Visual Tracking via Adaptive Tracker Selection with Multiple Features

Ju Hong Yoon[1], Du Yong Kim[2], and Kuk-Jin Yoon[1]

[1] Computer Vision Lab., Gwangju Institute of Science and Technology, Korea
[2] Applied Computing Lab., Gwangju Institute of Science and Technology, Korea
{jhyoon,kjyoon}@gist.ac.kr, duyongkim@gmail.com

Abstract. In this paper, a robust visual tracking method is proposed to track an object in dynamic conditions that include motion blur, illumination changes, pose variations, and occlusions. To cope with these challenges, multiple trackers with different feature descriptors are utilized, and each of which shows different level of robustness to certain changes in an object's appearance. To fuse these independent trackers, we propose two configurations, tracker selection and interaction. The tracker interaction is achieved based on a transition probability matrix (TPM) in a probabilistic manner. The tracker selection extracts one tracking result from among multiple tracker outputs by choosing the tracker that has the highest tracker probability. According to various changes in an object's appearance, the TPM and tracker probability are updated in a recursive Bayesian form by evaluating each tracker's reliability, which is measured by a robust tracker likelihood function (TLF). When the tracking in each frame is completed, the estimated object's state is obtained and fed into the reference update via the proposed learning strategy, which retains the robustness and adaptability of the TLF and multiple trackers. The experimental results demonstrate that our proposed method is robust in various benchmark scenarios.

Keywords: Visual tracking, multiple features, transition probability matrix, robust likelihood function, tracker interaction, appearance learning.

1 Introduction

Visual tracking is an important research topic in the field of computer vision because of its wide application in surveillance, robotics, human-computer interface, vehicle tracking, medical imaging, and so on. Due to the characteristics of the various vision applications, visual tracking is required to deal with practical challenges originating from dynamic circumstances such as object and/or background illumination changes, object pose variation, occlusions, and motion blur [22] as shown in Fig. 1. Therefore, many researchers have discussed how to improve the performance of visual trackers by using multiple features in an efficient manner [4–15]. Despite decades of research, how to use multiple features to achieve a robust visual tracking is still an open problem.

In this paper, we propose a new visual tracking framework that fuses multiple trackers and features intelligently. We assume that each feature shows strong discriminating power under the conditions to which it is best suited. For instance, the histogram of

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part IV, LNCS 7575, pp. 28–41, 2012.

(a) Pose and illumination changes (♯120,297) (b) Occlusion and motion blur (♯771,873)

Fig. 1. Example of our tracking results in *tiger1* and *liquor* seqs

oriented gradients (HOG) is robust to pose variation when the object's shape is consistent [21]; the Haar-like feature is robust to occlusion since it is part-based [20]; and the intensity is a good enough feature descriptor when there is small amount of pose variation and noise, because it contains redundant visual information [3].

In our method, each tracker is implemented with a different feature based on a particle filter. Our objective is then to integrate these multiple trackers and features to achieve robust visual tracking in dynamic environment changes. To achieve the efficient fusion, we propose two configurations, 1) Tracker Selection and 2) Tracker Interaction, in a Bayesian framework. The tracker selection chooses one of the tracking results from the multiple trackers according to tracker probability. The tracker that has the highest tracker probability is selected. The tracker interaction provides communication between the trackers based on a transition probability matrix (TPM) [2] with a conventional resampling technique [25]. The purpose of the tracker interaction is to prevent unreliable trackers from drifting. Since each tracker is implemented based on the particle filter, the interaction between the trackers is represented by three actions: keeping its own samples, taking samples from other trackers, and giving samples to other trackers. Here, the role of the TPM is to determine the aforementioned actions of each tracker.

The changes in an object's appearance affect the reliability of trackers. Hence, we need to reflect the variations in the reliability of trackers in the tracker fusion by updating the TPM and tracker probability. The update is executed in a recursive Bayesian form based on a tracker likelihood function (TLF) that measures the current fidelity of a tracking output from each tracker. We consider two terms, flexibility and stability when designing the TLF. This concept is successfully used in [4]. To embody this concept, we propose using two types of appearance models. The first focuses on flexibility and is computed based on recent object appearances that reflect an instantaneous object appearance. The second appearance model is obtained by using a reconstructed appearance based on an appearance dictionary, i.e., a set of representative appearance templates [16]. Due to the reconstructed appearance, we can measure robustly each tracker's reliability, although occlusion or outliers exist in the object's appearance. Hence, the latter appearance model is more stable and conservative than the former. These reference properties are maintained via the proposed learning strategy.

The contributions of this paper are summarized as follows. First, we propose a new tracking framework to integrate multiple trackers and features that consist of tracker selection and interaction. Second, a robust TLF is proposed to measure tracker reliability robustly even though occlusions or outliers occur in the object's appearance. Third,

a simple but effective learning strategy is proposed to maintain the references used in each tracker and the TLF.

The remainder of the paper is organized as follows. We explain the differences between our method and related studies in Section 2. The overall framework and its components are specifically explained in Section 3. Experimental results are shown in Section 4 with the performance evaluation of the proposed tracking method and comparison with the state-of-the-art trackers.

2 Related Work

During the last decade, many elaborate tracking frameworks have been proposed to achieve robust visual tracking by using multiple features [4–15]. Among these, some of studies that are closely most related to our approach are briefly explained in this section. The methods that integrate trackers or features have been proposed using Condensation [1] or other Bayesian filters; they can be categorized into three kinds: a single tracker with multiple observations [6–8, 14], and multiple trackers in parallel [9, 10, 15] or in cascade[11, 12].

In [6–8, 14], the multiple feature observations are fused into a product form within a single tracker framework, and the reliability of each feature is not measured. However, measuring the reliability of each feature is important since some features are very weak to specific changes in an object's appearance, such as motion blur, illumination change, etc. In our method, the current fidelity of each tracker with a different feature is reflected in a fusion process to achieve robustness. Du et al. [11] proposed using Linked Hidden Markov Models which enable the conjunction of particle filters with a belief propagation. Thus, trackers can interact with other trackers, and each tracker is connected with certain trackers in a fixed order to achieve robust performance. The approach in [12] sequentially estimates the rectangular template, color space, color distribution, and the contour of the object. Finally, all of the samples are unified to compute the final estimated state at each frame. The tracker order is critical to the performance of both methods: if the order is changed, then the performance will be degraded. However, in our method, each tracker operates in parallel and independently. Hence, other features or trackers can be added easily if the trackers are formulated within Condensation or Bayesian filters; moreover, they are fully connected and interact with each other via the TPM, and therefore, the order is not an issue. In [9, 10], the authors proposed combining two trackers based on tracker interaction, and fusing the tracking outputs. In contrast, our method provides a more general multiple tracker integration because it can fuse more than three trackers without modification. Kwon et al. [15] proposed using different trackers whose observations are hue, saturation, intensity, and edge, respectively. All trackers operate in parallel and interact with each other. However, it may seem ad-hoc because the interaction is conducted if the uniformly generated value is smaller than the selected threshold. In contrast, we try to avoid the heuristic interaction. In our method, the interaction is conducted based on the TPM which represents how trackers interact with other trackers; the TPM is recursively updated to cope with the current fidelity of each tracker, which may change at each frame.

3 Proposed Visual Tracking Framework

The purpose of visual tracking is to estimate an object motion state x_k in image sequences. To formulate this problem, we adopt Bayesian filtering in which the posterior probability $p(x_k|Z_k)$ is recursively updated as follows:

$$p(x_k|Z_{1:k}) \propto p(Z_k|x_k) \int p(x_k|x_{k-1}) p(x_{k-1}|Z_{1:k-1}) dx_{k-1}, \tag{1}$$

where the state x_k is represented as $x_k = [p_{X,k}, p_{Y,k}, \theta_k, s_k, \alpha_k, \phi_k]^{\mathrm{T}}$ where each parameter denotes (X, Y) position, rotation angle, scale, aspect ratio, and skew direction, respectively. Z_k denotes an observation. $p(x_k|x_{k-1})$ represents the object motion model that transits the previous state x_{k-1} to the new state x_k. $p(Z_k|x_k)$ is the observation likelihood that measure similarity between the state and the observation.

In our method, we utilize multiple observation models based on multiple features. Each observation(feature) model is assigned to one single tracker. If we use M features, M trackers are used totally. To efficiently unify M trackers into one framework, we consider two configurations, i.e., 1) Tracker Selection and 2) Tracker Interaction based on the interacting multiple model (IMM) filter [17] and (1) is reformulated as

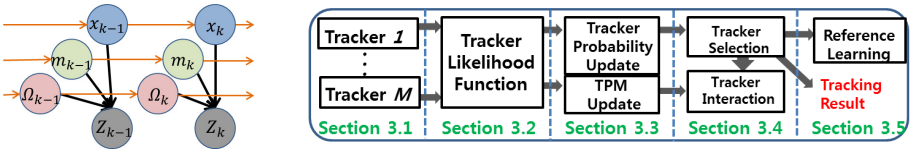
$$\begin{aligned} p(x_k|Z_{1:k}) &\triangleq \sum_{i=1}^{M} T_k^{(i)} \underbrace{p(x_k|Z_{1:k}, m_k = i)}_{\text{i-th tracker posterior probability}} \propto \sum_{i=1}^{M} T_k^{(i)} \underbrace{p(Z_k|x_k, m_k = i)}_{\text{observation likelihood model}} \\ &\times \int \underbrace{p(x_k|x_{k-1}, m_k = i)}_{\text{motion model}} \underbrace{\sum_{j=1}^{M} \omega_{k-1|k-1}^{(j,i)} p(x_{k-1}|Z_{1:k-1}, m_{k-1} = j)}_{\text{interacted prior}} dx_{k-1}, \end{aligned} \tag{2}$$

where $m_k \in \{1, ..., M\}$ is a tracker index and each tracker is formulated with the interacted prior, the motion model, and the observation likelihood. These trackers are integrated on interaction coefficients $\omega_{k-1|k-1}^{(j,i)} \triangleq P\{m_k = i|m_{k-1} = j, Z_{1:k-1}\}$ expressed in a form of matrix called a transition probability matrix (TPM) $\Omega_{k-1} = [\omega_{k-1}^{(j,i)}], i, j = 1, ..., M$. All tracker posterior probabilities are unified with tracker probabilities $T_k^{(i)} \triangleq P\{m_k = i|Z_{1:k}\}$ ($P\{\}$ denotes the discrete probability). Then, from (2), we can obtain the tracking result $\hat{x}_k$ as

$$\begin{aligned} &\hat{x}_k = \hat{x}_k^{(\hat{m}_k)}, \;\; \hat{m}_k = arg \max_i T_k^{(i)}, \\ &\hat{x}_k^{(i)} = arg \max_{x_k} p(x_k|Z_{1:k}, m_k = i), \;\; i = 1, ..., M, \end{aligned} \tag{3}$$

where M tracking outputs $\hat{x}_k^{(i)}$ are obtained by the maximum a posteriori estimate from the posterior probability of each tracker $p(x_k|Z_{1:k}, m_k = i)$.

To estimate the current object state x_k based on (2), we also need to estimate the TPM Ω_k and the selected tracker index m_k as shown in Fig. 2. For the practical implementation, we approximate multiple trackers based on the particle filter and estimate the object state $\hat{x}_k^{(i)}$ in 3.1. Since the object appearance and background continuously changes, the tracker probability and interaction coefficients are adaptively updated by



(a) Graphical model (b) Tracking procedure

Fig. 2. Left: (a) Graphical model of our method: Hidden(state x_k, tracker index m_k, TPM Ω_k); Observation (image frame, Z_k). Right: (b) Overall procedure of the proposed tracking algorithm

evaluating each tracker reliability measured by the robust TLF in 3.2 and 3.3. After that, we integrate multiple trackers based on both updated tracker probabilities and the TPM. We obtain one tracking result according to the tracker probabilities as in (3). The tracker interaction is conducted based on the updated TPM and the selected tracking result via the proposed tracker interaction in 3.4. The tracking result is fed into the reference update to reflect changes of the object appearance via the proposed learning strategy in 3.5. The overall procedure of the proposed method is shown in Fig. 2.

3.1 Single Tracker

Each tracker is formulated based on the interacted prior, motion model, and observation likelihood model as expressed in (2).

Interacted Prior: The interacted prior in (2) is computed based on the TPM via the proposed interaction method in 3.4.

Motion Model: To achieve a robust state motion transition $p(x_k|x_{k-1}, m_k = i)$, we simply adopt two motion models (zero- and first-order motion model) in terms of (X,Y) translation. The zero-order motion is identical to the random walk motion. The first-order motion utilizes the prior information of (X,Y) translation that is simply obtained by computing the difference between estimated X and Y positions at $k-1$ and $k-2$. More efficient usage of multiple motion models for visual tracking is referred to [23].

Observation Likelihood Model: A different feature is used to represent the object appearance in each tracker. The object appearance is extracted from the image as

$$Z_k^{(i)} = Vec(F^{(i)}(I(x_k))) + v_k^{(i)}, \quad i = 1, ..., M \tag{4}$$

where $Vec()$ is vectorization; $I(x_k)$ denotes an image template based on x_k; $F^{(i)}()$ is the i-th feature extraction; $v_k^{(i)}$ is unknown noise. To deal with this high dimensionality of appearance, we use the incremental PCA subspace learning method [3]. In the incremental PCA based observation likelihood model, we compute the mean and principal eigenvectors and incrementally update them to cope with the object appearance changes as proposed in 3.5. Based on the template mean $\bar{O}^{(i)}$ and L principal eigenvectors $g_l^{(i)}, l = 1, ..., L$, the observation likelihood based on i-th tracker is given as

$$\begin{aligned} &p(Z_k|x_k, m_k = i) = exp(-\rho_T \| Z_k^{(i)} - \textstyle\sum_l c_l g_l^{(i)} \|^2), \\ &c_l = (g_l^{(i)})^{\mathrm{T}}(Z_k^{(i)} - \bar{O}^{(i)}), \quad l = 1, ..., L, \end{aligned} \tag{5}$$

where ρ_T is the control parameter and c_l is the coefficient from the projection of the template mean to each principal eigenvector.

Particle Approximation: The i-th tracker posterior probability $p(x_k|Z_{1:k}, m_k = i)$ is approximated as a set of N samples as $\{x_{q,k}^{(i)}, w_{q,k}^{(i)}\}_{q=1}^{N}$ where $x_{q,k}^{(i)}$ and $w_{q,k}^{(i)}$ are the state sample and sample weight, respectively. Then, each tracker estimates the object state $\hat{x}_k^{(i)}$ in (3). As a result, we obtain M candidate states (i.e., $\hat{x}_k^{(i)}, i = 1, ..., M$) from M trackers.

In the next subsection, the tracker reliabilities are measured based on the M candidate states by using the robust TLF.

3.2 Robust Tracker Likelihood Function (TLF) on Flexibility and Stability

We can compute the normalized j-th feature appearance $z_k^{(i,j)}$ with respect to i-th tracker output as

$$z_k^{(i,j)} = \frac{Vec(F^{(j)}(I(\hat{x}_k^{(i)})))}{\|Vec(F^{(j)}(I(\hat{x}_k^{(i)})))\|}, \quad i, j = 1, ..., M \tag{6}$$

where $z_k^{(i,j)} \in \Re^{d^{(j)}}$ and $d^{(j)}$ is the dimension of j-th feature. In this section, we measure these appearances based on the i-th tracker output to analyze the tracker reliability and adaptively reflect the measured reliability in the tracker probability and TPM update in 3.3. This measure is called the tracker likelihood function (TLF) in (8) in which we consider two appearance models to manage the abrupt appearance changes of the object as well as the occlusion or outliers.

First, we assume that the recent object appearance is similar to the current object appearance. We call this reference template an "instantaneous reference" made of the recent object appearance and denote it as $f_{I,k}^{(j)}$ where j is the feature index. In this paper, we obtain this reference by simply averaging the object appearances in recent frames.

Secondly, to achieve stability in occlusions or other temporal outliers, we consider the reconstructed appearance that is a linear combination of the appearances called a "reconstructing reference" $f_{R,k}^{(j)}$ with coefficients $\alpha_k^{(i,j)}$ where i is the tracker index. To compute these coefficients, we adopt L1 minimization because it is robust to a wide range of image corruption, especially to occlusions [16, 19].

$$\min \|D_k^{(j)} c_k^{(i,j)} - z_k^{(i,j)}\|_2^2 + \lambda \|c_k^{(i,j)}\|_1 \tag{7}$$

where $D_k^{(j)} = [f_{R,k}^{(j)}, I^{(j)}]$ consists of a j-th feature dictionary and non-object (trivial) appearance template sets, i.e, $I^{(j)} \in \Re^{d^{(j)} \times d^{(j)}}$ [16]. The corresponding coefficients are represented as $c_k^{(i,\zeta)} = [\alpha_k^{(i,j)^{\mathrm{T}}}, \beta_k^{(i,j)^{\mathrm{T}}}]^{\mathrm{T}}$ where $\beta_k^{(i,j)} \in \Re^{d^{(j)}}$ are non-object coefficients. Here, $f_{R,k}^{(j)} = [f_{1,k}^{(j)}, ..., f_{r,k}^{(j)}] \in \Re^{d^{(j)} \times r}$ denotes the dictionary of j-th feature containing a set of r normalized representative appearance templates and $\alpha_k^{(i,j)} = [\alpha_{1,k}^{(i,j)}, ..., \alpha_{r,k}^{(i,j)}]^{\mathrm{T}} \in \Re^r$ denotes the object appearance coefficients. Then, we can obtain the j-th feature reconstructed appearance for the i-th tracker tracking result as $f_{R,k}^{(j)} \alpha_k^{(i,j)}$.

Based on the two appearance models, we calculate the TLF as follows:

$$\begin{aligned} &p(Z_k|m_k = i, \Omega_{k-1}, Z_{1:k-1}) \triangleq p_{TLF}(Z_k|\hat{x}_k^{(i)}) \approx p_I(Z_k|\hat{x}_k^{(i)})p_R(Z_k|\hat{x}_k^{(i)}) \\ &= \prod_{j=1}^{M} p(Z_k|\hat{x}_k^{(i)}, f_{I,k}^{(j)})p(Z_k|\hat{x}_k^{(i)}, f_{R,k}^{(j)}) \propto exp(-\rho(E_{I,k}^{(i)} + E_{R,k}^{(i)})), \end{aligned} \tag{8}$$

where ρ is the control parameter and

$$E_{I,k}^{(i)} = \sum_{j=1}^{M} (f_{I,k}^{(j)} - z_k^{(i,j)})^{\mathrm{T}} (f_{I,k}^{(j)} - z_k^{(i,j)}) \tag{9}$$

$$E_{R,k}^{(i)} = \sum_{j=1}^{M} (f_{R,k}^{(j)} \alpha_k^{(i,j)} - z_k^{(i,j)})^{\mathrm{T}} (f_{R,k}^{(j)} \alpha_k^{(i,j)} - z_k^{(i,j)}) \tag{10}$$

3.3 The Update of Tracker Probability and TPM

According to the tracker reliability affected by the dynamic environments in visual scene, the tracker probabilities and their interaction should changes. Thus, we update tracker probability and the TPM based on current tracker reliabilities (represented by the TLF $p_{TLF}(Z_k|\hat{x}_k^{(i)})$ in (8)) as follows.

Tracker Probability Update: The tracker probability update is defined by considering tracker reliabilities and the interactions between trackers as

$$T_k^{(i)} = C^{-1} p_{TLF}(Z_k|\hat{x}_k^{(i)}) \sum_{j=1}^{M} \omega_{k-1}^{(j,i)} T_{k-1}^{(j)}, \tag{11}$$

where C is the normalization term. Hence, the sum of all tracker probabilities is 1.

TPM Update: According to [2], the TPM is assumed to be an unknown random matrix with some prior distribution. Hence, in a Bayesian framework, the TPM posterior probability $p(\Omega|Z_{1:k})$ can be represented as a recursive form

$$p(\Omega|Z_{1:k}) = \frac{p(Z_k|\Omega, Z_{1:k-1})}{p(Z_k|Z_{1:k-1})} p(\Omega|Z_{1:k-1}) \tag{12}$$

For the practical implementation, the TPM posterior is approximated based on a first-order, second-order, or numerical integration (NI) approach. Among them, the NI is more robust and accurate than other approaches [2]. In the NI, the TPM posterior is expressed as the set of N_Φ fixed grid samples, Φ_q, i.e., $\{\Phi_q, \kappa_{q,k}\}_{q=1}^{N_\Phi}$ where $\kappa_{q,k}$ is the sample weight and updated as (The derivation of (13) is given in detail in the supplement material)

$$\begin{aligned} \kappa_{q,k} &= \frac{T_{k-1}^{\mathrm{T}} \Phi_q \Lambda_k}{T_{k-1}^{\mathrm{T}} \Omega_{k-1} \Lambda_k} \kappa_{q,k-1}, \\ \Omega_k &= \left(\sum_{g=1}^{N_\Phi} \kappa_{g,k} \right)^{-1} \sum_{q=1}^{N_\Phi} \kappa_{q,k} \Phi_q, \end{aligned} \tag{13}$$

where Ω_k is the estimated current TPM, $\Lambda_k = [p_{TLF}(Z_k|\hat{x}_k^{(1)}), ..., p_{TLF}(Z_k|\hat{x}_k^{(M)})]^{\mathrm{T}}$ is the set of the TLFs, and $T_{k-1} = [T_{k-1}^{(1)}, ..., T_{k-1}^{(M)}]^{\mathrm{T}}$ is the set of the tracker probabilities. Each value of the TPM samples $\phi_q^{(j,i)} \in \Phi_q$ is chosen within $[0, 1]$ while satisfying $\sum_{j=1}^{M} \phi_q^{(j,i)} = 1$. In the experiments, the 216 TPM samples are used and fixed for all benchmark sequences, and they are given in the supplementary material due to the limitation of the paper length.

Algorithm 1. Tracker Interaction

given $\{x_{q,k}^{(i)}, w_{q,k}^{(i)}\}_{q=1}^{N},\ i = 1, ..., M$ ▷ Sample representation of i-the tracker
given $\omega_k^{(j,i)} \in \Omega_k,\ \ j, i = 1, ..., M$ ▷ Updated TPM
for $i = 1 : M$ **do**
  for $q = 1 : N$ **do**
    $w_{q,k}^{*(i)} = w_{q,k}^{(i)} Kernel(Hx_{q,k}^{(i)} - Hx_k, R)$
  end for
  $w_{q,k}^{*(i)} := w_{q,k}^{*(i)} / \sum_{g=1}^{N} w_{g,k}^{*(i)},\ q = 1, ..., N$
end for
for $i = 1 : M$ **do**
  $\tilde{x}_k^{(i)} = \phi$
  for $j = 1 : M$ **do**
    $X = Resampling(\{x_{q,k}^{(j)}, w_{q,k}^{*(j)}\}_{q=1}^{N},\ \ N \times \omega_k^{(j,i)})$
    $\tilde{x}_k^{(i)} := \tilde{x}_k^{(i)} \cup X$
  end for
end for
Output $\{x_{q,k}^{(i)}, \frac{1}{N}\}_{q=1}^{N} := \{\tilde{x}_{q,k}^{(i)}, \frac{1}{N}\}_{q=1}^{N}, i = 1, ..., M$ ▷ Interacted prior of i-th tracker
$H = \begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \end{bmatrix}$, Range: $R = \sqrt{(2 \times q_{\mathrm{x}})^2 + (2 \times q_{\mathrm{y}})^2}$

3.4 Multiple Tracker Integration via Tracker Selection and Interaction

The multiple trackers are integrated via the tracker selection and interaction based on the updated tracker probability and TPM in Section 3.3.

Tracker Selection. The tracker selection picks one tracker whose tracker probability is the highest among updated tracker probabilities in (3). The output of the selected tracker $\hat{x}_k$ is the estimated object motion state at the current frame.

Tracker Interaction. The trackers interact with each other based on the TPM via the proposed tracker interaction in Algorithm 1. First, before the interaction based on the TPM, we remove the samples far from the selected tracking result $\hat{x}_k$ in terms of the position by using the uniform kernel with respect to the range R defined in Algorithm 1 where q_{x} and q_{y} are standard deviations that are set according to the object translation motion along x- and y- coordinates. In this paper, R is at most 12. H is the position conversion matrix that extracts position parameters by $[p_{X,k}, p_{Y,k}]^{\mathrm{T}} = H\hat{x}_k$. Then, each tracker interacts based on the TPM and the conventional resampling technique [25]. Here, N is the number of samples used in each tracker. The TPM provides the information that how many samples are transferred or kept. For instance, $N \times \omega_k^{(i,i)}$ represents that $N \times \omega_k^{(i,i)}$ samples are kept in the i-th tracker sample set after interaction. $N \times \omega_k^{(j,i)}$ represents that $N \times \omega_k^{(j,i)}$ samples from the j-th tracker are transferred to the i-th tracker. If the i-tracker is robust for some frames, then $\omega_k^{(i,i)}$ becomes greater than $\omega_k^{(j,i)}, j \neq i$ after the TPM update. Hence, most of the i-th tracker samples are kept and the i-th tracker obtain a few samples from other trackers. Finally, We select samples according to the interaction coefficients $\omega_{k-1}^{(j,i)}$ of the TPM via resampling technique that

is conventionally used in the particle filtering so that reliable samples with high weights in each tracker will survive.

3.5 Reference Learning

In this paper, we also propose a simple but effective reference learning strategy. The three kinds of reference (i.e., tracker reference, instantaneous reference, and reconstructing reference) are incrementally updated based on the estimated M features that is obtained by $\hat{f}_k^{(j)} = z_k^{(\hat{m},j)}, j = 1, ..., M$ in (6) where $\hat{m}$ is the index of the selected tracker in (3).

Tracker References: We update each tracker reference by using the incremental PCA [3]. Here, the reference of the selected tracker is not updated whereas the references of all other trackers are updated. This concept provides two benefits — sufficient learning and the alleviation of accumulation error in the reference. As mentioned in [4], the accumulation error is inevitable when the reference is updated. However, if the reference represents the object appearance properly, the reference does not need to be updated. In our tracking scheme, we assume that the reference of the selected tracker represents the current appearance of the object well; thus, we only update the references of other (not selected) trackers.

Instantaneous References: Each instantaneous reference is obtained by taking mean value of the recently estimated appearance. Hence, it is simply computed by $f_{I,k+1}^{(j)} = MEAN(\hat{f}_{k-\delta}^{(j)}, ..., \hat{f}_k^{(j)})$ where δ is a constant value[1].

Reconstructing References: Inspired from [18], the reconstructing references $f_{R,k}^{(j)}$ are updated by measuring noises of the estimated features. In [18], they decide whether the reference is updated or not by exploring the non-zero elements in the non-object reference coefficients, if there is occlusion, the reference vector contains many non-zero elements. In this paper, the noises are measured based on the non-object reference coefficient vector $\beta_k^{(\hat{m},j)} \in \Re^{d^{(j)}}$ in (7) where $\hat{m}$ is the index of the selected tracker. We count non-zero elements in $\beta_k^{(\hat{m},j)}$, and then compute a noise ratio $R_{noise}^{(j)}$ by $R_{noise}^{(j)} = B^{(j)}/d^{(j)}$ where $B^{(j)}$ is the number of non-zero elements. When the noise ratio $R_{noise}^{(m)}$ is smaller than the certain threshold[1] γ, one representative appearance template (i.e., $f_{i,k}^{(j)} \in f_{R,k}^{(j)}$) that has the lowest coefficient is replaced by $\hat{f}_k^{(j)}$.

4 Experimental Results

Using the benchmark sequences[2][3][4], we evaluate our tracking method, which is simply called "Adaptive Tracker Selection (ATS)". We employ three trackers with different features to implement the ATS: Tracker 1, 2, and 3 are associated with HOG, intensity, and Haar-like feature, respectively. As mentioned in the introduction, we select these

[1] We used the parameter $\delta = 10$ and the threshold $\gamma = 0.3$ in the experiments.

[2] `http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml`

[3] `http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/`

[4] `http://www.gpu4vision.org`

features because they are enough to deal with occlusion, motion blur, pose variation, and illumination changes. Each tracker is implemented based on a particle filter using 300 samples. To approximate the TPM, we use 216 TPM samples, which are given in the supplementary material. The initial tracker probability $T_0^{(m)}$ is set to 1/3. The reconstructing reference of each feature contains 25 appearance templates. The parameter ρ in (8) is set to 2.

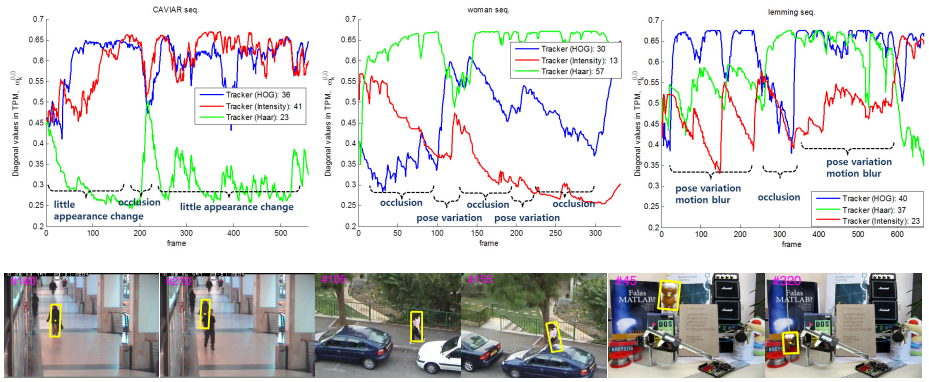
In 4.1, we discuss the computational time of our method. In 4.2, we analyze the ATS, focusing on the TPM, and show how the TPM manages multiple trackers of different features. Then, in 4.3, we present our comparative studies of a single tracker with multiple observations (S-MO) and a single tracker with a HOG, Intensity, and Haar-like observation, respectively (S-HOG, S-I, and S-Haar). It should be noted that they are implemented based on the IVT [3] framework with two motion models used in the ATS. Moreover, we compare our ATS to state-of-the-art trackers, i.e., MIL[27], TLD[28], L1Track[16], VTD[15], and PROST[4]. For the quantitative comparison, two performance indices are selected: mean distance errors and the percentage of correctly tracked frames according to a PASCAL score[4, 24]. The PASCAL score is obtained by evaluating to what extent the tracking template overlaps the ground truth template as a ratio [4]. Then, if the PASCAL score is greater than 0.5 in a certain frame, that frame is counted as a correctly tracked frame.

4.1 Computational Time

We implement our method using MATLAB 2010a. The most computation time is spent on feature extraction, especially for HOG and Haar features, and non-optimized code is used. The computation time of the S-MO method that uses the same features (HOG, Haar, Intensity) is comparable with that of the proposed ATS. When we use 900 samples, the S-MO takes about 2.76 sec/frame. The ATS takes about 3.26 sec/frame. Hence, it seems that our tracker integration scheme does not require a large amount of computational time.

4.2 Analysis on TPM

We explore how the TPM manages multiple trackers of different features for a certain change in object appearance. In Fig. 3, the changes in the values of diagonal coefficients ($\omega_k^{(i,i)}$) of the TPM are shown according to changes in object appearance over time. If $\omega_k^{(i,i)}$ decrease, then $\omega_k^{(j,i)}, n \neq m$ increases because $\sum_{j=1}^{M} \omega_k^{(j,i)} = 1$; hence, the i-th tracker becomes more dependent on other trackers. If the i-th tracker is not robust, then the $\omega_k^{(i,i)}$ value decreases. The maximum and minimum values of a diagonal coefficient are 0.7 and 0.2, respectively. Note that $\omega_k^{(1,1)}$, $\omega_k^{(2,2)}$, and $\omega_k^{(3,3)}$ denotes HOG tracker, Intensity tracker, and Haar tracker, respectively. In CAVIAR seq., there are few changes in appearance in many frames; hence, the Intensity tracker tracks the object most accurately. However, between ♯ 190 and ♯ 220, an occlusion occurs; thus, $\omega_k^{(1,1)}$ and $\omega_k^{(2,2)}$ decrease whereas $\omega_k^{(3,3)}$ increases. In woman seq. [26], an occlusion occurs repeatedly. When the occlusion first occurs, the $\omega_k^{(1,1)}$ and $\omega_k^{(2,2)}$ decrease and $\omega_k^{(3,3)}$ increases because the Haar feature is more robust than other features in occlusion. If there is only



(a) CAVIAR seq. (♯140,♯210) (b) woman seq. (♯105,♯155) (c) lemming seq. (♯45,♯320)

Fig. 3. 1) Changes of diagonal coefficients in the TPM: $\omega_k^{(1,1)}$(HOG:Blue), $\omega_k^{(2,2)}$(Intensity:Red), $\omega_k^{(3,3)}$(Haar:Green), 2) Numbers inside the box denote percentage of tracker selections

a small pose variation with no occlusion, the $\omega_k^{(1,1)}$ and $\omega_k^{(2,2)}$ start to increase. In lemming seq., there are frequent motion blurs and occlusions for short durations. In motion blurs, the HOG tracker is more robust than other features because the outer shape of the lemming is consistent. When the occlusion occurs around ♯300, $\omega_k^{(3,3)}$ increases.

4.3 Quantitative and Qualitative Evaluations

According to the overall results, the trackers that use on a single feature (i.e., MILTrack, TLD, L1Track) yield limited performances in various appearance changes as shown in Table 1 and Fig. 4. The trackers that use multiple featuers (i.e., VTD, PROST, S-MO, ATS) generate better results.

Occlusion: The target in the *CAVIAR* and *woman* seqs. undergoes heavy occlusions. As shown in Table 1, the S-Haar tracks the object perfectly because the Haar-like feature is robust when only occlusion exists. However, the S-HOG and S-I fail to track it since both these features are weak to occlusions. In Fig. 4, the VTD also fails to track in

Table 1. "A"("B"): "A"- the mean distance error in pixel; "B"- the percentage of correctly tracked frames based on Pascal score [24]. Red is the best result and blue is the second-best result.

Sequence	MIL	TLD	L1Track	VTD	PROST	S-I	S-HOG	S-Haar	S-MO	ATS
tiger1	15 (62)	12 (45)	44 (17)	44(21)	-	51 (37)	19 (66)	9 (80)	31 (39)	5 (94)
david	16 (62)	8 (96)	26 (58)	26 (68)	-	6 (90)	4 (91)	69 (36)	4 (100)	3 (100)
girl	27 (68)	26 (46)	13 (99)	15(98)	-	49 (50)	17 (87)	27 (74)	28 (76)	11 (100)
coke11	18 (32)	10 (48)	54 (5)	76(5)	-	63 (14)	9 (68)	12 (46)	10 (68)	7 (85)
CAVIAR	-	40 (19)	4 (100)	29 (41)	-	12 (65)	19 (41)	3 (100)	3 (100)	2 (100)
woman	-	-	252 (13)	108 (15)	-	92 (16)	124 (15)	4 (100)	4 (100)	2 (100)
board	115 (51)	142 (11)	255 (3)	83 (34)	39 (75)	146 (19)	16 (93)	35 (71)	84 (32)	16 (92)
box	196 (3)	17 (90)	150 (15)	66 (36)	13 (91)	104 (37)	10 (95)	69 (26)	86 (28)	9 (91)
lemming	15 (83)	146 (4)	212 (13)	83 (52)	25 (71)	20 (40)	68 (75)	174 (18)	111 (48)	11 (86)
liquor	165 (20)	20 (77)	181 (19)	103 (28)	21 (85)	521(22)	101 (29)	712 (22)	63 (23)	4 (98)

occlusions because it allows all the recent appearances including the occluded parts. In contrast, the ATS considers stability; hence, it can deal with the occlusion problem by measuring tracker reliability.

Pose Variation: In *girl* seq., the object repeatedly undergoes pose variation, but its outer shape is consistent. Thus, the S-HOG faithfully tracks the object as shown in Table1. The ATS successfully uses the HOG feature and shows a much better performance than the S-MO because the ATS measures the tracker reliability and updates the TPM. Based on the TPM, the more robust tracker can support other trackers.

Illumination Change: In *david* seq., the object appearance has illumination changes with little pose variation. The S-HOG robustly tracks the object because the outer shape of the object is consistent even when the illumination changes. The S-I also adapts well to the illumination changes because it is implemented based on the IVT [3]. Hence, the S-MO and ATS also perfectly track the object because these two features are robust.

Complex Changes: In practice, most image sequences contain various changes in appearance. Hence, the appropriate use of multiple features is very important. The *tiger1*

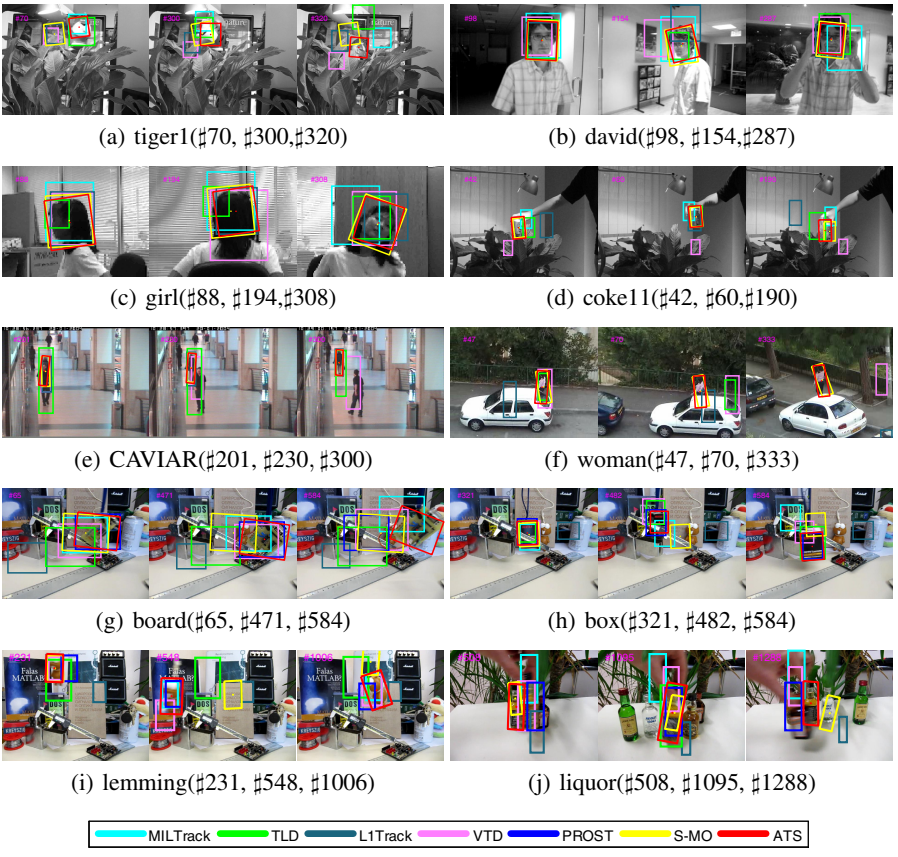


Fig. 4. Tracking results of different algorithms: ATS (the proposed method)

and *coke11* seqs. contain occlusions, illumination changes, and pose variations. The ATS measures the tracker reliability and updates the TPM as to which trackers can interact. Using this mechanism, the ATS shows better results than the S-MO, especially in these complicated situations. In the *board*, *box*, and *lemming* seqs., the object's appearance undergoes drastic motion blur and pose variation, but their shape is consistent. Hence, the S-HOG shows the best results in the *board* and *box* seqs. In terms of the PASCAL score, S-HOG faithfully tracks the object in the *lemming* seq. Overall, the ATS demonstrates the best performance because it utilizes not only the HOG feature but also other features in the appropriate situations. The *liquor* seq. contains the most severe motion blur and occlusions; hence, most of trackers fail to track the object. In contrast, PROST reliably tracks the object because it is designed to include flexibility, moderate adaption, and stability in the object appearance model to deal with various changes in appearance. Thus, the PROST tracker resembles the ATS, which also considers the flexibility and stability but in a different manner. In particular, in ATS the reliability and flexibility information are used more efficiently because it employs multiple features with multiple tracker based on the interactions. This leads to better results.

5 Conclusions

In this paper, we propose a robust visual tracking method that integrates multiple trackers based on multiple features via tracker interaction and selection. The tracker interaction is conducted based on the TPM and prevents individual tracker divergence. The TPM update and tracker selection are computed by investigating each tracker's reliability based on the TLF. To cover various kinds of changes in object appearance, the TLF is formulated based on instantaneous references for flexibility and reconstructing references for stability. Thus, the proposed tracking method can select the best among multiple trackers even if the object's appearance changes drastically. In addition, the proposed learning strategy enhances the performance of individual trackers and sustains the flexibility and stability of the two reference models in the TLF. The experimental results demonstrate that, in challenging sequences, the proposed tracking method tracks the object more robustly than other trackers.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2009-0065038).

References

1. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. IJCV 29(1), 5–28 (1998)
2. Jilkov, V.P., Li, X.R.: Online Bayesian estimation of transition probabilities for markovian jump systems. IEEE Transactions on Signal Processing 52(6), 307–315 (2004)
3. Ross, D., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. IJCV 77, 125–141 (2008)

4. Santner, J., Leistner, C., Saffari, A., Pock, T., Bischof, H.: Prost: Parallel robust online simple tracking. In: CVPR, pp. 723–730 (2010)
5. Spengler, M., Schiele, B.: Towards robust multi-cue integration for visual tracking. Machine Vision and Applications 14(1), 50–58 (2003)
6. Giebel, J., Gavrila, D.M., Schnörr, C.: A Bayesian Framework for Multi-cue 3D Object Tracking. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 241–252. Springer, Heidelberg (2004)
7. Brasnett, P., Mihaylova, L., Canagarajah, N., Mihaylova, L., Canagarajah, N., Bull, D.: Particle filtering with multiple cues For object tracking. In: Proc. of SPIE's Annual Symp. EI ST, pp. 430–441 (2005)
8. Wang, H., Suter, D.: Efficient visual tracking by probabilistic fusion of multiple cues. In: International Conference on Pattern Recognition, pp. 892–895 (2006)
9. Leichter, I., Lindenbaum, M., Rivlin, E.: A general framework for combining visual trackers - the "black boxes" approach. IJCV 67(3), 343–363 (2006)
10. Badrinarayanan, V., Perez, P., Clerc, F.L., Oisel, L.: Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues. In: ICCV, pp. 1–8 (2007)
11. Du, W., Piater, J.: A Probabilistic Approach to Integrating Multiple Cues in Visual Tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 225–238. Springer, Heidelberg (2008)
12. Moreno-Noguer, F., Sanfeliu, A., Samaras, D.: Dependent multiple cue integration for robust tracking. PAMI 30(4), 670–685 (2008)
13. Stenger, B., Woodley, T., Cipolla, R.: Learning to track with multiple observers. In: CVPR, pp. 2647–2654 (2009)
14. Zelniker, E.E., Hospedales, T.M., Gong, S., Xiang, T.: A unified Bayesian framework for adaptive visual tracking. In: BMVC, pp. 18.1–18.11 (2009)
15. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR, pp. 1269–1276 (2010)
16. Mei, X., Ling, H.: Robust visual tracking using l1 minimization. In: ICCV, pp. 1436–1443 (2009)
17. Bar-Shalom, Y., Li, X.R., Kirubarajan, T.: Estimation with applications to tracking and navigation. Wiley, New York (2001)
18. Mei, X., Ling, H., Wu, Y., Blasch, E., Bai, L.: Minimum error bounded efficient l1 tracker with occlusion detection. In: CVPR, pp. 1257–1264 (2011)
19. Kim, S.-J., Koh, K., Lustig, M., Boyd, S., Gorinevsky, D.: An interior-point method for large-scale l1 regularized least squares. IEEE Journal on Selected Topics in Signal Processing 1(4), 606–617 (2007)
20. Yang, M.-H.: Face detection. In: Encyclopedia of Biometrics, pp. 303–308 (2009)
21. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
22. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Computing Surveys 38(4) (2006)
23. Cifuentes, C.G., Sturzel, M., Jurie, F., Brostow, G.J.: Motion models that only work sometimes. In: BMVC (2012)
24. Everingham, M., Van Gool, L.J., Williams, C.K.I., Winn, J.M., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV 88(2), 303–338 (2010)
25. Doucet, A., Godsill, S., Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing 10(3), 197–208 (2000)
26. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR, pp. 798–805 (2006)
27. Babenko, B., Yang, M.-H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR, pp. 983–990 (2009)
28. Kalal, Z., Matas, J., Mikolajczyk, K.: P-n learning: bootstrapping binary classifiers by structural constraints. In: CVPR, pp. 49–56 (2010)