Interacting Multiview Tracker

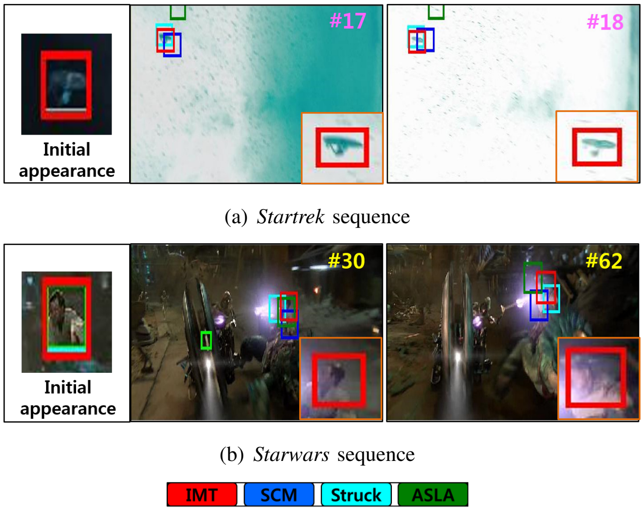
Ju Hong Yoon, Ming-Hsuan Yang, and Kuk-Jin Yoon

Abstract—A robust algorithm is proposed for tracking a target object in dynamic conditions including motion blurs, illumination changes, pose variations, and occlusions. To cope with these challenging factors, multiple trackers based on different feature representations are integrated within a probabilistic framework. Each view of the proposed multiview (multi-channel) feature learning algorithm is concerned with one particular feature representation of a target object from which a tracker is developed with different level of reliability. With the multiple trackers, the proposed algorithm exploits tracker interaction and selection for robust tracking performance. In the tracker interaction, a transition probability matrix is used to estimate dependencies between trackers. Multiple trackers communicate with each other by sharing information of sample distributions. The tracker selection process determines the most reliable one with the highest probability. To account for object appearance changes, the transition probability matrix and tracker probability are updated in a recursive Bayesian framework by reflecting the tracker reliability measured by a robust tracker likelihood function that learns to account for both transient and stable appearance changes. Experimental results on benchmark datasets demonstrate that the proposed interacting multiview algorithm performs robustly and favorably against state-of-the-art methods in terms of several quantitative metrics.

Index Terms—Object tracking, multiview representations, transition probability matrix, tracker interaction, multiple features.

1 Introduction

Visual tracking is an important and fundamental problem in computer vision, which finds a wide range of applications. For practical applications, it is essential for tracking algorithms to account for large appearance changes caused by illumination, pose variations, occlusions, and motion blurs [34] as shown in Figure 1. To cope with large appearance changes, numerous methods based on multiple features have been proposed for robust visual tracking where different types of features are used complementarily for different scenarios. However, although significant progress has been made in the past decade, it remains a difficult problem to exploit and integrate multiple features for robust visual tracking. The most essential task is how to combine features adaptively to account for appearance changes. Here, it should be noted that each feature has different characteristics against appearance changes. For instance, representations based on histogram of oriented gradients (HOG) [7] are robust to pose variations and appearance models based on Haar-like features [11] are effective to deal with occlusion.



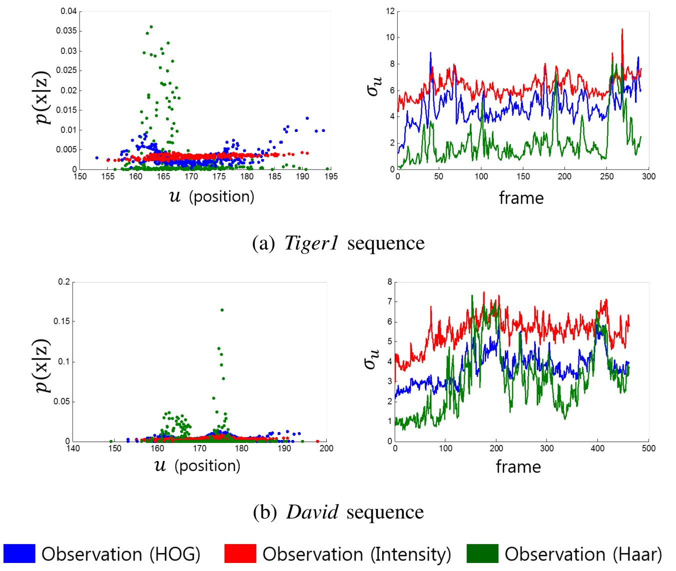
(a) *Startrek* sequence

(b) *Starwars* sequence

Fig. 1. Tracking results from videos with low contrast, drastic lighting changes, and pose variations (best viewed on high-resolution displays). The proposed algorithm (IMT) performs favorably against three top-ranked trackers (i.e., Struck [13], SCM [37], and ASLA [16]) from a recent benchmark study [31]. Quantitative results are presented in Table 3 and Figure 7.

In this paper, we propose a novel visual tracking algorithm that exploits and integrates multiple feature representations by considering their distinct characteristics to better account for appearance changes for robust tracking. Features with different and complementary representation strength are exploited, and multiple feature representations are used by trackers to describe object appearance. Each view (channel) of the multiview (multi-channel) feature learning framework is concerned with one particular representation of a target object [32]. Since each feature is defined in the different space, the likelihood probabilities by multiple trackers are computed at different scales. Consequently, the posterior distribution of each tracker is different even though the object state is defined in the same state space as illustrated in Figure 2. Hence, the scale difference should be taken into account when these posterior probabilities are used for object state estimation together. Nevertheless, it is difficult to assign the weights or to project different features to the same space. In this work, instead of

- *J. H. Yoon was with the School of Information and Communications, Gwangju Institute of Science and Technology, and he is currently with the Multimedia IP Center, Korea Electronics and Technology Institute, Seongnam-si, Gyeonggido, Republic of Korea. E-mail: jhyoon@keti.re.kr.*
- *M.-H. Yang is with in School of Engineering, University of California, Merced, USA. E-mail: mhyang@ucmerced.edu.*
- *K.-J. Yoon is with the School of Information and Communications, Gwangju Institute of Science and Technology, Gwangju, Republic of Korea. E-mail: kjyoon@gist.ac.kr.*



(a) *Tiger1* sequence

(b) *David* sequence

Fig. 2. As trackers are constructed using different features, corresponding posterior distributions ($p(\mathbf{x}|\mathbf{z})$) are of different scales. σ_u denotes the standard deviation of u.

combining multiple posterior distributions in a mixture form directly, we select the most reliable tracker at each instance. In addition, to prevent unreliable trackers from drifts, the trackers are designed to share their sample distribution information via interaction. Consequently, unreliable trackers receive more reliable samples from reliable ones.

The main components of the proposed algorithm are shown in Figure 3. At its core, a multiview feature representation [32] of a target object is proposed to account for appearance variations. Each tracker is developed based on one view (representation) of the target object. In addition, these trackers actively interact with each other to provide essential information of samples for effective visual tracking. To integrate multiple trackers for robust visual tracking, we propose the *tracker selection* and *tracker interaction* modules within a Bayesian framework. The tracker selection process determines the most reliable one in terms of tracker probabilities. The trackers share information of sample distributions through interaction based on a transition probability matrix and a resampling method to remove unreliable samples. Through this interaction, the visual drifting problem can be alleviated. In the proposed algorithm, we approximate the posterior distribution of each tracker by a set of samples. The interaction between trackers are implemented by two operations: retaining its own samples and receiving samples from other trackers. The objective of the transition probability matrix is to determine the number of samples for the aforementioned operations of each tracker.

In addition, to account for object appearance changes, we compute the tracker reliability and update the transition probability matrix to integrate trackers. The update of the transition probability matrix is formulated in a recursive Bayesian framework with a tracker likelihood function measuring each tracker reliability at each frame. The reliability of each tracker is used in the tracker interaction and selection processes. Both abrupt and stable appearance changes are considered in the tracker likelihood function. Abrupt appearance changes are modelled by multiple feature representations. On the other hand, stable appearance chances are described by a set of representative templates.

The contributions of the proposed interacting multiview tracking algorithm are as follows. First, we propose a novel tracking algorithm that integrates multiple trackers constructed by different feature representations via selection and interaction. Second, a robust likelihood function is proposed to measure tracker reliability which is of great importance for robust tracking. Third, a novel tracker interaction scheme is proposed by using the transition probability matrix with a resampling technique. Experimental results on large-scale benchmark datasets show that the proposed tracking algorithm performs favorably against state-of-the-art methods.

Preliminary results of this work were presented in [35]. In this paper, we provide more detailed descriptions and analysis of the proposed interacting multiview tracking algorithm with full derivation and detailed implementation. We compare with 10 top performing trackers on 51 benchmark sequences from [31]. In addition, three most related methods (CVT [22], MCS [4], and FCT [15]) are compared with detailed analysis. Furthermore, additional analysis is presented to demonstrate the effectiveness of the proposed interacting algorithm.

2 Related Work and Problem Context

Numerous tracking methods have been proposed using multiple features in the past decade. In this section, we discuss the approaches that are closely related to our work, where appearance models are constructed based on different features. The tracking algorithms that use multiple features can be categorized as a single tracker with multiple observations [6], [30], [36], cascade trackers [10], [26], and parallel trackers [22], [4], [3], [20].

Multiple Observations. Assuming that features are conditionally independent, multiple observations are combined in a product form for visual tracking [6], [30], [36]. However, reliability of each observation model (based on one different feature) in these approaches is not estimated for combination, which is of crucial importance as each feature is effective for describing certain appearance change (e.g., pose, illumination, and blur). In contrast, the reliability of each single tracker in this work is measured by the tracker likelihood function and reflected in the tracker integration process.

Cascade Trackers. In [10], a visual tracking method based on a coupled hidden Markov model to combine particle filters and visual cues is proposed. The approach in [26] sequentially estimates object states using the Kalman and particle filters with multiple features including rectangular shape, discriminative cues between foreground and background, color distribution, and object contour. The state predictions from the Kalman filter based on rectangular shape are passed to the other particle filters for sequential processing. These estimated states are combined in a Bayesian filter to determine the object location in each frame. In [26], the adopted features are dependent and

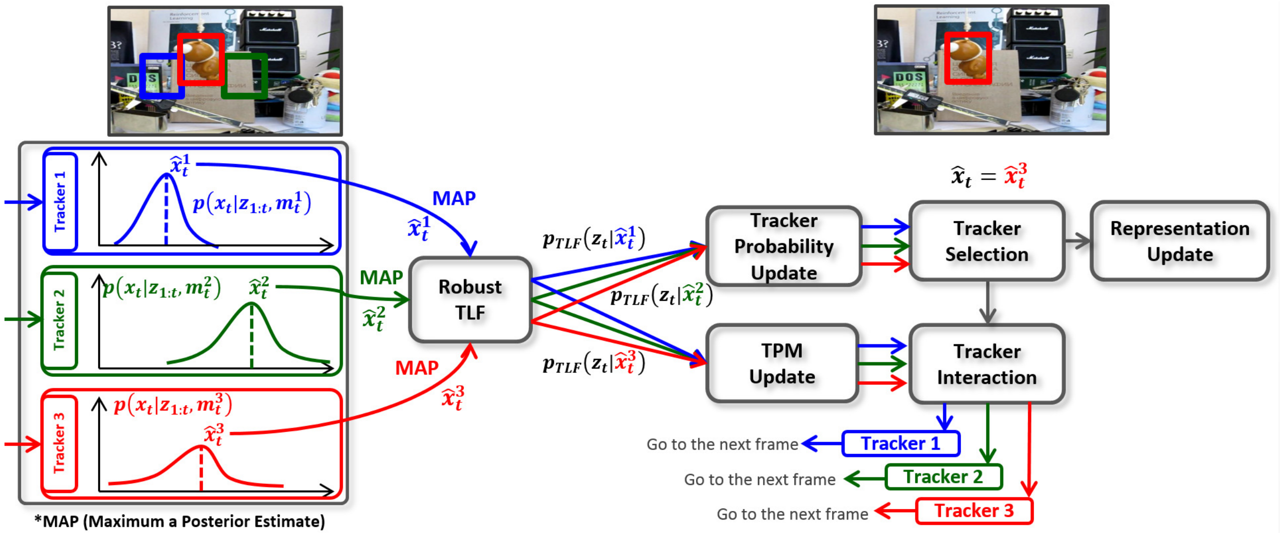


Fig. 3. Components of the proposed tracking algorithm.

the sequential state predictions from early stages are forwarded to the next stage for processing and integration. Thus, it is difficult to add new trackers using other features for different tasks. In the proposed algorithm, all trackers operate in parallel and interact with others, thereby facilitating addition of other trackers when necessary.

Parallel Trackers. In [22] and [4], two trackers with different features are combined and target locations are estimated by fusing tracking outputs [22] or selecting the most reliable one [4] based on covariance matrices of posterior distributions. However, a covariance matrix is not effective for measuring the reliability of a tracker when each posterior distributions are computed using observation models with different features (See Figure 2). Different from [22] and [4], the proposed algorithm selects the most reliable tracker via the proposed tracker likelihood function rather than covariance matrices. The tracker likelihood function is designed to deals with both abrupt and stable appearance changes. Furthermore, the proposed method provides a more general framework that accommodates more than two feature representations. In [20], multiple trackers constructed from four observation models (based on hue, saturation, intensity, and edge features) and two motion models are used to account for appearance and motion changes. While all trackers operate in parallel, the interaction among trackers is based on heuristics as uniform sampling is carried out with a threshold computed by a normalized likelihood ratio. In contrast, the proposed interaction scheme utilizes the transition probability matrix which represents probabilistic dependencies between trackers. Since the transition probability matrix is recursively updated by measuring the reliability of each tracker, unreliable trackers become more dependent on reliable ones to draw samples. As a result, the drifting problem with unreliable trackers is alleviated.

3 Algorithmic Overview

In the proposed algorithm, multiple interacting trackers based on different feature representations are used as shown in Figure 3. The reliability of trackers as well as their inter-dependencies are taken into account, and in turn the drawn samples from an individual tracker. First, each tracker estimates the object state independently, and then the reliability of each estimated object state is measured by the robust tracker likelihood function (TLF). These likelihoods are used to update the tracker probabilities to select the most reliable one. In addition, the result from the most reliable tracker is used to update the object appearance in the representation update. To compute the current dependencies of each tracker on other trackers, the transition probability matrix (TPM) is also updated by using the likelihoods from the TLF. By using the TPM, the tracker interaction makes unreliable trackers to depend more on the reliable ones to prevent the unreliable trackers from drifting. These interacted trackers are used to estimate the object state for the next frame.

4 State Estimation by Trackers

The goal of visual tracking is to estimate an object state given the observations $\mathbf{z}_{1:t} = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ up to time t. In this work, the object state is defined as $\mathbf{x}_t = [u_t, v_t, \theta_t, s_t, \alpha_t, \phi_t]^\top$ where (u_t, v_t), θ_t, s_t, α_t, and ϕ_t denote the position, rotation angle, scale, aspect ratio, and skew direction, respectively, to account for affine motion. To handle different kinds of appearance changes robustly, we exploit multiple features for observation models of multiple trackers. Let $m_t \in \{1, \ldots, M\}$ denote the index of M trackers constructed from M different features. For simplicity, we denote the i-th tracker index as $m_t^i \triangleq \langle m_t = i \rangle$. We propose algorithms for interaction and selection of M trackers. The tracker selection process determines the most reliable one at each frame. On the other hand, the drifting problem for the other M-1 trackers is alleviated via tracker interaction. Different from the method based on multiple models [5] where several motion predictions are used for feature point tracking, we exploit a number of representations in the proposed algorithm. Furthermore, we propose a novel tracker interaction approach using a particle filter.

The reliability of the i-th tracker is represented by the tracker probability $P\{m_t^i|\mathbf{z}_{1:t}\}$. The posterior distribution of object state $\mathbf{x}_t$ by the i-th tracker is computed by

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}, m_t^i) = \frac{p(\mathbf{z}_k|\mathbf{x}_t, m_t^i)p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, m_t^i)}{\int p(\mathbf{z}_t|\mathbf{x}_t, m_t^i)p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, m_t^i)d\mathbf{x}_t}, \quad (1)$$

where $p(\mathbf{z}_t|\mathbf{x}_t, m_t^i)$ is the observation model of the i-th tracker, and $p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, m_t^i)$ is a sample distribution by the i-th tracker given the observations up to time t-1 computed via interaction.

4.1 Tracker Interaction

The predicted distribution is computed with mixing probabilities $P\{m_{t-1}^j|m_t^i, \mathbf{z}_{1:t-1}\}$ by

$$\begin{aligned} &p(\mathbf{x}_t|\mathbf{z}_{1:t-1}, m_t^i) \\ &= \int p(\mathbf{x}_t|\mathbf{x}_{t-1}, m_t^i)\tilde{p}(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, m_t^i)d\mathbf{x}_{t-1}, \text{and} \\ &\tilde{p}(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, m_t^i) \\ &= \sum_{j=1}^{M} p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, m_{t-1}^j)P\{m_{t-1}^j|m_t^i, \mathbf{z}_{1:t-1}\}, \end{aligned} \quad (2)$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1}, m_t^i)$ is a motion model and $\tilde{p}(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, m_t^i)$ is an interacted prior distribution. The mixing probability is computed by

$$\begin{aligned} &P\{m_{t-1}^j|m_t^i, \mathbf{z}_{1:t-1}\} \\ &= \frac{P\{m_t^i|m_{t-1}^j, \mathbf{z}_{1:t-1}\}P\{m_{t-1}^j|\mathbf{z}_{1:t-1}\}}{\sum_{l=1}^{M} P\{m_t^i|m_{t-1}^l, \mathbf{z}_{1:t-1}\}P\{m_{t-1}^l|\mathbf{z}_{1:t-1}\}}. \end{aligned} \quad (3)$$

Note that both the tracker probability and interaction probability are defined by the discrete probability $P\{\cdot\}$ as the number of the trackers is finite, and they satisfy

$$\sum_i P\{m_t^i|\mathbf{z}_{1:t}\} = 1, \quad \sum_j P\{m_{t-1}^j|m_t^i, \mathbf{z}_{1:t-1}\} = 1.$$

Motion smoothness is a constraint often considered in feature point tracking [5] and thus model probabilities $P\{m_{t-1}^j|\mathbf{z}_{1:t-1}\}$ at time t-1 is useful. However, in visual tracking, it is not effective to use previous model (tracker) probabilities to compute an interacted prior distribution as occlusion, abrupt pose variations, or significant motion blurs scan cause abrupt appearance changes. Thus, we assume that all tracker probabilities are equal in the interaction scheme, and then approximate the mixing probability in (3) by

$$P\{m_{t-1}^j|m_t^i, \mathbf{z}_{1:t-1}\} \approx P\{m_t^i|m_{t-1}^j, \mathbf{z}_{1:t-1}\}, \quad (4)$$

where $P\{m_t^i|m_{t-1}^j, \mathbf{z}_{1:t-1}\}$ is an interaction probability.

4.2 Tracker Selection

We obtain the tracking result $\hat{\mathbf{x}}_t$ by selecting the most reliable tracker which has the highest tracker probability by

$$\begin{aligned} \hat{\mathbf{x}}_t &= \arg\max_{\mathbf{x}_t} \, p(\mathbf{x}_t|\mathbf{z}_{1:t}, \hat{m}_t), \\ \hat{m}_t &= \arg\max_{m_t^i} P\{m_t^i|\mathbf{z}_{1:t}\}, \ i = 1, \ldots, M. \end{aligned} \quad (5)$$

From (2), (4), and (5), both the tracker and interaction probabilities are utilized to estimate the object state and integrate multiple trackers. In addition, both tracker and interaction probabilities are updated.

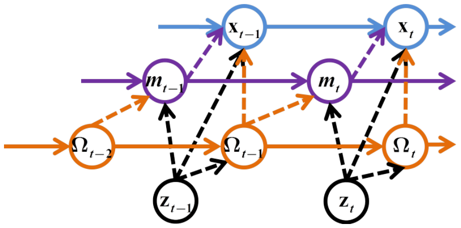


Fig. 4. Graphical model: Hidden variable (object state $\mathbf{x}_t$, a selected tracker index m_t, TPM $\mathbf{\Omega}_t$) and observation (observed image $\mathbf{z}_t$). 1) The TPM is updated using the current observation. 2) The tracker selection is conducted by updating the tracker probability based on the current observation and the TPM. 3) Each object state is estimated based on current observation, tracker selection, tracker interaction, and TPM.

5 Online Update

In contrast to existing methods based on multiple trackers [22], [4], we estimate not only object states but also the tracker and interaction probabilities for efficient and effective integration. Since different features are effective in accounting for certain appearance changes, multiple representations are used to construct trackers. In addition, the reliability of each tracker varies since each one is designed in different feature space. To achieve robust integration, we consider the reliability of each trackers in the interaction and selection processes.

For notation simplicity, we denote the tracker likelihood function of the i-th tracker as

$$p(\mathbf{z}_t|m_t^i, \mathbf{z}_{1:t-1}) \triangleq \Lambda_t^i. \quad (6)$$

Similarly, the notations of the tracker and interaction probabilities are denoted by

$$\begin{aligned} &P\{m_t^i|\mathbf{z}_{1:t}\} \triangleq T_t^i, \\ &P\{m_t^i|m_{t-1}^j, \mathbf{z}_{1:t-1}\} \triangleq \bar{\omega}_t^{j,i}. \end{aligned} \quad (7)$$

These notations are used in the following sections for update of tracker and interaction probabilities based on TLF.

5.1 Tracker Probability Update

The tracker probability is update as

$$\begin{aligned} P\{m_t^i|\mathbf{z}_{1:t}\} &= \frac{p(\mathbf{z}_t|m_t^i, \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} P\{m_t^i|\mathbf{z}_{1:t-1}\} \\ &= \frac{p(\mathbf{z}_k|m_t^i, \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \times \\ &\quad \sum_{j=1}^{M} P\{m_t^i|m_{t-1}^j, \mathbf{z}_{1:t-1}\}P\{m_{t-1}^j|\mathbf{z}_{1:t-1}\}, \end{aligned} \quad (8)$$

where the total probability $p(\mathbf{z}_t|\mathbf{z}_{1:t-1})$ is expressed by

$$\begin{aligned} p(\mathbf{z}_t|\mathbf{z}_{1:t-1}) &= \sum_{i=1}^{M} p(\mathbf{z}_t|m_t^i, \mathbf{z}_{1:t-1}) \times \\ &\quad \sum_{j=1}^{M} P\{m_t^i|m_{t-1}^j, \mathbf{z}_{1:t-1}\}P\{m_{t-1}^j|\mathbf{z}_{1:t-1}\}. \end{aligned} \quad (9)$$

Based on (8) with the notations in (6) and (7), the sequential tracker probability update is described by

$$T_t^i = \frac{\Lambda_t^i \sum_{l=1}^{M} \bar{\omega}_{t-1}^{l,i} T_{t-1}^l}{\sum_{j=1}^{M} \Lambda_t^j \sum_{l=1}^{M} \bar{\omega}_{t-1}^{l,j} T_{t-1}^l}. \tag{10}$$

5.2 Transition Probability Matrix Update

Figure 4 shows the graphical model of the proposed algorithm based on multiple interacting trackers. A set of interaction probabilities is expressed in a transition probability matrix $\mathbf{\Omega}$ which describes how trackers affect each other by

$$\mathbf{\Omega} = \left[\omega^{j,i}\right]_{M\times M} = \begin{bmatrix} \omega^{1,1} & \cdots & \omega^{1,M} \\ \vdots & \ddots & \vdots \\ \omega^{M,1} & \cdots & \omega^{M,M} \end{bmatrix}, \tag{11}$$

where $\mathbf{\Omega}$ is an unknown matrix from some given prior distributions. The estimated $\bar{\mathbf{\Omega}}_t$ is computed by the minimum mean squared error based on its posterior distribution,

$$\bar{\mathbf{\Omega}}_t = \left[\bar{\omega}_t^{j,i}\right]_{M\times M} \triangleq E[\mathbf{\Omega}|\mathbf{z}_{1:t}] = \int \mathbf{\Omega} p(\mathbf{\Omega}|\mathbf{z}_{1:t}) d\mathbf{\Omega}. \tag{12}$$

The goal is to estimate the posterior distribution of the TPM within the Bayesian framework [17],

$$p(\mathbf{\Omega}|\mathbf{z}_{1:t}) = \frac{p(\mathbf{z}_t|\mathbf{\Omega}, \mathbf{z}_{1:t-1})}{p_{\mathbf{\Omega}}(\mathbf{z}_t|\mathbf{z}_{1:t-1})} p(\mathbf{\Omega}|\mathbf{z}_{1:t-1}), \tag{13}$$

where the TPM observation model $p(\mathbf{z}_t|\mathbf{\Omega}, \mathbf{z}_{1:t-1})$ is derived in (16) by approximating the unknown $\mathbf{\Omega}$ with $\bar{\mathbf{\Omega}}_{t-1}$, and $\bar{\mathbf{\Omega}}_{t-1}$ is the best estimate of the unknown $\mathbf{\Omega}$ at time t-1 [17]. Thus, the TLF with the unknown $\mathbf{\Omega}$ is equal to the TLF in (6) and the tracker probability with the unknown $\mathbf{\Omega}$ is equal to the tracker probability in (7) as follows.

$$\begin{aligned} &p(\mathbf{z}_t|m_t^i, \mathbf{\Omega}, \mathbf{z}_{1:t-1}) \approx p(\mathbf{z}_t|m_t^i, \mathbf{z}_{1:t-1}) = \Lambda_t^i, \\ &P\{m_{t-1}^i|\mathbf{\Omega}, \mathbf{z}_{1:t-1}\} \approx P\{m_{t-1}^i|\mathbf{z}_{1:t-1}\} = T_{t-1}^i. \end{aligned} \tag{14}$$

With these approximations for $p(\mathbf{z}_t|\mathbf{\Omega}, \mathbf{z}_{1:t-1})$ in (16), the total probability $p_{\mathbf{\Omega}}(\mathbf{z}_t|\mathbf{z}_{1:t-1})$ is also approximated as described in (17). Based on (16) and (17), the sequential update of the TPM posterior distribution in (13) is expressed by

$$p(\mathbf{\Omega}|\mathbf{z}_{1:t}) \approx \frac{\mathbf{T}_{t-1}^{\top} \mathbf{\Omega} \mathbf{\Lambda}_t}{\mathbf{T}_{t-1}^{\top} \bar{\mathbf{\Omega}}_{t-1} \mathbf{\Lambda}_t} p(\mathbf{\Omega}|\mathbf{z}_{1:t-1}). \tag{15}$$

where $\mathbf{\Lambda}_t = [\Lambda_t^1, \ldots, \Lambda_t^M]^{\top}$ and $\mathbf{T}_{t-1} = [T_{t-1}^1, \ldots, T_{t-1}^M]^{\top}$.

Sample Approximation: For practical implementations, the TPM posterior distribution in (15) is approximated by first or second order numerical integration methods as they are shown to be more robust and accurate than other approaches [17]. In numerical integration, since prior information of probabilistic interactions between trackers is usually not given, the interaction probabilities are defined on a finite grid. Thus, the TPM prior distribution is approximated by set of samples $\{\mathbf{\Omega}^q | q = 1, \ldots, N_{\mathbf{\Omega}}\}$ with corresponding weights $\{p(\mathbf{\Omega}^q|\mathbf{z}_{1:t-1}) | q = 1, \ldots, N_{\mathbf{\Omega}}\}$, and the TPM posterior distribution in (15) is described by

$$p(\mathbf{\Omega}^q|\mathbf{z}_{1:t}) = \frac{\mathbf{T}_{t-1}^{\top} \mathbf{\Omega}^q \mathbf{\Lambda}_t}{\mathbf{T}_{t-1}^{\top} \bar{\mathbf{\Omega}}_{t-1} \mathbf{\Lambda}_{\mathbf{t}}} p(\mathbf{\Omega}^q|\mathbf{z}_{1:t-1}). \tag{18}$$

We obtain the updated TPM $\bar{\mathbf{\Omega}}_t$ at time t as

$$\bar{\mathbf{\Omega}}_t = \frac{1}{C} \sum_{q=1}^{N_{\mathbf{\Omega}}} \mathbf{\Omega}^q p(\mathbf{\Omega}^q|\mathbf{z}_{1:t}), \tag{19}$$

where $C = \sum_{q=1}^{N_{\mathbf{\Omega}}} p(\mathbf{\Omega}^q|\mathbf{z}_{1:t})$ is a normalization term and each TPM sample is expressed by $\mathbf{\Omega}^q = \left[\omega_q^{j,i}\right]_{M\times M}$. The interaction probabilities are chosen as $0 \leq \omega_q^{j,i} \leq 1$ and satisfy the condition $\sum_{j=1}^{M} \omega_q^{j,i} = 1$.

6 Robust Tracker Likelihood Function

The reliability of each tracker is used to update the tracker probability and TPM within the Bayesian framework. The tracker likelihood function computes the reliability of each one by measuring the tracking result individually. The estimated object state from the i-th tracker at time t is

$$\hat{\mathbf{x}}_t^i = \arg\max_{\mathbf{x}_t} p(\mathbf{x}_t|\mathbf{z}_{1:t}, m_t^i). \tag{20}$$

Since $\hat{\mathbf{x}}_t^i$ is obtained from the i-th tracker, the accuracy of $\hat{\mathbf{x}}_t^i$ is considered as the reliability of the i-th tracker. Hence, the TLF is expressed by

$$p(\mathbf{z}_t|m_t^i, \mathbf{z}_{1:t-1}) = p_{\text{TLF}}(\mathbf{z}_t|\hat{\mathbf{x}}_t^i). \tag{21}$$

For measuring the tracker reliability, we use instantaneous and reconstruction features to account for transient and stable appearance changes. These two representations are assumed to be independent and all M features ($\mathbf{f}^k, k = 1, \ldots, M$) are used for computing the TLF to measure the reliability of each tracker. Thus, the TLF is formulated by

$$\begin{aligned} &p_{\text{TLF}}(\mathbf{z}_t|\hat{\mathbf{x}}_t^i) \approx p_{\text{I}}(\mathbf{z}_t|\hat{\mathbf{x}}_t^i) p_{\text{R}}(\mathbf{z}_t|\hat{x}_t^i) \\ &= \prod_{k=1}^{M} p(\mathbf{z}_t|\hat{\mathbf{x}}_t^i, \mathbf{f}_{\text{I},t}^k) p(\mathbf{z}_t|\hat{\mathbf{x}}_t^i, \mathbf{f}_{\text{R},t}^k), \end{aligned} \tag{22}$$

where k is the feature index, $p_{\text{I}}(\mathbf{z}_t|\hat{\mathbf{x}}_t^i)$ is the TLF based on the instantaneous appearance model (IAM), and $p_{\text{R}}(\mathbf{z}_t|\hat{\mathbf{x}}_t^i)$ is the TLF based on the reconstruction appearance model (RAM). The instantaneous object appearance $\bar{\mathbf{f}}_{\text{I},t}^k$ is obtained from a set of recent observations $\mathbf{f}_{\text{I},t}^k$. The reconstructed object appearance $\bar{\mathbf{f}}_{\text{R},t}^{i,k}$ is computed from the stable appearance $\mathbf{f}_{\text{R},t}^k$ using the k-th feature and the tracking result $\mathbf{z}_t^{i,k}$ from the i-th tracker. Each TLF is computed by

$$p(\mathbf{z}_t|\hat{\mathbf{x}}_t^i, \mathbf{f}_{\text{I},t}^k) = \exp(-\rho\|\bar{\mathbf{f}}_{\text{I},t}^k - \mathbf{z}_t^{i,k}\|^2), \tag{23}$$

$$p(\mathbf{z}_t|\hat{\mathbf{x}}_t^i, \mathbf{f}_{\text{R},t}^k) = \exp(-\rho\|\bar{\mathbf{f}}_{\text{R},t}^{i,k} - \mathbf{z}_t^{i,k}\|^2), \tag{24}$$

where ρ is a control parameter and

$$\mathbf{z}_t^{i,k} = \frac{Vec(F^k(I(\hat{\mathbf{x}}_t^i)))}{\|Vec(F^k(I(\hat{\mathbf{x}}_t^i)))\|}, \tag{25}$$

where $Vec(\cdot)$ represents vectorization; $I(\mathbf{x}_t)$ denotes an image region based on a state vector $\mathbf{x}_t$; $F^k(\cdot)$ denotes the k-th

$$\begin{aligned} p(\mathbf{z}_t|\mathbf{\Omega}, \mathbf{z}_{1:t-1}) &= \sum_{i=1}^{M} p(\mathbf{z}_t|m_t^i, \mathbf{\Omega}, \mathbf{z}_{1:t-1}) P\{m_t^i|\mathbf{\Omega}, \mathbf{z}_{1:t-1}\} \\ &= \sum_{i=1}^{M} \underbrace{p(\mathbf{z}_t|m_t^i, \mathbf{\Omega}, \mathbf{z}_{1:t-1})}_{\approx \Lambda_t^i} \sum_{j=1}^{M} \underbrace{P\{m_t^i|m_{t-1}^j, \mathbf{\Omega}, \mathbf{z}_{1:t-1}\}}_{\triangleq \omega^{j,i}} \underbrace{P\{m_{t-1}^j|\mathbf{\Omega}, \mathbf{z}_{1:t-1}\}}_{\approx T_{t-1}^j} \\ &\approx \sum_{i=1}^{M} \Lambda_t^i \sum_{j=1}^{M} \omega^{j,i} T_{t-1}^j = \mathbf{\Lambda}_t^\top \mathbf{\Omega}^\top \mathbf{T}_{t-1} = \mathbf{T}_{t-1}^\top \mathbf{\Omega} \mathbf{\Lambda}_t, \end{aligned} \tag{16}$$

where

$$\mathbf{\Lambda}_t = [\Lambda_t^1, \ldots, \Lambda_t^M]^\top, \quad \mathbf{T}_{t-1} = [T_{t-1}^1, \ldots, T_{t-1}^M]^\top.$$

$$p_{\mathbf{\Omega}}(\mathbf{z}_t|\mathbf{z}_{1:t-1}) = \int \underbrace{p(\mathbf{z}_t|\mathbf{\Omega}, \mathbf{z}_{1:t-1})}_{\approx \mathbf{T}_{t-1}^\top \mathbf{\Omega} \mathbf{\Lambda}_t \text{in (16)}} p(\mathbf{\Omega}|\mathbf{z}_{1:t-1}) d\mathbf{\Omega} \approx \mathbf{T}_{t-1}^\top \left(\int \mathbf{\Omega} p(\mathbf{\Omega}|\mathbf{z}_{1:t-1}) d\mathbf{\Omega} \right) \mathbf{\Lambda}_t = T_{t-1}^\top \bar{\mathbf{\Omega}}_{t-1} \mathbf{\Lambda}_t. \tag{17}$$

feature extraction; and $\mathbf{z}_t^{i,k} \in \Re^{d^k}$ where d^k is the dimension of the k-th feature. The IAM and RAM are computed as follows.

Transient Object Appearance: The short-term object appearance changes are model by a set of recent object observations $\mathbf{f}_{\text{I},t}^k = [\mathbf{f}_{\text{I},t-l}^k, \ldots, \mathbf{f}_{\text{I},t-1}^k]$. The instantaneous appearance model $\bar{\mathbf{f}}_{\text{I},t}^k$ is obtained by averaging the recent L appearances as

$$\bar{\mathbf{f}}_{\text{I},t}^k = \frac{1}{L} \sum_{l=1}^{L} \mathbf{f}_{\text{I},t-l}^k. \tag{26}$$

Stable Object Appearance: The long-term object appearance $\mathbf{z}_t^{i,k}$ can be represented by a linear combination of stable features $\mathbf{f}_{\text{R},t}^k$ which are r representative features,

$$\mathbf{z}_t^{i,k} \approx \mathbf{f}_{\text{R},t}^k \boldsymbol{\alpha}_t^{i,k} = \mathbf{f}_{1,t}^k \alpha_{1,t}^{i,k} + \mathbf{f}_{2,t}^k \alpha_{2,t}^{i,k} + \ldots + \mathbf{f}_{r,t}^k \alpha_{r,t}^{i,k}, \tag{27}$$

where $\mathbf{f}_{\text{R},t}^k$=$[\mathbf{f}_{1,t}^k, \ldots, \mathbf{f}_{r,t}^k] \in \Re^{d^k \times r}$, $\boldsymbol{\alpha}_t^{i,k}$=$[\alpha_{1,t}^{i,k}, \ldots, \alpha_{r,t}^{i,k}]^\top \in \Re^r$ is an coefficient vector. By including the noise vector $\boldsymbol{\epsilon}^{i,k}$, we have

$$\mathbf{z}_t^{i,k} = \mathbf{f}_{\text{R},t}^k \boldsymbol{\alpha}_t^{i,k} + \boldsymbol{\epsilon}^{i,k} = \left[\begin{array}{cc} \mathbf{f}_{\text{R},t}^k & \mathbf{I}^k \end{array} \right] \left[\begin{array}{c} \boldsymbol{\alpha}_t^{i,k} \\ \boldsymbol{\beta}_t^{i,k} \end{array} \right]. \tag{28}$$

We use a set of non-target (trivial) templates from a d^k-dimensional identity matrix $\mathbf{I}^k \in \Re^{d^k \times d^k}$ [24] with a non-target coefficient vector $\boldsymbol{\beta}_t^{i,k} = [\beta_{1,t}^{i,k}, \beta_{2,t}^{i,k}, \ldots, \beta_{d^k,t}^{i,k}]^\top \in \Re^{d^k}$. If the observation contains little noise, then the non-target coefficient vector has only a few nonzero coefficients in $\boldsymbol{\beta}_t^{i,k}$.

In the proposed tracking algorithm, we obtain M tracking results at each frame, $\{\hat{\mathbf{x}}_t^i | i = 1, \ldots, M\}$. Based on the result of the i-th tracker, the candidate image region represented by the k-th feature is denoted as $\mathbf{z}_t^{i,k}$ in (25). The reconstructed appearance for $\mathbf{z}_t^{i,k}$ is denoted as $\mathbf{f}_{\text{R},t}^k \boldsymbol{\alpha}_t^{i,k}$. We obtain the coefficient vector $\boldsymbol{\alpha}_t^{i,k}$ by using ℓ_1 sparse coding as it is robust to wide range of image corruption, especially occlusions, [19], [24]. The coefficient vector $\mathbf{c}_t^{i,k}$ is computed by

$$\min_{\mathbf{c}_t^{i,k}} \|\mathbf{c}_t^{i,k}\|_1, \quad \text{s.t.} \quad \|\mathbf{z}_t^{i,k} - \mathbf{D}_t^k \mathbf{c}_t^{i,k}\|_2^2 \leq \lambda, \tag{29}$$

where $\lambda = 0.01$, and

$$\mathbf{D}_t^k = [\mathbf{f}_{\text{R},t}^k, \mathbf{I}^k], \quad \mathbf{c}_t^{i,k} = [(\boldsymbol{\alpha}_t^{i,k})^\top, (\boldsymbol{\beta}_t^{i,k})^\top]^\top. \tag{30}$$

The reconstructed object appearance $\bar{\mathbf{f}}_{\text{R},t}^{i,k}$ for $\mathbf{z}_t^{i,k}$ is computed as $\bar{\mathbf{f}}_{\text{R},t}^{i,k} = \mathbf{f}_{\text{R},t}^{i,k} \alpha_t^{i,k}$.

7 Representation Update

In this section, we present the update mechanisms for transient and stable object appearance as well as observation models for trackers based on M feature representations $\{\hat{\mathbf{f}}_t^k = \mathbf{z}_t^{\hat{m}_t,k} | k = 1, \ldots, M\}$ where $\mathbf{z}_t^{i,k}$ is from (25) and $\hat{m}_t$ is the index of the selected tracker in (5).

Transient Features: We use transient features to account for abrupt appearance changes of a target object. Each transient feature consists of the recently estimated observation as $\mathbf{f}_{\text{I},t+1}^k = \left[\mathbf{f}_{\text{I},t-\theta}^k, \ldots, \mathbf{f}_{\text{I},t}^k \right]$, where $\mathbf{f}_{\text{I},t}^k = \hat{\mathbf{f}}_t^k$ and θ is a variable that determines the duration.

Stable Features: Each stable feature $\mathbf{f}_{\text{R},t}^k$ is updated based on whether it can be sparsely represented by the current templates. Similar to [25], each feature is updated by analyzing the non-zero elements in the non-target coefficient vector $\boldsymbol{\beta}_t^{i,k}$. When occlusion occurs, a target object cannot be sparsely represented by the target template set. Consequently, there exist numerous non-zero coefficients corresponding to the non-target templates, and noise is measured by $\boldsymbol{\beta}_t^{\hat{m}_t,k} \in \Re^{d^k}$ in (29) where $\hat{m}_t$ is the index of the selected tracker. We count non-zero elements in $\boldsymbol{\beta}_t^{\hat{m}_t,k}$, and compute a noise ratio R_{noise}^k as $R_{\text{noise}}^k = B^k / d^k$ where B^k is the number of non-zero elements in $\boldsymbol{\beta}_t^{\hat{m}_t,k}$. If the noise ratio R_{noise}^k is smaller than a threshold, one feature $\mathbf{f}_{i,t}^k \in \mathbf{f}_{\text{R},t}^k$ with the lowest value is replaced by the feature of the estimated observation $\hat{\mathbf{f}}_t^k$.

Observation Model: In this work, the observation model for each tracker (i.e., $p(\mathbf{z}_t|\mathbf{x}_t, m_t^i)$ in (1)) is based on the incremental subspace model [27] for its computational efficiency over ℓ_1 sparse coding. For online tracking, it is known that error accumulation is inevitable when an appearance model is updated with new observations [12], [28]. Note that not every

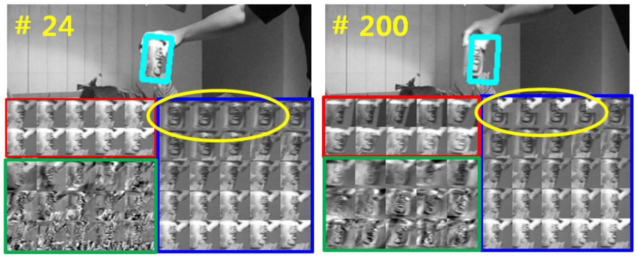


Fig. 5. Representation update examples. The transient and stable features are shown in the red and blue boxes, respectively. The learned principal components are shown in the green boxes. The yellow circles demonstrate the updated stable features at different frames.

observation model is updated at every frame. For the selected tracker of a given frame, the corresponding appearance model is not updated since it describes the target object well. On the other hand, the observation models of all the other trackers are updated with the new observation.

Examples of representation updates (i.e., transient and stable features as well as observation model discussed in Section 7) are shown in Figure 5. To show difference of each representation, we only show the intensity features for comparisons. In the *Coke* sequence, partial occlusions with illumination changes occur frequently. As introduced in Section 7, the transient features better account for frequent appearance changes of the object in such cases while the stable features are rarely updated. The principal components of the object appearance from an observation model are shown in green boxes. These principal components are incrementally updated in each observation model to account for appearance changes.

8 Interacting Multiview Tracker

The main components of the proposed interacting multiview tracker (IMT) are described in Figure 3 and Algorithm 1. We present the algorithmic details in this section.

8.1 Estimated Object States of Multiple Trackers

We use a particle filter for state prediction. The prior distribution of each tracker $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, m^i_{t-1})$ in (2) is approximated by a set of N samples as

$$p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, m^i_{t-1}) \approx \sum_{q=1}^{N} s^i_{q,t-1}\delta(\mathbf{x}^i_{q,t-1} - \mathbf{x}_{t-1}), \quad (31)$$

where $\delta(\cdot)$ is a delta function centered at a sample $\mathbf{x}^i_{q,t-1}$, and $s^i_{q,t-1}$ is a sample weight.

Interacted Prior via Tracker Interaction: At each frame, multiple trackers interact with each other by mixing their posterior distributions described in (2) based on the TPM. The interaction is efficiently carried out via the proposed interaction method by Algorithm 2, i.e.,

$$\left[\tilde{\mathcal{X}}^1_{t-1}, \ldots, \tilde{\mathcal{X}}^M_{t-1}\right] = \text{Tracker\_Interaction}\left[\bar{\mathbf{\Omega}}_{t-1}, \mathcal{X}^1_{t-1}, \ldots, \mathcal{X}^M_{t-1}\right], \quad (32)$$

where $\mathcal{X}^i_{t-1} = \{\mathbf{x}^i_{q,t-1}, s^i_{q,t-1}\}^N_{q=1}$ is the sample approximation of the prior distribution of the i-th tracker and $\tilde{\mathcal{X}}^i_{t-1}$ is the interacted prior distribution. The tracker interaction approach in this work is in spirit similar to [4], [1] where the posterior distribution of the unreliable tracker is replaced by the most reliable one. In addition, the reliability of the tracker is measured by exploring the covariance of the posterior distribution at each frame. However, the proposed interaction method enforces trackers interact with each other via the TPM. Hence, not all samples are transfered to other trackers.

Algorithm 2 Tracker Interaction:
$\left[\tilde{\mathcal{X}}^1_{t-1}, \ldots, \tilde{\mathcal{X}}^M_{t-1}\right] =$ **Tracker_Interaction** $\left[\bar{\mathbf{\Omega}}_{t-1}, \mathcal{X}^1_{t-1}, \ldots, \mathcal{X}^M_{t-1}\right]$

```
Input
Given $\{\mathcal{X}^i_{t-1} = \{\mathbf{x}^i_{q,t-1}, s^i_{q,t-1}\}^N_{q=1} | i = 1, \ldots, M\}$
▷Sample representation of a posterior distribution of $i$-the tracker

for $i = 1 : M$ do
    for $q = 1 : N$ do
        $s^{*i}_{q,t-1} = s^i_{q,t-1}\text{Kernel}(\mathbf{H}\mathbf{x}^i_{q,t-1} - \mathbf{H}\mathbf{x}_{t-1}, \mathbf{R})$
    end for
    $s^{*i}_{q,t-1} := s^{*i}_{q,t-1} / \sum_q s^{*i}_{q,t-1},\ q = 1, \ldots, N$
end for

Given $\bar{\omega}^{j,i}_{t-1} \in \bar{\mathbf{\Omega}}_{t-1}$        ▷TPM
for $i = 1 : M$ do
    $\tilde{\mathcal{X}}^i_{t-1} = \phi$
    for $j = 1 : M$ do
        $\mathcal{X} = \text{Resampling}(\{\mathbf{x}^j_{q,t-1}, s^{*j}_{q,t-1}\}^N_{q=1},\ N \times \bar{\omega}^{j,i}_{t-1})$
        $\tilde{\mathcal{X}}^i_{t-1} := \tilde{\mathcal{X}}^i_{t-1} \cup \mathcal{X}$
    end for
end for

Output
$\{\tilde{\mathcal{X}}^i_{t-1} = \{\tilde{\mathbf{x}}^i_{q,t-1}, \tilde{s}^i_{q,t-1} = \frac{1}{N}\}^N_{q=1} | i = 1, \ldots, M\}$
▷Sample representation of an interacted prior of $i$-th tracker

Given parameters
$\mathbf{H} = \begin{bmatrix} 1\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0 \end{bmatrix}$        ▷position conversion matrix
$\mathbf{R} = \sqrt{(2 \times q_v)^2 + (2 \times q_u)^2}$        ▷kernel range
```

The interacted prior distribution in (2) can be expressed by a sample representation as

$$\tilde{p}(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, m^i_{t-1}) \approx \sum_{q=1}^{N} \tilde{s}^i_{q,t-1}\delta(\tilde{\mathbf{x}}^i_{q,t-1} - \mathbf{x}_{t-1}). \quad (33)$$

By tracker interaction we first remove the samples far from the selected tracking result $\hat{\mathbf{x}}_{t-1}$ based on a kernel. As described in Algorithm 2, a uniform kernel is defined in terms of position with respect to range R with standard deviations (q_u, q_v) along u and v image coordinates. In addition, H is a transformation matrix that returns position parameters as from a previous state by $[p_{u,t-1}, p_{v,t-1}]^\top = H\hat{x}_{t-1}$. Second, multiple trackers interact with each other based on the TPM and a resampling technique [9]. The TPM contains information of how samples are transferred or retained. For instance, $N \times \bar{\omega}^{i,i}_{t-1}$ represents that the number of samples is retained in the i-th tracker sample set after interaction, and $N \times \bar{\omega}^{j,i}_{t-1}$ represents that the number of samples from the j-th tracker is transferred to the i-th tracker. If the i-tracker is effective for some frames, then $\bar{\omega}^{i,i}_{t-1}$ becomes greater than $\bar{\omega}^{j,i}_{t-1}$ $(j \neq i)$ due to update of the TPM. Hence, most samples of the i-th are retained, and the i-

Algorithm 1 Proposed interacting multiview tracker (IMT)

```
(Initial Step)
at time $t = 0$
The initial states of multiple trackers are set to $\{\mathbf{x}_0^i = \mathbf{x}_0 | i = 1, \ldots, M\}$.
The initial set of samples for the particle filter $\{\mathcal{X}_0^i = \{\mathbf{x}_{q,0}^i, s_{q,0}^i = \frac{1}{N}\}_{q=1}^N | i = 1, \ldots, M\}$.
The initial TPM is given by $\bar{\mathbf{\Omega}}_0 = \frac{1}{N_{\mathbf{\Omega}}} \sum_q \Omega^q$    ▷Section 9.2.
The initial tracker probability is set to $\{T_0^i = \frac{1}{M} | i = 1, \ldots, M\}$.
(Tracking Step)
for $t \geq 1$ do
    for $i = 1 : M$ do    ▷i is a tracker index
        1) Compute the interacted prior distribution $\tilde{\mathcal{X}}_{t-1}^i = \{\tilde{\mathbf{x}}_{q,t-1}^i, \tilde{s}_{q,t-1}^i\}_{q=1}^N$ using $\{\mathcal{X}_{t-1}^i | i = 1, \ldots, M\}$ with the TPM $\bar{\mathbf{\Omega}}_{t-1}$ and the tracker probability $\{T_{t-1}^i | i = 1, \ldots, M\}$ using Algorithm 2.
        2) Predict state samples $\{\mathbf{x}_{q,t}^i, s_{q,t|t-1}^i\}_{q=1}^N$ using (34).
        3) Update state samples $\{\mathbf{x}_{q,t}^i, s_{q,t}^i\}_{q=1}^N$ using (37).
        4) Obtain the estimated state $\hat{\mathbf{x}}_t^i$ from the $i$-th tracker using (39).
    end for
    5) Compute the TLFs $\{\Lambda_t^i | i = 1, \ldots, M\}$ using (22) using the set of $M$ estimated object states $\{\hat{\mathbf{x}}_t^i | i = 1, \ldots, M\}$.
    6) The tracker probability update with the TLFs $\{\Lambda_t^i | i = 1, \ldots, M\}$ using (10).
    7) The TPM update with the tracker probabilities $\{T_t^i | i = 1, \ldots, M\}$ and TLFs $\{\Lambda_t^i | i = 1, \ldots, M\}$ using (18) and (19).
    8) Compute the tracking result $\hat{\mathbf{x}}_t$ using (5).
    9) Update representations as described in Section 7.
end for
```

th tracker obtains a few samples from other trackers. Finally, we select samples according to the interaction probabilities, $\bar{\omega}_{t-1}^{j,i}$ of the TPM by resampling such that reliable samples with large weights in each tracker are retained.

Sampling via Motion Models: We draw new state samples from the interacted prior distribution $\tilde{p}(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}, m_{t-1}^i)$. In this work, we use the zero and first order motion models for state prediction $p(\mathbf{x}_t|\mathbf{x}_{t-1}, m_t^i)$. The zero-order motion is identical to the random walk motion, and the first-order motion utilizes the prior translation $\Delta\mathbf{x}_t = [\Delta u, \Delta v, 0, 0, 0, 0]^\top$ by computing the difference between estimated positions at time $t-1$ and $t-2$. Thus, samples are drawn based on

$$\begin{aligned} \mathbf{x}_{q,t}^i &\sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, m_t^i) \\ &= \begin{cases} \mathcal{N}(\mathbf{x}_{q,t-1}^i, \mathbf{Q}_0) & \text{if } \tau < 0.5 \\ \mathcal{N}(\mathbf{x}_{q,t-1}^i + \Delta\mathbf{x}_t, \mathbf{Q}_1) & \text{otherwise,} \end{cases} \end{aligned} \tag{34}$$

where $\mathbf{Q}_0$ and $\mathbf{Q}_1$ denotes the zero and first order motion covariances, respectively and given in Section 9.1. We use a uniform random variable τ distributed within $[0, 1]$ to select the motion model for drawing each sample. The set of the predicted samples is $\{\mathbf{x}_{q,t}^i, s_{q,t|t-1}^i\}_{q=1}^N$ where $s_{q,t|t-1}^i = \tilde{s}_{q,t-1}^i$.

Sample Update via Observation Models: An observation for the i-th tracker is expressed by

$$\mathbf{z}_t^i = Vec(F^i(I(\mathbf{x}_t))) + \mathbf{v}_t^i, \quad i = 1, \ldots, M, \tag{35}$$

where $I(\mathbf{x}_t)$ denotes an image template based on a state vector $\mathbf{x}_t$; $F^i(\cdot)$ represents the i-th feature extraction; and $\mathbf{v}_t^i$ is noise. In the incremental subspace based observation model [27], we compute the mean and principal eigenvectors with updates for the appearance model in each tracker. Based on the template mean $\bar{O}^i$ and L principal eigenvectors $\mathbf{g}_l^i$, $l = 1, \ldots, L$, the i-th observation model based on the i-th feature is given by

$$\begin{aligned} &p(\mathbf{z}_t|\mathbf{x}_t, m_t^i) = \exp(-\rho_T \|\mathbf{z}_t^i - \textstyle\sum_l \mathbf{c}_l \mathbf{g}_l^i\|^2), \\ &\mathbf{c}_l = (\mathbf{g}_l^i)^\top (\mathbf{z}_t^i - \bar{O}^i), \quad l = 1, \ldots, L, \end{aligned} \tag{36}$$

where ρ_T is a control parameter and $\mathbf{c}_l$ is the coefficient from the projection of the template onto each principal eigenvector (16 eigenvectors are used for each observation model).

We note that the TLF in (22) is not related to the observation model in (36). The TLF is only used to update the tracker probability and TPM. The reason being that it is time-consuming to measure all particle samples if we use TLF instead of (36). For the efficient implementation, we use (36) as an observation model to measure particle samples of a single tracker as it can be computed efficiently to adapt object appearance changes. Based on (35) and (36), the weight of each sample is updated by

$$s_{q,t}^i = \frac{p(\mathbf{z}_t|\mathbf{x}_{q,t}^i, m_t^i) s_{q,t|t-1}^i}{\sum_{q=1}^N p(\mathbf{z}_t|\mathbf{x}_{q,t}^i, m_t^i) s_{q,t|t-1}^i}. \tag{37}$$

With the samples and weights in (34) and (37), we obtain the sample representation of the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t}, m_t^i)$ in (1) as

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}, m_t^i) \approx \sum_{q=1}^N s_{q,t}^i \delta(\mathbf{x}_{q,t}^i - \mathbf{x}_t), \tag{38}$$

which is described by a set of samples with weights $\{\mathbf{x}_{q,t}^i, s_{q,t}^i\}_{q=1}^N$.

Estimated Object States: From the updated posterior distributions, we obtain a set of M estimated states using the maximum a posterior estimates $(i = 1, \ldots, M)$,

$$\hat{\mathbf{x}}_t^i = \mathbf{x}_{\hat{q},t}^i, \quad \hat{q} = \arg\max_q(\{s_{q,t}^i | q = 1, \ldots, N\}). \tag{39}$$

8.2 Tracker Selection and TPM update

To select the most reliable tracker and update the TPM, we compute the reliability of trackers using the TLF $p_{\text{TLF}}(\mathbf{z}_t|\hat{\mathbf{x}}_t^i) = \Lambda_t^i$ in (22) and M estimated states, $\{\hat{\mathbf{x}}_t^i | i = 1, \ldots, M\}$ from M multiple trackers. With the TLFs $\{\Lambda_t^i | i = 1, \ldots, M\}$, we update the tracker probability using (10) and obtain updated tracker probabilities $\{T_t^i | i = 1, \ldots, M\}$. By selecting the highest tracker probability, we obtain the tracking result $\hat{\mathbf{x}}_t$ as described in (5). The tracking result $\hat{\mathbf{x}}_t$ is then used for representation update (See Section 7). After computing the set of the updated tracker probabilities $\{T_t^i | i = 1, \dots, M\}$ and the set of TLFs $\{\Lambda_t^i | i = 1, \dots, M\}$, we update the TPM $\bar{\mathbf{\Omega}}_t$ using (18) and (19).

9 Experiments

We evaluate the proposed IMT algorithm with the state-of-the-art methods using several benchmark datasets [2] (http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml) and [31] (http://visual-tracking.net), and our own sequences (i.e., *Startrek* and *Starwars*). In this work, we use three trackers with different feature representations based on HOG, intensity, and Haar-like features, which have been shown to be effective for handling occlusions, motion blurs, pose variations, and illumination changes. We discuss motion parameter settings in Section 9.1, sampling scheme for the TPM in Section 9.2, and feature extraction in Section 9.3. We analyze the TPM and show how it is used by multiple trackers in Section 9.4. In Section 9.5, we demonstrate the effects of the proposed TLF, and in Section 9.6, we compare the proposed IMT algorithm with other tracking methods based on one single feature representation of HOG, intensity, and Haar-like features (denoted as SHOG, SI, and SHaar methods). We evaluate the proposed algorithm in Section 9.7, with methods based on multiple trackers or representations including the approaches with combination of visual trackers (CVT) [22], the multi-cue switching tracker (MCS) [4], and a single tracker with multiple observation models (SMO) similar to [29] where the tracker reliability is not measured. For fair comparisons, each single tracker of the proposed IMT algorithm and parameters are the same as the ones used in the CVT, MCS, SMO, SHOG, SI, and SHaar methods. Furthermore, in Section 9.8, we compare the IMT algorithm with state-of-the-art trackers including the MIL [2], TLD [18], VTD [20], VTS [21], Struck [13], ASLA [16], SCM [37], CXT [8], LSK [23], CSK [14], and KCF [15] methods.

For quantitative comparisons, we present the success rate rather than center location error as it does not fully reflect especially after tracking drifting [31]. The code and datasets are available at https://cvl.gist.ac.kr/project/imt.html and http://faculty.ucmerced.edu/mhyang/project/imt.html.

9.1 Motion Parameters

In this work, an object state is expressed by six parameters of the affine transformation [27] based on a diagonal covariance matrix $\mathbf{Q} = \text{diag}(q_u^2, q_v^2, q_\theta^2, q_s^2, q_\alpha^2, q_\phi^2)$ where q_u and q_v are standard deviations of position; q_θ, q_s, q_α, and q_ϕ are standard deviations of rotation angle, scale, aspect ratio, and skew, respectively. For all the experiments, we fix four parameters as $q_\theta = 0.02$, $q_s = 0.01$, $q_\alpha = 0$, $q_\phi = 0.001$. The translation standard deviation of the zero-order motion $\mathbf{Q}_0$ are fixed as $q_u = q_v = 6$. The translation standard deviation of the first-order motion $\mathbf{Q}_1$ are fixed as $q_u = q_v = 3$. Since the SI, SHOG, SHaar, SMO, MCS, CVT, and IMT methods are based on the same single tracker [27], we use the same parameter settings as mentioned above. We note that the results in [35] are based on optimized parameters for each sequence, whereas in this work the parameters are fixed for all experiments.

TABLE 1
Interaction probability basis of the i-th tracker
$\boldsymbol{\omega}_s^i = [\omega_s^{1,i}, \omega_s^{2,i}, \omega_s^{3,i}]$ where s denotes the basis index.

Tracker 1	Tracker 2
$\boldsymbol{\omega}_1^1 = [\mathbf{0.7},\ 0.15,\ 0.15]^\top$	$\boldsymbol{\omega}_1^2 = [0.15,\ \mathbf{0.7},\ 0.15]^\top$
$\boldsymbol{\omega}_2^1 = [\mathbf{0.6},\ 0.20,\ 0.20]^\top$	$\boldsymbol{\omega}_2^2 = [0.20,\ \mathbf{0.6},\ 0.20]^\top$
$\boldsymbol{\omega}_3^1 = [\mathbf{0.5},\ 0.25,\ 0.25]^\top$	$\boldsymbol{\omega}_3^2 = [0.25,\ \mathbf{0.5},\ 0.25]^\top$
$\boldsymbol{\omega}_4^1 = [\mathbf{0.4},\ 0.30,\ 0.30]^\top$	$\boldsymbol{\omega}_4^2 = [0.30,\ \mathbf{0.4},\ 0.30]^\top$
$\boldsymbol{\omega}_5^1 = [\mathbf{0.3},\ 0.35,\ 0.35]^\top$	$\boldsymbol{\omega}_5^2 = [0.35,\ \mathbf{0.3},\ 0.35]^\top$
$\boldsymbol{\omega}_6^1 = [\mathbf{0.2},\ 0.40,\ 0.40]^\top$	$\boldsymbol{\omega}_6^2 = [0.40,\ \mathbf{0.2},\ 0.40]^\top$

Tracker 3
$\boldsymbol{\omega}_1^3 = [0.15,\ 0.15,\ \mathbf{0.7}]^\top$
$\boldsymbol{\omega}_2^3 = [0.20,\ 0.20,\ \mathbf{0.6}]^\top$
$\boldsymbol{\omega}_3^3 = [0.25,\ 0.25,\ \mathbf{0.5}]^\top$
$\boldsymbol{\omega}_4^3 = [0.30,\ 0.30,\ \mathbf{0.4}]^\top$
$\boldsymbol{\omega}_5^3 = [0.35,\ 0.35,\ \mathbf{0.3}]^\top$
$\boldsymbol{\omega}_6^3 = [0.40,\ 0.40,\ \mathbf{0.2}]^\top$

TABLE 2
Average tracking success rate on 16 benchmark sequences in Table 3. The IMTs with different initial TPM settings show similar performance.

	IMT with TPM$_{\text{ave}}$	IMT with TPM$_{\text{naive}}$
average success rate	92	90

9.2 TPM Setting for Three Trackers

As discussed in Section 5.2, we approximate the TPM posterior distribution by a set of TPM samples $\{\mathbf{\Omega}^q | q = 1, \dots, N_{\mathbf{\Omega}}\}$. To construct the TPM, we use the interaction probability basis defined on a finite grid in Table 1 where each vector represents the interaction probabilities describing how samples are retained and transferred. For instance, if we use 600 state samples for each tracker, the interaction probability basis $\boldsymbol{\omega}_1^1$ represents that $\boldsymbol{\omega}_1^1 \times 600 = [420,\ 90,\ 90]^\top$ where the first tracker retains its own 420 samples and receives 90 samples from the second and 90 samples from the third trackers, respectively. In this work, we only set the maximum and minimum values for the diagonal entries of the TPM. The diagonal values are set to $\omega_s^{i,i} \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7\}$ to make each tracker retain at most 70% of its own samples and at least 20% of its own samples. The off-diagonal values of the TPM are set with a given diagonal value by $\omega_s^{j,i} = \frac{1-\omega_s^{i,i}}{2}$ (See Table 1). Using the interaction probability basis in Table 1, we obtain a TPM sample as

$$\mathbf{\Omega}^q = \left[\boldsymbol{\omega}_{s_1}^1, \boldsymbol{\omega}_{s_2}^2, \boldsymbol{\omega}_{s_3}^3\right], \quad s_1, s_2, s_3 = 1, \dots, 6.$$

Consequently, 216 TPM samples $\{\mathbf{\Omega}^q | q = 1, \dots, 216\}$ are generated by considering all combinations of the basis in Table 1. These TPM samples are fixed in all experiments. The initial TPM $\bar{\mathbf{\Omega}}_0$ is obtained by averaging all of TPM samples. Note that the TPM method is not sensitive to initial values as it is updated at each frame. To demonstrate this, we compare the performance of the IMT method with two different initial TPMs (TPM$_{\text{ave}}$ and TPM$_{\text{naive}}$) as shown in Table 2, where TPM$_{\text{ave}}$ is the obtained by averaging all of TPM samples as discussed above, and TPM$_{\text{naive}}$ is a matrix whose elements are equally set to $\frac{1}{3}$.

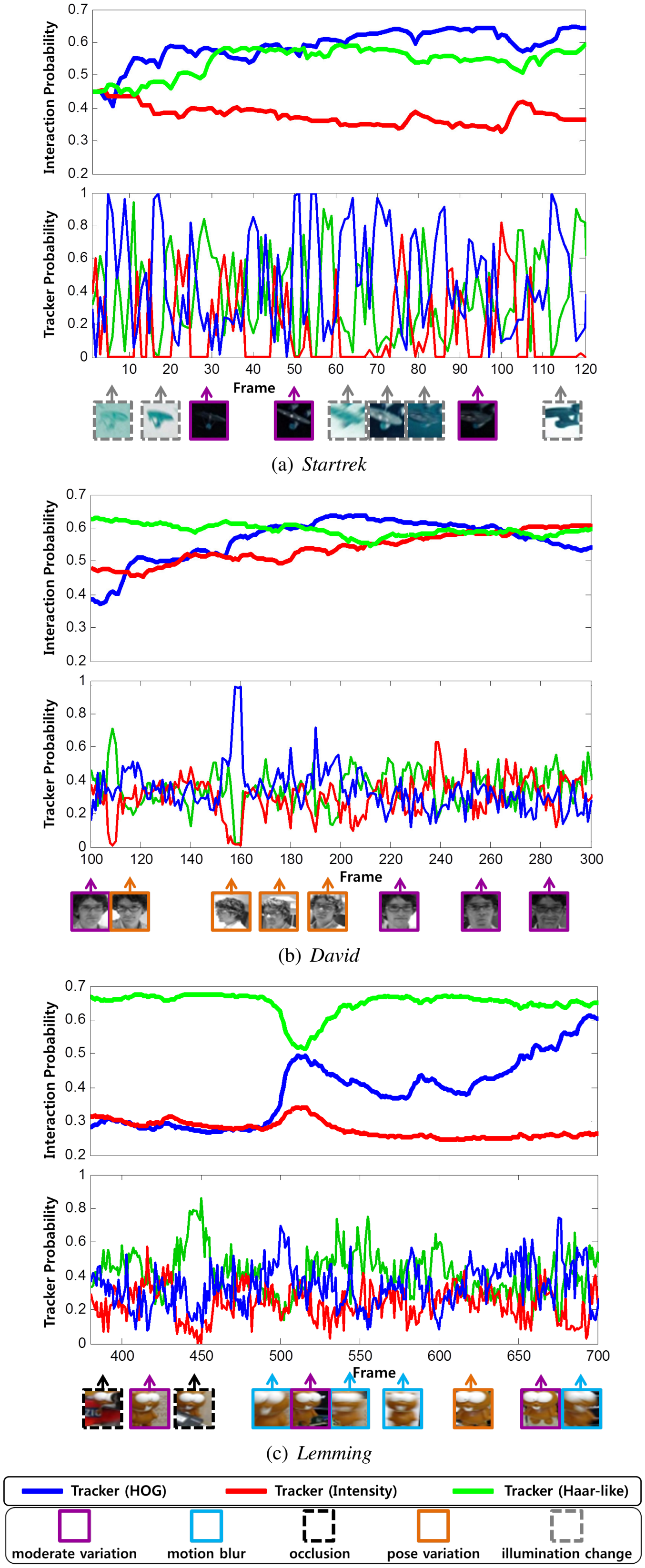
9.3 Feature Extraction

In this work, the size of an image template is 32-by-32 pixels from which a 1024-dimensional intensity feature vector is formed. To generate HOG features, we use 36 blocks and each block has 4 cells within an image template, and the dimension of HOG feature for each block is 36 (i.e., each HOG feature vector is of 1296 dimensions). The Haar-like features are generated with two horizontal and vertical edge filters within a 32-by-32 template to 1760-dimensional vectors.

9.4 Analysis of TPM and Tracker Probability

We analyze how TPM is used among multiple trackers to account for different object appearance changes. In Figure 6, the diagonal interaction probabilities ($\bar{\omega}_t^{i,i}$) of the TPM and tracker probabilities are shown according to object appearance changes over time. When the diagonal entry $\bar{\omega}_t^{i,i}$ decreases, then the off-diagonal entries $\bar{\omega}_t^{j,i}, j \neq i$ increases (as $\sum_{j=1}^{M} \bar{\omega}_t^{j,i} = 1$). The increase of the off-diagonal entries represent that the i-th tracker becomes more dependent on other trackers. It also shows that when the i-th tracker probability continues to be the highest, the diagonal interaction probability $\bar{\omega}_t^{i,i}$ of the TPM tends to increase. The increase of the diagonal entry represents that the i-th tracker becomes less dependent on other trackers.

In the *Startrek* sequence (See Figure 6(a)), both object and background appearances are drastically changed due to abrupt illumination variations. In such scenarios, the tracker based on intensity features is not reliable and hence its tracker probability is usually low, and likewise its interaction probability is consistently low. In the *David* sequence, the tracker based on HOG features is more robust than others when large pose variations occur, which can be explained by that face contour is more effective for tracking in such scenarios (See Figure 6(b)). On the other hand, trackers based on all the other features perform well when moderate appearance changes occur. In the *Lemming* sequence (See Figure 6(c)), when the target object undergoes partial occlusions, the interaction probability for the tracker with Haar-like feature increases and its tracker probability is greater than that of other trackers. When the motion blurs suddenly occur, the interaction probability for the tracker based on HOG features increases and the interaction probabilities of other trackers decreases. Similarly, the tracker probability of the tracker based on HOG features is greater than that of other trackers as the shape of the object is consistent. The tracker based on Haar-like features adaptively learns the appearance changes. As a result, its interaction and tracker probabilities increase after a few frames.



(a) *Startrek*

(b) *David*

(c) *Lemming*

Fig. 6. Changes of interaction probabilities on the diagonal of the TPM and tracker probabilities. Each color line represents one type of trackers. Each color box represents one type of appearance changes. The results are obtained by running the IMT 10 times.

9.5 Analysis of TLF

To show the effectiveness of combination of the instantaneous and reconstruction appearance models in the TLF (See Section 6), we evaluate the tracking results using three combinations. The first one is the IMT-all which uses both IAM and RAM together as proposed in this work; the second one is the IMT-IAM which uses only the instantaneous appearance model; and the third one is the IMT-RAM which utilizes only the reconstruction appearance model. As shown in Table 3, the IMT-all achieves more robust and consistent performance than the other two alternatives.

9.6 Comparison with Single-Feature Trackers

Table 3 shows the results of three trackers based on one single feature (i.e., SI, SHOG, and SHaar). These trackers are the same as the single tracker used in the IMT, and their observation models are described in (36). Overall, the proposed multiview tracking algorithm performs better than these trackers with a single feature. In addition, the trackers based on multiple features (i.e., SMC, MCS, and CVT) perform better than the SI, SHOG and SHaar methods. These results demonstrate the merits of using multiple features for robust object tracking.

9.7 Comparison with Most Related Trackers

The SMO, CVT [22], and MCS [4] methods are related to the proposed method, but the integration approach of multiple features are different as discussed in Section 2. As shown in Table 3, the proposed IMT algorithm performs favorably against these tracking algorithms.

The SMO tracker exploits multiple observation models in a particle filter framework. However, it does not perform well as all observation models contribute equally to estimation of object states without considering their reliability. Hence, posterior distributions and tracking performance may be affected by one tracker with an unreliable observation model.

The CVT method fuses tracking results from multiple trackers with their reliability weight where each one is determined solely by covariance information of its posterior distribution. As discussed in Section 2 and shown in Figure 2, the covariance-based approach may not achieve reliable results as the covariance of each posterior distribution does not well represent tracker reliability because each one is constructed from different feature space (i.e., no calibration of tracking results). In addition, similar to SMO, it does not consider the reliability information in the interaction step, which has the interaction scheme in computing the likelihood.

In contrast, the MCS method selects the most reliable tracker at each frame where the reliability is determined by the acceptance ratio using the covariance of the posterior and prior distributions. If the acceptance ratio is below the threshold (e.g., 0.2 in the experiments), the tracker is considered as an unreliable one. In the sampling stage, the MCS method simply replaces the probability distribution of unreliable trackers by that of the most reliable tracker. However, the covariance information is not reliable as discussed above. This sampling process is likely to cause tracking failure as it does consider all information of unreliable trackers which can be incorrectly selected due to inaccurate covariance information.

Different from the MCS, CVT, and SMO methods, each tracker of the proposed IMT algorithm generates tracking results independently, and the most reliable one is selected using the TLF (that measures the tracker reliability robustly at each frame as shown in Figure 6) by considering stability and effectiveness of feature representations (See also Figure 5). In addition, the reliability information is effectively utilized in the tracker interaction process. Thus, the IMT algorithm performs favorably against these methods based on multiple trackers.

9.8 Comparison with State-the-of-Art Trackers

Benchmark Dataset: We compare the proposed IMT algorithm with 29 state-of-the-art trackers using a large benchmark dataset [31] which contains 51 sequences. Three evaluation metrics are used to evaluate whether the tracking algorithms are sensitive to different initial settings. For the one-pass evaluation (OPE), we use a ground truth bounding box in the first frame for initialization. For the temporal robustness evaluation (TRE), we initialize each tracker with ground truth locations at different frames. For the spatial robustness evaluation (SRE), we use the perturbed ground truth locations in the first frames for experiments. The top 10 tracking algorithms are shown in Figure 7 for presentation clarity. Figure 7 shows that the IMT algorithm performs robustly and favorably against the top 9 trackers using all the evaluation metrics (OPE, TRE, and SRE).

Startrek* and *Starwars: The target objects undergo drastic illumination changes and motion blurs in low resolution and contrast image sequences. As shown in Table 3, Figure 8(a), and Figure 8(b), most of trackers do not perform well. On the other hand, the IMT algorithm tracks the objects well in both sequences due to the use of tracker reliability to weigh less on the unreliable tracker (i.e., a tracker with intensity feature) and more on reliable trackers in the tracker integration scheme (via tracker selection and interaction) as shown in Figure 6(a).

David*, *Girl*, and *Football: The objects in these sequences undergo large pose variations with occlusions. The VTD method drifts away from the target objects when large appearance changes occur (e.g., #167 in Figure 8(c)). When the target object is partially occluded by other similar objects (e.g., #441 in Figure 8(d) and #297 in Figure 8(e)), the VTD, MIL, and TLD methods do not perform well. Although, the KCF track the object center location well, but it cannot estimate the size of the objects. The IMT algorithm tracks the target objects reliably as different trackers are selected to handle different tracking scenarios as shown in Figure 6(b).

Woman* and *CAVIAR: The objects in both sequences undergoes heavy occlusions. In addition, the scale of the object in the *CAVIAR* sequence changes significantly as shown in Figure 8(f). The Struck and TLD methods do not perform well when large scale change occurs. Due to significant scale changes in the *CAVIAR* sequence, the KCF shows limited tracking performance. When heavy occlusions occur in the *Woman* sequence (#60 in Figure 8(g)), the MIL and VTD methods start to drift away from the target object. On the other hand, the IMT algorithm tracks the target objects well by using Haar-like features efficiently, which are more robust for handling occlusion than other feature as shown in Table 3.

***Singer1*, *Sylv*, and *Trellis*:** The objects in these sequence undergo large appearance changes due to illumination and pose variations. As shown in Figure 8(h)-8(l), the MIL methods do not perform well. The VTD, ASLA, TLD, and Struck

TABLE 3
Success rate using ***the same default parameters***. The top and second best results are denoted by red and blue.

	IMT	IMT	IMT	SI	SHOG	SHaar	SMO	MCS	CVT	Struck	ASLA	SCM	KCF	MIL	TLD	VTD
	-RAM	-IAM	-All					[4]	[22]	[13]	[16]	[37]	[15]	[2]	[18]	[20]
Startrek	91	47	86	1	12	36	44	56	76	78	1	56	74	36	3	89
Starwars	86	83	90	2	75	13	20	19	79	40	85	68	92	45	1	40
David	98	100	99	34	99	34	99	62	100	67	97	95	75	62	96	68
Girl	97	80	98	28	87	85	73	73	81	100	74	99	84	68	46	98
Football	86	79	87	64	76	59	73	57	64	66	65	57	70	73	41	76
CAVIAR	100	99	100	49	44	89	100	100	100	41	97	100	38	38	19	41
Woman	98	93	100	16	9	100	92	97	67	100	100	100	100	16	31	15
Singer1	100	66	100	39	50	98	94	63	70	29	99	100	29	27	99	43
Sylv	68	81	77	45	72	44	45	63	75	92	74	88	81	54	92	80
Trellis	93	99	98	36	68	82	90	62	89	78	85	85	84	24	47	50
Deer	100	100	100	32	98	98	77	33	2	100	2	2	82	12	73	4
Jumping	96	92	95	21	28	7	70	17	10	79	16	12	28	47	84	11
Board	80	89	86	10	77	70	65	50	52	70	71	89	86	51	11	34
Lemming	72	68	85	23	52	17	46	39	38	80	69	30	44	83	4	52
Tiger1	90	87	96	10	50	47	35	42	43	84	83	52	69	62	45	85
Coke	75	59	75	3	44	48	68	58	57	78	69	69	69	32	48	7

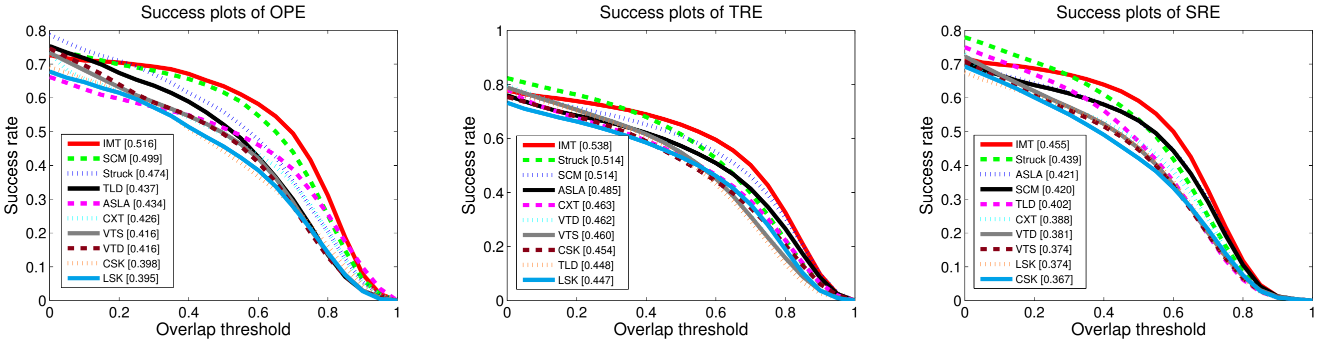


Fig. 7. The area under curve (AUC) of each success plot [31]. OPE: Running the trackers throughout each sequence with initializations of the ground truth positions. TRE: Running the trackers with initialization from the ground truth position at different frames. SRE: Running the trackers with initialization from the different bounding boxes at the first frame. In all evaluation metrics, the IMT performs well against the other state-of-the-art methods.

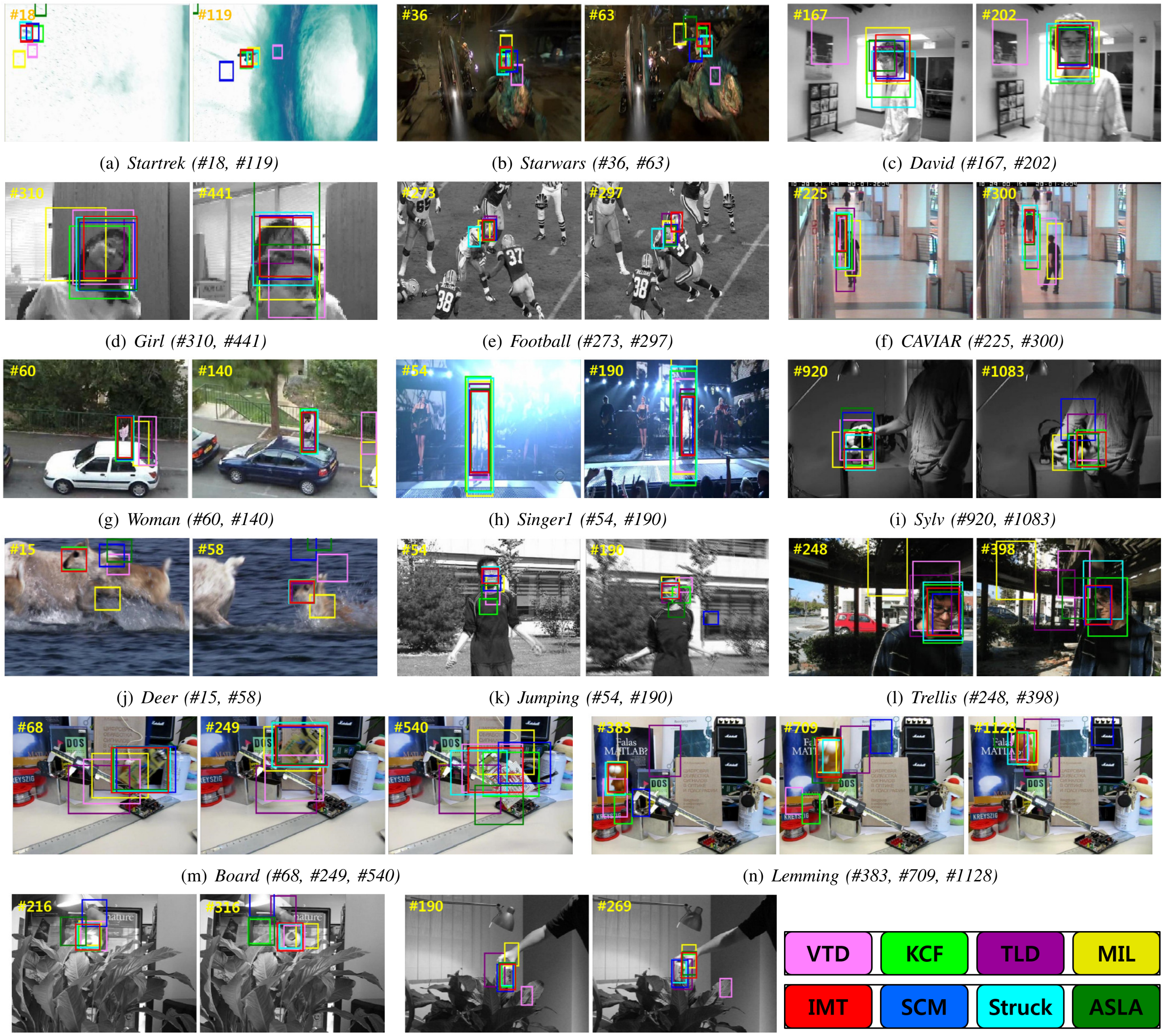
approaches do not track the object reliably when illumination and pose variations occur together (#248 and #398 in Figure 8(l)). In addition, the Struck and VTD methods do not perform well when scale and large illumination changes occur simultaneously (#54 and #190 in Figure 8(h)). The KCF does not deal with large scale changes well as shown in *Singer1* sequence. Different from other tracking methods, the IMT algorithm tracks the object favorably by using complementary features for various appearance changes.

Jumping **and** ***Deer***: The object appearances change significantly due to fast motion and blurs with noise in both sequences. Except for the IMT, Struck, and TLD methods, other trackers do not handle drastic motion blurs well as shown in Table 3 and Figure 8(j) as well as 8(k). The IMT algorithm effectively uses shape features (HOG) to deal with motion blurs. Table 3 shows that better results are obtained by trackers based on SHOG features. Furthermore, by using stable features in the TLF, large noise caused by motion blurs is well handled by the IMT algorithm especially in the *Jumping* sequence.

Tiger1**,** ***Coke*****,** ***Board*****, and** ***Lemming***: The target objects in these sequences undergo various appearance changes including motion blurs, illumination changes, occlusions, and pose variations. When the target object undergoes motion blurs and illumination changes simultaneously in the *Coke* sequence (#190 and #216 in Figure 8(p)), the ASLA, SCM, and KCF methods do not perform well. When frequent partial occlusions occur (e.g., #316 in Figure 8(o) and #190 in Figure 8(p), the ASLA, TLD, KCF, and MIL methods drift away from the target objects. On the other hand, the TLD, VTD, and MIL methods fail to track the objects well (#68 and #249 in Figure 8(m) and #383 and #709 in Figure 8(n)) when motion blurs occur. The ASLA and Struck methods do not perform well when large pose changes occur (#540 in the *Board* sequence and #1128 in the *Lemming* video). In contrast, the IMT algorithm performs well which can be attributed to adaptive use of HOG features to handle motion blurs and Haar-like features to deal with occlusions as shown in Figure 6(c). As the IMT algorithm utilizes transient and stable features for tracker selection and interaction, it is more robust in dealing with large object appearance changes.

9.9 Run Time Performance

We implement the proposed and evaluated methods (i.e., IMT, MCS, and CVT) using MATLAB. For each method, we use



(a) *Startrek (#18, #119)*
(b) *Starwars (#36, #63)*
(c) *David (#167, #202)*
(d) *Girl (#310, #441)*
(e) *Football (#273, #297)*
(f) *CAVIAR (#225, #300)*
(g) *Woman (#60, #140)*
(h) *Singer1 (#54, #190)*
(i) *Sylv (#920, #1083)*
(j) *Deer (#15, #58)*
(k) *Jumping (#54, #190)*
(l) *Trellis (#248, #398)*
(m) *Board (#68, #249, #540)*
(n) *Lemming (#383, #709, #1128)*
(o) *Tiger1 (#216, #316)*
(p) *Coke (#190, #269))*
(q) Tracker Label

Fig. 8. Experimental results of state-of-the-art tracking methods.

600 samples for every tracker. The most time-consuming part of the proposed IMT algorithm is to extract multiple features. As the MCS and CVT methods use the same features (HOG, Haar-like, and intensity), the run time performance is comparable to that of the IMT algorithm (0.8 seconds versus 1.4 seconds per frame). The run time of the IMT is higher as it entails solving an ℓ_1 minimization problem for computing the TLF using (29), which can be further reduced by recent efficient ℓ_1 solvers [33].

10 Conclusions

In this paper, we propose a robust visual tracking algorithm that integrates multiple trackers based on different feature representations via tracker interaction and selection. The tracker interaction is carried out based on the transition probability matrix which is designed to alleviate the drifting problems of less reliable tracking methods. The update of transition probability matrix and tracker selection are computed based on the reliability of each tracker via the proposed tracker likelihood function. To better account for abrupt and gradual appearance changes, each likelihood function is formulated based on transient and stable features. The proposed tracking algorithm selects the best one among multiple trackers to account for object appearance changes. Experimental results on benchmark datasets demonstrate that the proposed tracking algorithm performs favorably against state-of-the-art methods.

References

[1] S. Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261–271, 2007.

[2] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1619–1632, 2011.

[3] V. Badrinarayanan. *Probabilistic graphical models for visual tracking of objects*. PhD thesis, Ph.D thesis, INRIA Rennes Bretagne-Atlantique, 2009.

[4] V. Badrinarayanan, P. Perez, F. L. Clerc, and L. Oisel. Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[5] Y. Bar-Shalom, T. Kirubarajan, and X.-R. Li. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[6] P. A. Brasnett, L. Mihaylova, N. Canagarajah, and D. Bull. Particle filtering with multiple cues for object tracking. In *Proc. of SPIE's Annual Symp. EI ST*, pages 430–441, 2005.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[8] T. B. Dinh, N. Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1177–1184, 2011.

[9] A. Doucet, S. Godsill, and C. Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[10] W. Du and J. Piater. A probabilistic approach to integrating multiple cues in visual tracking. In *European Conference on Computer Vision*, European Conference on Computer Vision, pages 225–238, 2008.

[11] H. Grabner and H. Bischof. On-line boosting and vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 260–267, 2006.

[12] H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *European Conference on Computer Vision*, pages 234–247, 2008.

[13] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *IEEE International Conference on Computer Vision*, pages 263–270, 2011.

[14] J. a. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European Conference on Computer Vision*, pages 702–715, 2012.

[15] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[16] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1822–1829, 2012.

[17] V. P. Jilkov and X. R. Li. Online bayesian estimation of transition probabilities for markovian jump systems. *IEEE Transactions on Signal Processing*, 52:1620–1630, 2004.

[18] Z. Kalal, J. Matas, and K. Mikolajczyk. P-n learning: Bootstrapping binary classifiers by structural constraints. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–56, 2010.

[19] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine Learning Research*, 2007, 2007.

[20] J. Kwon and K. M. Lee. Visual tracking decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1269–1276, 2010.

[21] J. Kwon and K. M. Lee. Tracking by sampling trackers. In *IEEE International Conference on Computer Vision*, pages 1195–1202, 2011.

[22] I. Leichter, M. Lindenbaum, and E. Rivlin. A general framework for combining visual trackers — the "black boxes" approach. *International Journal of Computer Vision*, 67(3):343–363, 2006.

[23] B. Liu, J. Huang, L. Yang, and C. Kulikowsk. Robust tracking using local sparse appearance model and k-selection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1313–1320, 2011.

[24] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2259–2272, 2011.

[25] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai. Minimum error bounded efficient 1 tracker with occlusion detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1257–1264, 2011.

[26] F. Moreno-Noguer, A. Sanfeliu, and D. Samaras. Dependent multiple cue integration for robust tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:670–685, 2008.

[27] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3):125–141, 2008.

[28] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. Prost: Parallel robust online simple tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 723–730, 2010.

[29] M. Spengler and B. Schiele. Towards robust multi-cue integration for visual tracking. *Machine Vision and Applications*, 14(1):50–58, 2003.

[30] H. Wang and D. Suter. Efficient visual tracking by probabilistic fusion of multiple cues. In *International Conference on Pattern Recognition*, pages 892–895, 2006.

[31] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[32] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *CoRR*, abs/1304.5634, 2013.

[33] A. Y. Yang, A. Ganesh, Z. Zhou, S. Sastry, and Y. Ma. A review of fast l1-minimization algorithms for robust face recognition. *CoRR*, abs/1007.3753, 2010.

[34] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Survey*, 38(4), 2006.

[35] J. H. Yoon, D. Y. Kim, and K.-J. Yoon. Visual tracking via adaptive tracker selection with multiple features. In *European Conference on Computer Vision*, volume 7575, pages 28–41, 2012.

[36] E. Zelniker, T. M. Hospedales, S. Gong, and T. Xiang. A unified bayesian framework for adaptive visual tracking. In *British Machine Vision Conference*, 2009.

[37] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1838–1845, 2012.

Ju Hong Yoon received the B.S. degree in electrical and electronic engineering from Sungkyunkwan University in 2008 and the M.S. and Ph.D degrees from the Gwangju Institute of Science and Technology in 2009 and 2014, respectively. He is currently a senior researcher at Korea Electronics Technology Institute. His current research includes multi-object tracking, stereo vision, filtering theory, etc.

Ming-Hsuan Yang is an associate professor in Electrical Engineering and Computer Science at University of California, Merced. He received the PhD degree in computer science from the University of Illinois at Urbana-Champaign in 2000. Prior to joining UC Merced in 2008, he was a senior research scientist at the Honda Research Institute working on vision problems related to humanoid robots. Yang served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence from 2007 to 2011, and is an associate editor of the International Journal of Computer Vision, Image and Vision Computing and Journal of Artificial Intelligence Research. He received the NSF CAREER award in 2012, the Senate Award for Distinguished Early Career Research at UC Merced in 2011, and the Google Faculty Award in 2009. He is a senior member of the IEEE and the ACM.

Kuk-Jin Yoon received the B.S., M.S., and Ph.D. degrees in Electrical Engineering and Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in 1998, 2000, 2006, respectively. He was a post-doctoral fellow in the PERCEPTION team in INRIA-Grenoble, France, for two years from 2006 and 2008 and joined the School of Information and Communications in Gwangju Institute of Science and Technology (GIST), Korea, as an assistant professor in 2008. He is currently an associate professor and a director of the Computer Vision Laboratory in GIST.