Bayesian Multi-Object Tracking Using Motion Context from Multiple Objects

Ju Hong Yoon
KETI
jhyoon@keti.re.kr

Ming-Hsuan Yang
UC Merced
mhyang@ucmerced.edu

Jongwoo Lim
Hanyang University
jlim@hanyang.ac.kr

Kuk-Jin Yoon
GIST
kjyoon@gist.ac.kr

Abstract

Online multi-object tracking with a single moving *camera is a challenging problem as the assumptions of 2D conventional motion models (e.g., first or second order models) in the image coordinate no longer hold because of global camera motion. In this paper, we consider motion context from multiple objects which describes the relative movement between objects and construct a Relative Motion Network (RMN) to factor out the effects of unexpected camera motion for robust tracking. The RMN consists of multiple relative motion models that describe spatial relations between objects, thereby facilitating robust prediction and data association for accurate tracking under arbitrary camera movements. The RMN can be incorporated into various multi-object tracking frameworks and we demonstrate its effectiveness with one tracking framework based on a Bayesian filter. Experiments on benchmark datasets show that online multi-object tracking performance can be better achieved by the proposed method.*

1. Introduction

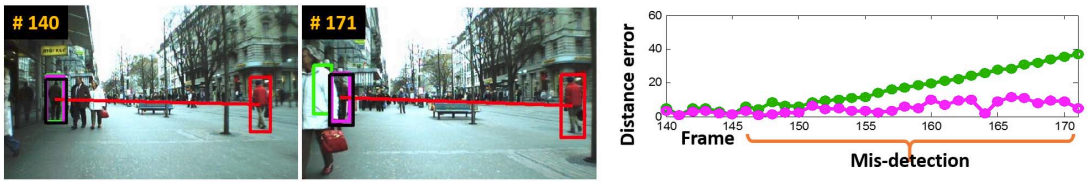
Multi-object tracking (MOT) is of great importance for numerous computer vision tasks with applications such as surveillance, traffic safety, automotive driver assistance systems, and robotics. Thanks to advances of object detectors [3, 4], detection-based MOT methods have been extensively studied in recent years. In this approach, the goal is to determine the trajectories and identities of target instances throughout an image sequence using the detection results of each frame as observations.

In general, *detection-based* tracking methods can be categorized into online and batch methods. The online methods solve the MOT problem using only the past frames up to the current frame [15, 20, 19, 2]. In contrast, the batch or delayed-output methods utilize the visual information in the entire sequence or the future frames; hence, they iteratively optimize the detection assignment of the current frame using the future information [16, 24, 23, 1, 13]. In terms of tracking accuracy, the methods in the second group are usually more accurate as forward-backward visual information is available for disambiguation. However, for online applications such as driver assistance systems and service robots, the online approach is more suitable since the tracking results in the current frame are available instantly. In this paper, we address this *online* MOT problem.

In *detection-based* MOT, as each trajectory is constructed by matching multiple detected objects of the same class across frames, data association plays an essential role for robust tracking. For data association, both appearance and motion models are typically used and thus of critical importance. In many situations, appearance models alone are not adequate to discriminate objects, particularly for separating instances of the same class (e.g., pedestrians), since their shape and texture look similar. This problem is more critical in *online* MOT methods because the information to reduce such ambiguities is limited compared to batch MOT methods.

With such ambiguities in appearance, motions and positions must be used to correctly associate the confusing detections to the objects. In previous works, 2D object motion in image plane is typically described by a first or second order model based on the past tracking results [15, 20, 19, 2]. These 2D conventional motion models are effective when the objects are continuously detected and when the camera is stationary or slowly moving, e.g., the objects in the red boxes in Fig. 1. However, they quickly become unreliable when objects are occluded or undetected for several frames and at the same time the camera moves or fluctuates. In many MOT application scenarios, the camera may be on a moving platform, such as a vehicle or a pan-tilt unit, and the unpredicted global motion cause many existing MOT algorithms to fail because the predicted object position by the simplistic motion models is often far from the re-detected object position, e.g., the green box in Fig. 1. Nevertheless, considerably less attention has been paid to motion modeling than appearance models in MOT, especially for scenes with moving cameras. To resolve this problem, in this paper, we propose a novel method for online MOT in complex moving scenes, which can be applied to various scenarios without knowing either scene dynamics or camera

978-1-4799-6683-7/15 $31.00 © 2015 IEEE
DOI 10.1109/WACV.2015.12



(a) *Bahnhof* seq.(# 140, # 171)

(b) *Jelmoli* seq.(# 256, # 278)

Figure 1. Examples for effectiveness of a relative motion. *Black box*: ground truth, *Red box*: a well-tracked object, *Magenta box*: prediction based on the well-tracked object with the proposed RMN, *Green box*: prediction by conventional self-motion model. The distance error in each graph shows that the prediction based on well-tracked object with the RMN is much more accurate than the prediction based on the conventional self-motion model.

motion. The proposed Relative Motion Network (RMN) algorithm accounts for the motion context from multiple moving objects which are mostly insensitive to unexpected camera motions. Two examples of successful predictions using the proposed RMN algorithm are shown in Fig. 1 where the tracks of occluded or undetected objects are recovered after several frames. Note that the proposed algorithm does not consider very abrupt camera motions and fluctuations, but consider realistic scenarios where the camera moves at a moderate speed with some fluctuations (as in the supplementary video) where at least a few objects can be tracked with continuous detections and predicted well, e.g., the objects in the red boxes of Fig. 1. In such situations, the RMN helps tracking other undetected objects after the objects are re-detected again. Furthermore, we also incorporate the proposed RMN into Bayesian framework to formulate general online MOT algorithm.

Numerous quantitative evaluations against several state-of-the-art methods on benchmark datasets show that the proposed online MOT algorithm can handle aforementioned challenges effectively.

2. Related Works and Problem Context

We introduce representative MOT algorithms that focus on motion models, which can be categorized based on static or dynamic camera assumptions as considered in this work.

Static camera: Takala et al. [20] propose to measure directional smoothness and speed of each object based on the current location and the past trajectory to track multiple objects. In [23], nonlinear motion patterns of each object and the entry/exit maps are trained by exploiting past and future object trajectories. The trained trajectory information is used in correcting mis-detections In [2], object velocity is used to generate confidence maps of future trajectories for tracking. Despite demonstrated success in multi-object tracking, the aforementioned methods only works with static cameras.

Dynamic camera: Duan et al. [5] use both individual and mutual relation models to handle occlusions and large shape changes but they are not used in disambiguating the objects in data association. Their mutual relation model has a limitation that it only works when the objects move in the same direction. In [24], the pairwise relative object motion model is developed as an additional similarity function for MOT, which uses both past and future tracklets. Leal-Taixé *et al.* [11] propose robust motion models which are trained in offline manner based on motion training samples. On the contrary, our method does not need training data for constructing motion models.

In this paper, we exploit *all* relative motions between objects for *online* MOT. For single object tracking, the relative distance between feature points in image has been used to reduce tracking drifts during occlusion or drastic appearance change [7]. For multi-object tracking, [25] utilizes relative spatial constraint between objects. However, they do not utilize a detector; hence, the data association between objects and detections is not considered. However, different from [7, 25], the proposed MOT algorithm considers data association problem between detections and objects, and the RMN is used to enhance the data association performance. In addition, we design relative motion weights to consider different contributions from other objects, and object states and relative motion weights are continuously estimated within the Bayesian framework.

3. Relative Motion Network for Tracking

In this paper, the state of object i (i.e. i-th object) is defined as $\mathbf{x}_t^i = [u_t^i, v_t^i, \dot{u}_t^i, \dot{v}_t^i, w_t^i, h_t^i]^\top$, where (u_t^i, v_t^i)

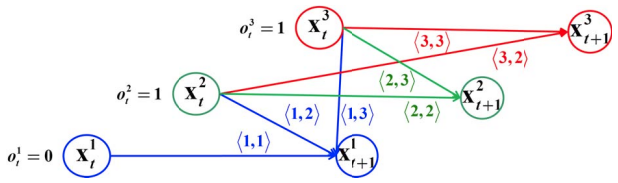


Figure 2. Prediction of object state transition based on RMN and (4): $\mathcal{R}_t = \cup_{i=1}^{3} \mathcal{R}_t^i$ where $\mathcal{R}_t^1 = \{\langle 1,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle\}$, $\mathcal{R}_t^2 = \{\langle 2,2\rangle, \langle 2,3\rangle\}$, $\mathcal{R}_t^3 = \{\langle 2,3\rangle, \langle 3,3\rangle\}$. A detection event of the i-th object is defined as $o_t^i \in \{0,1\}$ in (14). When the object is detected $o_t^i = 1$; otherwise, $o_t^i = 0$. In this example, the 1-st object is not detected ;hence, the relative motion models from the 1-st object are not included in $\mathcal{R}$.

and $(\dot{u}_t^i, \dot{v}_t^i)$ denote the center position and velocity, respectively; w_t^i, and h_t^i represent the width and height of the object bounding box; and t is the frame or time index. The proposed online MOT algorithm uses the relative motion between two objects i and j based on the position and velocity difference as

$$\begin{aligned} \mathbf{r}_t^{\langle i,j \rangle} &\triangleq \mathbf{r}(\mathbf{x}_t^i, \mathbf{x}_t^j) \\ &= [\mu_t^{\langle i,j \rangle}, \nu_t^{\langle i,j \rangle}, \dot{\mu}_t^{\langle i,j \rangle}, \dot{\nu}_t^{\langle i,j \rangle}]^\top \\ &= [u_t^i - u_t^j, v_t^i - v_t^i, \dot{u}_t^i - \dot{u}_t^j, \dot{v}_t^i - \dot{v}_t^j]^\top, \end{aligned} \tag{1}$$

where $(\mu_t^{\langle i,j \rangle}, \nu_t^{\langle i,j \rangle})$ and $(\dot{\mu}_t^{\langle i,j \rangle}, \dot{\nu}_t^{\langle i,j \rangle})$ represent the spatial and velocity difference, respectively. The i-th object has a set of relative motion vectors with respect to other objects and each one is used as a motion model for the i-th object. When we have N objects, a relative motion network (RMN) $\mathcal{R}_t$ is defined as

$$\begin{aligned} \mathcal{R}_t &= \cup_{i=1}^{N} \mathcal{R}_t^i, \\ \mathcal{R}_t^i &= \{\langle i,i \rangle\} \cup \{\langle i,j \rangle |\, o_t^j = 1, 1 \le j \le N\}, \end{aligned} \tag{2}$$

where the RMN represents a set of linked edges between objects. Here, we include the $\langle i,j \rangle$ relative motion model in $\mathcal{R}_t^i$ only if the j-th object is detected at frame t, which is represented by the detection event $o_t^j = 1$ (defined in (14)). Otherwise, we do not include the $\langle i,j \rangle$ relative motion model in $\mathcal{R}_t$ because detection failures are caused by various reasons such as disappearance and occlusion. The self motion model (i.e., denoted by $\langle i,i \rangle$) is always included in case there exists only one object. Since the motion correlations between pairs of objects are different, we consider the motion correlations by using relative weights $\theta_t^{\langle i,j \rangle}$ as

$$\Theta_t^i = \{\theta_t^{\langle i,j \rangle} | \langle i,j \rangle \in \mathcal{R}_t^i, 1 \le j \le N\}, \quad \sum_{\langle i,j \rangle \in \mathcal{R}_t^i} \theta_t^{\langle i,j \rangle} = 1, \tag{3}$$

where we set initial relative weights uniformly based on the set $\mathcal{R}_t^i$ as $\theta_t^{\langle i,j \rangle} = \frac{1}{|\mathcal{R}_t^i|}$ ($|\mathcal{R}_t^i|$ denotes the cardinality of a set $\mathcal{R}_t^i$). With the relative motion in the RMN, we design the object motion model in (4) that enables the i-th object state transition from the previous j-th object state selected from $\langle i,j \rangle \in \mathcal{R}_t$. One example is shown in Fig. 2. With one of relative motion models from the RMN, the motion transition is formulated by

$$\begin{aligned} \mathbf{x}_{t+1}^i &= f(\mathbf{x}_t^j, \langle i,j \rangle) + \mathbf{w} \\ &= \mathbf{F}[[u_t^j, v_t^j, \dot{u}_t^j, \dot{v}_t^j] + \mathbf{r}_t^{\langle i,j \rangle^\top}, w_t^i, h_t^i]^\top + \mathbf{w}, \end{aligned} \tag{4}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

is the transition matrix based on a constant velocity motion model; the object width and height are independent from the relative motion; and $\mathbf{w}$ represents the assumed white Gaussian noise model in this paper.

4. Online Multi-object Tracking with RMN

As mentioned, the goal of online MOT is to estimate a set of object states $\mathbf{X}_t$ from a set of current observations $\mathbf{Z}_t$. The set of N object states and the set of M observations are expressed as $\mathbf{X}_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^N\}$ and $\mathbf{Z}_t = \{\mathbf{z}_t^1, \ldots, \mathbf{z}_t^M\}$, respectively. In this paper, we utilize the RMN to achieve robust multi-object state estimation.

4.1. Bayesian Filter with the RMN

We solve the multi-object tracking problem within the Bayesian framework. Assuming that object states are independent of each other in a way similar to other existing methods [2, 8], the goal is to maximize the posterior probability of the object state $\mathbf{x}_t^i$ given the observation history $\mathbf{Z}_{0:t} = \{\mathbf{Z}_0, \ldots, \mathbf{Z}_t\}$ and the RMN. The posterior probability is defined by

$$\begin{aligned} &p(\mathbf{x}_t^i | \mathbf{Z}_{0:t}, \mathcal{R}_t^*) \\ &\propto \sum_{\langle i,j \rangle \in \mathcal{R}_t^*} \theta_t^{\langle i,j \rangle} p(\mathbf{Z}_t | \mathbf{x}_t^i, \langle i,j \rangle) p(\mathbf{x}_t^i | \mathbf{Z}_{0:t-1}, \langle i,j \rangle), \end{aligned} \tag{5}$$

where the RMN before the update is defined by $\mathcal{R}_t^* = \mathcal{R}_{t-1}$. The posterior probability is decomposed with the relative motion models and their weights $\theta_t^{\langle i,j \rangle}$. The prior probability of each object state is modeled with the relative motion model based on a first-order Markov chain:

$$\begin{aligned} &p(\mathbf{x}_t^i | \mathbf{Z}_{0:t-1}, \langle i,j \rangle) \\ &= \int p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^j, \langle i,j \rangle) p(\mathbf{x}_{t-1}^j | \mathbf{Z}_{0:t-1}, \langle i,j \rangle) d\mathbf{x}_{t-1}^j, \end{aligned} \tag{6}$$

where the transition density $p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^j, \langle i,j \rangle)$ is described by the motion model in (4). The likelihood function $p(\mathbf{Z}_t | \mathbf{x}_t^i, \langle i,j \rangle)$ is designed by considering association events that the k-th observation is assigned to the i-th object with the $\langle i,j \rangle$ relative motion model. This event probability is denoted by $P_k(E_t^{\langle i,j \rangle})$. We also consider the event

that none of the observation is associated with the i-th object, which is denoted by $P_0(E_t^{\langle i,j\rangle})$. Then, the likelihood function is composed of the probability of these association events by

$$p(\mathbf{Z}_t|\mathbf{x}_t^i,\langle i,j\rangle) \triangleq P_0(E_t^{\langle i,j\rangle})+\sum_k P_k(E_t^{\langle i,j\rangle})p(\mathbf{z}_t^k|\mathbf{x}_t^i,\langle i,j\rangle), \tag{7}$$

where the likelihood function of the k-th observation and the i-th object is $p(\mathbf{z}_t^k|\mathbf{x}_t^i,\langle i,j\rangle)$, which is used for object state update via a Kalman filter in Algorithm 1.

The relative weight $\theta_t^{\langle i,j\rangle}$ in (5) is also updated with the event probabilities and observations by

$$\begin{aligned}\theta_t^{\langle i,j\rangle} &\triangleq P(\langle i,j\rangle|\mathbf{Z}_{0:t}) = \frac{P(\langle i,j\rangle|\mathbf{Z}_{0:t-1})P(\mathbf{Z}_t|\langle i,j\rangle)}{\sum_{\langle i,j\rangle\in\mathcal{R}_t^*}P(\langle i,j\rangle|\mathbf{Z}_{0:t-1})P(\mathbf{Z}_t|\langle i,j\rangle)},\\ P(\mathbf{Z}_t|\langle i,j\rangle) &= P_0(E^{\langle i,j\rangle}) + \textstyle\sum_k P_k(E_t^{\langle i,j\rangle})P(\mathbf{z}_t^k|\langle i,j\rangle),\end{aligned} \tag{8}$$

where the prior relative weight is defined by $\theta_{t-1}^{\langle i,j\rangle} \triangleq P(\langle i,j\rangle|\mathbf{Z}_{0:t-1})$ and the model likelihood function is computed by the PASCAL score [18] as

$$P(\mathbf{z}_t^k|\langle i,j\rangle) = \frac{\text{area}(T(\mathbf{z}_t^k)\cap T(\mathbf{x}_t^{\langle i,j\rangle}))}{\text{area}(T(\mathbf{z}_t^k)\cup T(\mathbf{x}_t^{\langle i,j\rangle}))}, \tag{9}$$

where the i-th object state from the j-th object state is computed by $\mathbf{x}_t^{\langle i,j\rangle} = f(\mathbf{x}_{t-1}^j,\langle i,j\rangle)$ from (4) and $T(\cdot)$ denotes a bounding box of the state vector $\mathbf{x}$ or the observation $\mathbf{z}$.

To update the object states and relative weights, the event probabilities are determined by solving the data association in the next section.

4.2. Event Probability via Data Association

To solve the online MOT problem, we need to associate each observation to an object. The similarity function for the data association is defined as

$$\begin{aligned}\Lambda(\mathbf{z}_t^k,\mathbf{x}_t^i) &\triangleq P(\mathbf{z}_t^k|\mathbf{x}_t^i,\mathcal{R}_t^*)\\ &= P_m(\mathbf{z}_t^k|\mathbf{x}_t^i,\mathcal{R}_t^*)P_s(\mathbf{z}_t^k|\mathbf{x}_t^i)P_a(\mathbf{z}_t^k|\mathbf{x}_t^i),\end{aligned} \tag{10}$$

where we also consider the size similarity $P_s(\mathbf{z}_t^k|\mathbf{x}_t^i)$ (defined in (17)) and appearance similarity $P_a(\mathbf{z}_t^k|\mathbf{x}_t^i)$ (defined in (16)) together in a way similar to existing MOT methods. In the proposed algorithm, we consider the motion similarity $P_m(\mathbf{z}_t^k|\mathbf{x}_t^i,\mathcal{R}_t^*)$ based on the RMN. We select the most important and contributive relative motion model of the i-th object to the k-th observation according to the updated relative weight $\theta_t^{\langle i,j\rangle}(\mathbf{z}_t^k)$ to minimize the cost function in (12). This is because that the contributions from the other objects are not equal, and the predicted states from less contributive relative motion models are less reliable and less related to the k-th observation. Thus, by selecting the most contributive relative motion model, we exclude the predicted states from the less contributive relative motion models in data association,

$$\begin{aligned}&P_m(\mathbf{z}_t^k|\mathbf{x}_t^i,\mathcal{R}_t^*) \approx P(\mathbf{z}_t^k|\langle i,j\rangle_k),\\ &\langle i,j\rangle_k = \arg\max_{\langle i,j\rangle\in\mathcal{R}_t^*}\theta_t^{\langle i,j\rangle}(\mathbf{z}_t^k),\\ &\theta_t^{\langle i,j\rangle}(\mathbf{z}_t^k) = \frac{P(\mathbf{z}_t^k|\langle i,j\rangle)\theta_{t-1}^{\langle i,j\rangle}}{\sum_{\langle i,j\rangle\in\mathcal{R}_t^*}P(\mathbf{z}_t^k|\langle i,j\rangle)\theta_{t-1}^{\langle i,j\rangle}},\end{aligned} \tag{11}$$

where the motion similarity is computed by the PASCAL score $P(\mathbf{z}_t^k|\langle i,j\rangle)$ from (9) and $\langle i,j\rangle_k$ is the selected relative motion model index. In this paper, we solve the data association problem as a bijective matching task by using Hungarian algorithm. Based on the similarity function in (10), we obtain the cost function between the i-th object and the k-th observation as $C_t^{i,k} = -\ln\Lambda(\mathbf{z}_t^k,\mathbf{x}_t^i)$. We compute the assignment matrix $\mathcal{A} = [a^{i,k}]_{N\times M}$ that minimizes the cost

$$\begin{aligned}&\arg\min_{\mathcal{A}}\sum_{i,k}C_t^{i,k}a^{i,k},\\ &\text{s.t.}\ \sum_i a^{i,k} = 1,\forall i\ \text{ and }\ \sum_k a^{i,k} = 1,\forall k,\end{aligned} \tag{12}$$

where an assignment indicator is defined as $a^{i,k}\in\{0,1\}$. The association between objects and observations are determined as follows. When $a^{i,k}=1$, the observation assignment is obtained by following two cases.

(I) $C_t^{i,k} < \tau$, the i-th object $\mathbf{x}_t^i$ is associated with the k-th observation $\mathbf{z}_t^k$. Then, the assignment observation is $\gamma_t^{i,k} = 1$. (Note that we empirically select the threshold τ as [2] and fix it in all the experiments.)

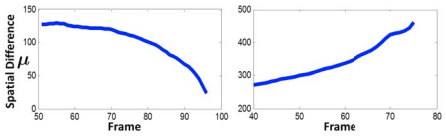
(II) $C_t^{i,k} > \tau$, the i-th object is not associated with the k-th observation. The observation assignment is $\gamma_t^{i,k} = 0$.

Since the detection event represents the association between the i-th object and the k-th observation, we utilize the observation assignment $\gamma_t^{i,k}$ from the data association in computing the event probability. The event probability is computed by

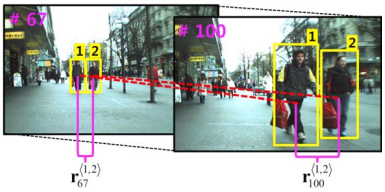
$$P_k(E_t^{\langle i,j\rangle}) = \frac{\gamma_t^{i,k}}{|\mathcal{R}_t^{i*}|},\ P_0(E_t^{\langle i,j\rangle}) = \frac{1}{|\mathcal{R}_t^{i*}|} - \sum_k P_k(E_t^{\langle i,j\rangle}), \tag{13}$$

where $\mathcal{R}_t^{i*}\subset\mathcal{R}_t^*$ and we divide the event probability by $|\mathcal{R}_t^{i*}|$ (the cardinality of a set of the relative motion models used for the i-th object) to make the total sum of event probabilities along the relative motion models $\langle i,j\rangle\in\mathcal{R}_t^{i*}$ always be 1. These event probabilities are used for the update of object states in (7) and relative weights in (8). If the i-th object is associated with any observations, the i-th object is successfully detected. Hence, the detection event is simply obtained by the event probabilities as follows

$$o_t^i = \sum_{\langle i,j\rangle\in\mathcal{R}^*}\sum_k P_k(E_t^{\langle i,j\rangle}), \tag{14}$$



(a)



(b)

Figure 3. (a) Examples of time-varying relative motion in a moving camera: Although the camera moves or fluctuates, the spatial difference changes following a certain dynamic model. (b) Objects moving in a group: Although two objects move coherently in a group, their spatial difference changes according to the geometric relation between the objects and a camera.

where the detection event of the i-th object is consequently defined as $o_t^i \in \{0, 1\}$. If the object is associated with any observation after the data association, then $o_t^i = 1$; otherwise, $o_t^i = 0$. As a result, we obtain a set of detection events $\{o_t^i | \, i = 1, \ldots, N\}$ with which the links between the objects in the RMN are updated based on (2). This updated RMN $\mathcal{R}_t$ is used as $\mathcal{R}_{t+1}^* = \mathcal{R}_t$ for the next time step. All non-associated observations are used for new object initialization.

5. Relative Motion Update

Existing MOT methods [5, 24] assume that the relative motion between two objects is static. However, this assumption does not generally hold because of different directions and speed of object motion or geometric relation with a camera as shown in Fig. 3. For these reasons, in this work, we consider the time-varying relative motion to deal with general situations. The relative motion typically changes in piecewise linear patterns as shown in Fig. 3(a) and thus we model their variation with a constant velocity model and update the relative motion using a Kalman filter [22] with the following transition and observation models,

$$\begin{aligned} \mathbf{r}_t^{\langle i,j \rangle} &= \mathbf{F}_\mathbf{r} \mathbf{r}_{t-1}^{\langle i,j \rangle} + \mathbf{v}_\mathbf{r} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{r}_{t-1}^{\langle i,j \rangle} + \mathbf{v}_\mathbf{r}, \\ \mathbf{y}_t^{\langle i,j \rangle} &= [u_{\mathbf{z},t}^{k_i} - u_{\mathbf{z},t}^{k_j}, \; v_{\mathbf{z},t}^{k_i} - v_{\mathbf{z},t}^{k_j}]^\top \\ &= \mathbf{H}_\mathbf{r} \mathbf{r}_t^{\langle i,j \rangle} + \mathbf{w}_\mathbf{r} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{r}_t^{\langle i,j \rangle} + \mathbf{w}_\mathbf{r}, \end{aligned} \tag{15}$$

where $(u_\mathbf{z}, v_\mathbf{z})$ is an observation position; k_i represents the associated observation index with the i-th object when $\gamma^{i,k} = 1$ from the data association; $\mathbf{F}_\mathbf{r}$ denotes a constant velocity motion model; $\mathbf{H}_\mathbf{r}$ converts the relative motion to the relative observation; and $\mathbf{v}_\mathbf{r}$ and $\mathbf{w}_\mathbf{r}$ are the assumed white Gaussian noise terms. If one of objects is not detected, then the relative motion is simply estimated by prediction using the motion model $\mathbf{r}_t^{\langle i,j \rangle} = \mathbf{F}_\mathbf{r} \mathbf{r}_{t-1}^{\langle i,j \rangle}$.

6. Implementation

Since our algorithm is formulated based on the Bayesian framework, it can be implemented with one of various filtering methods such as Kalman and particle filters. In this paper, we adopt and modify a Kalman filter to approximate the proposed tracking algorithm. The Kalman filter have been applied to multi-object tracking to estimate an object trajectory or motion [8] with object detectors. The main steps of our algorithm are summarized in Algorithm 1.

Appearance and size similarity: The appearance of an object is represented by a color histogram, and the similarity between detections and objects are computed by Bhattacharyya coefficient [15].

$$P_a(\mathbf{z}_t^k, \mathbf{x}_t^i) = \sum_{j=1}^{N_\mathcal{H}} \sqrt{\mathcal{H}^j(\mathbf{z}_t^k) \mathcal{H}^j(\mathbf{x}_t^i)}, \tag{16}$$

where $\mathcal{H}^j(\mathbf{z}_t^k)$ and $\mathcal{H}^j(\mathbf{x}_t^i)$ denotes histogram of the i-th object and the k-th detection; j represents the j-th bin; and $N_\mathcal{H}$ is the number of bins. In this work, we use 64 bins for each color space. Hence, 192 bins are totally used to represent object appearance.

We assume the aspect ratio between width and height is constant. The size similarity is computed by

$$P_s(\mathbf{z}_t^k, \mathbf{x}_t^i) = 1 - \frac{|h_{t-1}^i - h_{\mathbf{z},t}^k|}{h_{t-1}^i + h_{\mathbf{z},t}^k}, \tag{17}$$

where h_{t-1}^i denotes the height of the i-th object state and $h_{\mathbf{z},t}^k$ represents the height of the k-th observation.

Initialization and termination: In this work, objects are managed in a way similar to [2], and a relative motion model is created when a new object is initialized. If previous and current observations have overlaps for a few frames and are not associated with the other existing objects, a new instance is created. Relative motion models between the new object with all the others are then generated. If an object is not associated with any observation for a certain period, it is terminated and the corresponding relative motion models are removed.

7. Experiments

Datasets: We use 7 benchmark sequences to demonstrate the effectiveness of the proposed RMN algorithm. Five benchmark sequences were recored by a moving camera, i.e., ETH dataset [6] (*Bahnhof*, *Sunnyday*, and *Jelmoli* [1]) and two sequences from Youtube (i.e., *Marathon1*,

[1] http://www.vision.ee.ethz.ch/∼aess/dataset/

Algorithm 1 Online RMN Multi-Object Tracking (RMOT)

Input:
– RMN: $\mathcal{R}_t^* = \{\mathcal{R}_t^{1*}, \ldots, \mathcal{R}_t^{N*}\}$ where $\mathcal{R}_t^* = \mathcal{R}_{t-1}$,
– Relative Weight: $\Theta_{t-1} = \{\Theta_{t-1}^1, \ldots, \Theta_{t-1}^N\}$ where $\Theta_{t-1}^i = \{\theta_{t-1}^{\langle i,j\rangle} \,|\langle i,j\rangle \in \mathcal{R}_{t-1}^i, 1 \le j \le N\}$.
– Object: $\mathbf{x}_{t-1}^i \sim \mathcal{N}(\bar{\mathbf{x}}_{t-1}^i, \mathbf{P}_{t-1}^i),\ i = 1, \ldots, N,$ ▷ mean $\bar{\mathbf{x}}_{t-1}^i$ and covariance $\mathbf{P}_{t-1}^i$ of the i-th object state
– Observation: $\mathbf{Z}_t = \{\mathbf{z}_t^1, \ldots, \mathbf{z}_t^M\}$,
• **Object State Prediction with RMN**
  for $i = 1 : N$
    for $\langle i,j\rangle \in \mathcal{R}_t^*$
      – $\bar{\mathbf{x}}_{t|t-1}^{\langle i,j\rangle} = f(\bar{\mathbf{x}}_{t-1}^j, \langle i,j\rangle)$ in (4)
      – $\bar{\mathbf{P}}_{t|t-1}^{\langle i,j\rangle} = \mathbf{F}\bar{\mathbf{P}}_{t-1}^i\mathbf{F}^\top + \mathbf{Q}$ ▷ $\mathbf{F}$ is from (4) and covariance $\mathbf{Q}$
    end for
    – $\chi^i = \{(\bar{\mathbf{x}}_{t|t-1}^{\langle i,j\rangle}, \mathbf{P}_{t|t-1}^{\langle i,j\rangle}) | \langle i,j\rangle \in \mathcal{R}_t^{i*}\}$
  end for
• **Data Association**
  – Using χ^i and Θ_{t-1}, computing the cost matrix $[C_t^{i,k}]_{N\times M}$ from the similarity function in (10)-(11)
  – Observation assignments $[\gamma_t^{i,k}]_{N\times M}$ are obtained from data association (12).
  – Event probabilities (i.e., $P_k(E_t^{\langle i,j\rangle})$ and $P_0(E_t^{\langle i,j\rangle})$ in (13) and detection events $\{o_t^i | i = 1, \ldots, N\}$ in (14)
• **Update of Object States and RMN**
  for $i = 1 : N$
    – A set of relative weights $\theta_t^{\langle i,j\rangle} \in \Theta_t^i$ is updated in (8).
    if $o_t^i = 1$
      – $\langle i,j\rangle^* = \max_{\langle i,j\rangle \in \mathcal{R}_t^*} \theta_t^{\langle i,j\rangle}$ from (8).
      (Kalman filter update)
      – $\mathbf{K}_t^i = \mathbf{P}_{t|t-1}^{\langle i,j\rangle^*}\mathbf{H}^\top(\mathbf{H}\mathbf{P}_{t|t-1}^{\langle i,j\rangle^*}\mathbf{H}^\top + \mathbf{R})^{-1}$ ▷ Noise covariance $\mathbf{R}$
      – $\bar{\mathbf{x}}_t^i = \bar{\mathbf{x}}_{t|t-1}^{\langle i,j\rangle^*} + \sum_k \gamma_t^{i,k}\mathbf{K}_t^i(\mathbf{z}_t^k - \mathbf{H}\bar{\mathbf{x}}_{t|t-1}^{\langle i,j\rangle^*})$
      – $\mathbf{P}_t^i = \mathbf{P}_{t|t-1}^{\langle i,j\rangle^*} - \mathbf{K}_t^i(\mathbf{H}\mathbf{P}_{t|t-1}^{\langle i,j\rangle^*}\mathbf{H}^\top + \mathbf{R})\mathbf{K}_t^{i\top}$
    else
      – $\bar{\mathbf{x}}_t^i = \bar{\mathbf{x}}_{t|t-1}^{\langle i,i\rangle},\ \mathbf{P}_t^i = \mathbf{P}_{t|t-1}^{\langle i,i\rangle}$
    end if
  end for
  – Updating the RMN $\mathcal{R}_t$ in (2) with detection events $\{o_t^i | i = 1, \ldots, N\}$.
• **Relative Motion Update**
  – Updating each relative motion with the transition and the observation model in (15) via a Kalman filter for a given relative observation.
Parameter: The observation matrix $\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$. $\mathbf{R}$ and $\mathbf{Q}$ are given in the supplementary material.

Marathon2). We consider real situations where the camera moves at a reasonable speed and some jitters (as in the supplementary video). These sequences contain fluctuated scenes as a result of camera motion. The other two sequences are *TUD* and *PETL1* dataset which were obtained by a static camera [2]. For the ETH dataset, we only use sequences from the left camera without any information regarding depth, scene dynamics, and camera motion. Detection results and the ground truth of the *Bahnhof*, the *Sunnyday*, the *TUD*, and the *PETL1* sequences from the website [2]. For the *Jelmoli* sequence, we use the detector from [4] for tests. To generate detections for the *Marathon1* and *Marathon2* sequences, we use a face detector [21].

Trackers: We compare the proposed MOT algorithm with a baseline tracker which utilizes the self motion model (SMM), and the baseline tracker is named as SMM-MOT (SMOT). For fair comparisons, the SMOT is also implemented based on the same MOT framework described in Algorithm 1 but with the self motion model. We also compare our method (RMOT) with other state-of-the-art methods, i.e., two *Online* methods (StructMOT [9] and MOTTBD [17]) and four *Offline* methods (PRIMPT [10], OnlineCRF [24], and CemTracker [13]), we use the reported results in their paper. For new benchmark sequences (Jelmoli, Marathon1, Marathon2), we compare the proposed RMOT with the SMOT. To achieve fair comparisons, we use the same detection results and same ground truth.

To facilitate understanding of the proposed RMOT, the MATLAB code, datasets, ground truth data will be made available to the public `https://cvl.gist.ac.kr/project/rmot.html`.

Runtime: All the experiments are carried out on a Intel 3.4 GHz PC with 8 G memory. Given the detections, the average computation time of the current MATLAB implementation is approximately 2.64×10^{-2} seconds to obtain the tracking results without any code optimization. To be specific, for K objects, the RMN update takes approximately $6.0 \times 10^{-4} \times \frac{K(K-1)}{2}$ seconds. The object state estimation approximately $5.1 \times 10^{-5} \times K$ seconds. Therefore, the proposed algorithm can be applied to online and real-time applications.

Evaluation metrics and software: For evaluation, we use well-known metrics which are widely used in MOT evaluation [12], which consists of Recall (correctly tracked objects over total ground truth), Precision (correctly tracked objects over total tracking results), and false positives per frame (FPF). We also report the number of identity switches (IDS) and the number of fragmentations (Frag) of ground truth trajectories. The ratio of tracks with success- fully tracked parts for more than 80% (mostly tracked (MT)), less than 20% (mostly lost (ML)), or less than 80 % and more than 20 % (partially tracked (PT)). The number of ground truth (GT) is reported in Table 2 and 2. We utilize the same evaluation software [2] used in the other previous papers [10, 9, 17, 24, 13] because different evaluation softwares measure performance differently [14].

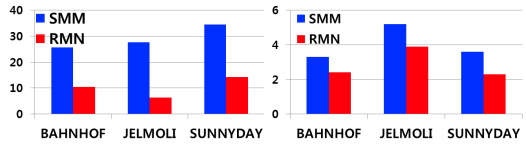
7.1. Comparison with the Baseline Tracker

We evaluate the accuracy of the proposed RMN algorithm against a conventional self motion model (SMM) in terms of distance error on the ETH dataset as shown in Fig. 4. We note that numerous online MOT methods [15, 19, 2] are based on SMM. The errors are computed by the distance between a predicted object position and a ground truth position based on certain criteria. We compute two kinds of mean distance errors. The first one (D_1) is evaluated from error distances that are measured when an object is associated with a detection again after it is not associated with

[2] http://iris.usc.edu/people/yangbo/downloads.html

Dataset	Method	Recall	Precision	FPF	GT	MT	PT	ML	Frag	IDS
ETHZ	RMOT	78.2 %	83.6 %	1.16	164	61.6 %	31.0 %	7.3 %	58	57
Bahnhof, Sunnyday, and Jelmoli	SMOT	73.8 %	75.5 %	1.82	164	51.8 %	39.0 %	9.1 %	142	94
	RMOT	73.9 %	72.6 %	1.58	15	60.0 %	26.7 %	13.3 %	14	3
Marathon1 and Marathon2	SMOT	68.4 %	68.4 %	1.84	15	33.3 %	60.0 %	6.7 %	42	4

Table 1. Comparison with the SMOT (i.e., a baseline tracker) on datasets from a moving camera.



(a) Distance D_1 (b) Distance D_2

Figure 4. (a) Prediction accuracy of RMN and SMM after long-term mis-detections. (b) Prediction accuracy of RMN and SMM when objects are well tracked.

any observation for more than 5 frames as shown in Fig. 1. The second one (D_2) is evaluated from error distances that are measured when the object is well tracked. As shown in Fig. 1 and 4, the prediction based on the RMN is more accurate than that by the SMM, and its efficiency is greater when mis-detections occur.

As shown in Table 1, the RMOT outperforms the SMOT in most of metrics because the RMOT can overcome camera motion problems using the RMN as shown in Fig. 1. Examples of our qualitative results are shown in Fig. 5. When mis-detections occur with camera motions, the self motion model used in the SMOT method becomes unreliable because the SMOT cannot compensate the motion changes caused by the camera without associated observations. Hence, even if the corresponding object is re-detected, the tracker cannot locate the object near the detection response with the prediction based on the SMM due to inaccurate prediction as shown in Fig. 1 and 4.

7.2. Comparison with the State-of-the-art Tracker

Online methods: Table 2 demonstrates quantitative results. The StructMOT and MOT-TBD methods are also designed for online MOT, and they also do not require known camera motion either. However, different from the RMOT and the MOT-TBD, the StructMOT uses the cost function that should be trained in offline manner. Multiple features (i.e., LBP, 3D RGB, HOF), 2D motion information, bounding box, and centroid Euclidean distance are used to train the cost function. Although the RMOT only utilizes RGB histogram and motion information, and does not require any trained cost functions, the RMOT shows the comparable performance in most of metrics. For the sequences from a moving camera (the ETHZ datasets), the RMOT shows better performance in Recall, Precision, MT, ML and Frag because when the long-term mis-detections occur due to occlusions or detection failures, the RMN model helps better predict object states from the other well-tracked objects in data association.

Offline methods: Accroding to the results in Table 2, although the RMOT is an online method, it shows comparable performance in most of metrics in comparison with the offline methods except for the Frag and IDS. The RMOT tends to have more fragments and ID switches compared to those of the OnlineCRF and the PRIMPT. This is natural because our method is an online method which does not uses any future information. Therefore, some short tracks are not fused together leading to a higher number of Frag, and some of tracks sometimes follow same objects causing a higher number of IDS.

8. Conclusion

In this paper, we exploit the motion context from multiple objects, which describes the relative movements between objects to account for camera motions and mis-detections. From the tracked objects, we obtain a set of relative motion and construct the RMN model which in turn helps predict the object states and associate observations for tracking under camera motions with natural fluctuations. For concreteness, we incorporate the RMN model within the Bayesian filtering framework and a data association method for online multi-object tracking. Experimental results on challenging sequences demonstrate that the proposed algorithm achieves favorable and comparable performance over several state-of-the-art methods.

Acknowledgment. This work was partially supported by ICT R&D program of MSIP/IITP [14-824-09-006, Novel Computer Vision and Machine Learning Technology with the Ability to Predict and Forecast], the Center for Integrated Smart Sensors as Global Frontier Project (CISS-2011-0031868), and the IT R&D Program of MKE/KEIT (10040246). M.-H. Yang is supported in part by NSF CAREER Grant #1149783 and NSF IIS Grant #1152576.

References

[1] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *CVPR*, 2012.

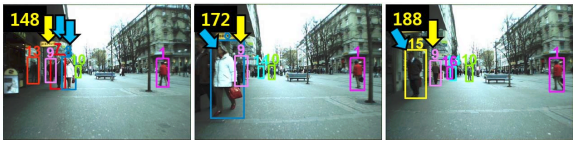
[2] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *PAMI*, 33(9):1820–1833, 2011.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

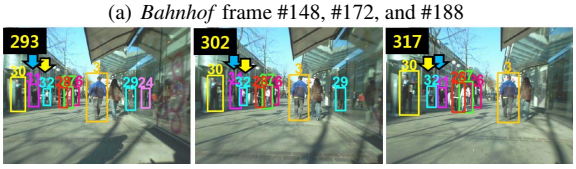
[4] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010.

Dataset	Method	Recall	Precision	FPF	GT	MT	PT	ML	Frag	IDS
PETS-S2L1	RMOT	96.9 %	97.4 %	0.15	19	89.5 %	10.5 %	0.0 %	7	2
	StructMOT [9]	97.2 %	93.7 %	0.38	19	94.7 %	5.3 %	0.0 %	19	4
	PRIMPT [10]	89.5 %	99.6 %	0.02	19	78.9 %	21.1 %	0.0 %	23	1
	CemTracker [13]	-	-	-	19	94.7 %	5.3 %	0.0 %	15	22
TUD-Stadtmitte	RMOT	87.9 %	96.6 %	0.19	10	80.0 %	20.0 %	0.0 %	7	6
	StructMOT [9]	83.3 %	95.4 %	0.25	10	80.0 %	20.0 %	0.0 %	11	0
	PRIMPT [10]	81.0 %	99.5 %	0.028	10	60.0 %	30.0 %	10.0 %	0	1
	OnlineCRF [24]	87.0 %	96.7 %	0.18	10	70.0 %	30.0 %	0.0 %	1	0
	CemTracker [13]	-	-	-	10	40.0 %	60.0 %	0.0 %	13	15
ETHZ Bahnhof and Sunnyday	RMOT	81.5 %	86.3 %	0.98	124	67.7 %	27.4 %	4.8 %	38	40
	StructMOT [9]	78.4 %	84.1 %	0.98	124	62.7 %	29.6 %	7.7 %	72	5
	MOT-TBD [17]	78.7 %	85.5 %	-	125	62.4 %	29.6 %	8.0 %	69	45
	PRIMPT [10]	76.8 %	86.6 %	0.89	125	58.4 %	33.6 %	8.0 %	23	11
	OnlineCRF [24]	79.0 %	85.0 %	0.64	125	68.0 %	24.8 %	7.2 %	19	11
	CemTracker [13]	77.3 %	87.2 %	-	124	66.4 %	25.4 %	8.2 %	69	57

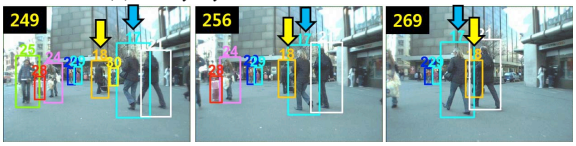
Table 2. Comparison with the state-of-the-art trackers. The trackers in gray color are an offline method. The evaluation results of the RMOT is obtained by using the detection results and ground truth which are the same as used in the state-of-the-art trackers.



(a) *Bahnhof* frame #148, #172, and #188



(b) *Sunnyday* frame #293, #302 and #317



(c) *Jelmoli* frame #249, #256, and #269

(d) *Marathon1* frame #3, #13, and #22

(e) *Marathon2* frame #3, #9, and #21

Figure 5. Example of our results. As shown in (a), (b), (c), and (e), the objects pointed by the yellow arrow are occluded by the objects with the blue arrow. Due to occlusions, these objects are missing for a while, but after occlusion, the objects are correctly re-tracked using the RMN. As shown in (d), The object with the yellow arrow is blurred due to motion fluctuations, but it is tracked robustly with the RMN in spite of unreliable self motion model and severe appearance changes. crossing.

[5] G. Duan, H. Ai, S. Cao, and S. Lao. Group tracking: exploring mutual relations for multiple object tracking. In *ECCV*, 2012.

[6] A. Ess, B. Leibe, K. Schindler, , and L. V. Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008.

[7] H. Grabner, J. Matas, L. J. V. Gool, and P. C. Cattin. Tracking the invisible: Learning where the object might be. In *CVPR*, 2010.

[8] J. F. Henriques, R. Caseiro, and J. Batista. Globally optimal solution to multi-object tracking with merged measurements. In *ICCV*, 2011.

[9] S. Kim, S. Kwak, J. Feyereisl, and B. Han. Online multi-target tracking by large margin structured learning. In *ACCV*, 2012.

[10] C.-H. Kuo and R. Nevatia. How does person identity recognition help multi-person tracking? In *CVPR*, 2011.

[11] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *CVPR*, 2014.

[12] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybrid-boosted multi-target tracker for crowded scene. In *CVPR*, 2009.

[13] A. Milan, S. Roth, and K. Schindler. Continuous energy minimization for multitarget tracking. *PAMI*, 36:58–72, 2014.

[14] A. Milan, K. Schindler, and S. Roth. Challenges of ground truth evaluation of multi-target tracking. In *CVPRW*, 2013.

[15] K. Okuma, A. Taleghani, N. D. Freitas, O. D. Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.

[16] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.

[17] F. Poiesi, R. Mazzon, and A. Cavallaro. Multi-target tracking on confidence maps: An applicaion to people tracking. *CVIU*, 117(10):1257–1272, 2013.

[18] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof. In *CVPR*.

[19] X. Song, J. Cui, H. Zha, and H. Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In *ECCV*, 2008.

[20] V. Takala and M. Pietikäinen. Multi-object tracking using color, texture and motion. In *CVPR*, 2007.

[21] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2), 2004.

[22] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[23] B. Yang and R. Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *CVPR*, 2012.

[24] B. Yang and R. Nevatia. An online learned crf model for multi-target tracking. In *CVPR*, 2012.

[25] L. Zhang and L. van der Maaten. Structure preserving object tracking. In *CVPR*, 2013.