Robust Appearance-based Tracking using a sparse Bayesian classifier

Shu-Fai Wong
Department of Engineering
University of Cambridge
sfw26@eng.cam.ac.uk

Kwan-Yee Kenneth Wong
Computer Science Department
Hong Kong University
kykwong@cs.hku.hk

Roberto Cipolla
Department of Engineering
University of Cambridge
cipolla@eng.cam.ac.uk

Abstract

An appearance-based approach to track an object that may undergo appearance change is proposed. Unlike recent methods that store a detailed representation of object's appearance, this method allows an appearance feature with a reduced dimension to be used. Through the use of a sparse Bayesian classifier, high classification and detection accuracy can be maintained even if a reduced feature vector is used. In addition, the classifier allows online-training which enables online-updating of the original classification model and provides better adaptability. Experiments show that the method can be used to track targets undergo appearance change due to the change in view-point, facial expression and lighting direction.

1   Introduction

Useful vision-based applications such as video surveillance system usually involve solving visual tracking problem where a moving object is located through a video sequence. This problem can be difficult because the appearance of the moving object and the background may change from frame to frame. These variations, however, are common in most application situations where lighting condition and camera position are not under control.

Recently, an *adaptive* tracking approach (e.g. [4, 10, 5, 1, 6]) has been proposed to reduce the vulnerability of a *template-based* tracking system (e.g. [2]) to changes in objects' appearance and background. By *continuously updating* the *appearance* model of the object being tracked, the object can be located reliably in a given video sequence with reference to the 'latest' model (which is usually an image patch as in [4, 10, 5]). Although this adaptive approach can give a reliable tracking result when capturing condition changes (e.g. there is a change in view-point), this approach usually suffers from high computational complexity. This is mainly because methods using this approach either work on an image-based representation of the target or adopt a complicated updating process. Besides, this approach may adapt to new features while forgetting some original features and this implies such an adaptive approach may fail to locate the object when there is a sudden occlusion or a sudden change in view-point.

Unlike other recent methods that use a high-resolution image patch as a matching template, this work allows the use of a feature [1] of a lower dimensional size. The reduction in appearance information due to a lower dimensional size is compensated by the adoption of a powerful classifier for the matching process. The classifier used in this work is a sparse Bayesian classifier (or a relevance vector machine) and it provides a mean to make a matching decision based on a set of prototypical samples instead of a single template. Incremental learning approach, which evaluates new samples and updates classification model (i.e. the set of prototypes) on-the-fly, is adopted to increase the adaptability and efficiency of the proposed tracking system.

The proposed system performs better than previous work in three ways. First, the low dimensional size of the appearance feature and the sparsity of the classifier provides an *efficient* and *fast* matching result. Second, the Bayesian classification scheme used is more *robust* and *reliable* than simple matching schemes based on a single template (which is continuously updated), especially when there is a sudden change and recovery of appearance. Third, the incremental learning scheme enables the classifier to *adapt* to long lasting appearance change so that it can be applied to a wider range of environment. It is particularly useful in building a general object (e.g. face) tracking system which may need to adapt to a specific user. A detailed description on the proposed system will be given in the next section.

2   Robust Appearance Model

2.1   Modelling by a Bayesian Classifier

Appearance-based tracking involves *matching* between a candidate image patch and a template image so that the

---

[1]A set of 50 gabor features (5 scales $\times$ 10 orientations) has been used.

likelihood of a predicted location can be estimated and this matching process can be done by a *classifier*. In this work, a sparse Bayesian classifier or relevance vector machine (RVM) [7] is used as the classifier to give the *probability* of an image patch containing the target.

RVM has been exploited in visual tracking in recent research [9]. In their work, RVM was used as a regressor which infers the best location of the target given an image patch. In other words, RVM was used to model the association between an image patch and the location parameter(s). In this work, RVM is used as a classifier to *model* the *appearance* of the target without directly predicting the best location. The location prediction is done separately. The advantage of this usage of RVM is that the probability value given by the classifier can be used in high-level inference tasks such as object identification. In the proposed system, this probability value is used to determine the necessity of model updating so that the frequency of updating tasks can be reduced.

RVM classifier is a simple *binary classifier* and is used to *discriminate* between target image patches and background image patches in this work. Consider a training set that consists of N appearance feature vectors (e.g. pixel intensities of an image patch) and their corresponding labels (i.e. target class and background class), $\{\mathbf{x_n}, t_n\}_{n=1}^{N}$. The problem of learning a binary RVM classifier can be expressed as that of learning a function f so that the input feature $\mathbf{x_n}$ will map onto their correct classification label t_n, and the probability of $\mathbf{x_n}$ being classified as the target class (i.e. $t_n = 1$) equals to $\sigma(y_n) = 1/(1 + e^{-y_n})$ where $y_n = f(\mathbf{x_n})$. The function f can be written as a sparse model where $M \ll N$ [7]:

$$f(\mathbf{x_n}) = \sum_{m=1}^{M} \omega_m \phi_m(\mathbf{x_n}) + \omega_0 \qquad (1)$$

where $\omega = (\omega_0, ...\omega_M)^T$ are the weights and $\phi_m(\mathbf{x_n}) = K(\mathbf{x_n}, \mathbf{x_m})$ with $K(\cdot, \cdot)$ being a positive definite kernel function (Gaussian Kernel is used in our system) and $\mathbf{x_m}$ being a prototype (or a relevance vector) from the training set. Under the RVM framework where hyperparameters $\alpha = \{\alpha_0, ..., \alpha_M\}$ are introduced, learning f from the training data means inferring ω from the data $\mathbf{t} = \{t_1, ..., t_N\}$ such that the posterior probability over the weights, $p(\omega \mid \mathbf{t}, \alpha)$, is maximised. Given $\mathbf{A} = \text{diag}(\alpha_0, \alpha_1, ..., \alpha_N)$, $\mathbf{B_{nn}} = \sigma\{y_n\}[1 - \sigma\{y_n\}]$ and $\mathbf{\Phi}$ is the $N \times (N + 1)$ design matrix, the optimal values of the weights can be estimated by using an iterative procedure [7], where the inverse of a Hessian matrix at 'most probable' weight (ω_{MP}), $\nabla\nabla \log p(\mathbf{t}, \omega \mid \alpha)|_{w_{MP}} = -(\mathbf{\Phi^T B\Phi} + \mathbf{A})$, have to be computed for a current fixed values of α at each loop. The values of α can be inferred from the training data such that the marginal likelihood $p(\mathbf{t} \mid \alpha)$ is maximised. The iterative procedure for estimating ω and α is repeated until some suitable convergence criteria are satisfied.

In a *batch-learning* approach, all training examples will be considered as relevance vectors at the initial stage and the irrelevance vectors will be 'pruned' after re-evaluation of α in each iteration. In other words, every α_i has a finite value at the beginning and the Hessian matrix to be computed in each estimation loop has a size of $(N+1) \times (N+1)$ initially, where N is the number of training samples. Since inversion of Hessian matrices is involved in the learning algorithm, the overall training complexity is $O(N^3)$. This implies that if the initial sample size is huge, the learning algorithm may take a long time to converge.

2.2 Incremental Training Scheme

According to [8], we can also start with an initially small model and sequentially 'add' relevance vectors to increase the marginal likelihood. Considering the marginal likelihood, or equivalently, its logarithm $\mathcal{L}(\alpha)$:

$$\mathcal{L}(\alpha) = \log p(\mathbf{t} \mid \alpha) = -\frac{1}{2}[N \log 2\pi + \log \mid \mathbf{C} \mid + \mathbf{t}^T \mathbf{C}^{-1}\mathbf{t}] \qquad (2)$$

with $\mathbf{C} = \mathbf{B}^{-1} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T$. From the analysis given in [8], $\mathbf{C}$ can be rewritten in this way: $\mathbf{C} = \mathbf{B}^{-1} + \sum_{m \neq i} \alpha_m^{-1} \phi_m \phi_m^T + \alpha_i^{-1} \phi_i \phi_i^T = \mathbf{C_{-i}} + \alpha_i^{-1} \phi_i \phi_i^T$, where $\mathbf{C_{-i}}$ is $\mathbf{C}$ without basis vector i. $\mathcal{L}(\alpha)$ can be therefore rewritten as:

$$\mathcal{L}(\alpha) = \mathcal{L}(\alpha_{-i}) + \frac{1}{2}[\log \alpha_i - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}] \quad (3)$$

where $s_i = \phi_i^T \mathbf{C}_{-i}^{-1} \phi_i$ and $q_i = \phi_i^T \mathbf{C}_{-i}^{-1} \mathbf{t}$. From [8], estimation of α, which gives maximum value of marginal likelihood, can be computed directly from:

$$\begin{aligned} \alpha_i &= \frac{s_i^2}{q_i^2 - s_i}, & \text{if} \quad q_i^2 > s_i, \\ \alpha_i &= \infty, & \text{if} \quad q_i^2 \leq s_i, \end{aligned} \qquad (4)$$

The implication of this evaluation method for α is that we can make discrete changes to the model while we are guaranteed to increase the marginal likelihood. This means we can start from an initially small model and test the 'relevance' of each new input vector i sequentially. When vector i is in the model (i.e. $\alpha_i < \infty$) but $q_i^2 \leq s_i$, then vector i should be removed (i.e. α_i set to ∞); When vector i is not in the model (i.e. $\alpha_i = \infty$) and $q_i^2 > s_i$, vector i should be added (i.e. α_i set to a certain optimal value). Classification model is thus *built incrementally*.

By adopting this incremental training approach, computational complexity is $O(M^3)$ where M is the number of relevance vectors and $M \ll N$. In other words, the training time can be reduced dramatically. In addition, this learning approach allows any new input to be evaluated on the fly

and to be added to the model if certain criteria are fulfilled. This property can be used to develop an adaptive tracking system where online updating of a classification model is needed. The full tracking algorithm will be presented in the next section.

3 Probabilistic Tracking

3.1 Probabilistic Model

In this work, we adopt a *stochastic* tracking approach where a particle filter is used to predict the best location of the target from previous observations. The stochastic approach is chosen because it allows probabilistic analysis based on multiple hypotheses and therefore gives a more robust and reliable result.

Under this approach, visual tracking can be formulated as inferring a motion state θ_t from a series of noisy observations (or raw images) $I_{1:t} = \{I_1, ..., I_t\}$. The state variable used in this work describes the location and the scale of a bounding box (B_t) containing the target and the variable consists of four components: $(C_{x,t}, C_{y,t})$ representing the centre position of B_t, r_t representing the rotation angle of B_t, and s_t representing the scaling factor of B_t. Given a captured image or an observation I_t, and the bounding box B_t with a certain state θ_t, a cropped image x_t containing the target can be obtained, and this image is used to represent the observed appearance of the target in this work.

In the above context, the motion state is a hidden variable (i.e. a variable we want to estimate) while the observations are measurable variables. At each time frame t, a motion state θ_t is associated with an observation I_t and this motion state is a state transition result from a previous motion θ_{t-1}. Therefore, the inference task can be considered as finding a certain state, θ_t, such that $p(\theta_t \mid I_{1:t})$ is maximised, and this probabilistic term can be expanded using Bayes' rule:

$$p(\theta_t \mid I_{1:t}) \propto p(I_t \mid \theta_t) \int p(\theta_t \mid \theta_{t-1}) p(\theta_{t-1} \mid I_{1:t-1}) d\theta_{t-1}$$

(5)

In other words, the inference problem can now be tackled by estimating an observation likelihood term (i.e. observation model) $p(I_t \mid \theta_t)$, and a state transition term (i.e. dynamic model) $p(\theta_t \mid \theta_{t-1})$ at each time frame. Particle filter or CONDENSATION algorithm [3] can be applied to perform the estimation from frame to frame.

In this work, a particle filter is used to approximate the posterior distribution, $p(\theta_t \mid I_{1:t})$, by a set of weighted particles, $S_t = \{\theta_t^{(j)}, \gamma_t^{(j)}\}_{j=1}^J$ with $\sum_{j=1}^J \gamma_t^{(j)} = 1$. In a certain time frame $t-1$, $\theta_t^{(j)}$ is properly weighted with respect to $p(\theta_t^{(j)} \mid I_{1:t-1})$. At time t, state samples $\{\theta_{t-1}^{(j)}\}_{j=1}^J$ are drawn from the particle set, S_{t-1}, and each sample is propagated from $\theta_{t-1}^{(j)}$ to $\theta_t^{(j)}$ with a probability of $p(\theta_t \mid \theta_{t-1})$.

By verifying the likelihood of each sample state using the observation model, $p(I_t \mid \theta_t)$, the posterior probability at time t is updated and a new particle set is weighted again using the updated posterior. The whole process repeats for coming time frames.

To simplify the inference task, the dynamic model used, $p(\theta_t \mid \theta_{t-1})$, is modelled as a Gaussian distribution with Covariance Ψ:

$$p(\theta_t \mid \theta_{t-1}) = \mathcal{N}(\theta_t; \theta_{t-1}, \Psi)$$

(6)

3.2 Observation Model and Appearance Update

The main usage of the observation model is to compute the likelihood of a certain motion state, θ_t. As explained previously, given an observation, I_t, and a bounding box, B_t, with a certain state, θ_t, a cropped image, x_t, can be obtained. By using a pre-stored appearance model, the likelihood of the motion state can be *approximated* by a matching score between the cropped image, x_t, and a template image from the appearance model. In this work, the matching score is obtained by using the sparse Bayesian classifier (or RVM classifier) described in Section 2.

In order to adapt to substantial and long lasting appearance change, the appearance model represented in the RVM classifier can be updated using the incremental learning method described in Section 2.2. Since the probabilistic output given by the RVM classifier indicates the probability of an input image being the target class, this probabilistic value can be used as an indicator to determine the necessity of an appearance update. In this work, if this value lies between 0.4 and 0.6, an appearance update is performed. The whole tracking algorithm is summarised in Algorithm 1.

4 Experimental Results

The proposed method was implemented using unoptimised C++ code and the OpenCV library. The experiment was executed on a P4 2.4GHz computer with 1G memory. In the experiment, video samples exploited in [6] were used. The size of each sample image is 320×240.

In the experiment, several images from the testing video were selected randomly to train the classifier. The average number of training samples (both positive and negative samples) used is 300 and the initial training takes around 30s. The average number of relevance vectors (or prototypes) retained in the classification model is 30. Given the average length of the testing video is 1000 frames, the average number of re-training process invoked is 5 times. Given the average number of particles used is 50, the average time taken for each tracking iteration (without retraining) is 63 ms (i.e. 16 frames per second) and retraining process takes around

Algorithm 1 Appearance-based Tracking Algorithm

- Initialise a sample set, $S_0 = \{\theta_0^{(j)}, \gamma_0^{(j)} = \frac{1}{J}\}_{j=1}^J$
- Train a RVM classifier, $f(x)$, using target and background patches

for $t = 1$ to T **do**
   for $j = 1$ to J **do**
     - Obtain a sample state, $\theta_{t-1}^{(j)}$, from a sample set S_{t-1}
     - Obtain a propagated state, $\theta_t^{(j)}$, from $\theta_{t-1}^{(j)}$ according to $p(\theta_t^{(j)} \mid \theta_{t-1}^{(j)})$
     - Obtain a cropped image, $x_t^{(j)}$, from the current image frame I_t using $\theta_t^{(j)}$
     - Evaluate the likelihood of $\theta_t^{(j)}$ using the classifier, $f(x_t^{(j)})$
     - Update the weight, $\gamma_t^{(j)}$, of the state using $1/(1 + e^{-f(x_t^{(j)})})$
   end for
   - Display a tracking box using $\theta_t^{(j')}$ where $j' = \text{arg-max}_j\{\gamma_t^{(j)}\}$
   if $\max_j\{\gamma_t^{(j)}\}$ within a certain threshold **then**
     - Retrain the classifier using the corresponding image patch, $x_t^{(j)}$
   end if
   - Normalise the weight by $\gamma'^{(j)}_t = \gamma_t^{(j)} / \sum_{j=1}^J \gamma_t^{(j)}$ and generate the sample Set, S_t, using the updated weight
end for

1 second to complete (where 1 positive and 9 arbitrary negative samples are added each time). Two sample tracking sequences are shown in Figure 1. Figure 2 illustrates the effectiveness of the proposed method under several difficult conditions such as conditions involve the change in viewing angle, facial expression and lighting direction.
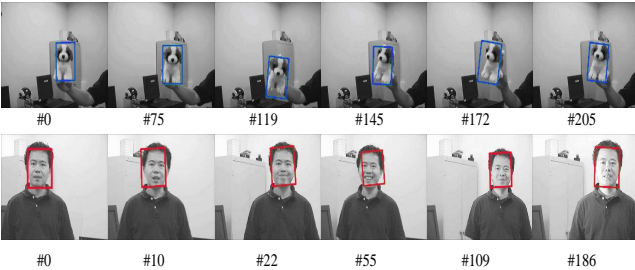

| #0 | #75 | #119 | #145 | #172 | #205 |

| #0 | #10 | #22 | #55 | #109 | #186 |

Figure 1: This figure shows two sample tracking sequences generated using the proposed method.

5 Conclusion

In this paper, a new appearance-based method is proposed to track an object which may undergo appearance change. The proposed method performs better than recent methods in three ways. First, by reducing the dimensional size of the appearance feature and using a sparse Bayesian classifier to ensure high classification accuracy, this method *relaxes* the *tradeoffs* between time complexity and accuracy. Second, this method detects and tracks an object based on more than one prototype, and this means it is more *robust* to the appearance change of the object. Third, this
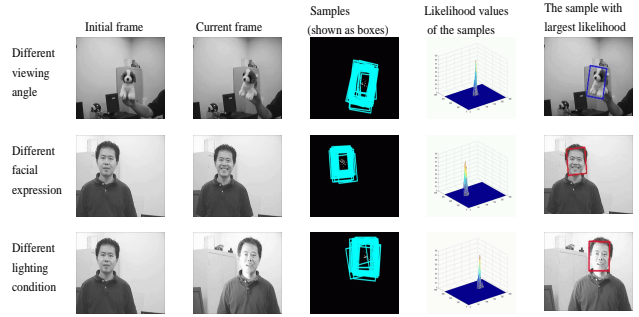

Figure 2: This figure shows the likelihood histograms generated by the proposed method and the corresponding tracking results under several difficult conditions.

method supports incremental learning which enables *online updating* of an appearance model. The updating process, however, takes longer time than the classification task and may deteriorate tracking performance. Fortunately, given a good initial classification model, the target can be tracked reliably without performing any updating task. Further investigation on relaxing this constraint will be done.

References

[1] F. de la Torre, S. Gong, and S. J. McKenna. View-based adaptive affine tracking. In *Proc. ECCV*, pages 828–842, 1998.

[2] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10):1025–1039, 1998.

[3] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. ECCV*, pages 343–356, 1996.

[4] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10):1296–1311, 2003.

[5] H. T. Nguyen and A. W. M. Smeulders. Fast occluded object tracking by a robust appearance filter. *PAMI*, 26(8):1099–1104, 2004.

[6] D. Ross, J. Lim, and M.-H. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Proc. ECCV*, pages 470–482, 2004.

[7] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

[8] M. E. Tipping and A. C. Faul. Fast marginal likelihood maximization for sparse bayesian models. In *Proc. of the Int. Workshop on Artificial Intelligence and Statistics*, 2003.

[9] O. Williams, A. Blake, and R. Cipolla. Sparse bayesian learning for efficient visual tracking. *PAMI*, 27(8):1292–1304, 2005.

[10] S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE TIP*, 13(11):1491–1506, 2004.