IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 20, 2020, accepted April 9, 2020, date of publication May 11, 2020, date of current version May 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994000

Online Multi-Object Tracking With Visual and Radar Features

SEUNG-HWAN BAE, (Member, IEEE)
Department of Computer Engineering, Inha University, Incheon 22212, South Korea
e-mail: shbae@inha.ac.kr

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant NRF-2018R1C1B6003785, and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) (Artificial Intelligence Convergence Research Center (Inha University)) under Grant 2020-0-01389.

ABSTRACT Multi-object tracking (MOT) constructs multiple object trajectories by associating detections between consecutive frames while maintaining object identities. In many autonomous systems equipped with a camera and a radar, an amplitude and visual features can be measured. Therefore, our goal is to solve a MOT problem by associating detections with both features. To achieve it, we propose a unified MOT framework based on object model learning and confidence-based association. For improving discriminability between different objects, we present a method to learn several visual and amplitude object models during online tracking. By applying the learned object models for the affinity evaluation, we improve the confidence-based association further. In addition, we present a practical track management method to initialize and terminate tracks, and eliminate duplicated false tracks. We implement several MOT systems with different object model learning and association methods, and compare our system with them on challenging visual MOT datasets. We further compare our method with the recent deep appearance learning methods. These comparisons verify that our method can achieve the competitive tracking accuracy while maintaining a low MOT complexity.

INDEX TERMS Object tracking, sensor fusion, visual/amplitude features, object model learning, affinity evaluation, confidence-based data association, surveillance system.

I. INTRODUCTION

Multi-object tracking (MOT) is to find states (*i.e.* positions, velocities, or sizes) of multiple objects in consecutive frames (*or* scans) while conserving their identifications. Over the past decades, it has been extensively studied in autonomous, robot, and computer vision research areas since it is used as a core algorithm to understand and predict behaviors of dynamic objects. However, it is still a difficult problem due to inaccurate detections, abrupt changes of object motion or appearance, and frequent occlusion by clutter or other objects.

To resolve this problem, a tracking-by-detection approach has been flourished. Given object detections (*or* measurements) from a radar and a camera, it builds trajectories by linking detections between consecutive frames. Therefore, automated tracking can be achieved by initializing and terminating tracks with provided detections. In addition, tracking accuracy can be improved because it can recover track fragments and identity switches by matching tracks with the corresponding detections.

In tracking-by-detection, a data association between tracks and detections is crucial, and a lot of methods have been developed. Greedy-based association methods such as the nearest neighborhood [1] and the strongest neighbor [2] show the high speed, but reduce the accuracy often when many matching combinations exist. Joint probabilistic data association (JPDA) [1] and multiple hypothesis tracking (MHT) [3] can determine the optimal assignments between tracks and detections during single and multiple frames, respectively. However, they increase the association complexity combinatorially as the number of possible assignments between tracks and detections increases linearly. To reduce the complexity of JPDA, [4] leverage the m-best solutions of an integer programming. Also, [5] show the classical MHT method using online appearance learning can be comparable to recent MOT methods.

However, in recent years, many autonomous systems (*e.g.* vehicles, mobile robots, and unnamed aerial vehicles) use a camera and a radar together for more accurate and stable

The associate editor coordinating the review of this manuscript and approving it for publication was Ivan Lee.

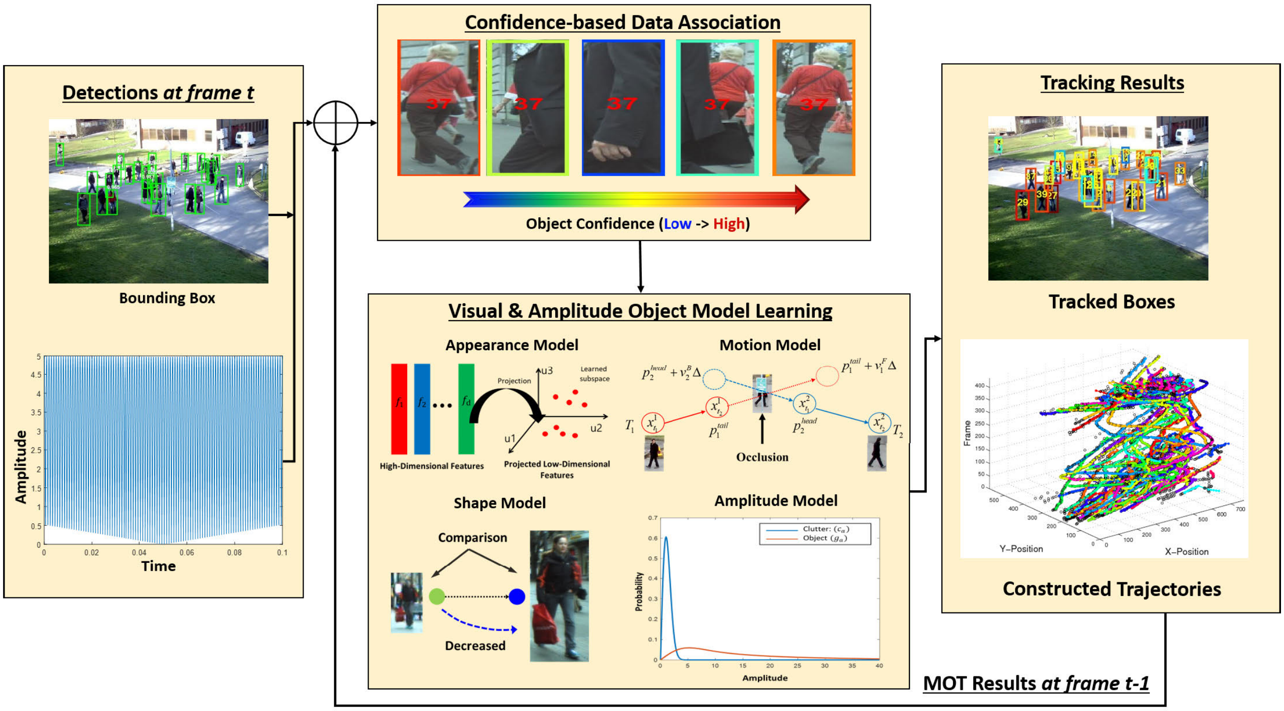


FIGURE 1. The overall framework of our approach for tracking objects with visual and amplitude features. When detection bounding boxes and amplitudes are provided, multiple objects are tracked with the leaned object models at previous frame and confidence-based data association. Then, visual/amplitude object models and object trajectories are updated in online by association results. Updated models and trajectories are used as inputs of the subsequent frame.

object detection and tracking. In many practical scenarios, combining different types of features can improve the accuracy and robustness of MOT since sensors are complementary to each other [6]. Therefore, [7] design object dynamic and measurement models based on EKF to fuse radar, image, and ego vehicle odometry measurements. Provided that a scene geometry, [8], [9] present a method to align camera and radar features on global cartesian coordinates, and use the aligned features for object detection. Reference [10] present an overall system for detecting and tracking moving objects by combining different measurements from radars, Lidars, and a camera. Reference [11] model measurements of a radar and a stereo camera in polar coordinates as a member of Lie Groups and perform object state filtering on Lie groups. Most of them have focused on fusing heterogeneous features effectively by developing object dynamic and measurement models [7], [10], [11] or sensor alignments [8], [9]. Then, they use the aligned or fused features for improving the estimation of the object state [7], [8], [10], [11].

Similar to aforementioned other works, we also leverage visual and radar features for more robust MOT. Compared to those works [7]–[11], our work, however, is more focused on improving MOT accuracy and speed by improving the data association. Because the core of the data association is the affinity evaluation, we propose effective object affinity models and an accurate affinity evaluation measure. In other word, our work aims at learning the various object models efficiently with the visual and amplitude features, and modeling the affinity measure to make the learned models applicable for the data association. As a result, we can improve the online MOT accuracy while maintain a low tracking complexity.

To this end, we propose an overall MOT system which can exploit both features effectively as shown in Fig. 1. The proposed system is based on object model learning and confidence-based data association. We first evaluate confidence scores of tracks, and then categorize them into tracks with low confidence and tracks with high confidence. For the tracks with high confidence, we perform a local association to associate them with detections at a current frame. As a result, we can sequentially grow tracks with online provided detections using this frame-by-frame association. On the other hand, we regard tracks with low confidence as fragmented ones, and perform a global association between tracks with low confidence and other tracks with high confidence or detections. From this global association, we can build long trajectories under occlusions.

For reliable association, accurate affinity (*or* likelihood) evaluation between tracks and detections is essential. For a track and a detection from the same object, their affinity score should be high. Otherwise, the affinity should be low. From visual features, we learn object appearance, motion and shape models during tracking, whereas learn an amplitude model from a radar feature. Using the learned object models, we can evaluate affinity scores more accurately although many tracks and detections exist, and use the evaluated scores for the confidence-based association. For automated MOT, it is usually required to initialize tracks using detections and terminate tracks according to their status. In addition, in many cases, duplicated tracks which follow the same object are generated. To handle these issues, we also present an effective track management method.

On the challenging visual surveillance benchmark datasets for MOT, we thoroughly evaluate our methods in terms of a standard evaluation metric in radar-based MOT. In particular, we implement different versions of MOT systems with different object models and association methods, and compare their MOT performance under several clutter densities. In addition, we compare our method with the state-of-the-art MOT methods using deep learning. In this evaluation, we compare these methods using the common evaluation metrics in vision-based MOT. From these comparisons, we prove the benefits of our methods on several datasets.

The key contribution of this paper can be summarized as follows:

- A unified MOT framework which can leverage visual and radar features effectively.
- Presenting a variety of visual and amplitude object models to learn object models more accurately.
- Enhancing the confidence-based association by applying the several object models for affinity evaluation.
- Extensive implementation and evaluation for various MOT systems on the challenging visual MOT datasets.
- Achieving the state-of-the-art performance comparable with recent deep learning methods while remaining a low tracking complexity.

II. RELATED WORK

In this section, we discuss previous study on radar-based and vision-based MOT.

A radar usually provides a spatial detection (*or* measurement) including a range and bearing. In many practical cases, origins of detections are unknown because a returned signal of a radar is mixed by objects and clutters. Therefore, many data association methods have been developed in order to assign a measurement to a corresponding track. Simple greedy association methods such as the nearest neighborhood [1] and the strongest neighbor [2] association are presented. Although these methods have a low association complexity, incorrect associations occur when tracks are spatially located close together. For handling this joint track-to-measurement assignment problem within a single-frame or a multi-frame search, joint probabilistic data association (JPDA) [1] and multiple hypothesis tracking (MHT) [3] methods are proposed. For reducing the joint association complexity, linear multi-target integrated probabilistic data association (LMIPDA) [12] is also developed. For handling nonlinear dynamics of multiple objects, sequential Monte Carlo (SMC) methods [13]–[16] for MOT are developed. To estimate object states and cardinality simultaneously, joint probabilistic probability densities of multiple objects are modeled in [13], [14]. However, the computational complexity of these methods increases exponentially as the number of hypotheses increases. To alleviate this problem, the data association and state estimation are treated as a separated problem in [15], [16].

However, the spatial feature is not sufficient for the association cases where objects are closely spaced or clutter is densely distributed in the object vicinity. Therefore, for more accurate association, an amplitude is used as an extra feature in [17]–[21]. The basic idea of these methods is that an amplitude from an object is usually stronger than it from a clutter. The extended MHT [17] and Viterbi data association [18] using the amplitude are provided. In order to exploit the amplitude without the pre-knowledge of signal-to-noise ratios (SNRs), a marginalization method [19] which computes an object amplitude likelihood within any SNR boundary is presented. For estimating objects' states and SNR jointly, SMC-based [20] and MAP-based SNR estimation [21] methods are proposed.

In vision-based MOT, tracking by detection methods have flourished for achieving automated and robust MOT. In general, they builds trajectories by associating (*or* linking) detections. They can be divided into batch and online tracking methods according to the association manner. Batch tracking methods [5], [22]–[24] usually build trajectories by using a global association of detections of whole frames. They produce better MOT results than online methods in most cases. However, they cannot be applied for real-time or casual systems because they construct a batch of detections beforehand, and build trajectories by linking whole detections by an iterative global association. On the other hand, online tracking methods [25]–[30] build trajectories by using a frame-by-frame association of past and current detections. Therefore, they can be suitable for real-time applications. However, they tend to yield identity switches and track fragments by long-term occlusions since detections of future frames are not used.

Because both tracking methods build trajectories by local or global associations, an affinity evaluation between tracks and detections is important for the accurate association. To this end, object affinity models using object appearance, motion, and shape cues [5], [25]–[28] are also developed. Due to the recent advances of deep learning, deep learning-based affinity models [28], [31]–[33] have been presented. References [31], [34] use an autoencoder and a convolutional neural network (CNN) as deep appearance models for learning more rich representation. References [28], [35] exploit the Siamese network [36] to calculate the affinity between an object pair from the network output directly. References [32], [33] learn temporal dynamics of tracked objects using a recurrent neural network and CNN. References [24], [30] learns a deep distance metric by aggregating appearance and motion cues. Although the deep learning can improve a model discriminability, many training samples and costly GPUs are required. In this work, we introduce an amplitude affinity model for vision-based MOT, and show that MOT accuracy can be enhanced by using the new and simple amplitude affinity model. We thus argue that the main benefit of our method can improve tracking accuracy while maintaining a low tracking complexity. We also prove this by comparing our method with recent MOT methods using deep learning on the challenging visual MOT datasets.

III. VISUAL AND AMPLITUDE OBJECT MODELS

A. OBJECT DYNAMICS AND MEASUREMENT MODELS

We represent the state of an object i is represented as $\mathbf{x}_t^i = \left[\mathbf{p}_t^i, \mathbf{v}_t^i, \mathbf{s}_t^i, d_t^i\right]^T \in \mathbb{R}^7$, where $\mathbf{p}_t^i = \left[x_{1,t}^i, x_{2,t}^i\right]$, $\mathbf{v}_t^i = \left[x_{3,t}^i, x_{4,t}^i\right]$, $\mathbf{s}_t^i = \left[x_{5,t}^i, x_{6,t}^i\right]$, and $d_t^i = \left[x_{7,t}^i\right]$ are the position, velocity, size, and expected (or mean) SNR. A nonlinear discrete-time dynamic motion is used to model the behavior of an object i as follows:

$$\mathbf{x}_t^{i,m} = f_t(\mathbf{x}_{t-1}^{i,m}) + \mathbf{q}_{t-1}, \quad t = 1, 2, \ldots \tag{1}$$

where $\mathbf{x}_t^{i,m} = \left[\mathbf{p}_t^i, \mathbf{v}_t^i\right]^T \in \mathbb{R}^4$ means dynamic states of an object i at frame t composed of positions and velocities along with x and y coordinate, respectively. f_t is a nonlinear function of the motion state $\mathbf{x}_{t-1}^{i,m}$ and $\mathbf{q}_{t-1} \sim \mathcal{N}(0, Q_t)$ is white Gaussian system noise. The initial states $\mathbf{x}_0^{i,m}$ is assumed to be Gaussian $\mathcal{N}(\mu_0^{i,m}, P_0)$ with the covariance P_0, where $\mu_0^{i,m} = E(\mathbf{x}_0^{i,m})$ and $P_0 = \mathrm{cov}\{\mathbf{x}_0^{i,m}, \mathbf{x}_0^{i,m}\}$.

In general, a detection set obtained at a frame is composed of many detections originated from multiple objects and clutter (*or* background) [1], [37]. Let us denote a set of detections at frame t as $\mathbb{Z}_t = \left\{\mathbf{z}_t^j\right\}_{j=1}^{m_t}$. Each detection $\mathbf{z}_t^j$ from a camera and a radar is represented as $\left[\mathbf{b}_t^j, d_t^j\right]^T$, where $\mathbf{b}_t^j = [b_{t,x}^j, b_{t,y}^j, b_{t,w}^j, b_{t,h}^j]$ are x and y positions, a width, and a height of a detection box obtained from a camera. Also, d_t^j is an amplitude measurement from a radar. Even though a range and bearing features can be detected from a radar, we use an amplitude feature of it only because a camera usually provides more accurate locations and sizes in a real-world environment [6].

Furthermore, an object-originated measurement $\xi_{j,t}^i$ is modeled by a linear measurement model as

$$\xi_{j,t}^i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{x}_t^{i,m} + \begin{bmatrix} w_{x,t} \\ w_{y,t} \end{bmatrix}, \tag{2}$$

where the noise $w_{x,t} \sim \mathcal{N}(0, \sigma_x^2)$ and $w_{y,t} \sim \mathcal{N}(0, \sigma_y^2)$ for localization errors are uncorrelated Gaussian noise sequence. Here, it is assumed that the visual $\mathbf{b}_{j,t}^i$ and amplitude $a_{j,t}^i$ measurements are independent each other.

B. AFFINITY EVALUATION MODELS

We then define a track (*or* trajectory) T^i as a set of states up to frame t as $T^i = \{\mathbf{x}_k^i | v^i(k) = 1, 1 \leq t_s^i \leq k \leq t_e^i \leq t\}$, where t_s^i and t_e^i are the time stamps of the start- and end-frame of the track. If an object i appears at frame t, we denote it by using a binary function as $v^i(t) = 1$. Otherwise, $v^i(t) = 0$.

In addition, we then describe a track T^i with four elements $\{A^i, S^i, M^i, P^i\}$, where A^i, S^i, M^i, and P^i represent appearance, shape, motion, and amplitude models, respectively. Then, an affinity measure to determine how well two objects are matched is defined as

$$\Lambda(u, z) = \Lambda^A(u, z)\, \Lambda^S(u, z)\, \Lambda^M(u, z)\, \Lambda^P(u, z), \tag{3}$$

where u and z can be a track or a detection. Each affinity is computed as follows:

$$\begin{aligned}
\Lambda^A(u, z) &= \max\left(\cos\left(\mathbf{f}_{proj}^u, \mathbf{f}_{proj}^z\right), 0\right), \\
\Lambda^S(u, z) &= \exp\left(-\left\{\frac{h^u - h^z}{h^u + h^z} + \frac{w^u - w^z}{w^u + w^z}\right\}\right), , \\
\Lambda^M(u, z) &= \mathcal{N}\left(\mathbf{p}_{tail}^u + \mathbf{v}_F^u \Theta; \mathbf{p}_{head}^z, O_F\right) \\
&\quad \times \mathcal{N}\left(\mathbf{p}_{head}^z + \mathbf{v}_B^z \Theta; \mathbf{p}_{tail}^u, O_B\right), \\
\Lambda^P(u, z) &= g_a^{DT}(\bar{a}^u | \hat{d}^z) \times g_a^{DT}(\bar{a}^z | \hat{d}^u)
\end{aligned} \tag{4}$$

For the appearance affinity $\Lambda^A(u, z)$, we use the subspace learning using partial least square (PLS) [38]. We first extract an averaged RGB color histogram $\mathbf{f}_{hist}^u$ of each track for Δ_{new}[1] frames. We then project $\mathbf{f}_{hist}^u$ on the learned PLS subspace W^u (*i.e.* $\mathbf{f}_{proj}^u = W^u \mathbf{f}_{hist}^u$) from (14) to produce a compact and discriminative feature $\mathbf{f}_{proj}^u$. The appearance affinity is the cosine similarity between $\mathbf{f}_{proj}^u$ and $\mathbf{f}_{proj}^z$. More details of learning W are given in Sec. III-D.

The shape affinity $\Lambda^S(u, z)$ is calculated with their updated height h and width w. $\Lambda^M(u, z)$ is the motion affinity between u tail (*i.e.* the last refined position) and z head (*i.e.* the first refined position) with the frame gap Θ. The forward velocity $\mathbf{v}_F^u$ is evaluated from the head to the tail of u, while the backward velocity $\mathbf{v}_B^z$ is evaluated from the tail to the head of z. We use the Kalman filtering for updating the velocities. The difference between the predicted position computed with the velocity and the refined position is assumed to follow a Gaussian distribution. The forward motion is used only when evaluating affinity between a track and a detection.

The amplitude affinity $\Lambda^P(u, z)$ is evaluated with the averaged amplitude scores $\bar{a}^u$ and $\bar{a}^z$ for associated amplitude measurements up to a current frame, and their estimated SNRs, $\hat{d}^u$ and $\hat{d}^z$. In the next section, we discuss an amplitude likelihood model g_a^{DT} and a method to estimate $\hat{d}^u$ and $\hat{d}^z$.

C. AMPLITUDE MODEL AND UNKNOWN SNR ESTIMATION

1) OBJECT AMPLITUDE MODEL

We assume the probability density of an amplitude a follows a Rayleigh distribution as discussed in [39]. We then define the expected (or mean) SNR[2] $d = S/N_0$, where S is the signal power and d can be treated as the expected object signal power because $N_0 = 1$. In addition, a slow Rayleigh fading amplitude-modulated narrowband signal is considered in the presence of narrowband noise. In this case, the signal returned from the object is expressed as the sum of

[1] More details can be found in Sec. V-A.

[2] The SNR is represented in log scale: SNR (dB) $= 10\log_{10}(d)$.

the transmitted signal and the narrow band noise. The background noise is normalized as in [39]. This means that the expected noise power N_0 is unity. Therefore, the amplitude density function of an object follows the Rayleigh distribution with the variance $1 + d$ (*i.e.,* the signal-plus-noise to noise ratio):

$$p(a, d) = \frac{2a}{1+d} \cdot \exp\left(\frac{-a^2}{1+d}\right). \tag{5}$$

However, to evaluate the signal power S from the object amplitude distribution (5), the expected object SNR d is required to estimate because

$$S = \mathrm{E}\left[a^2\right] = \int_0^\infty a^2 p(a, d)da. \tag{6}$$

Let us next consider the case in which the amplitude a exceeds a detection threshold DT, *i.e.*, $a \geq DT$. Then, the amplitude density of the object becomes

$$\begin{aligned} p^{DT}(a, d) &= \frac{1}{P_D} p(a, d) \\ &= \frac{2a}{1+d} \cdot \exp\left(\frac{DT^2 - a^2}{1+d}\right), \quad a \geq DT, \end{aligned} \tag{7}$$

where the object detection probability P_D used for normalization is calculated as

$$P_D = \int_{DT}^\infty p_1(a, d)da = \exp\left(\frac{-DT^2}{(1+d)}\right). \tag{8}$$

When the object SNR d is known, the amplitude likelihoods of an object can then be computed as

$$Object : g_a^{DT}(a|d) = p^{DT}(a, d), \tag{9}$$

2) SNR ESTIMATION

To exploit $g_a^{DT}(a|d)$, we estimate object SNR d using the MAP method [21]. We model the prior $p(d)$ with a Gaussian random walk model. In other word, we consider that the SNR is randomly fluctuated in the vicinity of the previously estimated (*or* initial) SNR $\hat{d}^i_{t-1}$. Then, $p(d)$ can be represented with the estimated $\hat{d}_{t-1}$ at frame $t - 1$ and variance σ_d^2 as follows:

$$p(d) = \mathcal{N}\left(d; \hat{d}^i_{t-1}, \sigma_d^2\right), \quad \hat{d}^\tau_k \geq 0. \tag{10}$$

To estimate an unknown SNR more accurately, one can use several amplitude measurements. In other words, rather than inferring the object SNR with an instant amplitude feature a_t^i of the object i at frame t, it can be estimated with a set of amplitude features stacked during Δ frames.

Let us denote the stacked amplitude measurements from time $t - \Delta + 1$ to time t as $a^i_{t-\Delta+1:t}$.[3] The MAP problem of finding an optimal SNR with respect to the collection of amplitudes $a^i_{t-\Delta+1:t}$ can be modeled by

$$\begin{aligned} \hat{d}^i_t &= \underset{d}{\operatorname{argmax}} \prod_{a^i \in a^i_{t-\Delta+1:t}} p\left(a^i, d\right), \quad d \geq 0, \\ &= \underset{d}{\operatorname{argmax}} \prod_{a^i \in a^i_{t-\Delta+1:t}} p\left(a^i | d\right) p(d), \\ &= \underset{d}{\operatorname{argmax}} \sum_{a^i \in a^i_{t-\Delta+1:t}} \log\left(p(a^i|d)\right) + \log\left(p(d)\right), \end{aligned} \tag{11}$$

where the first likelihood term $p(a^i|d)$ is given by (9). By substituting the SNR prior of (11) with (10), the following objective function can be derived:

$$\begin{aligned} \hat{d}^i_t &= \underset{d}{\operatorname{argmax}} \sum_{a^i \in a^i_{t-\Delta+1:t}} \log\left(p(a^i|d)\right) + \log(c), \\ c &= \mathcal{N}\left(d; \hat{d}^i_{t-1}, \sigma_d^2\right). \end{aligned} \tag{12}$$

We solve this nonlinear least-squares problem using the Levenberg-Marquardt method [40].

D. ONLINE APPEARANCE LEARNING

1) SAMPLE GENERATION

Given an associated detection box $b^i = \left[b_x, b_y, b_w, b_h\right]$ for an object i, we can generate some positive sample boxes by rescaling b^i with a scaling factor ψ. We denote a rescaled box as $\mathbf{b}^i_{res} = [b_x, b_y, b_w \cdot \psi, b_h \cdot \psi]$. We initially set to $\psi = 0.7$ and increase ψ with the interval 0.1 until an overlap ratio α_{over} for an intersection region over an union region between $\mathbf{d}^i$ and $\mathbf{d}^i_{res}$ is below to 0.75. We generate a set of positive boxes $B_t^{i,+} = \left\{\mathbf{b}^i, \left\{\mathbf{b}^{i,k}_{res}\right\}_{k=1}^{g^+-1}\right\}$, where $\mathbf{b}^{i,k}_{res}$ has $\alpha_{over} \geq 0.75$ over $\mathbf{b}^i$.

For improving appearance discriminability between an object and other objects nearby or scene clutter, we collect negative sample boxes around an object. Given an object bounding box $\mathbf{b}^i$, we define a negative sample box as $\mathbf{b}^{i,k}_{neg} = \left[b_x + \beta\cos(\omega), b_y + \beta\sin(\omega), b_w/\zeta_w, b_h/\zeta_h\right]$. Here, $\beta = \frac{\rho\sqrt{b_w^2+b_h^2}}{2}$ and $\omega = \frac{2\pi k}{g^-}$. $k \in \{1, \ldots, g^-\}$ is a negative sample index. In our experiment, we set ρ, ζ_w and ζ_h to 1.2, 2 and 4, respectively. As a result, a negative sample set $B_t^{i,-} = \left\{b^{i,k}_{neg}\right\}_{k=1}^{g^-}$ is generated by collecting $b^{i,k}_{neg}$ with different k.

Once box sets $B_t^{i,+}$ and $B_t^{i,-}$ are generated, we collect positive $Z_t^{i,+} = \left\{\left(\mathbf{f}^k_{hist}, +1\right)\right\}_{k=1}^{g^+}$ and negative $Z_t^{i,-} = \left\{\left(\mathbf{f}^k_{hist}, -1\right)\right\}_{k=1}^{g^-}$ sample sets. Here, g^+ and g^- are the number of positive and negative samples. $\mathbf{f}^k_{hist}$ is a color histogram feature with dimension ϱ extracted from positive $B_t^{i,+}$ and negative $B_t^{i,-}$ box sets.

2) PARTIAL LEAST SQUARE (PLS) SUBSPACE LEARNING

To discriminate appearance features of different objects, we learn projection spaces using the PLS since the

[3] To determine a_t^i at frame t, we first filter out measurements using the track gating technique and amplitude thresholding. We then select the amplitude with the maximum strength among filtered measurements, and consider it as a_t^i. More details can be founded in [21].

appearance learning using PLS shows the more discriminability than PCA and color histogram features [38]. We denote a sample set of the i-th track collected from $t-\Delta+1$ to t frames as $Z^i_{t-\Delta+1:t}$, where $Z^i_{t-\Delta+1:t}$ consists of $Z^{i,+}_{t-\Delta+1:t}$ and $Z^{i,-}_{t-\Delta+1:t}$ as defined in Sec. III-D1. Using the NIPALS algorithm, we learn a new PLS weight vector $\mathbf{w}$ with dimension ϱ at each iteration as follows:

$$\mathbf{w}=\frac{F^T\mathbf{e}}{\mathbf{e}^T\mathbf{e}},\quad \mathbf{w}=\frac{\mathbf{w}}{\|\mathbf{w}\|},$$
$$\mathbf{r}=F\mathbf{w},\quad p=\frac{\mathbf{o}^T\mathbf{r}}{\mathbf{r}^T\mathbf{r}},\quad \mathbf{e}=\frac{\mathbf{o}p}{\sqrt{p^Tp}},\tag{13}$$

where $F=\{\mathbf{f}^1_{hist},\mathbf{f}^2_{hist},\ldots,\mathbf{f}^g_{hist}\}$ is the appearance feature matrix with dimension $g\times\varrho$ consisting of g histogram features with dimension ϱ for $Z^i_{t-\Delta+1|t}$. $\mathbf{r}$, $\mathbf{o}$ and $\mathbf{e}$ are g-dimensional feature score, label, and label score vectors, respectively. p is a label loading value. By learning $\mathbf{w}$ for τ iterations, we can produce a PLS weight matrix $W=\{\mathbf{w}^1,\mathbf{w}^2,\ldots,\mathbf{w}^\tau\}^T$.

Then, a weight matrix W^i for the i-th object can be learned with $Z^i_{t-\Delta+1|t}$ using (13). For updating W^i during tracking, we first generate a W^i_{new} with $Z^i_{t-\Delta+1|t}$, and combine W^i_{new} with the learned W^i and balancing weight $\upsilon=0.5$:

$$W^i\longleftarrow\upsilon W^i_{new}+(1-\upsilon)W^i,\tag{14}$$

Once W^i is learned, we can generate a projected PLS feature $\mathbf{f}^i_{proj}=W^i\mathbf{f}^i_{hist}$ and use $\mathbf{f}^i_{proj}$ for affinity evaluation in (4). In our case, we set ϱ and τ to 144 and 40. This means that tracking speed can be improved because the dimension of $\mathbf{f}^i_{proj}$ is much more than the dimension of the original feature $\mathbf{f}^i_{hist}$.

IV. DATA ASSOCIATION

We define T^i in Sec. III-B. Then, a set of trajectories of all objects up to frame t can be denoted as $\mathbb{T}_{1:t}$. We denote a set of trajectories existing at frame t as $\left\{T^i\right\}^N_{i=1}$. Using the confidence measure [28], we then evaluate a track confidence in consideration of the length and continuity of a track and the affinity with an associated detection as follows:

$$conf\left(T^i\right)\models\left(\frac{1}{L}\sum_{k\in[t^i_s,t^i_e],v^i(k)=1}\Lambda\left(T^i,\mathbf{z}^i_k\right)\right)\times\left(1-\exp^{-\beta\cdot\sqrt{(L-w)}}\right),\tag{15}$$

where L is the length of a track χ^i as $L=\left|T^i\right|$, and w is the number of frames in which the object i is missing due to occlusion by other objects or unreliable detection as $\lambda=t^i_e-t^i_s+1-L$. β is a control parameter relying on the performance of a detector. When a detector shows high accuracy, β should be set to a large value (β is set to 1.2 as done in [28]). The average affinity $\Lambda\left(T^i,\mathbf{z}^i_k\right)$ between the track and detection is computed by (3).

Once the confidence scores of tracks are computed by (15), local and global association are adaptively performed according to track confidence. A track with high confidence $T^{i(hi)}$ is considered as a reliable track, and is locally associated with a detection in order to grow it progressively. When h track with high confidence and a detection set $Z_t=\{\mathbf{z}^j_t\}^m_{j=1}$ are given at frame t, we compute a local association score matrix S as

$$S=[s_{ij}]_{h\times m},\quad s_{ij}=-\Lambda(T^{i(hi)},\ \mathbf{z}^j_t),\ \mathbf{z}^j_t\in Z_t,\tag{16}$$

where the affinity $\Lambda(T^{i(hi)},\mathbf{z}^j_t)$ is computed by (3). Then, track-detection pairs which maximize the total affinity in $S_{h\times n}$ are determined by using the Hungarian algorithm [41]. When the association cost of a pair is less than a pre-defined threshold, $-\log(\theta)$, $\mathbf{z}^j_t$ is associated with $T^{i(hi)}$. For the track $T^{i(hi)}$ associated with detection $\mathbf{z}^j_t$, states and confidence of the track are updated with the association results as follows:

- The position and the velocity of a track are updated with the associated $\mathbf{z}^j_t$. The size of the object is also updated by averaging the sizes of associated detections of recent past frames.
- $conf(T^i)$ is updated using $\mathbf{z}^j_t$ by (15).

On the other hand, a tracks with low confidence $T^{i(lo)}$ is considered as a fragmented trajectory by occlusions. To link fragmented tracks into one, we associate $T^{i(lo)}$ with $T^{i(hi)}$ or a detection $\mathbf{y}^j_t$ not associated with any $T^{i(hi)}$ in the local association. Assume that there exist η non-associated detections ($\eta\leq m$), and h and l tracks with high and low confidence, respectively. Then, we perform global association by considering following events:

- Event A: $T^{i(lo)}$ is associated with $T^{j(hi)}$,
- Event B: $T^{i(lo)}$ is terminated,
- Event C: $T^{i(lo)}$ is associated with $\mathbf{y}^j_t$.

We then define a global association score matrix G for all the events as follows:

$$G_{(l+\eta)\times(h+l)}=\begin{bmatrix}A_{l\times h} & B_{l\times l}\\ -\theta_{\eta\times h} & C_{\eta\times l}\end{bmatrix},\tag{17}$$

Here, $A=[a_{ij}]$ represents the event A, where $a_{ij}=-\Lambda(T^{i(lo)},T^{j(hi)})$ is the association cost computed by the affinity between them using (3). $B=\operatorname{diag}[b_1,\ldots,b_l]$ models the event B, where $b_i=-(1-conf(T^{i(lo)}))$ is the cost to terminate $T^{i(lo)}$, and $C=\left[c_{ij}\right]$ represents the event C, where $c_{ij}=-\Lambda(T^{i(lo)},\mathbf{y}^j_t)$ is the association cost computed by (3). A threshold θ is used to select reliable association pairs having high affinity scores.

Once G matrices are computed, we determine optimal matching pairs using the Hungarian algorithm such that the total affinity score in the matrix is maximized. Then, detections of the associated pairs are linked each other in a sequential manner, and confidences of all existing tracks are updated by (15).

V. TRACK MANAGEMENT AND UPDATE

For achieving automated MOT, managing tracks appropriately is also important. In this section, we briefly discuss some tasks which are contained in the track management.

In general, a track initialization is required to generate a new track with detection responses. Once a track is generated, it tracks an object. However, a track could not often follow an non-object (*e.g.* clutter) due to occlusions and inaccurate detections. In this case, we need to eliminate this false track to correct the tracking failure. In some cases, track duplication, which more than two tracks follow a same object, can be occurred by inaccurate track initialization and tracking failures. In the next section, we provide our track initialization, termination, and merging method to deal with those difficulties.

A. TRACK INITIALIZATION AND TERMINATION

The problem of initiating a new track can be transformed as a problem to find consecutive and similar detection responses during a certain Δ_{new} frames. In general, detections of a new track should not be associated with any existing tracks in the local and global association stages. We define a set of non-associated detections from $t-\Delta_{new}+1$ to t as $\mathbf{Y}_{t-\Delta_{new}+1:t}$. It means that the candidates for new tracks are reduced from $\mathbf{Z}_{t-\Delta_{new}+1:t}$ to $\mathbf{Y}_{t-\Delta new+1:t}$, where $\mathbf{Y}_{t-\Delta new+1:t} \subseteq \mathbf{Z}_{t-\Delta_{new}+1:t}$. We define a new track $T^{new} = \{\mathbf{y}_k^{new} | t-\Delta_{new}+1 \leq k \leq t\}$, where $t-\Delta_{new}+1$ and t are the time stamps of the start- and end-frame of the new trajectory, and $\mathbf{y}_t^{new} = [\mathbf{b}_t, a_t]^T$. Now, we define an affinity score for the new track initialization $\Lambda^N(T^{new})$ as follows:

$$\Lambda^N(T^{new}) \models \frac{1}{\Delta_{new}-1} \sum_{k=t-\Delta_{new}+2}^{t} \Lambda^S(\mathbf{y}_k, \mathbf{y}_{k-1}) \times \Lambda^V(\mathbf{y}_k, \mathbf{y}_{k-1}), \; \mathbf{y}_k \in \mathbf{Y}_{t-\Delta_{new}+1:t} \quad (18)$$

where $\Lambda^S(\mathbf{y}_t, \mathbf{y}_{t-1})$ is the shape affinity defined in (4). Also, we evaluate the spatial affinity $\Lambda^N(\mathbf{y}_k, \mathbf{y}_{k-1})$ by evaluating spatial distances along x and y coordinates. $\Lambda^V(\mathbf{y}_k, \mathbf{y}_{k-1}) = \mathbf{N}(\xi_k; \xi_{k-1}, \Sigma_v)$, where $\xi_k = [b_{k,x}, b_{k,y}]$. The covariance Σ_v is then determined by the maximum velocities of an object along with x and y coordinates and the unbiased converted covariance $\mathbf{R}_t^c$ [42], such that

$$\Sigma_p = \begin{bmatrix} \left(V_{x,max}T_s + 2\sqrt{R_t^{11}}\right)^2 & 0 \\ 0 & \left(V_{y,max}T_s + 2\sqrt{R_t^{22}}\right)^2 \end{bmatrix}. \quad (19)$$

Then, we generate a new track when $\Lambda_N(T^{new})$ exceeds the track initialization probability ϑ_I.

The desirable track termination method should identify and eliminate false tracks, *i.e.*, those that do not follow true objects. In our case, we evaluate the reliability of a track using the track confidence model $conf\ (T_i)$, and some tracks which have confidences lower than ϑ_T are eliminated. Using the track initialization and termination methods we can generate new tracks and eliminate false tracks efficiently by considering affinities between detections and the track reliability.

B. TRACK MERGING

In MOT problems, several tracks often follow a same object due to inaccurate track initialization or tracking failures. It is called a track duplication. In [20], we presented a track merging method based on a mean shift algorithm. In brief, we classify and group tracks $\{T^i\}_{i=1}^N$ according to the recent states $\{\hat{\mathbf{x}}_t^i\}_{i=1}^N$. Using the mean shift, the m_c modes of clusters $\{C_q\}_{1,\ldots,m_c}$ are then determined. Once clusters $\{C_q\}_{1,\ldots m_c}$ are generated, track q and its components such as track states $\hat{\mathbf{x}}_t^q$, covariance $\mathbf{P}_{t|t}^q$, track confidence $conf\ (T^i)$, and object models are determined as follows:

- The track state $\hat{\mathbf{x}}_{t|t}^q$ is the mode of the cluster $\{C_q\}_{1,\ldots,m_c}$.
- The covariance $\mathbf{P}_{t|t}^q$ is the min $\left(\mathbf{P}_{t|t}^{\{C_q\}}\right)$.
- The confidence $conf\ (T^q)$ is the max $\left(conf\ \left(T^{\{C_q\}}\right)\right)$.
- The object models $\{A^q, S^q, M^q, P^q\}$ are the models of the track q^*, where $\underset{q^*}{\text{argmax}} \left(conf\ \left(T^{\{C_q\}}\right)\right)$.

VI. EXPERIMENTAL RESULTS

On the challenging visual surveillance datasets, we evaluate our MOT method. For more comparisons, we implement and compare different MOT methods.

A. IMPLEMENTATION

To verify our affinity models using visual/amplitude features and the confidence-based association method, we have implemented and compared several multi-object tracking systems (M1-M4) using different object models and data association methods. For this comparison, based on the Algorithm 1, we have implemented the following MOT systems by combining different methods:

- (M1) without visual models;
- (M2) without an amplitude model;
- (M3) with all models and LMIPDA-AI association [20];
- (M4) with all models and confidence-based association;

Here, the system (M1) only uses the range, bearing, and amplitude features of radars. For affinity evaluation of (M1), we therefore use the object motion Λ^M and amplitude Λ^P models. On the other hand, (M2) do not exploit an amplitude feature, and exploit visual models Λ^A, Λ^S, and Λ^M for affinity evaluation. For (M3) and (M4), we use the all models of a camera and radar, but different association methods are applied for each system. In (M3), we use the LMIPDA-AI association method. In this association, a track existence probability should be computed in order to evaluate the posterior association probability $\beta_{j,t}^i$ between a track i and a measurement j which is within a gate of the track i. For a fair comparison, we replace a track existence probability with a track confidence. In addition, (M3) leverages all the affinity models when evaluating the $\beta_{j,t}^i$. When estimating an object SNR in (M1), (M3), (M4), we set the variance σ_d^2 and Δ to 5 and 5 when solving the object function (12).

For (M3), we use the gating technique to reduce matching combinations between tracks and measurements as done in [12], [20]. Using the gating technique, m_t^i validated

Algorithm 1 The Overall Algorithm for Implementing MOT Systems With Different Association Methods and Affinity Models

```
Input  : A set of measurements: $\mathbb{Z}_t$ and a set of trackers
         $\left\{T^i\right\}_{i=1}^{N_t}$, where each tracker is composed of $T^i =$
         $\{A^i, S^i, M^i, P^i, \mathbf{x}_k^i | v^i(k) = 1, 1 \le t_s^i \le k \le t_e^i \le t\}$.
Output: Updated trackers $\left\{T^i\right\}_{i=1}^{N_t}$
for M1 to M4 do
    // Step1: Select a set of validated measurements $\mathbf{Z}_t^i$;
    if M3 then
        for $i \leftarrow 1$ to $N_t$ do
            $\mathbf{Z}_t^i = \{\mathbf{z}_{j,t}^i : (\mathbf{v}_{j,t}^i)^{\mathrm{T}}(\mathbf{S}_t^i)(\mathbf{v}_{j,t}^i) \le \gamma, a_{j,t}^i \ge DT\}$
            by (20)
        end
    end
    // Step2: Data association;
    // Confidence-based association;
    if M1 or M2 or M4 then
        for $i \leftarrow 1$ to $h$ do
            for $j \leftarrow 1$ to $m$ do
                Generate a local association matrix
                $S = [s_{ij}]_{h \times m}$ by (16)
            end
            Local association by optimizing $S$
        end
        for $i \leftarrow 1$ to $l + \eta$ do
            for $j \leftarrow 1$ to $h + l$ do
                Generate a global association matrix
                $G_{(l+\eta)\times(h+l)}$ by (17)
            end
            Global association by optimizing $G$
        end
    end
    // LMIPDA-AI association;
    if M3 then
        for $i \leftarrow 1$ to $N_t$ do
            for $j \leftarrow 1$ to $m_k^\tau$ do
                Evaluate a posterior association
                probability $\beta_{j,t}^i$ by [20]
            end
        end
    end
    // Step3: Model and confidence update;
    for $i \leftarrow 1$ to $N_t$ do
        if M1 then
            Update $\{M^i, P^i\}$ by (12)
        end
        if M2 then
            Update $\{A^i, S^i, M^i\}$ by (14)
        end
        if M3 or M4 then
            Update $\{A^i, S^i, M^i, P^i\}$ by (12) and (14)
        end
        Update $conf\left(T^i\right)$ by (15)
    end
    // Step4: Track management;
    for $i \leftarrow 1$ to $N_t$ do
        Terminate $T^i$ when $conf\left(T^i\right) \le \vartheta_T$
    end
    Define $\{C_q\}_{1,\ldots m_c}$ and merge $T^{\{C_q\}_{1,\ldots m_c}}$
    Initialize $T^{new}$ when $\Lambda^N(T^{new}) \ge \vartheta_I$
end
```

measurements in the gate of the track i are determined by

$$\mathbb{Z}_t^i = \left\{\mathbf{z}_{j,t}^i : \left(\mathbf{v}_{j,t}^i\right)^T \left(\mathbf{S}_t^i\right)^{-1} \left(\mathbf{v}_{j,t}^i\right) \le \gamma\right\}, \quad i = 1, \ldots, m_t^i, \tag{20}$$

where γ is a gate threshold and m_t^i is the number of measurements in the gate of the track i; $\mathbf{v}_{j,t}^i = \xi_{j,t}^i - \bar{\xi}_{t|t-1}^i$, is a zero-mean Gaussian residual with a covariance $\mathbf{S}_k^i$.

Given the gated measurements, amplitude thresholding is exploited to filter out false alarms with the threshold DT because the amplitude from an object is usually stronger than false alarms [20].

B. EVALUATION METRIC

As a performance measure, the optimal subpattern assignment (OSPA) metric [43] is used. Given the true and estimated sets composed of states of multiple objects, we measure the *localization distance* and *cardinality distance*. The localization distance evaluates the state similarities between matched pairs of the true and estimated sets. On the other hand, the cardinality distance evaluates how well the number of existing tracks matches the number of true objects. As an overall performance measure, the *OSPA distance* representing the total error is calculated by summing both the localization and cardinality distances. For all the distance metrics, a smaller distance indicates better results.

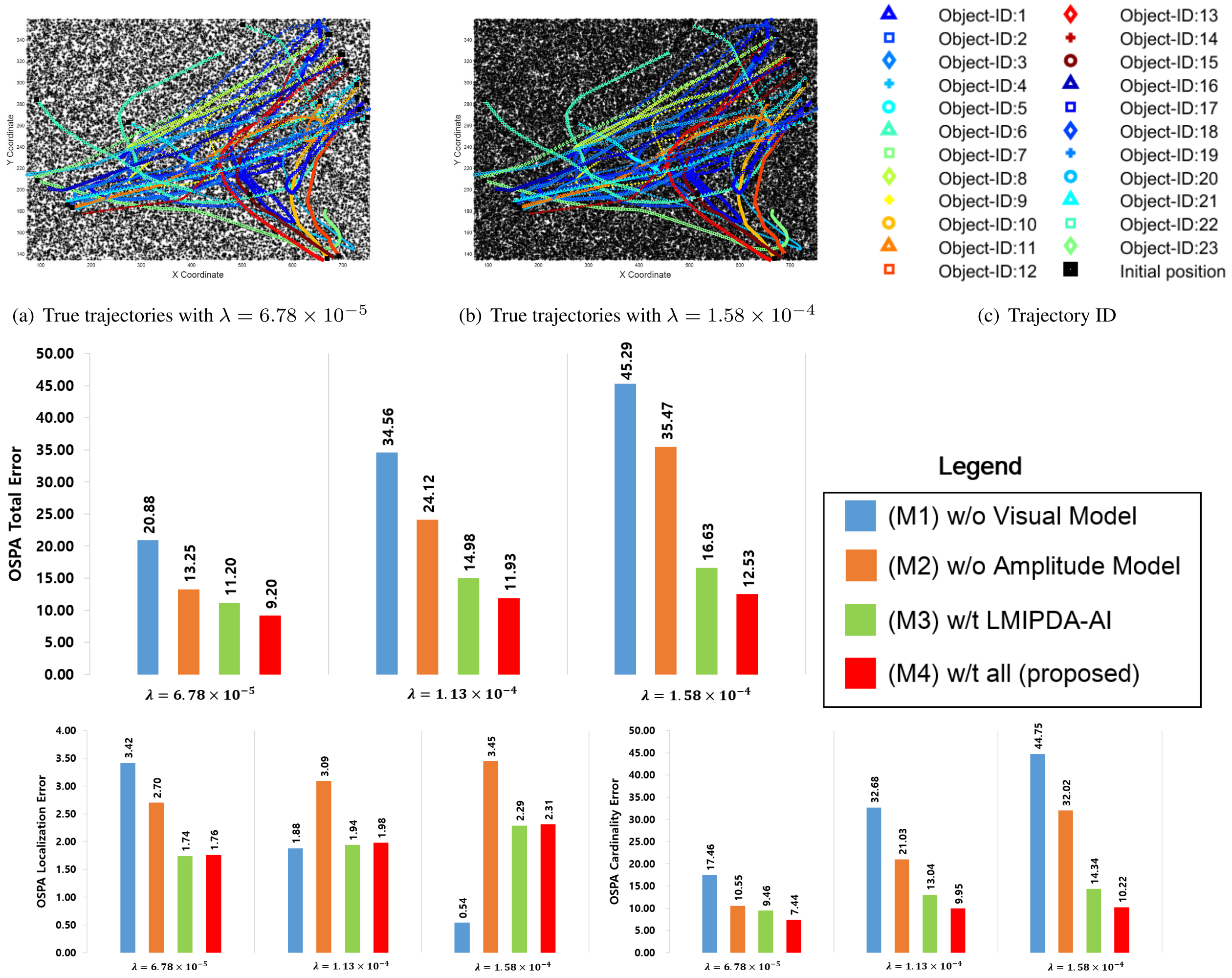
In the OSPA metric, the cut-off parameter is set to $c = 100$, which determines the relative weighting of penalties assigned to the cardinality and localization errors. The order parameter then is set to $p = 1$ which determines the sensitivity of the metric to outliers.

C. VISUAL MULTI-OBJECT TRACKING DATASET

To compare the systems (M1-M4) in real MOT environment, we use the publicly available VS-PETS 2009 benchmark dataset [44]. In the dataset, PETS S2.L1 and PETS S2.L2 sequences for multi-object tracking evaluation are exploited. PETS S2.L1 and S2.L2 sequences consist of 795 and 436 frames and the resolution of each image is 768(pixels) × 576(pixels). 23 and 74 objects exist for PETS S2.L1 and S2.L2, respectively. As shown in Fig 2(a) and Fig. 3(a), the trajectories of multiple objects are complicated. In particular, PETS S2.L2 sequence is very challenging because many objects are moving and interacting with each other. For more evaluation, we compare (M1)-(M4) on the Town Centre dataset. This dataset consists of 4500 frames and each frame is a full HD image of 1920(pixels) × 1080(pixels) resolution. 230 objects are moving and interacting as shown in Fig. 4(a). We allocate each object to an initial SNR within [5dB, 20dB], and the object SNRs fluctuate at each scan according to the Gaussian distribution (10) with the variance $\sigma_d = 10$.

D. DETECTION

For PETS and Town Centre datasets, we use the public available detections from [45] and [46] which exploit the



(a) True trajectories with $\lambda = 6.78 \times 10^{-5}$

(b) True trajectories with $\lambda = 1.58 \times 10^{-4}$

(c) Trajectory ID

(d) Comparison of different methods in terms of OSPA metrics

FIGURE 2. **Comparisons with different MOT systems (M1-M4) for the PETS S2.L1 over 795 frames. In 2(a) and 2(b), $*$ represents false positives from clutter.**

HOG detector [47] and its variant [48], respectively Measurements of objects are assumed to detect with $P_D = 0.95$ and some detections for the objects are removed according to P_D. From each detection, spatial locations (*i.e.* x and y positions) and sizes (*i.e.* width and height) are obtained. For each object SNR at frame t, amplitude measurements are generated according to Rayleigh distribution (6).

For each sequence, we generate more clutters with various clutter density λ (measurements/frame/pixel^2): For PETS S2.L1, $\lambda = 6.78 \times 10^{-5}$, $\lambda = 1.13 \times 10^{-4}$ and $\lambda = 1.58 \times 10^{-4}$; For PETS S2.L2, $\lambda = 4.52 \times 10^{-5}$, $\lambda = 9.04 \times 10^{-5}$ and $\lambda = 1.36 \times 10^{-4}$; For Town centre, $\lambda = 9.65 \times 10^{-6}$, $\lambda = 1.93 \times 10^{-5}$ and $\lambda = 2.89 \times 10^{-5}$. As a result, from 20 to 70 clutters are produced randomly at each frame. In Fig. 2(a) - 2(b), Fig. 3(a) - 3(b), and Fig. 4(a) - 4(b), we represent false detections from clutter with *.

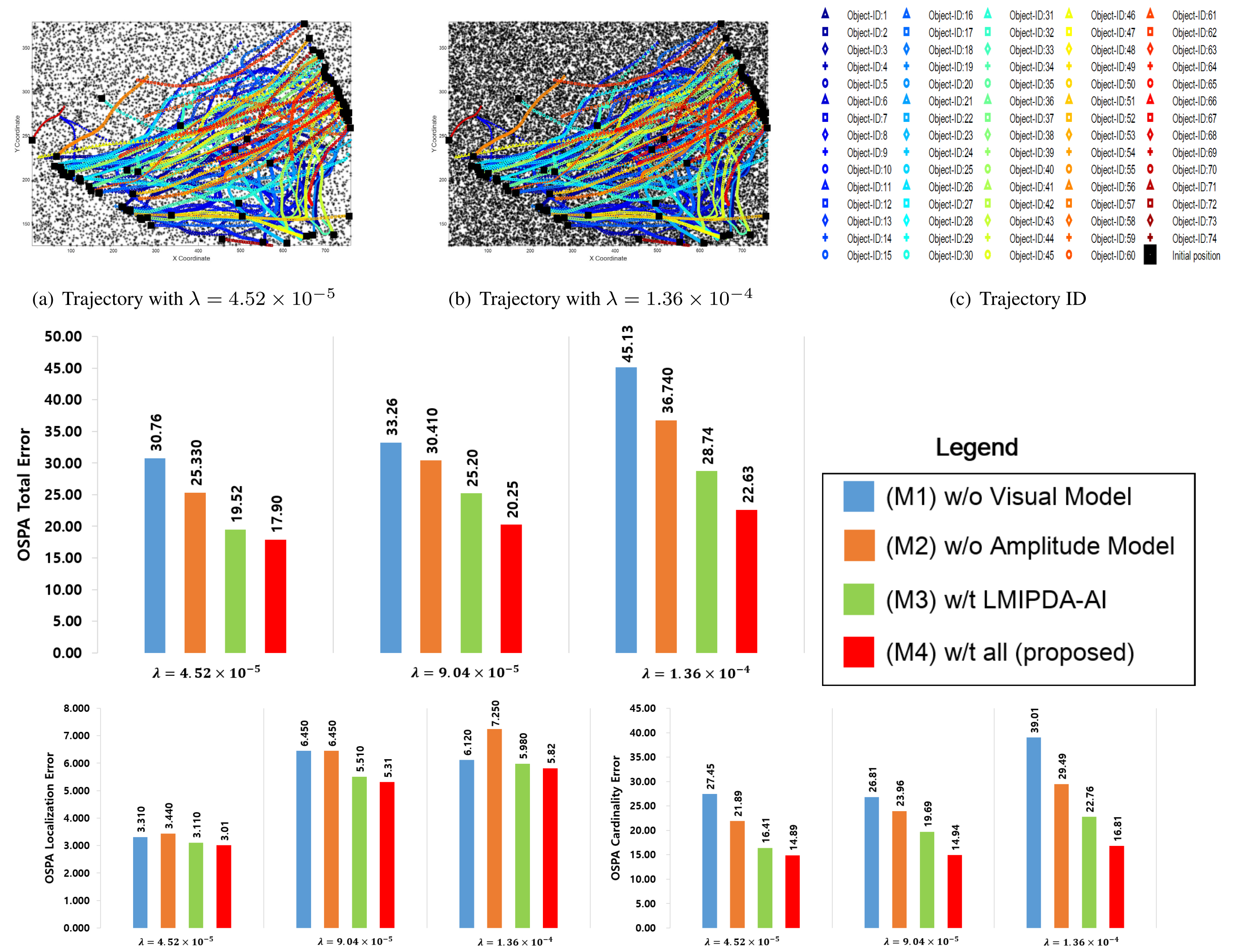
E. TRACKING PARAMETERS

For a fair comparison, all the systems (M1)-(M4) use the same detections and tracking parameters. From an extensive evaluation, we know that most parameters do not affect the overall system performance significantly. In the affinity model in (3), all parameters (*i.e.* positions, sizes and velocities) are automatically determined by tracking results except for O^F and O^B, which are set to diag[16^2 32^2]. The same threshold $\theta = 0.4$ is used for the local and global associations. For M3, we set the thresholds of a gate and amplitude to $\gamma = 15$ and $DT = 0.7$. For initializing a new track, Δ_{new} and ϑ_I is set to 5 and 0.3. The maximum velocities $V_{x,max}$ and $V_{y,max}$ are set to 20 (m/s). For track termination, ϑ_T is 0.05.

F. QUANTITATIVE RESULTS

1) PETS S2.L1 SEQUENCE

In Fig. 2, we compare the MTT systems (M1)-(M4). Figure 2(a) - 2(b) show the true trajectories of all the objects and detections for this sequence. In Fig. 2(d), we show the OSPA total, cardinality, and location error rates for all the systems. The OSPA total errors of systems (M1) and (M2) using visual or amplitude models only are much higher than



(a) Trajectory with $\lambda = 4.52 \times 10^{-5}$

(b) Trajectory with $\lambda = 1.36 \times 10^{-4}$

(c) Trajectory ID

(d) Comparison of different methods in terms of OSPA metrics

FIGURE 3. Comparisons with different MOT systems (M1-M4) for the PETS S2.L2 over 436 frames. In 3(a) and 3(b), $*$ represents false positives from clutter.

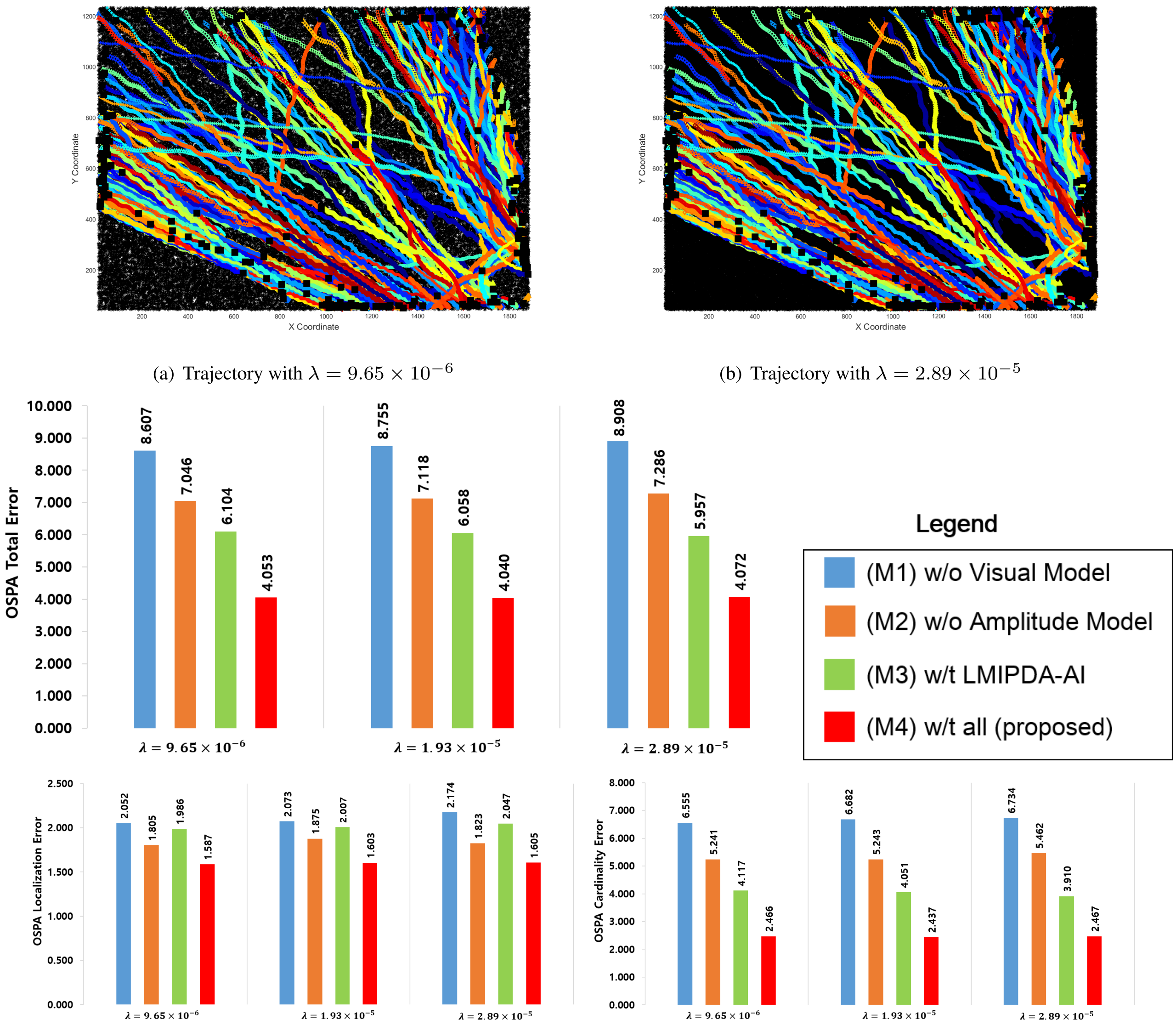
those of (M3)-(M4) using all the models. As clutter density λ gets higher, OSPA distances of (M1) and (M2) increases considerably. This is because that more clutter is likely to be generated nearby objects as λ increases. Therefore, exploiting one of visual and amplitude models only reduces the discriminability of object affinity evaluation. In particular, OSPA errors of (M1) without visual feature is higher than those of (M2) without amplitude feature. The OSPA localization error of (M1) is lower than that of (M2). The reason is that (M1) generates fewer tracks than (M2) as shown in its higher cardinality error.

On the other hand, (M3)-(M4) show the better accuracy than (M1) and (M2), and maintain their performance for high λ. This indicates that using both features can enhance the association accuracy and is more effective in the heavy cluttered environment. When comparing (M3) and (M4) using different association methods, using confidence-based association shows the better results than LMIPDA-AI. This means that adaptive local and global association based on track confidence can determine the association pairs more accurately.

2) PETS S2.L2 SEQUENCE

Figure 3 demonstrates the tracking results of the (M1)-(M4) on the PEST S2.L2 sequence. This sequence is very challenging because of the complex motions of objects and many interactions between many objects. Therefore, the overall performance of all the systems is degraded over their performance on PETS S2.L1. In particular, the localization errors of systems increase due to inaccurate detections and many false detections.

From the OSPA results shown in Fig. 3(d), we also confirm that exploiting both visual and amplitude models indeed is beneficial to reduce OPPA errors when comparing (M1)/(M2) and (M3)/(M4). In addition, (M1) without



(a) Trajectory with $\lambda = 9.65 \times 10^{-6}$

(b) Trajectory with $\lambda = 2.89 \times 10^{-5}$

(c) Comparison of different methods in terms of OSPA metrics

FIGURE 4. Comparisons with different MOT systems (M1-M4) for the Town Centre over 4500 frames. In 4(a) and 4(b), $*$ represents false positives from clutter.

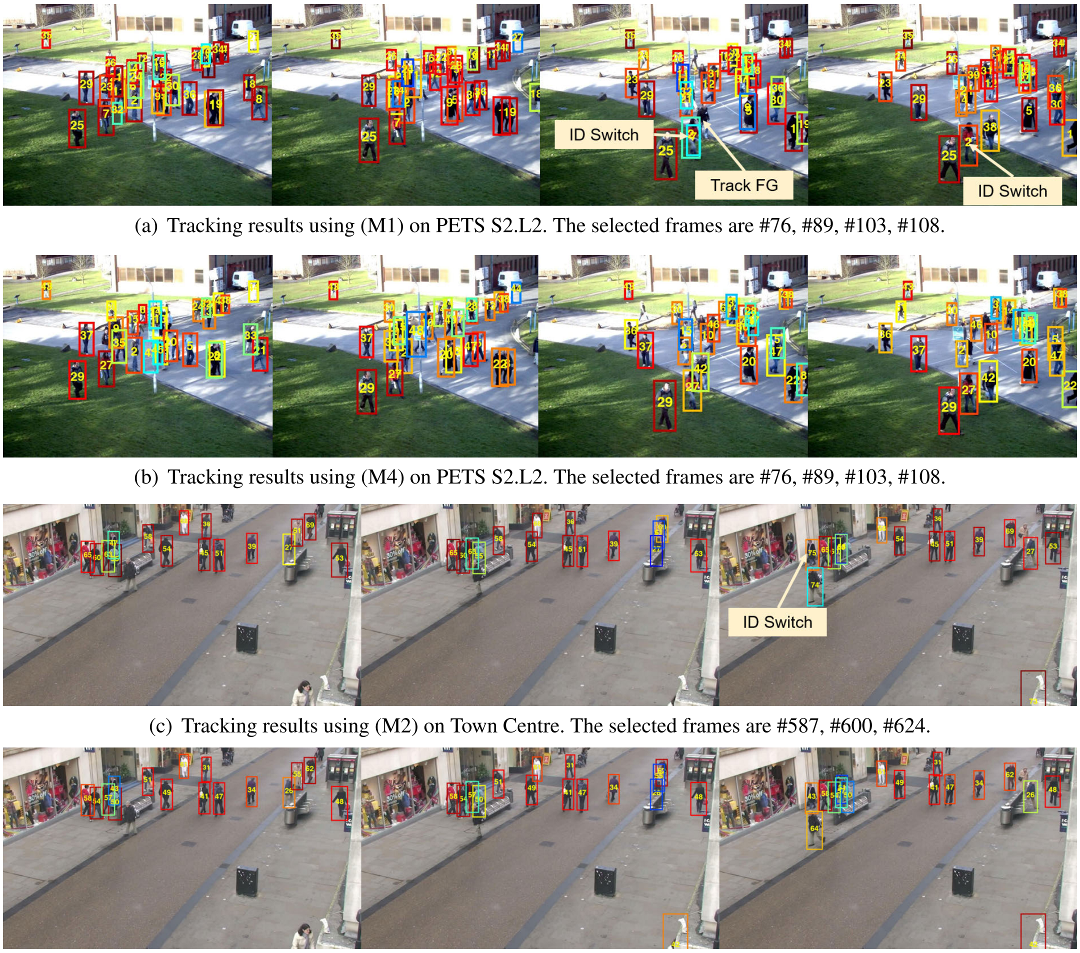
visual feature shows the lowest accuracy. Using the amplitude model reduces OSPA error about 10 when comparing (M2) and (M4). In particular, the effect of using the amplitude model P^i increases as λ increases. In this evaluation, our (M4) achieves the best accuracy. In particular, the cardinality errors of (M4) are not sensitive to λ. The low cardinality errors reflect that the number of generated tracks is close to the number of true objects.

3) TOWN CENTRE SEQUENCE

We further compare (M1)-(M4) on the Town centre sequence as shown in Fig. 4. This sequence is very long and contains many objects. However, the performance of all systems is better than other two sequences. We also obtain the better results by using both visual and amplitude models. In addition, confidence-based association shows the lower OPSA errors than LMIPDA-AI. From the quantitative results on PETS S2.L1, PETS S2.L2, and Town Centre, we prove that our affinity models and association method contribute to increase MOT accuracy, and the performance gain of using our methods gets higher as clutter density increases.

G. QUALITATIVE EVALUATION

In Fig. 5, we compare tracking results of (M1), (M2), and (M4). Figure 5(a) and 5(b) compare (M1) without visual feature and (M4) using both features. We found that some track fragment (FG) and identity switch (ID Switch) are caused by inaccurate association of (M1) when tracked objects are occluded. Furthermore, we show that (M2) without the amplitude model produces an ID switch as shown in Fig. 5(c).



(a) Tracking results using (M1) on PETS S2.L2. The selected frames are #76, #89, #103, #108.

(b) Tracking results using (M4) on PETS S2.L2. The selected frames are #76, #89, #103, #108.

(c) Tracking results using (M2) on Town Centre. The selected frames are #587, #600, #624.

(d) Tracking results using (M4) on Town Centre. The selected frames are #587, #600, #624

FIGURE 5. On PETS S2.L2 and Town Centre, comparison results between (M1)/(M2) and our (M4) systems. They use different object affinity models.

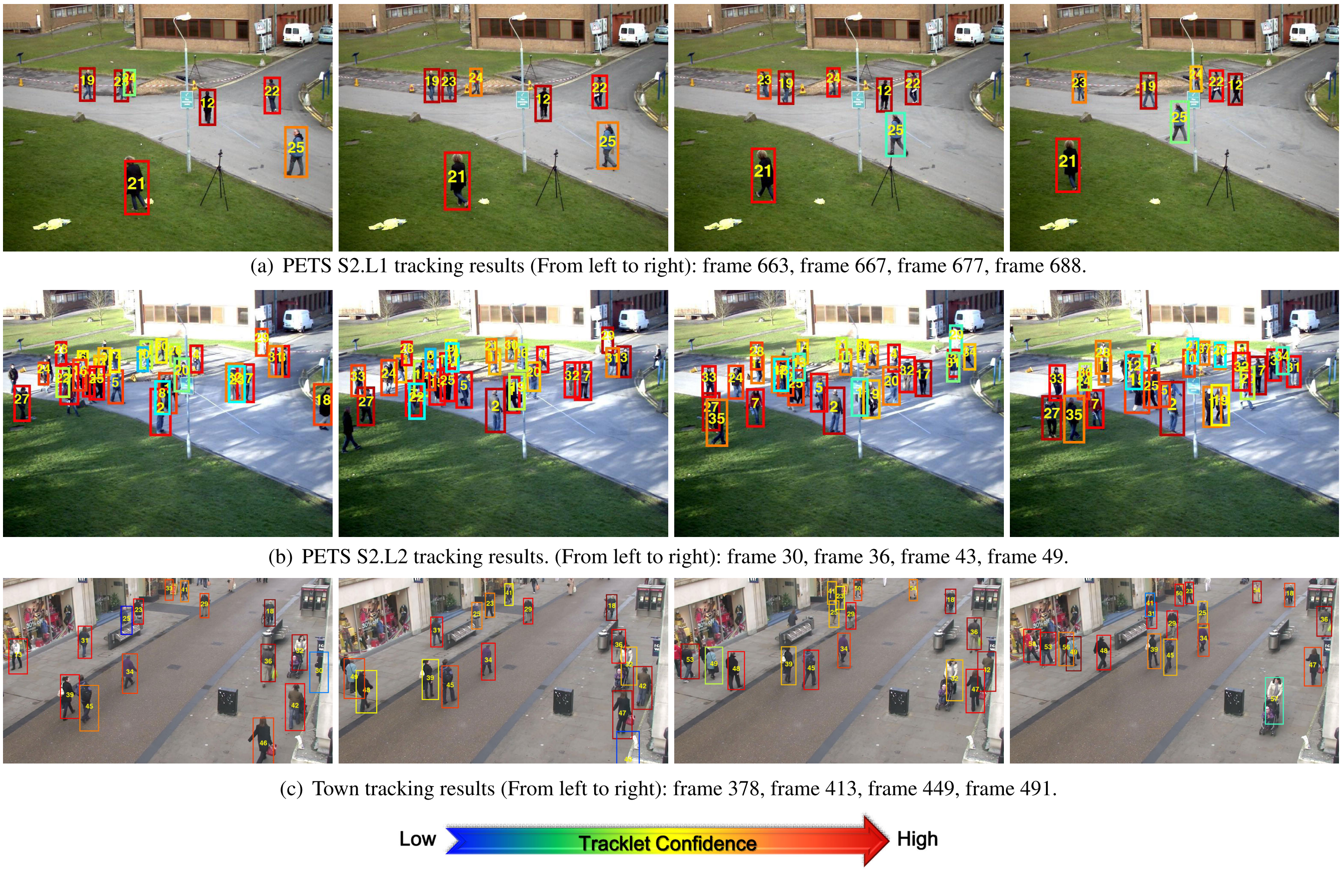
Figure 6 shows the track results of our (M4) on the several datasets. Our (M4) can track multiple objects successfully even in complex scenes.

H. COMPARISON WITH DEEP APPEARANCE LEARNING

To show the benefits and effects of our method more, we compare our method with recent MOT tracking systems using deep appearance learning [28], [31], [36]. For a fair comparison, we implement all other systems on the same framework shown in Fig. 1, and replace the appearance model (4) with their deep appearance models. We use the public available codes for [28], [31], [36], and train deep appearance models on the CUHK02 [49] person re-identification dataset. The dataset contains 7,262 image patches for 1,816 different persons captured from 10 camera views. We resize a color image patch of a person to 128×64, and use the resized patches as an input of deep appearance models. We obtain detection boxes by applying a Mask R-CNN [50] detector for each image. We also generate amplitude measurements with the Rayleigh distribution (6).

In addition, for this comparison we use the common evaluation metrics in vision-based MOT: the multiple object tracking accuracy (MOTA↑), multiple object tracking precision (MOTP↑), the ratio of mostly tracked trajectories (MT↑), the ratio mostly lost trajectories (ML↓), the number of track fragment (FG↓), recall (REC ↑), precision (PRE ↑), false alarms per frame (FAF↓), the number of identity switches (IDS↓) and tracker speed in frames per second (Hz↑). Here, ↑ and ↓ represent that higher and lower scores are better results, respectively.

Table 1 shows the evaluation results on PETS S2.L1, PETS S2.L2, and Town centre datasets. As shown, the proposed methods are comparable with other MOT systems [28], [31], [36] using deep learning. Although the recent method [28] shows the best tracking accuracy, the proposed method with amplitude feature shows the better MOTA, IDG, FG, REC, PRE, FAF scores than other deep learning-based MOT trackers [31], [36]. In addition, we also know that using the amplitude feature can improve the MOTA score, which is the most important metric, by 1.58% when comparing our


(a) PETS S2.L1 tracking results (From left to right): frame 663, frame 667, frame 677, frame 688.

(b) PETS S2.L2 tracking results. (From left to right): frame 30, frame 36, frame 43, frame 49.

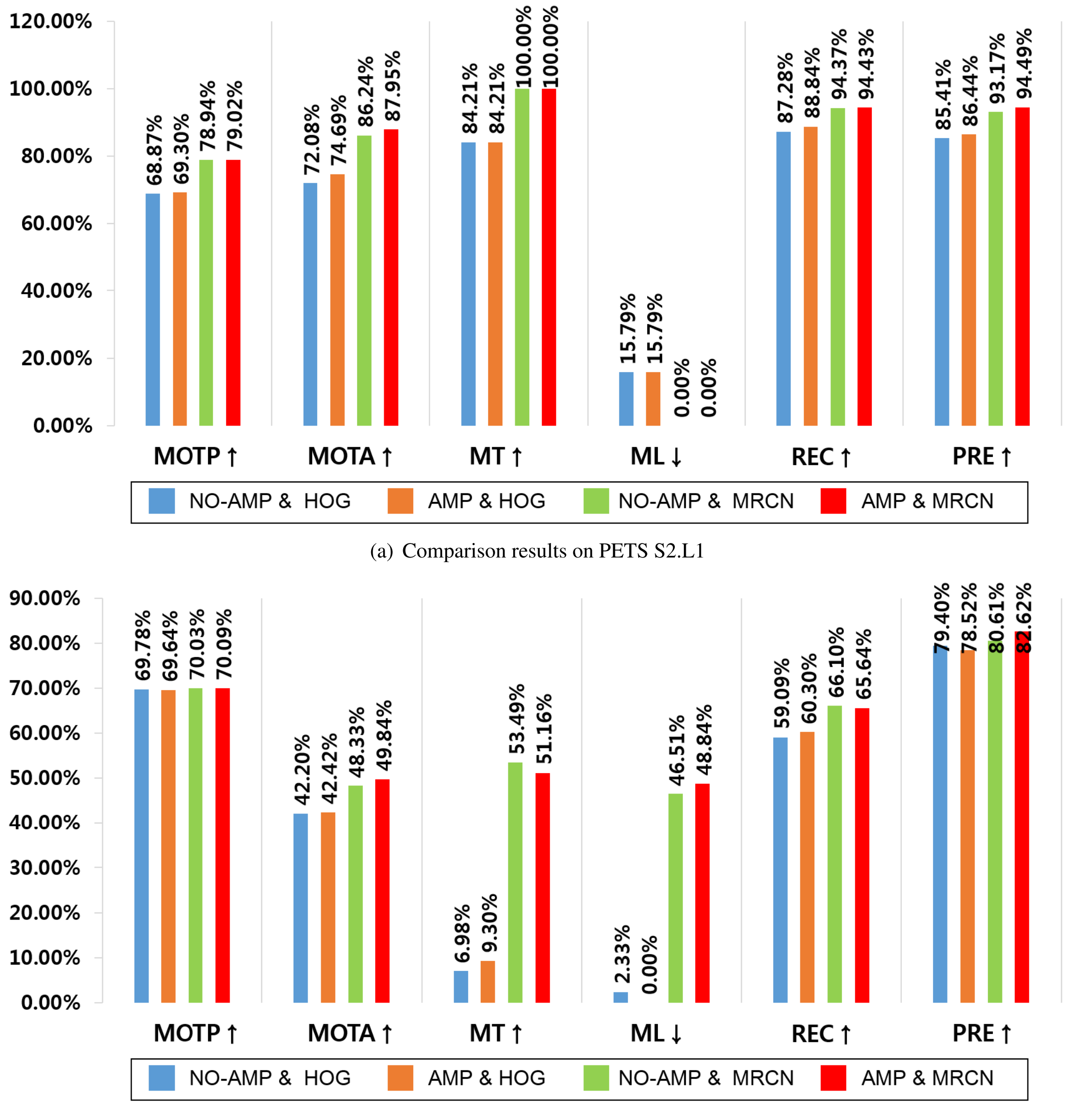
(c) Town tracking results (From left to right): frame 378, frame 413, frame 449, frame 491.

FIGURE 6. On PETS S2.L1, PETS S2.L2, and Town datasets, tracking results of our MOT system. Here, box edge color indicates confidence of a track at each frame.

TABLE 1. Performance comparison with proposed systems and other MOT systems using deep appearance learning. The same detections and ground truth are used on PETS, ETHMS and Town Centre datasets.

Dataset	Method	MOTP ↑	MOTA ↑	GT	MT ↑	PT	ML ↓	IDS ↓	FG ↓	REC ↑	PRE ↑	FAF ↓	Speed (spf)↑
PETS L1	Siamese deep network [36]	78.21%	84.01%	19	100.00%	0.00%	0.00%	18	20	93.41%	91.94%	0.48	0.49
	Deep learning tracker [31]	78.59%	88.17%	19	100.00%	0.00%	0.00%	8	15	94.58%	85.15%	0.33	2.66
	Discriminative Deep Learning [28]	80.54%	90.96%	19	100.00%	0.00%	0.00%	4	8	95.30%	94.66%	0.31	0.68
	Proposed (w/o amplitude)	78.94%	86.24%	19	100.00%	0.00%	0.00%	20	25	94.37%	93.17%	0.40	0.04
	Proposed (w/t amplitude)	79.02%	87.95%	19	100.00%	0.00%	0.00%	19	26	94.43%	94.49%	0.319	0.06
PETS L2	Siamese deep network [36]	69.48%	43.21%	43	53.49%	46.51%	0.00%	216	248	63.85%	77.52%	4.37	1.56
	Deep learning tracker [31]	69.80%	46.13%	43	51.16%	48.84%	0.00%	167	208	63.40%	80.21%	3.69	25.25
	Discriminative Deep Learning [28]	70.27%	50.77%	43	42.86%	57.14%	0.00%	113	173	64.42%	83.69%	2.96	1.81
	Proposed (w/o amplitude)	70.03%	48.33%	43	53.49%	0.00%	46.51%	188	228	66.10%	80.61%	3.75	0.11
	Proposed (w/t amplitude)	70.09%	49.84%	43	51.16%	0.00%	48.84%	206	248	65.64%	82.62%	3.259	0.42
Town Centre	Siamese deep network [36]	70.20%	54.70%	230	46.09%	43.48%	10.43%	311	914	75.10%	79.00%	3.16	2.76
	Deep learning tracker [31]	70.20%	56.90%	230	48.26%	40.87%	10.87%	242	861	75.30%	80.60%	2.87	7.79
	Discriminative Deep Learning [28]	76.50%	64.40%	230	63.04%	27.39%	9.57%	164	226	83.90%	81.30%	3.05	1.45
	Proposed (w/o amplitude)	73.31%	57.28%	230	46.96%	43.04%	10%	186	334	75.69%	80.66%	2.883	0.15
	Proposed (w/t amplitude)	73.30%	58.79%	230	46.09%	44.78%	9.13%	178	321	75.28%	82.26%	2.578	0.32
Average Results	Siamese deep network [36]	72.63%	60.64%	292	50.69%	33.10%	16.22%	545	1182	77.45%	82.82%	2.67	1.51
	Deep learning tracker [31]	72.86%	63.73%	292	52.05%	31.50%	16.45%	417	1084	77.76%	81.99%	2.30	11.90
	Discriminative Deep Learning [28]	75.77%	68.71%	292	64.38%	23.91%	11.70%	281	407	81.21%	86.55%	2.11	1.31
	Proposed (w/o amplitude)	74.09%	63.95%	292	51.37%	33.30%	15.33%	394	587	78.72%	84.82%	2.35	0.10
	Proposed (w/t amplitude)	74.14%	65.53%	292	50.34%	35.37%	14.29%	403	595	78.45%	86.46%	2.05	0.27

systems with/without amplitude. However, the best benefit of our method is the tracking speed as shown. Indeed, our methods can greatly reduce the run time compared to [28], [31], [36]. Note that we achieve this performance without using the person re-identification dataset for appearance learning. These comparison results indicate that our method can work very fast while keeping high MOT accuracy.



(a) Comparison results on PETS S2.L1

(b) Comparison results on PETS S2.L2

FIGURE 7. Comparisons of our methods by using different detectors on PETS S2.L1 and PETS S2.L2 datasets. Here, AMP NO-AMP mean our methods with/without the amplitude affinity model. HOG and MRCN are the HOG [48] and Mask R-CNN [50] detectors, respectively.

I. EVALUATION USING DIFFERENT DETECTORS

In order to investigate how a detector accuracy affects the MOT performance, we evaluate our MOT system over different detection responses on PETS S2.L1 and PETS S2.L2. We use the public available HOG detections from [45], [46] and detections by applying the Mask RCNN [50] detector. We compare our systems with/without the amplitude affinity models.

Figure 7 compares the performance of both systems in terms of several MOT metrics. Since the detection accuracy affects the precision and recall of a tracker the most, we compute recall, precision, and other metrics related to these. For all the metrics, our trackers yield the better scores by using the recent Mask RCNN detector than using a HOG detector. In particular, the gap of MOTA scores is large. Since this metric represents the overall tracking accuracy, it turns out that MOT performance is affected by a detection quality. When comparing our systems with/without the affinity model, exploiting the amplitude affinity produces better rates for all the metrics. Thus, we also prove that the proposed amplitude affinity model can indeed enhance the MOT performance regardless of the detector performance.

VII. CONCLUSION

In recent years, many autonomous systems exploit camera and radar sensors for achieving stable and accurate multi-object tracking. In this study, we have proposed a unified framework to exploit visual and amplitude features effectively for MOT. The proposed framework is based on object model learning and data association methods.

We have learned visual and amplitude models during tracking. In particular, we have learned an object appearance model using discriminative subspace learning, and an amplitude model using the MAP-based SNR estimation. By combining these affinity models with the confidence-based association, we have enhanced the MOT performance significantly. Furthermore, we have presented a practical track management method to deal with track initialization and duplication.

In order to show the benefits of our methods, we have implemented several MOT systems using different affinity models and association methods, and compare their performance extensively on several challenging visual MOT datasets. In addition, we have compared our method with state-of-the-art MOT methods using deep appearance learning. The comparison proves that our method achieves the high tracking accuracy which is comparable to the recent methods. In particular, the best benefit of our method is the low MOT complexity. We greatly reduce the run-time against the deep learning methods.

REFERENCES

[1] Y. Bar-Shalom, T. E. Fortmann, and M. Scheffe, "Joint probabilistic data association for multiple targets in clutter," *Inf. Sci. Syst.*, vol. 24, pp. 843–854, 1980.

[2] X. R. Li, "Tracking in clutter with strongest neighbor measurements. I. Theoretical analysis," *IEEE Trans. Autom. Control*, vol. 43, no. 11, pp. 1560–1578, Nov. 1998, doi: 10.1109/9.728872.

[3] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[4] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic data association revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3047–3055.

[5] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4696–4704.

[6] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.

[7] F. Tango, E. Richter, U. Scheunert, and G. Wanielik, "Advanced multiple objects tracking by fusing radar and image sensor data—Application on a case study," in *Proc. Int. Conf. Inf. Fusion*, Jun./Jul. 2008, pp. 1–7.

[8] M. Schikora and B. Romba, "A framework for multiple radar and multiple 2D/3D camera fusion," in *Proc. 4th German Workshop Sensor Data Fusion, Trends, Solutions, Appl.*, 2009, pp. 2358–2364.

[9] T. Wang, N. Zheng, J. Xin, and Z. Ma, "Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications," *Sensors*, vol. 11, no. 9, pp. 8992–9008, 2011.

[10] H. Cho, Y.-W. Seo, B. V. K. V. Kumar, and R. R. Rajkumar, "A multisensor fusion system for moving object detection and tracking in urban driving environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1836–1843.

[11] J. Ćesić, I. Marković, I. Cvišić, and I. Petrović, "Radar and stereo vision fusion for multitarget tracking on the special Euclidean group," *Robot. Auto. Syst.*, vol. 83, pp. 338–348, Sep. 2016.

[12] D. Musicki and B. La Scala, "Multi-target tracking in clutter without measurement assignment," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 3, pp. 877–896, Jul. 2008.

[13] C. Kreucher, K. Kastella, and A. O. Hero, "Multitarget tracking using the joint multitarget probability density," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1396–1414, Oct. 2005.

[14] M. R. Morelande, C. M. Kreucher, and K. Kastella, "A Bayesian approach to multiple target detection and tracking," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1589–1604, May 2007.

[15] P. Chavali and A. Nehorai, "Concurrent particle filtering and data association using game theory for tracking multiple maneuvering targets," *IEEE Trans. Signal Process.*, vol. 61, no. 20, pp. 4934–4948, Oct. 2013.

[16] R. Jinan and T. Raveendran, "Particle filters for multiple target tracking," *Procedia Technol.*, vol. 24, pp. 980–987, 2016.

[17] G. Van Keuk, "Multihypothesis tracking using incoherent signal-strength information," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 3, pp. 1164–1170, Jul. 1996.

[18] B. F. L. Scala, "Viterbi data association tracking using amplitude information," in *Proc. 7th Int. Conf. Inf. Fusion*, 2004, pp. 1–5.

[19] D. Clark, B. Ristic, B.-N. Vo, and B. T. Vo, "Bayesian multi-object filtering with amplitude feature likelihood for unknown object SNR," *IEEE Trans. Signal Process.*, vol. 58, no. 1, pp. 26–37, Jan. 2010.

[20] S. H. Bae, D. Y. Kim, J. H. Yoon, V. Shin, and K.-J. Yoon, "Automated multi-target tracking with kinematic and non-kinematic information," *IET Radar, Sonar Navigat.*, vol. 6, no. 4, pp. 272–281, 2012.

[21] S.-H. Bae, J. Park, and K.-J. Yoon, "Joint estimation of multi-target signal-to-noise ratio and dynamic states in cluttered environment," *IET Radar, Sonar Navigat.*, vol. 11, no. 3, pp. 539–549, Mar. 2017.

[22] M. Keuper, S. Tang, Z. Yu, B. Andres, T. Brox, and B. Schiele, "A multi-cut formulation for joint segmentation and tracking of multiple objects," *CoRR*, vol. abs/1607.06317, 2016.

[23] X. Wang, B. Fan, S. Chang, Z. Wang, X. Liu, D. Tao, and T. S. Huang, "Greedy batch-based minimum-cost flows for tracking multiple objects," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4765–4776, Oct. 2017.

[24] L. Chen, X. Peng, and M. Ren, "Recurrent metric networks and batch multiple hypothesis for multi-object tracking," *IEEE Access*, vol. 7, pp. 3093–3105, 2019.

[25] S.-H. Bae and K.-J. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1218–1225.

[26] B. Wang, G. Wang, K. L. Chan, and L. Wang, "Tracklet association with online target-specific metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1234–1241.

[27] J. Wei, M. Yang, and F. Liu, "Learning spatio-temporal information for multi-object tracking," *IEEE Access*, vol. 5, pp. 3869–3877, Mar. 2017.

[28] S.-H. Bae and K.-J. Yoon, "Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 595–610, Mar. 2018.

[29] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: Multi-object tracking and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7934–7943.

[30] J. Xiang, G. Zhang, and J. Hou, "Online multi-object tracking based on feature representation and Bayesian filtering within a deep learning architecture," *IEEE Access*, vol. 7, pp. 27923–27935, 2019.

[31] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. NIPS*, 2013, pp. 809–817.

[32] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Proc. AAAI*, Feb. 2017, pp. 1–8.

[33] K.-R. Kim, W. Choi, Y. J. Koh, S.-G. Jeong, and C.-S. Kim, "Instance-level future motion estimation in a single image based on ordinal regression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 273–282.

[34] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 4846–4855.

[35] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3701–3710.

[36] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 539–546.

[37] M. I. Skolnik, *Introduction to Radar Systems*, 2nd ed. New York, NY, USA: McGraw-Hill, 1980.

[38] S.-H. Lee, M.-Y. Kim, and S.-H. Bae, "Learning discriminative appearance models for online multi-object tracking with appearance discriminability measures," *IEEE Access*, vol. 6, pp. 67316–67328, 2018.

[39] S.-H. Bae, "Survey of amplitude-aided multi-target tracking methods," *IET Radar, Sonar Navigat.*, vol. 13, no. 2, pp. 243–253, Feb. 2019.

[40] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[41] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows—Theory, Algorithms and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.

[42] M. Longbin, S. Xiaoquan, Z. Yiyu, S. Zhong Kang, and Y. Bar-Shalom, ''Unbiased converted measurements for tracking,'' *IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 3, pp. 1023–1027, Jul. 1998.

[43] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, ''A consistent metric for performance evaluation of multi-object filters,'' *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.

[44] (2009). *PETS2009*. P. dataset. [Online]. Available: http://www.cvg.reading.ac.uk/pets2009/a.html

[45] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, ''MOTChallenge 2015: Towards a benchmark for multi-target tracking,'' *CoRR*, 2015.

[46] B. Benfold and I. Reid, ''Stable multi-target tracking in real-time surveillance video,'' in *Proc. CVPR*, Jun. 2011, pp. 3457–3464.

[47] N. Dalal and B. Triggs, ''Histograms of oriented gradients for human detection,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[48] P. Dollar, R. Appel, S. Belongie, and P. Perona, ''Fast feature pyramids for object detection,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[49] W. Li and X. Wang, ''Locally aligned feature transforms across views,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3594–3601.

[50] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, ''Mask R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.

SEUNG-HWAN BAE (Member, IEEE) received the B.S. degree in information and communication engineering from Chungbuk National University, in 2009, and the M.S. and Ph.D. degrees in information and communications from the Gwangju Institute of Science and Technology (GIST), in 2010 and 2015, respectively. He was a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI), South Korea, from 2015 to 2017. He was an Assistant Professor with the Department of Computer Science and Engineering, Incheon National University, South Korea, from 2017 to 2020. He is currently an Assistant Professor with the Department of Computer Engineering, Inha University, South Korea. His research interests include multi-object tracking, object detection, deep learning, dimensionality reduction, medical image analysis, generative adversarial networks, and image forensics.