detail our approach to perform discriminative generative analysis with this generative model. This is followed by a discussion on how our discriminative generative model can be updated on line. Our tracking algorithm is summarized in Section 4 and experiments are presented in Section 5. Finally, we conclude this paper with comments and remarks on future work.

2 PROBABILISTIC TRACKING ALGORITHM

We formulate the object tracking problem as a state estimation problem in a way similar to [5] [9]. Denote o_t as an image region observed at time t and $O_t = \{o_1, \ldots, o_t\}$ is a set of image regions observed from the beginning to time t. An object tracking problem is a process to infer state s_t from observation O_t, where state s_t contains a set of parameters referring to the tracked object's 2-D position, orientation, and scale in image o_t. Assuming a Markovian state transition, this state estimation can be formulated as a recursive equation:

$$p(s_t|O_t) = kp(o_t|s_t) \int p(s_t|s_{t-1})p(s_{t-1}|O_{t-1})ds_{t-1} \quad (1)$$

where k is a constant, and $p(o_t|s_t)$ and $p(s_t|s_{t-1})$ correspond to the observation model and dynamic model, respectively.

In (1), $p(s_{t-1}|O_{t-1})$ is the state estimation given all the prior observations up to time $t-1$, and $p(o_t|s_t)$ is the likelihood that observing image o_t at state s_t. Combining these two together, the posterior estimation $p(s_t|O_t)$ can be computed efficiently. For object tracking, an ideal distribution of $p(s_t|O_t)$ should peak at o_t, i.e., s_t matching the observed object's location o_t. While the integral in (1) predicts the regions where object is likely to appear given all the prior observations, the observation model $p(o_t|s_t)$ determines the most likely state that matches the observation at time t.

In our formulation, $p(o_t|s_t)$ measures the probability of observing o_t as a sample being generated by the tracked object class. Note that O_t is an image sequence and if the images are acquired at high frame rate, it is expected that the difference between o_t and o_{t-1} is small though object's appearance might vary according to different of viewing angles, illuminations, and possible self-deformation. Instead of adopting a complex static model to learn $p(o_t|s_t)$ for all possible o_t, a simpler model can be taken on for the same task by adapting this model to account for the object appearance change. In addition, since video frames o_t and o_{t-1} are most likely similar and computing $p(o_t|s_t)$ depends on $p(o_{t-1}|s_{t-1})$, the prior information $p(o_{t-1}|s_{t-1})$ can be used to enhance the distinctiveness between the object and its background in $p(o_t|s_t)$.

The idea of using an adaptive observation model for object tracking and then applying discriminative analysis to enhance the validation performance is the focus of the rest the paper. The observation model we use is based on probabilistic principle component analysis (PPCA) [10]. Object Tracking using PCA models have been well exploited in the computer vision community [2]. Nevertheless, most existing tracking methods do not update the observation models as time progresses. In this paper, we follow the work by Tipping and Bishop [10] and propose an adaptive observation model based on PCA within a formal probabilistic framework. Our result is a generalization of conventional Fisher Linear Discriminant with rigid probabilistic interpretation.

3 A DISCRIMINATIVE GENERATIVE OBSERVATION MODEL

In this work, we track the object based on its observed appearances in the videos, i.e. o_t. Since the size of image region o_t might change according to different s_t, we first convert o_t to a standard size and use it for tracking. In the following, we denote y_t as the standardized appearance vector of o_t.

The dimensionality of the appearance vector y_t is usually high. In our experiments, the standard fixed image size is a 19×19 rectangular image and thus y_t is a 361-dimensional vector. We thus model the appearance vector with a graphical model of low-dimensional latent variables.

3.1 A Generative Model with Latent Variables

A latent model relates a n-dimensional appearance vector y to a m-dimensional vector of latent variables x:

$$y = Wx + \mu + \epsilon \quad (2)$$

where W is a $n \times m$ projection matrix associating y and x, μ is the mean of y, and ϵ is an additive noise. As commonly assumed in factor analysis [1] and graphical models [6], the latent variables x are independent with unit variance, $x \sim \mathcal{N}(0, I_m)$, where I_m is an m-dimensional identity matrix, and ϵ is a zero mean Gaussian noise, $\epsilon \sim \mathcal{N}(0, \sigma^2 I_n)$ where I_n is an n-dimensional identity matrix. Since x and ϵ are both Gaussians, it follows that y is also a Gaussian distribution, $y \sim \mathcal{N}(\mu, C)$, where $C = WW^T + \sigma^2 I_n$. Together with (2), we get a generative observation model:

$$p(o_t|s_t) = p(y_t|W, \mu, \epsilon) \sim \mathcal{N}(y_t|\mu, WW^T + \sigma^2 I_n) \quad (3)$$

This latent variable model follows the form of probabilistic principle component analysis, and its parameters can be estimated from a set of examples [10]. Given a set of appearance samples $Y = \{y_1, \ldots, y_N\}$, the covariance matrix of Y is denoted as $S = \frac{1}{N}\sum(y-\mu)(y-\mu)^T$. Let $\{\lambda_i | i = 1, \ldots, N\}$ be the eigenvalues of S arranged in descending order, i.e. $\lambda_i \geq \lambda_j$ if $i < j$. Also, define diagonal matrix $\Sigma_m = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ and U_m the eigenvectors that corresponds to the eigenvalues in Σ_m. Tipping and Bishop show that the the maximum likelihood estimation of μ, W and ϵ [10] as follows:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad (4)$$

$$\hat{W} = U_m(\Sigma_m - \sigma^2 I_m)^{1/2} R \quad (5)$$

$$\hat{\sigma}^2 = \frac{1}{n-m}\sum_{i=m+1}^{n} \lambda_i \quad (6)$$

where U_m is a matrix of m column eigenvectors of S with corresponding largest m eigenvalues in the $m \times m$ diagonal matrix Σ_m, and R is an arbitrary $m \times m$ orthogonal rotation matrix.

To model the possible appearance variation of a target object (due to pose, illumination and view angle change), one could resort to mixtures of PCA for the task. However, it is not only time consuming to estimate the model parameters but also leads to other serious questions such as the number of components as well as under-fitting or over-fitting. On the other hand, at any given time a linear PCA model suffices to model gradual appearance variation if the model is constantly updated. In this paper, we use a single PCA, and dynamically adapt the model parameters W, μ, and σ^2.

3.1.1 Inference with Probabilistic PCA

Once the model parameters are known, we can infer the likelihood of a vector y being a sample of this generative appearance model. From (6), the log likelihood is computed by

$$\log p(y|W,\mu,\sigma^2) = -\frac{1}{2}\left(N\log 2\pi + \log|C| + \overline{y}^T C^{-1}\overline{y}\right) \quad (7)$$

where $\overline{y} = y - \mu$. Neglecting the constant terms, the likelihood is determined by $\overline{y}^T C^{-1}\overline{y}$. Together with $C = WW^T + \sigma^2 I_n$ and (6), it follows that

$$\overline{y}^T C^{-1}\overline{y} = \overline{y}^T U_m \Sigma_m^{-1} U_m^T \overline{y} + \frac{1}{\sigma^2}\overline{y}^T (I_n - U_m U_m^T)\overline{y} \quad (8)$$

Here $\overline{y}^T U_m \Sigma_m^{-1} U_m^T \overline{y}$ is the Mahalanobis distance of y in the subspace spanned by U_m, and $\overline{y}^T(I_n - U_m U_m^T)\overline{y}$ is the shortest distance from y to this subspace spanned by U_m. Usually σ is set to a small value, and consequently the likelihood will be determined solely by the distance to the subspace. However, the choice of σ is not trivial. From (8), if the σ is set much smaller than the actual value, the distance to the subspace will be over-emphasized and ignore the contribution of Mahalanobis distance, thereby rendering an inaccurate estimate. The choice of σ is even more critical in situations where the appearance change dynamically and require σ should be adjusted accordingly. This topic will be further examined in the following section.

3.1.2 Online Learning of Probabilistic PCA

Unlike the analysis in the previous section where model parameters are estimated based on a fixed set of training examples, our generative model has to learn and update its parameters online. Starting with a single example (the appearance of the tracked object in the first video frame), our generative model constantly updates its parameters as new observations arrive.

The equations for updating parameters are derived from (6). The update of U_m and Σ_m is complicated since it involves the computations of eigenvalues and eigenvectors. Here we use a forgetting factor γ to put more weights on the more recent data. Denote the newly arrived samples at time t as $Y = (y^1, \ldots, y^M)$, and assume the mean μ is fixed, U_m^t and Σ_m^t can be obtained by performing singular value decomposition (SVD) on

$$[\sqrt{\gamma}U_{m,t-1}(\Sigma_{m,t-1})^{1/2} | \sqrt{\frac{(1-\gamma)}{M}}\overline{Y}] \quad (9)$$

where $\overline{Y} = [y^1 - \mu, \ldots, y^M - \mu]$. $\Sigma_{m,t}^{1/2}$ and $U_{m,t}$ will contain the m-largest singular values and corresponding singular vectors respectively at time t. This update procedure can be efficiently implemented using an incremental SVD algorithm, e.g., [7].

If the mean μ constantly changes, the above update can not be applied. We recently propose a method to compute SVD with correct updated mean in which $\Sigma_{m,t}^{1/2}$ and $U_{m,t}$ can be obtained by computing SVD on

$$\left[\sqrt{\gamma}U_{m,t-1}(\Sigma_{m,t-1})^{1/2} \left| \sqrt{\frac{(1-\gamma)}{M}}\overline{Y} \right| \sqrt{(1-\gamma)\gamma}(\mu_{t-1} - \mu_Y)\right] \quad (10)$$

where $\overline{Y} = [y^1 - \mu_Y, \ldots, y^M - \mu_Y]$ and $\mu_Y = \frac{1}{M}\sum_{i=1}^M y^i$. This formulation is similar to the SVD computation with the fixed mean case, and the same incremental SVD algorithm can be used to compute $\Sigma_{m,t}^{1/2}$ and $U_{m,t}$ with an extra term shown in (10).

Computing and updating σ is more difficult than the form in (10). In the previous section, we show that an inaccurate value of σ will severely affect the likelihood estimation. In order to have a accurate estimation of σ based on (6), a large set of training examples is usually required. Our generative model starts with a single example and gradually adapts the model parameters. If we update σ based on (6) we will start with a very small value of σ and the algorithm could quickly lose track of the target because of incorrect likelihood estimate. Since the training examples are not permanently stored in memory, λ_i in (6) and consequently σ may not be accurately updated if the number of drawn samples is in sufficient. These constraints lead us to develop a method that adaptive adjusts σ according to the newly arrived samples, which will be explained in the next section.

3.2 Discriminative Generative Model

As is observed in Section 2, the object's appearance at o_{t-1} and o_t does not change much. Therefore, we can use the observation at o_{t-1} to boost the likelihood measurement in o_t. That is, we draw a set samples (i.e., image patches) parameterized by $\{s_{t-1}^i | i = 1,...,k\}$ in o_{t-1} that have large $p(o_{t-1}|s_{t-1}^i)$, but the low posterior $p(s_{t-1}^i|O_{t-1})$. These are treated as the negative samples (i.e., samples that are not generated from the class of the target object) that the generative model is likely to confuse at O_t.

Given a set samples $Y' = \{y^1, \ldots, y^k\}$ where y^i is the appearance vector collected in o_{t-1} based on state parameter s_{t-1}^i, we want to find a linear projection V^* that projects Y' onto a subspace such that the likelihood estimation of Y' in the subspace is minimized. Let V be a $p \times n$ matrix and since $p(y|W,\mu,\sigma)$ is a Gaussian, $p(Vy|V,W,\mu,\sigma) \sim \mathcal{N}(V\mu, VCV^T)$ is a also Gaussian. The log likelihood can be defined as:

$$\begin{aligned}&\log p(VY'|V,W,\mu,\sigma) = \\ &-\tfrac{k}{2}\left(p\log(2\pi) + \log|VCV^T| + tr((VCV^T)^{-1}VS'V^T)\right)\end{aligned} \quad (11)$$

where $S' = \frac{1}{k}\sum_{i=1}^k (y^i - \mu)(y^i - \mu)^T$.

To facilitate the following analysis we first assume V project Y' to a 1-D space, i.e. $V = v^T$. Note that $v^t Cv$ is the variance of the object samples in the projected space, and we need to impose a constraint $v^t Cv = 1$ to ensure that the minimum likelihood solution of v doesn't increase the variance in the projected space. Let and $v^t Cv = 1$, the optimization problem becomes

$$\begin{aligned} v^* &= \arg\max_{\{v|v^T Cv=1\}} v^T S' v \\ &= \arg\max_v \tfrac{v^T S' v}{v^T C v}\end{aligned} \quad (12)$$

Therefore, we obtain an equation exactly like Fisher discriminant analysis for a binary class problem. As can be seen in (12), v is a projection that keep the object's sample in the projected space close to the center (with variance $v^T Cv = 1$), while keeping negative samples in Y' away from μ in the projective space. The optimal value of v is the generalized eigenvector of S' and C that corresponds to largest eigenvalue. In a general case, it follows that

$$V^* = \arg\max_V \frac{|VS'V^T|}{|VCV^T|} \quad (13)$$

where V^* can be obtained by solving a generalized eigenvalue problem of S' and C. By projecting observation samples onto a low-dimensional subspace, we enhances the discriminative power of the generative model. In the meanwhile, we reduce the computation time for the inference processes, which is also a critical improvement for real time applications like object tracking.

3.2.1 Online Update of Discriminative Analysis

The computation of the projection matrix V depends on matrices C and S'. In section 3.1.1, we have shown the procedures to update C. The procedures can be used to update S'. Let $\mu_{Y'} = \frac{1}{k}\sum_{i=1}^{k} y^i$ and $S_{Y'} = \frac{1}{k}\sum_{i=1}^{k}(y^i - \mu_{Y'})(y^i - \mu_{Y'})^T$,

$$\begin{aligned} S' &= \tfrac{1}{k}\sum_{i=1}^{k}(y^i-\mu)(y^i-\mu)^T \\ &= S_y + (\mu - \mu_{Y'})(\mu - \mu_{Y'})^T \end{aligned} \tag{14}$$

In other words, we can use the same procedures to update C and S_y, and then use (14) to compute S'.

Given S' and C, computing V is a generalized eigenvalue problem. If we decompose $S' = A^T A$ and $C = B^T B$, then we can find V more efficiently using generalized singular value decomposition. Denote U_y and Σ_y as the matrices we used to track S_y, it follows that by letting $A = [U_y \Sigma_y^{1/2} \mid (\mu - \mu_y)]^T$ and $B = [U_m \Sigma_m^{1/2} \, |\sigma^2 I]^T$, $S' = A^T A$ and $C = B^T B$.

As is detailed in [4] , V can be computed by first performing a QR factorization:

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} Q_A \\ Q_B \end{bmatrix} R \tag{15}$$

and then computing the singular value decomposition of Q_A

$$Q_A = U_A D_A V_A^T \tag{16}$$

and, then $V = R^{-1} V_A$. When rank(A) is small, the computation of V is fast. See also [4] for more detail on solving generalized SVD problem.

4 PROPOSED TRACKING ALGORITHM

In this section, we describe the proposed tracking algorithm in greater detail and demonstrate how the abovementioned learning and inference algorithms are incorporated. Our algorithm localizes the tracked object in each video frame using a rectangular window. A state s is a 5-tuple vector, $s = (x, y, \theta, w, h)$, that parameterizes the windows position (x, y), orientation (θ) and width and height (w, h). The proposed algorithm is based on maximum likelihood estimate (i.e., the most probable location of the object) given all the observations up to that time instance, $s_t^* = \arg\max_{s_t} p(s_t|O_t)$.

We assume that state transition is a Gaussian distribution, i.e.,

$$p(s_t|s_{t-1}) \sim \mathcal{N}(s_{t-1}, \Sigma_s)$$

where Σ_s is a diagonal matrix. That is, we assume the parameters of a state variable are independent. According to this distribution, the tracker then draws N samples $S_t = \{c_1, \ldots, c_N\}$ which represent the possible locations of the target. Denote y_t^i as the appearance vector of o_t, and $Y_t = \{y_t^1, \ldots, y_t^N\}$ as a set of appearance vectors that corresponds to the set of state vectors S_t. The posterior probability that the tracked object is at c_i in video frame o_t is then defined as

$$p(s_t = c_i|O_t) = \kappa p(y_t^i|V, W, \mu, \sigma) p(s_t = c_i|s_{t-1}^*)$$

where κ is a constant. Therefore, $s_t^* = \arg\max_{c_i \in S_t} p(s_t = c_i|O_t)$.

Once s_t^* is determined, the corresponding observation y_t^* will be a new example to update W and μ. Appearance vector y_t^i that produces large value in $p(y_t^i|V, W, \mu, \sigma)$ but whose corresponding state parameters c_i are away from s_t^* will be used as new examples to update V.

Our tracking assumes o_1 and s_1^* are given (through object detection) and thus obtains the first appearance vector y_1 which in turns is used an the initial value of μ, but V and W are unknown at the outset. When V and W are not available, our tracking algorithm is based on template matching (with μ being the template). The matrix W is computed after a small number of appearance vectors are observed, With W, we can then start to compute and update V accordingly.

As mentioned in the Section 3.1.1, it is difficult to obtain an accurate estimate of σ. In our tracking the system, we adaptively adjust σ according to Σ_m in W. We set σ be a fixed fraction of the smallest eigenvalues in Σ_m This will ensure the distance measurement in 8 will not be over-emphasized on either the Mahalanobis distance in the subspace or the distance to the subspace.

5 EXPERIMENTAL RESULTS

Figure 1. A target undergoes with appearance deformation and large lighting variation. The bottom row of each panel shows the samples selected by our method that best separates the target from background classes. These samples are then used to update the observation model of our method, and thereby facilitates object tracking in an ever-changing environment.

We tested the proposed algorithm with face tracking experiments. To examine whether our model is able to adapt and track faces in the dynamically changing environment, we design the testing videos to have appearance deformation, large illumination change, and large pose variations.

All the image sequences consist of 320×240 pixel grayscale videos, recorded at 30 frames/second and 256 gray-levels per pixel. The forgetting term is empirically selected as 0.85, and the batch size for update is set to 5 as a trade-off of computational efficiency as well as effectiveness of modeling appearance change due to fast motion.

Figures 1 and 2 show snapshots of the tracking results. There are two rows of images below each video frame. The first row shows the sampled images in the current frame that have the largest likelihoods of being the target locations according our discriminative generative model. The second row shows the sample images in the current video frame that are selected online for updating the discriminative generative model.

The first four video frames in Figure 1 show the tracked object undergoes a series of appearance deformation, and our model is capable of adapting to these changes and correctly locates the tracked object in these frames. The rest four video frames in Figure 1 demonstrate our model adapts and keeps tracks the objects when there is a large illumination and shape variation.

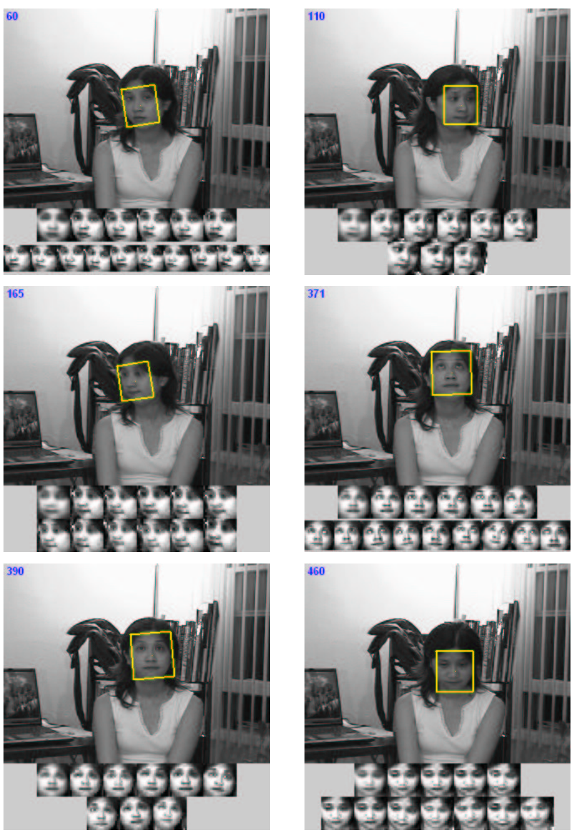

Figure 2. A target undergoes large pose variation. The bottom row of each panel shows the samples selected by our method that best separates the target from background classes. These samples are then used to update the observation model of our method, and thereby facilitates object tracking in an ever-changing environment.

The results in Figure 2 show that our model is able to locate the object accurately though there the object appearance undergoes large and quick pose variation.

We have also tested these two sequences with conventional view-based eigentracker [2] or template-based method. The empirical results show that such methods do not perform well as it does not adapt as the appearance change.

6 CONCLUSION

We have presented a discriminative generative framework that generalizes the conventional Fisher Linear Discriminant algorithm with a generative model and renders a proper probabilistic interpretation. For object tracking, we aim to find a discriminative generative model that best separates the target class from the background. With a computationally efficient algorithm that constantly update this discriminative model as time progresses, our method adapts the discriminative generative model to account for appearance variation of the target and background, thereby facilitating the tracking task in different situations. Our experiments show that the proposed model is able to learn a discriminative generative model for tracking target objects undergoing large pose and lighting changes. We also plan to apply the proposed method to other problems that deal with nonstationary data stream in our future work.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work is supported in part by Honda Research Institute.

REFERENCES

[1] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, Wiley, New York, 1984.

[2] M. J. Black and A. D. Jepson, 'Eigentracking: Robust matching and tracking of articulated objects using view-based representation', in *Proceedings of the Fourth European Conference on Computer Vision*, eds., B. Buxton and R. Cipolla, LNCS 1064, pp. 329–342. Springer Verlag, (1996).

[3] R. T. Collins and Y. Liu, 'On-line selection of discriminative tracking features', in *Proceedings of the Ninth IEEE International Conference on Computer Vision*, volume 1, pp. 346–352, (2003).

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1996.

[5] M. Isard and A. Blake, 'Contour tracking by stochastic propagation of conditional density', in *Proceedings of the Fourth European Conference on Computer Vision*, eds., B. Buxton and R. Cipolla, LNCS 1064, pp. 343–356. Springer Verlag, (1996).

[6] *Learning in Graphical Models*, ed., Michael I. Jordan, MIT Press, 1999.

[7] A. Levy and M. Lindenbaum, 'Sequential Karhunen-Loeve basis extraction and its application to images', *IEEE Transactions on Image Processing*, **9**(8), 1371–1374, (2000).

[8] B. D. Lucas and T. Kanade, 'An iterative image registration technique with an application to stereo vision', in *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 674–679, (1981).

[9] D. Ross, J. Lim, and M.-H. Yang, 'Adaptive probabilistic visual tracking with incremental subspace update', in *Proceedings of the Eighth European Conference on Computer Vision*, eds., Tomás Pajdla and Jiři Matas, LNCS 3022, pp. 470–482. Springer Verlag, (2004).

[10] M. E. Tipping and C. M. Bishop, 'Probabilistic principal component analysis', *Journal of the Royal Statistical Society, Series B*, **61**(3), 611–622, (1999).