Open access • Proceedings Article • DOI:10.1109/CVPR.2012.6247898

Robust visual tracking using autoregressive hidden Markov Model — Source link

Dong Woo Park, Junseok Kwon, Kyoung Mu Lee

Institutions: Seoul National University

Published on: 16 Jun 2012 - Computer Vision and Pattern Recognition

Topics: Active appearance model, Hidden Markov model, Video tracking, Autoregressive model and Probabilistic logic

Robust Visual Tracking using Autoregressive Hidden Markov Model

Dong Woo Park, Junseok Kwon, and Kyoung Mu Lee
Department of EECS, ASRI, Seoul National University, 151-742, Seoul, Korea
kospi1981@gmail.com, {paradis0, kyoungmu}@snu.ac.kr, http://cv.snu.ac.kr

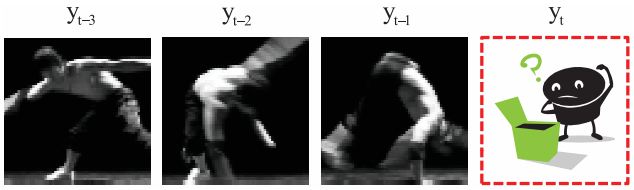
Abstract

Recent studies on visual tracking have shown significant improvement in accuracy by handling the appearance variations of the target object. Whereas most studies present schemes to extract the time-invariant characteristics of the target and adaptively update the appearance model, the present paper concentrates on modeling the probabilistic dependency between sequential target appearances (Fig. 1-(a)). To actualize this interest, a new Bayesian tracking framework is formulated under the autoregressive Hidden Markov Model (AR-HMM), where the probabilistic dependency between sequential target appearances is implied. During the learning phase at each time step, the proposed tracker separates formerly seen target samples into several clusters based on their visual similarity, and learns cluster-specific classifiers as multiple appearance models, each of which represents a certain type of the target appearance. Then the dependency between these appearance models is learned. During the searching phase, the target state is estimated by inferring the most probable appearance model under the consideration of its dependency on formerly utilized appearance models. The proposed method is tested on 12 challenging video sequences containing targets with abrupt appearance variations, and demonstrates that it outperforms current state-of-the-art methods in accuracy.

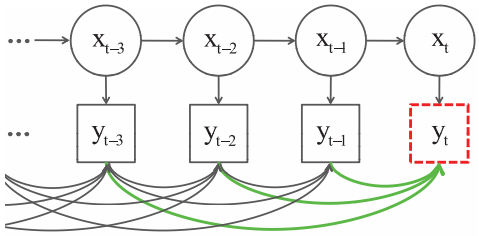
1. Introduction

One of the major challenges in visual tracking comes from dealing with the appearance variations of the target object over time. Since the variations are attributed to various factors (*e.g.*, pose change, shape deformation, illumination change, occlusion, camera viewpoint change, *etc.*) and cannot be seen beforehand, especially when tracking generic objects, adopting an appropriate appearance model at each time step is difficult.

The most basic scheme reported to handle this difficulty, thus far, is to adaptively update the single appearance model at each frame: learn a new appearance model with time-invariant characteristics extracted from formerly observed target samples, and adopt the model to the current frame.



(a) Sequential target appearances in *b-boy* sequence.



(b) Autoregressive Hidden Markov Model

Figure 1. (a) Key question the current paper attempts to answer: how to infer the current target appearance, considering its dependency on formerly observed target appearances. (b) The n th-order AR-HMM when $n = 3$. The AR-HMM implies the dependency between sequential target appearances, which is different from the standard HMM.

Collins *et al.* [5] first emphasize this scheme by adaptively changing the color features that distinguish the target from the background. Ross *et al.* [16] utilize the incremental learning strategy to adaptively update subspaces that compose the target appearance. In [2, 3], a binary classifier is learned via the Support Vector Machine and AdaBoost algorithm, respectively, to represent the target. In [8], an online Boosting algorithm is proposed to update an appearance model when formerly observed samples are discarded. Furthermore, [9, 18] combine the semi-supervised learning with [8] to handle noisy target samples. Babenko *et al.* [4] apply the Multiple Instance Learning to the Boosting algorithm to resolve the sample ambiguity problem. However, due to the lack of time-invariant characteristics that cover all appearance variations shown beforehand, such methods update the model with characteristics more representative of the recent target samples by adjusting the learning rate [16, 8, 4] or learning from the subset of formerly seen samples [5, 2, 3, 9, 18]. Therefore, such schemes are intolerant

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

of abrupt appearance variations in short time intervals.

Several studies have resolved the limitation of the basic schemes by constructing the multiple appearance models, each of which represents a certain type of the appearance. Kwon *et al.* [11] decompose features from the previously observed target samples via SPCA. Then each feature is injected into one of the multiple trackers as an appearance model. In [10], the MCBoost is used to jointly learn the target sample clusters and cluster-specific classifiers as appearance models. In [14, 13], the sparse coding scheme is used to extract the multiple templates from given training samples. However, none of the methods considers the dependency between sequential target appearances when selecting an appropriate appearance model at each time step; thus, they are limited in inferring the most probable appearance model with high accuracy.

Another research effort to cope with abrupt appearance variations is to model the mapping function between sequential target appearances. In [6, 12], a nonlinear mapping function from the geometrical transformation to the appearance is modeled via the manifold learning. When a target image patch is given at the searching step, the geometrical transformation settings of a learned mapping function are recovered. Hence, an appearance model for the next frame is determined. However, when all kinds of variational factors are considered simultaneously, finding an appropriate mapping function is difficult. Furthermore, because they select an appearance model for the next frame in a deterministic manner, such trackers are robust only when the target objects experience cyclic appearance variations.

In the current work, a new Bayesian tracking framework is formulated under the AR-HMM that implies the probabilistic dependency between sequential target appearances, as shown in Fig. 1-(b). Under this framework, the proposed tracker performs the learning and searching in consecutive order at each time step. During the learning phase, our tracker jointly separates formerly observed target samples into several clusters based on their visual similarity, and approximates the appearances of the target samples included in each cluster as a cluster-specific classifier (appearance model). Then the dependency between the multiple appearance models is learned. During the searching phase, the tracker estimates the target state by inferring the most probable appearance model under the consideration of its dependency on previously utilized appearance models.

The major contribution of the present study is twofold: 1) The AR-HMM is first adopted for the visual tracking framework. The posterior probability of the target state is derived under the AR-HMM. The resulting formulation indicates that considering the dependency between the sequential target appearances is equal to modeling the prior of the target appearance and adopting it to control the degree of belief in the likelihood term. 2) By slightly modifying the learning scheme in [17], jointly clustering target samples and learning cluster-specific appearance models are performed in a fully unsupervised manner. Different from the previous work [10] related to this topic, which requires the off-line setups to construct the cluster priors, the learning scheme adopted in the present paper automatically determines the number of clusters based on the amount of variations formerly shown by the target. This property makes the proposed tracker more practical in tracking generic objects with different amounts of appearance variations.

2. Tracking Framework under AR-HMM

From hence, the target appearance at time t, $\mathbf{y}_t$, is dealt as a random vector. Note that, under the standard HMM used by previous studies, $\mathbf{y}_t$ is a deterministic vector. For ease of implementation, $\mathbf{y}_t$ is assumed as discrete, that is, $\mathbf{y}_t \in \{\mathbf{o}^k, k = 1, ..., K\}$, where $\mathbf{o}^k$ is an appearance model by which the target may be represented, and K is the number of appearance models at time t. However, the proposed formulation works even when $\mathbf{y}_t$ is continuous. A goal of the proposed tracker is to estimate the target state at time t, $\mathbf{x}_t$, under the maximum-a-posteriori (MAP) criterion by employing the most probable appearance model $\mathbf{o}^k$.

2.1. Formulation

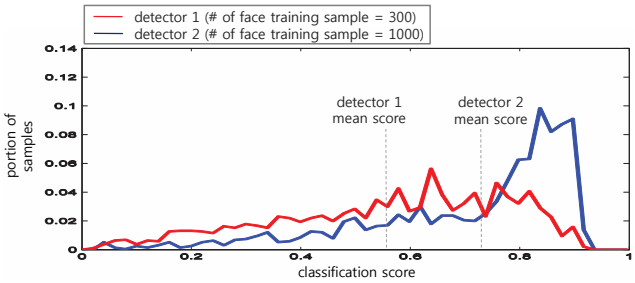
The posterior probability of $\mathbf{x}_t$ is formulated under the AR-HMM shown in Fig. 1-(b). By initially applying the Bayesian theorem, the posterior probability is given by

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_t|\mathbf{y}_{1:t-1})}, \tag{1}$$

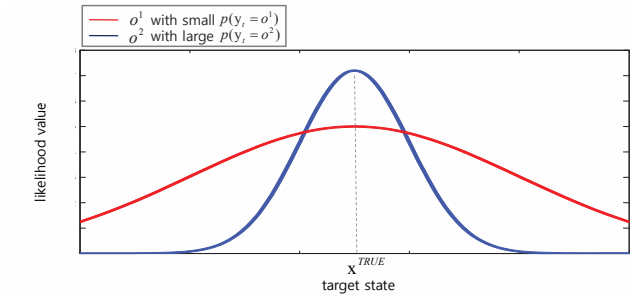
where $\mathbf{y}_{t_1:t_2} \equiv \{\mathbf{y}_\tau, \tau = t_1, ..., t_2\}$ is a set of appearances from time t_1 to t_2. Under the standard HMM, $p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{x}_t) = p(\mathbf{y}_t|\mathbf{x}_t)$, because $\mathbf{y}_t$ and $\mathbf{y}_{1:t-1}$ are mutually independent. In contrast, this independency does not hold under the AR-HMM. When the n th-order AR-HMM is assumed, two following properties hold: 1) $\mathbf{y}_t$ and $\mathbf{y}_{t-n:t-1}$ are mutually dependent, 2) $\mathbf{y}_{t-n:t-1}$ and $\mathbf{x}_t$ are mutually independent. Given that $\mathbf{y}_t$ is random, the second property can be proved using the Bayes ball algorithm. Based on these two properties, two terms $p(\mathbf{y}_t|\mathbf{y}_{1:t-1})$ and $p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{x}_t)$ in Eq. 1 can be simplified as follows:

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1}) = p(\mathbf{y}_t|\mathbf{y}_{t-n:t-1}), \tag{2}$$

$$\begin{aligned} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathbf{x}_t) &= p(\mathbf{y}_t|\mathbf{y}_{t-n:t-1}, \mathbf{x}_t) \\ &= \frac{p(\mathbf{y}_{t-n:t-1}|\mathbf{x}_t, \mathbf{y}_t)p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_{t-n:t-1}|\mathbf{x}_t)} \\ &= \frac{p(\mathbf{y}_{t-n:t-1}|\mathbf{y}_t)p(\mathbf{y}_t|\mathbf{x}_t)}{p(\mathbf{y}_{t-n:t-1})}. \end{aligned} \tag{3}$$



(a)



(b)

Figure 2. (a) Distribution of classification scores of the face test samples. (b) Expected distribution of likelihood values at possible target states.

By substituting Eqs. 2 and 3 into Eq. 1, a simple and clear posterior probability is finally achieved as

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{t-n:t-1}|\mathbf{y}_t)p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{p(\mathbf{y}_{t-n:t-1})p(\mathbf{y}_t|\mathbf{y}_{t-n:t-1})} \quad (4)$$

$$= \frac{1}{p(\mathbf{y}_t)}p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1}). \quad (5)$$

A remarkable point is that not only $p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$, but also $1/p(\mathbf{y}_t)$ should be considered when calculating $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. In addition, the form of the posterior probability remains equally irrelevant to the order n of the AR-HMM. In summary, Eq. 5 indicates that coping with the dependency between sequential appearances is equal to modeling $1/p(\mathbf{y}_t)$ and applying it when calculating the posterior.

2.2. Discussion

Eq. 5 gives a meaningful message about the relation between the appearance prior $p(\mathbf{y}_t)$ and the likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$: the degree of belief in $p(\mathbf{y}_t|\mathbf{x}_t)$ should be controlled by $p(\mathbf{y}_t)$. Specifically, if $p(\mathbf{y}_t = \mathbf{o}^k)$ is large, $p(\mathbf{y}_t = \mathbf{o}^k|\mathbf{x}_t)$ should output a correspondingly large value to insist that the target is at a certain state $\mathbf{x}_t$, with the appearance that can be approximated by $\mathbf{o}^k$, and vice versa.

This message is in alignment with a theory of Machine Learning [15]: the size of the gap between the training error and the test error of a certain hypothesis is inversely related to the size of the training set. To verify, two face detectors are trained via the AdaBoost algorithm [7], using different sizes of training sets (300 and 1000 face training samples are given for each detector, respectively, and 1000 nonface training samples are given for both). Then these detectors are tested on 1000 face test samples, and the distribution of their classification scores by each detector is shown in Fig. 2-(a). Although the classification scores of the face training samples are mostly around 1 in both detectors (no training error), more test sample scores shift to smaller values (larger test error) when the smaller number of face training samples are given for training.

Reminding the theory, let there exist two appearance models for tracking, $\mathbf{o}^1$ and $\mathbf{o}^2$, whose likelihood values monotonically increase when the classification score increases. In addition, let $p(\mathbf{y}_t = \mathbf{o}^2) \gg p(\mathbf{y}_t = \mathbf{o}^1)$. We can expect that the target appearances, which can be represented by $\mathbf{o}^2$, have been more frequently observed than that of $\mathbf{o}^1$, so that $\mathbf{o}^2$ is approximated with more samples than $\mathbf{o}^1$. Then, under an assumption that $\mathbf{o}^1$ and $\mathbf{o}^2$ are both unbiased, a likelihood value returned by $\mathbf{o}^2$ at its truth state $\mathbf{x}^{TRUE}$ can be expected to be larger than that of $\mathbf{o}^1$, as shown in Fig. 2-(b). Hence, if $\mathbf{o}^1$ and $\mathbf{o}^2$ return same maximum likelihood values at different $\mathbf{x}_t$s, which result should be believed? If the target is guaranteed to exist in the scene and its appearance can be represented either by $\mathbf{o}^1$ or $\mathbf{o}^2$, the result of $\mathbf{o}^1$ should be believed. The appearance model $\mathbf{o}^2$ should have returned a larger likelihood value to insist that its state is the truth state, when $\mathbf{o}^1$ insists the same based on an identical likelihood value. Consequently, the inverse relation between $p(\mathbf{y}_t)$ and $p(\mathbf{y}_t|\mathbf{x}_t)$ is reasonable.

3. Learning under AR-HMM

During the learning phase, $p(\mathbf{y}_t|\mathbf{x}_t)$ and $1/p(\mathbf{y}_t)$ are modeled in consecutive order. To model $p(\mathbf{y}_t|\mathbf{x}_t)$, formerly seen target samples are separated into several clusters and cluster-specific classifiers are learned to discriminate the target from the background. These classifiers are used as the probable appearance models by which the target may be represented. Then $1/p(\mathbf{y}_t)$ is modeled among the classifiers. The proposed tracker learns those two terms in a fully incremental manner, that is, it learns using the currently given samples and updates them when the new samples arrive at the next time step.

3.1. Learning $1/p(\mathbf{y}_t)$

Since the target appearances observed from the initial frame to the current are the subset of the whole target appearances, as shown in Fig. 3, exact modeling of $1/p(\mathbf{y}_t)$ is unrealistic. Under the n th-order AR-HMM, $1/p(\mathbf{y}_t)$ can be approximated as follows (see Eqs. 4 and 5):

$$\frac{1}{p(\mathbf{y}_t)} = \frac{1}{p(\mathbf{y}_{t-n:t-1})}\frac{p(\mathbf{y}_{t-n:t-1}|\mathbf{y}_t)}{p(\mathbf{y}_t|\mathbf{y}_{t-n:t-1})}. \quad (6)$$

This approximation provides two advantages: 1) $1/p(\mathbf{y}_{t-n:t-1})$ can be neglected under the MAP criterion,

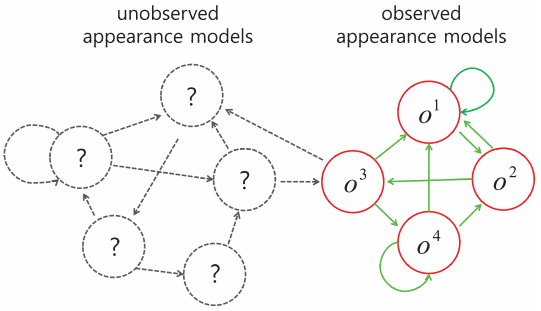


Figure 3. Example of observed and unobserved appearance models at a certain frame.

because it is constant at time t. 2) Since $p(\mathbf{y}_{t-n:t-1}|\mathbf{y}_t)$ and $p(\mathbf{y}_t|\mathbf{y}_{t-n:t-1})$ are the probabilities conditioned on neighbor appearance models in the time domain, each term can be approximated even when the whole set of the appearances is not observed. Thus, when the number of appearance models, K, is fixed and target samples are allocated to one of the appearance models through learning $p(\mathbf{y}_t|\mathbf{x}_t)$, we can model $p(\mathbf{y}_{t-n:t-1}|\mathbf{y}_t)$ and $p(\mathbf{y}_t|\mathbf{y}_{t-n:t-1})$ by the maximum likelihood estimation: counting the observed number of transitions between the appearance models and normalizing it with the number of total transitions. Although appearance models, which may be dependent on unobserved appearance models (*e.g.*, $\mathbf{o}^3$ in Fig. 3), may have an error, it is neglected as the factor that cannot be handled. Nevertheless, as shown in the experimental results, the proposed tracker operates well. In the current implementation, we set $n = 1$, which allows us to consider only adjacent neighbor appearance models to approximate $1/p(\mathbf{y}_t)$.

3.2. Learning $p(\mathbf{y}_t|\mathbf{x}_t)$

To learn $p(\mathbf{y}_t|\mathbf{x}_t)$, we adopt the learning scheme in [17] with slight modifications. Although they propose the scheme to cluster the object classes that may share the classification knowledge, we use it to automatically cluster target samples and learn cluster-specific appearance models, so that every target samples can be well discriminated from the nontarget samples by at least one of the multiple appearance models. As mentioned earlier, this scheme is more practical than that in [10] in solving the multi-modality problem they tackle, in the sense that the scheme automatically decides the number of clusters based on the amount of appearance variations formerly shown by the target. In contrast to [10], none of the off-line setups to construct the cluster priors is required. Due to the lack of space, a brief introduction of the learning scheme is provided. Readers are referred to [17] for a more detailed explanation.

To notate, let $\mathbf{m}$ be the instance (image patch) and ℓ be the corresponding binary label, i.e., $\ell \in \{0, 1\}$. At time t, instances are extracted from the estimated states at previous frames, and a target training set $\mathbf{D}_{pos} \equiv \{(\mathbf{m}^i, \ell^i = 1, s^i), i = 1, ..., I_{pos}\}$ is constructed, where I_{pos} is the number of target samples and $s^i \in \{1, ..., K\}$ is the cluster index of $\mathbf{m}^i$. The number of I_{neg} nontarget samples are randomly extracted around the estimated state at the most recent frame $t-1$, and a nontarget training set $\mathbf{D}_{neg} \equiv \{(\mathbf{m}^i, \ell^i = 0), i = 1, ..., I_{neg}\}$ is constructed. A training set for cluster k, $\mathbf{D}^k$, is composed by unifying $\mathbf{D}_{neg}$ and $\mathbf{D}^k_{pos}$, where $\mathbf{D}^k_{pos}$ consists of target samples with the cluster index $s^i = k$ in $\mathbf{D}_{pos}$. The size of $\mathbf{D}^k$ and $\mathbf{D}^k_{pos}$ are notated as I^k and I^k_{pos}.

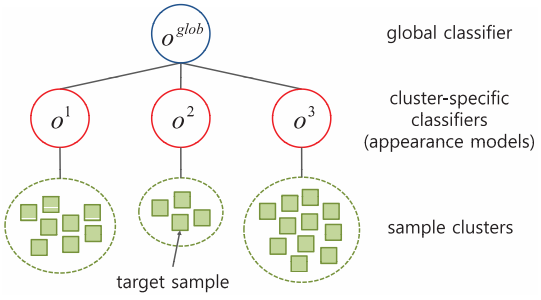


Figure 4. Structure of multiple appearance models.

An appearance model is modeled as a linear binary classifier. Thus, the classification score of $\mathbf{m}$ by $\mathbf{o}^k$ is $\mathbf{o}^k(\mathbf{m}) = \sum_{j=1}^{J} \beta^k_j \mathbf{h}_j(\mathbf{m})$, where $\mathbf{h}_j(\cdot)$ is the classification score by the j th feature of $\mathbf{o}^k$, $\mathbf{h}_j$, and β^k_j is the corresponding weight coefficient. (The feature $\mathbf{h}_j$ is not indexed by k, because the cluster-specific classifiers are constrained to share the same features. Based on our experience, this constraint helps the classifiers to be unbiased when the number of target samples is small.) By defining $\mathbf{m}_t$ as the instance extracted from $\mathbf{x}_t$, and ℓ_t as the corresponding label, the likelihood can be expressed as $p(\mathbf{y}_t = \mathbf{o}^k|\mathbf{x}_t) = p(\ell_t = 1|\mathbf{m}_t, \mathbf{o}^k)$. Using the logistic regression model,

$$p(\ell_t = 1|\mathbf{m}_t, \mathbf{o}^k) = \frac{1}{1 + exp(-\mathbf{o}^k(\mathbf{m}_t))}. \quad (7)$$

The structure of the multiple appearance models is shown in Fig. 4. Under a global classifier $\mathbf{o}^{glob}$, there exist multiple cluster-specific classifiers $\mathbf{o}^k$s. Each target sample is allocated to one of the cluster-specific classifiers. Different from [17], our cluster-specific classifiers do not share the weight coefficients, but the features selected by a global classifier $\mathbf{o}^{glob}$ from the feature pool $\mathbf{P}$.

A pseudo-code to learn such models is shown in Algorithm 1. A key idea is to alternate between learning the appearance models $\{\mathbf{o}^k, k = 1, ..., K\}$ and determining the cluster index vector $\mathbf{s} = [s^1, ..., s^{I_{pos}}]$. To begin, the number of J features are selected from the feature pool $\mathbf{P}$ through $\mathbf{o}^{glob}$, and the cluster index vector $\mathbf{s}$ is initialized. The weight coefficient vectors $\beta^k = [\beta^k_1, ..., \beta^k_J]$, $k = 1, ..., K$, are then determined by maximizing

$$\ln p(\beta^k|\mathbf{D}^k) \propto \sum_{i=1}^{I^k} \ln p(\ell^i|\mathbf{m}^i, \beta^k) + \ln p(\beta^k). \quad (8)$$

Algorithm 1 Learning multiple appearance models

Input: $\mathbf{D}_{pos}, \mathbf{D}_{neg}, \mathbf{P}, U, J, \gamma$
Output: $\mathbf{s}, \{\mathbf{o}^k, k = 1, ..., K\}$

Select J features from $\mathbf{P}$ by training $\mathbf{o}^{glob}$ with $\mathbf{D}_{pos}$ and $\mathbf{D}_{neg}$, using AdaBoost.
$s^1 \leftarrow 1, K \leftarrow 1, s^i \leftarrow 0$ for $i = 2, ..., I_{pos}$
for $i = 2$ to I_{pos} **do**
  Randomly draw s^i under Eq. 10.
  if $s^i = K + 1$ **then**
    $K \leftarrow K + 1$
  end if
end for
for 1 to U **do**
  for $k = 1$ to K **do**
    Learn β^k for J features with $\mathbf{D}^k_{pos}$ and $\mathbf{D}_{neg}$ by maximizing Eq. 8, using AdaBoost.
  end for
  for $i = 1$ to I_{pos} **do**
    Learn β^{K+1} for J features with $(\mathbf{m}^i, \ell^i = 1)$ and $\mathbf{D}_{neg}$ by maximizing Eq. 8, using AdaBoost.
    Decide s^i by maximizing Eq. 9.
    if $s^i = K + 1$ **then**
      $K \leftarrow K + 1$
    else
      Delete β^{K+1}.
    end if
  end for
  for $k = 1$ to K **do**
    if $I^k_{pos} = 0$ **then**
      Delete β^k, $K \leftarrow K - 1$
    end if
  end for
  Rearrange $\mathbf{s}$, so that they range from 1 to K.
end for

Assuming that $p(\beta^k)$ is under the uniform distribution, the problem is solved by the AdaBoost algorithm [7]. After learning the cluster-specific classifiers, the cluster index vector $\mathbf{s} = [s^1, ..., s^{I_{pos}}]$ is determined by maximizing

$$\ln p(s^i|\beta, \mathbf{s}^{\sim i}, \ell^i, \mathbf{m}^i) \propto \ln p(\ell^i|\beta^{s^i}, s^i, \mathbf{m}^i) + \ln p(s^i|\mathbf{s}^{\sim i}), \quad (9)$$

where $\mathbf{s}^{\sim i}$ denotes a vector $\mathbf{s}$, but with s^i omitted. The term $\beta = [\beta^1, ..., \beta^{K+1}]$, where $K + 1$ is the new cluster index. The likelihood $p(\ell^i|\beta^{s^i}, s^i, \mathbf{m}^i)$ is given by Eq. 7. The prior $p(s^i|\mathbf{s}^{\sim i})$ is given by the Chinese Restaurant Process (CRP),

$$p(s^i = k|\mathbf{s}^{\sim i}) = \begin{cases} \frac{I^k_{pos}}{I_{pos}-1+\gamma}, & 1 \le k \le K \\ \frac{\gamma}{I_{pos}-1+\gamma}, & k = K + 1 \end{cases}, \quad (10)$$

where γ is the concentration parameter. These two processes are iterated for the prefixed number of U times.

As shown in Eq. 9, a remarkable point is that the number of clusters is automatically determined considering the trade-off between the classification error and the cost for splitting the cluster. The cost is determined by the CRP shown in Eq. 10 (when $k = K + 1$). Thus, none of the naive cluster priors should be given through the off-line setups.

4. Searching under AR-HMM

After the learning phase, a target state $\widehat{\mathbf{x}}_t = arg\max_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{y}_{1:t})$ is searched via the Metropolis-Hastings algorithm, as in [11]. The sampling is composed of two basic steps: the proposal step and the acceptance step. In the proposal step, a new sample state $\mathbf{x}^{(r+1)}_t$ is proposed from the current sample state $\mathbf{x}^{(r)}_t$ by the proposal density function $p(\mathbf{x}^{(r+1)}_t|\mathbf{x}^{(r)}_t)$, where r is the sample index. The Normal distribution $\mathcal{N}(\mathbf{x}^{(r)}_t, \sigma)$, with the mean $\mathbf{x}^{(r)}_t$ and the covariance matrix σ, is used as $p(\mathbf{x}^{(r+1)}_t|\mathbf{x}^{(r)}_t)$. After the new state $\mathbf{x}^{(r+1)}_t$ is proposed, the acceptance ratio η is calculated as follows:

$$\eta = \min[1, \frac{\max_{\mathbf{y}^{(r+1)}_t} p(\mathbf{x}^{(r+1)}_t|\mathbf{y}_{1:t-1}, \mathbf{y}^{(r+1)}_t)}{\max_{\mathbf{y}^{(r)}_t} p(\mathbf{x}^{(r)}_t|\mathbf{y}_{1:t-1}, \mathbf{y}^{(r)}_t)}], \quad (11)$$

where the posterior of each sample state is given by Eq. 5. (Under the AR-HMM, the prior $p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = p(\mathbf{x}_t)$. We approximate $p(\mathbf{x}_t) \approx p(\mathbf{x}_t|\widehat{\mathbf{x}}_{t-1})$, assuming that $p(\mathbf{x}_t)$ is time-varying. The Normal distribution $\mathcal{N}(\widehat{\mathbf{x}}_{t-1}, \xi)$, with the mean $\widehat{\mathbf{x}}_{t-1}$ and the covariance matrix ξ, is used for the state transition prior $p(\mathbf{x}_t|\widehat{\mathbf{x}}_{t-1})$.) With this η, the tracker decides whether to accept the new state $\mathbf{x}^{(r+1)}_t$ or not. After iterating these two steps for the prefixed number of times, the tracker can finally select the most probable state $\widehat{\mathbf{x}}_t$. Notably, the proposed tracker implicitly deals with the dependency between sequential target appearances by calculating the posterior under the consideration of the appearance prior of multiple appearance models.

5. Experimental Results

The proposed tracker is tested on 12 video sequences. Eight videos (*girl, david, tiger1, tiger2, faceocc1, faceocc2, sylvester*, and *shaking*) are collected from the public dataset, and four videos (*b-boy, cheetah, lighting,* and *horse-race*) are collected for ourselves. For cross-validation, the center position error is compared with that of current state-of-the-art methods (VTD [11], MIL [4], IVT [16], and FRAGT [1]), the executable codes of which are accessible on their own web pages.

In the current settings, the instance $\mathbf{m}$ is set as the rectangular gray-scaled image, and the feature $\mathbf{h}$ is set as the Haar-like feature [19]. The target state is defined as $\mathbf{x}_t \equiv [x_t, y_t, \mu_t, \theta_t]$, where $[x_t, y_t]$ is the 2D center position, μ_t is the scale, and θ_t is the rotation. For the parameters, we set the order of the AR-HMM, n, to 1, the size of the feature pool $\mathbf{P}$ to 450, the number of features for cluster-specific classifiers, J, to 150, the concentration parameter γ to 0.2, and the number of learning iterations, U, to 3. Since the number of sample clusters, K, can be increased infinitely overtime under the current framework, the maximum value

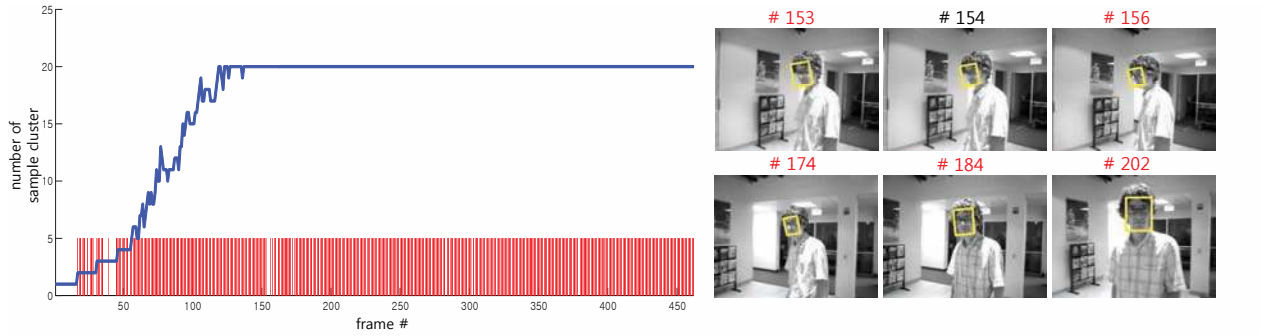


Figure 5. Tracking results in the *david* sequence. The graph on the left shows the number of learned sample clusters over the frames (blue line). A red impulse is drawn on frames where the model transition occurs. The estimated states in several frames are depicted on the image sequences on the right. The frames with the model transition are indicated with a red frame index.

of K is constrained to 20 by forcing the probability of the CRP for splitting the cluster to zero, when K reaches its maximum value. Such settings are fixed through all the experiments shown below. Furthermore, all target samples extracted from previous frames are utilized in learning the likelihood. Although the learning time increases linearly with time t in this setting, it is neglected because the focus of the present paper is to confirm the significance of the dependency between sequential target appearances. Learning time can be bounded simply by discarding old samples, or adopting the online learning scheme, such as that in [8]. The related issues will be studied for future research.

5.1. Qualitative Evaluation

Fig. 5 shows the tracking results of the proposed tracker in the *david* sequence. As shown in the graph on the left, the number of sample clusters increases when the amount of appearance variations shown by the target becomes larger over time. It can also be recognized that the model transition (selecting the most probable appearance model which is different from that of the most recent frame) occurs more frequently than expected. Although the *david* sequence does not contain abrupt appearance variations, the model transition occurs in the frame of 92% of the total 462 frames. This result indicates that, even when the variations are quite smooth (*e.g.*, rotating the face, as shown in the right of Fig. 5), an appropriate appearance model for each frame may be different from that of the adjacent frames.

5.2. Quantitative Evaluation

For the quantitative evaluation, the mean center position error per frame is calculated for each tracker. Each tracker is tested five times per video sequence.

To demonstrate that the inverse relation between the appearance prior $p(\mathbf{y}_t)$ and the likelihood $p(\mathbf{y}_t|\mathbf{x}_t)$ is reasonable, the proposed tracker is tested in four cases: 1) when two terms are in inverse relation, as derived in Eq. 5 (ART), 2) when the prior term is neglected (ARTWOP), 3) when two terms are in proportional relation (ARTWPR), and 4) when K is constrained to 1 (ARTSAP). ARTWOP and ARTSAP can be regarded as tracking under the standard HMM with multiple appearance models and single appearance model, respectively. In addition to the formerly mentioned settings, the size of the nontarget training set, I_{neg}, is set to 300. Such settings are identical for all cases. The results are shown in the four columns on the right in Table 1. As can be seen, ART shows the best accuracy for all test videos. When the appearance prior and the likelihood are forced to have a proportional relation (ARTWPR), the accuracy is even lower than that of the tracker using the single appearance model (ARTSAP) for most test videos. When the appearance prior is omitted (ARTWOP), the results are better than that of ARTWPR and ARTSAP, but ARTWOP never outperforms ART for any video. In summary, accuracy is improved when multiple appearance models are used. The accuracy is even better when the dependency between sequential target appearances is also considered.

The results of cross validation are shown in the first five columns in Table 1. In this case, the number of 1000 nontarget training samples are given to the proposed tracker to show its fully maximized accuracy (ARTOPT). Notably, however, ART shows outperforming accuracy, compared with that of other state-of-the-art methods, except in the *tiger2* and *shaking* sequences. In the *girl, david, tiger1, tiger2,* and *sylvester* sequences, the targets experience pose changes under constrained illumination changes. In the *faceocc1* and *faceocc2* sequences, the targets are smoothly occluded by the books. In such videos, ARTOPT shows almost perfect tracking accuracy through all trials, whereas other methods experience drifting and shrinking of the tracking window. To evaluate for more challenging environment, the methods are tested on the *b-boy, cheetah, lighting, shaking*, and *horse-race* sequences. The estimated target states are shown in Fig. 6. In the *b-boy* sequence, severe pose changes are shown by a dancing person. Although the scene is challenging, ARTOPT tracks the target well for most trials, whereas other methods cause drifting. In the *cheetah* sequence, abrupt shape deformations of

	ARTOPT	VTD	MIL	IVT	FRAGT	ART	ARTWOP	ARTWPR	ARTSAP
girl	10.6	14.6	33.1	55.6	20.4	12.9	15.8	26.6	16.4
david	3.3	46.9	24.9	42.8	27.5	3.7	4.1	5.8	4.1
tiger1	4.9	44.5	30.3	55.7	24.8	12.6	23.1	27.4	20.6
tiger2	5.4	53.1	11.9	48.9	36.7	16.3	24.2	49.7	27.6
faceocc1	8.1	9.8	35.4	10.9	9.3	8.5	9.4	15.0	12.0
faceocc2	6.0	54.1	15.5	12.8	63.9	12.1	15.9	18.9	18.7
sylvester	5.9	23.1	14.8	120.9	15.7	8.7	10.1	11.2	11.5
shaking	7.7	78.2	42.7	96.5	194.4	50.3	158.9	185.6	199.9
b-boy	34.9	147.1	65.9	244.3	152.3	55.1	64.5	63.4	66.7
cheetah	17.0	31.6	230.8	126.3	132.0	17.4	20.8	49.8	22.1
lighting	5.1	105.0	153.5	52.3	120.8	7.6	27.4	235.9	169.0
horse-race	12.5	121.5	51.8	37.5	81.0	25.4	79.9	136.4	85.9

Table 1. Mean center position errors in pixels. Red and blue indicate the best and second best accuracy, respectively, at each sequence.

a running cheetah occur. Although ARTOPT cannot contain the whole body of the cheetah for several frames in the last portion, it still can follow the center of the cheetah, whereas other trackers lose the target in most trials. In the *lighting* and *shaking* sequences, there are abrupt illumination changes on the guitarists, but the situation is accurately handled by ARTOPT. In the *horse-race* sequence, the head of a jockey is occluded frequently and cyclically by the two horses, but ARTOPT handles this situation well by choosing the most probable appearance model at each frame. Through all the test videos, ARTOPT outperforms other state-of-the-art methods in accuracy.

Lastly, ARTOPT requires around 8 seconds for the mean processing time per frame on 12 video sequences under the current test environment (implementation in C codes, Intel Q9550 2.83GHz CPU). Most of the time is spent learning the likelihood term, which may be greatly reduced after optimizing the current implementation in the code level. Furthermore, as mentioned in [17], the adopted learning scheme has the structure where parallel-processing is applicable. After several optimizations, the proposed tracker is expected to run in real time.

6. Conclusion

In the current paper, we proposed a new Bayesian tracking framework under the AR-HMM, to deal with the dependency between sequential target appearances. A new form of the posterior probability of the target state was derived. Through the derivation, the importance of modeling the appearance prior and its inverse relation with the likelihood were shown. Additionally, a new learning scheme to jointly learn sample clusters and cluster-specific classifiers was adopted. Since it does not require the off-line setups to construct the cluster priors, the proposed tracker is more practical than the existing method in tracking generic objects. In various test videos, the inverse relation between the appearance prior and the likelihood was demonstrated, and the outperforming accuracy of the proposed tracker was compared with that of existing state-of-the-art methods. Future efforts will be focused on reducing the processing time by optimizing the source code, and adopting online learning and parallel-processing schemes to the current framework.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. *In CVPR*, 2006. 5

[2] S. Avidan. Support vector tracking. *In PAMI*, 26(8):1064–1072, 2004. 1

[3] S. Avidan. Ensemble tracking. *In CVPR*, 2005. 1

[4] B. Babenko, M. H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. *In CVPR*, 2009. 1, 5

[5] R. T. Collins and Y. Liu. On-line selection of discriminative tracking features. *In PAMI*, 27(10):1631–1643, 2005. 1

[6] A. Elgammal. Learning to track: conceptual manifold map for closed-form tracking. *In CVPR*, 2005. 2

[7] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *In Journal of Computer and System Sciences*, 55:119–139, 1997. 3, 5

[8] H. Grabner and H. Bischof. Online boosting and vision. *In CVPR*, 2006. 1, 6

[9] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. *In ECCV*, 2008. 1

[10] T. K. Kim, T. Woodley, B. Stenger, and R. Cipolla. Online multiple classifier boosting for object tracking. *In CVPR Workshop on OLCV*, 2010. 2, 4

[11] J. Kwon and K. M. Lee. Visual tracking decomposition. *In CVPR*, 2010. 2, 5

[12] H. Lim, V. Morariu, O. I. Camps, and M. Sznaier. Dynamic appearance modeling for human tracking. *In CVPR*, 2006. 2

[13] B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski. Robust and fast collaborative tracking two stage sparse optimization. *In ECCV*, 2010. 2

[14] X. Mei and H. Ling. Robust visual tracking using l1 minimization. *In ICCV*, 2009. 2

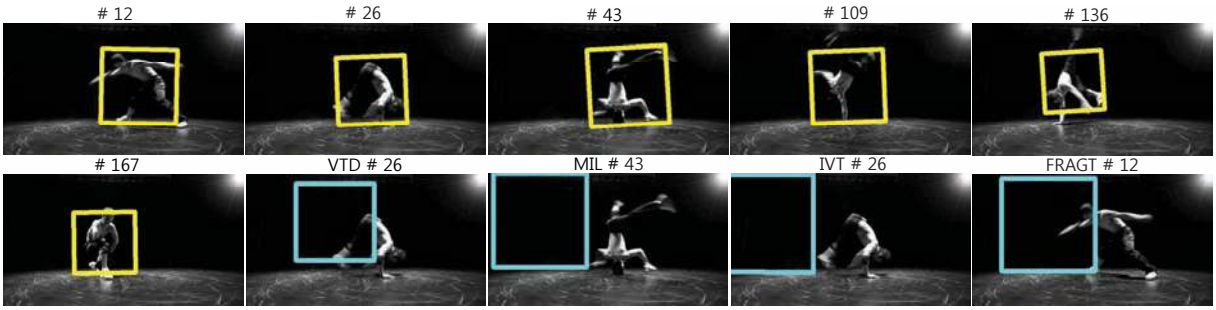
[15] T. M. Mitchell. Machine learning. *McGraw-Hill International Edition*, 1997. 3

[16] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang. Incremental learning for robust visual tracking. *In IJCV*, 77:125–141, 2008. 1, 5

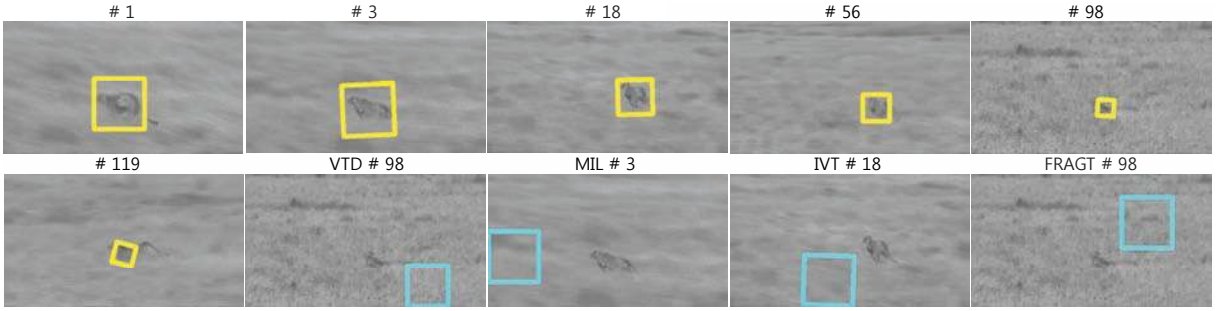
[17] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. *In CVPR*, 2011. 2, 4, 7

[18] S. Stalder, H. Grabner, and L. V. Gool. Beyond semi-supervised tracking: tracking should be as simple as detection, but not simpler than recognition. *In CVPR Workshop on OLCV*, 2009. 1

[19] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *In CVPR*, 2001. 5

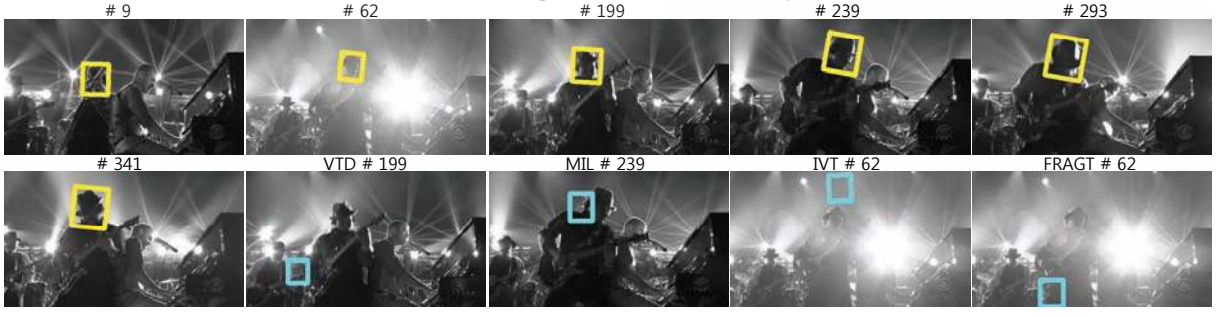


(a) *b-boy* sequence (pose change)

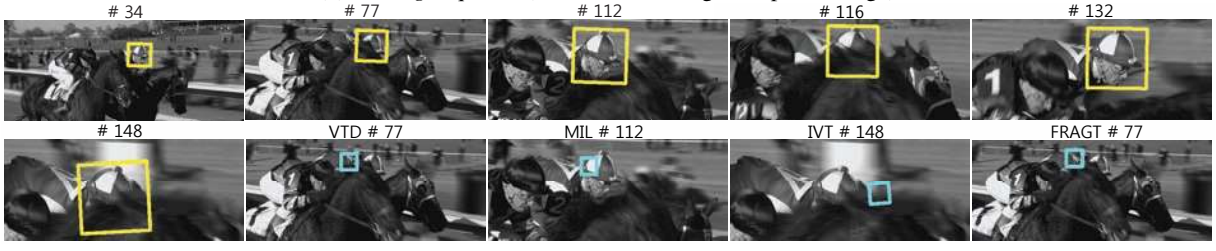


(b) *cheetah* sequence (shape deformation)

(c) *lighting* sequence (illumination change)



(d) *shaking* sequence (illumination change and pose change)



(e) *horse-race* sequence (occlusion)

Figure 6. Tracking results in five challenging video sequences. The estimated states are depicted using yellow rectangles for the proposed method (ARTOPT), and blue rectangles for other state-of-the-art methods (VTD, MIL, IVT, and FRAGT).