IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 March 2022, accepted 22 April 2022, date of publication 2 May 2022, date of current version 13 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3171565

Multi-Object Tracking and Segmentation With Embedding Mask-Based Affinity Fusion in Hierarchical Data Association

YOUNG-MIN SONG[1], (Graduate Student Member, IEEE), YOUNG-CHUL YOON[2], KWANGJIN YOON[3], HYUNSUNG JANG[4], NAMKOO HA[4], AND MOONGU JEON[1], (Senior Member, IEEE)

[1]School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea
[2]Hyundai Motor Company, Uiwang-si 16082, South Korea
[3]SI Analytics Company Ltd., Daejeon 34051, South Korea
[4]LIG Nex1 Company Ltd., Yongin-si 16911, South Korea

Corresponding author: Moongu Jeon (mgjeon@gist.ac.kr)

ABSTRACT In this paper, we propose a highly feasible fully online multi-object tracking and segmentation (MOTS) method that uses instance segmentation results as an input. The proposed method is based on the Gaussian mixture probability hypothesis density (GMPHD) filter, a hierarchical data association (HDA), and a mask-based affinity fusion (MAF) model to achieve high-performance online tracking. The HDA consists of two associations: segment-to-track and track-to-track associations. One affinity, for position and motion, is computed by using the GMPHD filter, and the other affinity, for appearance is computed by using the responses from single object trackers such as kernalized correlation filter, SiamRPN, and DaSiamRPN. These two affinities are simply fused by using a score-level fusion method such as min-max normalization referred to as MAF. In addition, to reduce the number of false positive segments, we adopt mask IoU-based merging (mask merging). The proposed MOTS framework with the key modules: HDA, MAF, and mask merging, is easily extensible to simultaneously track multiple types of objects with CPU-only execution in parallel processing. In addition, the developed framework only requires simple parameter tuning unlike many existing MOTS methods that need intensive hyperparameter optimization. In the experiments on the two popular MOTS datasets, the key modules show some improvements. For instance, ID-switch decreases by more than half compared to a baseline method in the training sets. In conclusion, our tracker achieves state-of-the-art MOTS performance in the test sets.

INDEX TERMS Multi-object tracking, instance segmentation, tracking by segmentation, online approach, Gaussian mixture probability hypothesis filter, affinity fusion.

I. INTRODUCTION

Multi-object tracking and segmentation (MOTS) has recently become one of challenging research fields which has been proposed for pixelwise intelligent systems beyond 2D bounding boxes. This new vision task MOTS has been extended from a conventional task multi-object tracking (MOT) with segmentation. Since the MOT benchmark datasets [1]–[3] were released, the tracking-by-detection paradigm has been the mainstream, and breakthroughs [4], [5] in object detection have been achieved by many deep neural network (DNN)-based detectors [6]–[10] from various sensor domains, such as color cameras (2D images) and LiDAR (3D point clouds). For instance, the detection responses of [7], [8] are 2D bounding boxes and those of [6], [9], [10] are 3D boxes. In addition, He *et al.* [11] introduced a pixelwise classification and detection method represented by instance segmentation, which has motivated many segmentation-based studies including MOTS.

The MOTS task was first introduced in Voigtlaender *et al.* [12] with a new baseline method, new evaluation measures, and a new MOTS dataset extended

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

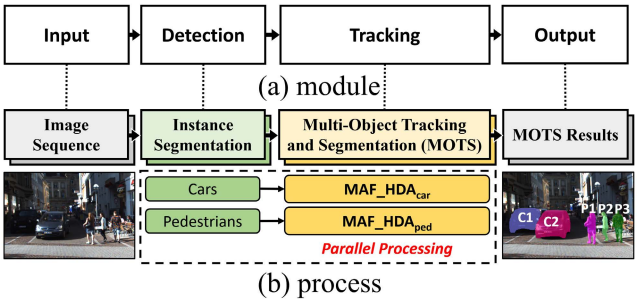


FIGURE 1. Flow chart of parallel multi-object tracking and segmentation (MOTS) processing, which receives two classes of objects, i.e., cars and pedestrians. (a) Modules (input, detection, tracking, and output) are implemented as (b) processes (image sequencing, instance segmentation, MOTS, and MOTS results). Our proposed MOTS framework is denoted by MAF_HDA.

from MOTChallenge [3] and KITTI [2] image sequences. To solve this task, most of state-of-the-art methods [13]–[18] have exploited multi-stage approaches that separate detection (instance segmentation) and tracking modules while some one-stage methods [12], [19], [20] have been rarely proposed. Luiten *et al.* [16] and Kim *et al.* [17] have proposed MOTS methods that use a fusion of 2D box detection, 3D box detection, and instance segmentation results. Xu *et al.* [13] exploited spatial-embedding [21] to raise instance segmentation quality which performs instance segmentation without bounding box (bbox) proposals so runs faster than bbox proposal based instance segmentation like Mask RCNN [11]. Also, they devised a point-wise representative feature extraction from input segments which can consider foreground and background information. Yang *et al.* [14] focused refining the segmentation quality fusing two difference Mask RCNN implementations in offline. These state-of-the art methods [13], [14], [16], [17], [19], [20] have improved MOTS performance through raising the detection quality: refinement or fusion of multi-type detections, and they necessarily involve exhausted additional learning step. Different to those state-of-the-art MOTS works, in this paper, we propose an easily feasible online MOTS method without the exhausted learning but also our method achieves competitive MOTS performances.

Our contributions are summarized as follows:

1) We propose an easily feasible online MOTS method consisting of (a) two-step hierarchical data association (HDA), (b) mask-based affinity fusion (MAF), and (c) mask merging. These key modules can run with CPU-only execution.

2) Particularly among the key modules, (b) MAF effectively fuses ''position and motion'' affinity with a Gaussian mixture probability density (GMPHD) filter [22] and ''appearance'' with single object tracker such as KCF [23], SiamRPN [24], and DaSiamRPN [25] to improve the MOTS performance compared to a baseline method using only one-step GMPHD filter association.

3) Additionally, the tracking part of proposed method can run with CPU-only process when KCF is used so that it can run in parallel to simultaneously track multiple types of objects: cars and pedestrians, in this paper (see Figure 1).

4) Finally, we evaluate the proposed method on state-of-the-art datasets [2], [3], [12]. The results on the training sets show that the developed key modules efficiently improve MOTS performances compared to the baseline method. In the results on the test sets, our method not only shows competitive performance against state-of-the-art published methods but also achieves state-of-the-art level performance against state-of-the-art unpublished methods that are available at the leaderboards of the MOTS20 and KITTI-MOTS websites.

The proposed method has high applicability due to a feasible combination of existing models [22]–[25] and simple parameter tuning unlike many state-of-the-art DNN based tracking methods [12]–[20]. In addition, in the experiments, our method shows state-of-the-art level performance. We present the works related to the proposed method in Section II and the details of our method are covered in Section III. Additionally, we discuss the experimental results in Section IV and conclude the paper in Section V. In what follows, we use MAF_HDA as the abbreviation for our proposed method.

II. RELATED WORKS

A. TRACKING MODELS WITH A PHD FILTER

The PHD filter [22], [26], [27] was originally designed to deal with radar and sonar data-based MOT systems. Mahler [26] proposed recursive Bayes filter equations for the PHD filter that optimize multi-target tracking processes where states and observations are defined with a random-finite set. Following this theory, Vo *et al.* [27] implemented the PHD filter by using a sequential Monte Carlo method with particle filtering and clustering, named the SMCPHD filter, and proposed the governing equations by using a Gaussian mixture model with a closed-form recursion method named the Gaussian mixture probability hypothesis density (GMPHD) filter. Since the GMPHD filter is tractable in implementing online and real-time trackers, it has been recently extended and exploited as a famous tracking model in video-based systems. While the radar and sonar sensors receive massive number of false positives but rarely miss any observations, visual object detectors yield many fewer false positives and more missed detections. Thus, in video-based tracking, noise control processes for the original domains are simplified and many additional techniques for visual objects have been developed.

1) POSITION AND MOTION MODELS

Song *et al.* [28] combined the GMPHD filter and data association processes with two-step hierarchy to recover lost tracks IDs. They designed an affinity model considering

position and linear motion between the tracks in the second step association. This approach is an intuitive implementation of the GMPHD filter to reconnect lost tracks. In addition, they presented an energy minimization model based on occluded objects group to correct the false associations that already occurred in the first step association between detections and tracks. Sanchez-Matilla *et al.* [29] proposed detection confidence-based data association schemes with a PHD filter. Strong (high-confidence) detections initiate and propagate tracks, but weak (low-confidence) detections propagate only existing tracks. This scheme works well when the detection results are reliable. However, the tracking performance depends on the detection performance and is especially weak for long-term missed detections. Sanchez-Matilla *et al.* [30] utilized long short-term memory (LSTM) models to design a global motion model for MOT.

2) APPEARANCE MODELS

More intensive tracking solutions [31], [32] have been proposed with appearance models. Kutschbach *et al.* [31] combined a naive GMPHD filtering process and kernelized correlation filters (KCF) [23] that can update appearance online and discriminate occluded objects. They proposed robust online appearance learning to refind the IDs of lost tracks. However, updating the appearance of every object in every frame requires heavy computing resources and inevitably increases the runtime. In Fu *et al.* [32], the GMPHD filter is equipped with an online group-structured dictionary for appearance learning and an adaptive gating technique, which is an advanced tracking process suitable for video-based MOT.

These online MOT methods based on the PHD filter have successfully improved the tracking performance by using motion and appearance learning models. We exploit the GMPHD filter in hierarchical data association of [28] to consider temporal information. In addition, to efficiently apply single object tracking (SOT) as appearance affinity in MOT, different to [31] that they simply used KCF [23] to propagate detection loss tracks in, we devise a simple and efficient affinity fusion model which fuse the GMPHD filter based position and motion affinity and SOT based appearance affinity. As shown in Figure 2, three representative SOT methods: KCF, SiamesRPN [24], and DaSiamRPN [25], can discriminate falsely classified pedestrian (true class: car) and true pedestrian since their class-independent feature extraction abilities.

B. STATE-OF-THE-ART MOTS METHODS

Conventional MOT methods [28]–[36] have exploited the tracking-by-detection paradigm, where the detectors [7], [8], [37]–[39] generate 2D bounding box (bbox) results and the trackers assign tracking identities (IDs) to the bounding boxes via data association. Refs. [28]–[32] successfully extended the GMPHD filter into visual object tracking as we discussed in the previous subsection II-A. Li *et al.* [34]

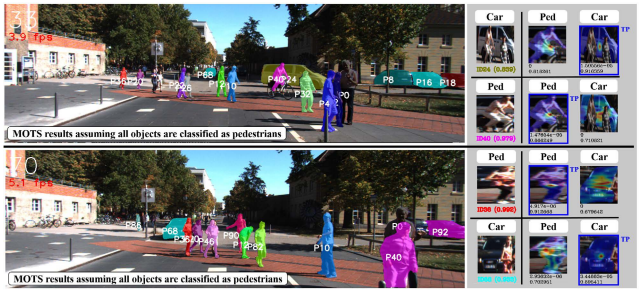


FIGURE 2. Demonstration of class-independent discrimination ability of a single object tracker: KCF [23], in KITTI-MOTS test 0028 sequence.

proposed a dissimilarity cost computation in form of weight multiplied summation of appearance, structure, motion and size based on 2D bbox. Muresan and Nedevschi [33] used a sophisticated MOT method that aggregates features of 2D image and 3D point cloud spaces by projecting 2D pixels into 3D point cloud map. Unlike MOT, MOTS uses pixelwise instance segmentation results as a tracking input beyong 2D bbox results. Voigtlaender *et al.* [12] first introduced the MOTS task. They extended the popular MOT datasets such as MOTChallenge [3] and KITTI [2] with instance segmentation results by using a fine-tuned MaskRCNN [11] for the same image sequences, and proposed a new soft multi-object tracking and segmentation accuracy (sMOTSA) measure that can be used to evaluate MOTS methods. In addition, they presented a new MOTS baseline method named TrackRCNN, which was extended from MaskRCNN with 3D convolutions to deal with temporal information. Inspired by the new task, state-of-the-art MOTS methods [13], [14], [16], [19] have been proposed very recently. MOTSFusion [16] proposes a fusion-based MOTS method exploiting bounding box detection [40] and instance segmentation [41]. It estimates a segmentation mask for each bounding box and builds up short tracklets using 2D optical flow, then fuses these 2D short tracklets into dynamic 3D object reconstructions hierarchically. The precise reconstructed 3D object motion is used to recover missed objects with occlusions in 2D coordinates. PointTrack [13] devises a new feature extractor based on PointNet [42] to appropriately consider both foreground and background features. This is motivated by the fact that the inherent receptive field of convolution-based feature extraction inevitably confuses up the foreground and background features. PointNet is used to randomly sample feature points considering the offsets between foreground and background regions, the colors of those regions, and the categories of segments. Then the context-aware embedding vectors for association are built after concatenation of the separately computed position difference vectors. In addition PointTrack exploited spatial-embedding [21] to raise instance segmentation quality which performs instance segmentation without bounding box. Similarly, CPPNet [19] trained their segmentation model by using [21] and proposed copy-paste data argumentation technique. ReMOTS [14] focused on

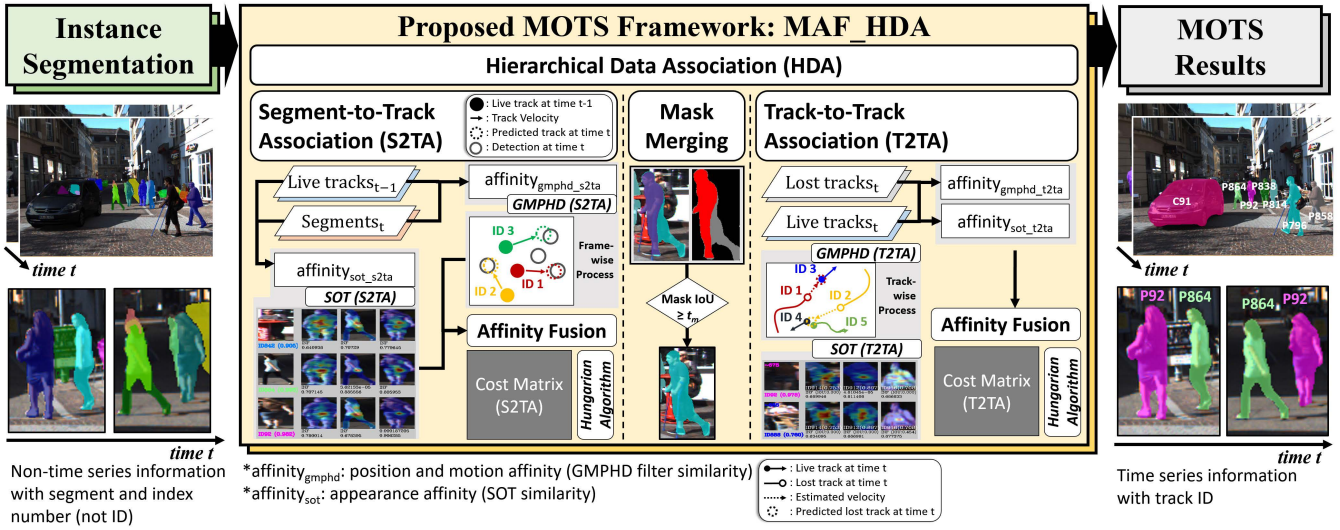


FIGURE 3. Detailed processing pipeline of MAF_HDA with input (images and instance segmentation results) and output (MOTS results). The key components are hierarchical data association (HDA), mask merging, and mask-based affinity fusion (MAF). HDA has two association steps: S2TA and T2TA. MAF executes each affinity fusion in each association step while mask merging runs once between S2TA and T2TA.

refining the segmentation quality fusing two difference Mask RCNN implementations in offline.

Many of state-of-the-art MOTS studies [13], [14], [16], [19] have performed with their own pixelwise segmentation quality increase techniques. Different to them, we propose an easily feasible MOTS method without intensive or additional learning techniques that can give high applicability to research community. Our proposed method, named MAF_HDA, exploits the tracking-by-instance-segmentation paradigm, which performs the MOTS task by using two popular filtering methods: the GMPHD filter [22] and the KCF [23]. We build a two-step hierarchical data association (HDA) strategy to handle tracklet loss and ID switches. In each association step, position and motion affinity are calculated by the GMPHD filter, and appearance affinity is calculated by the KCF. To appropriately consider these two affinities, we devise a mask-based affinity fusion (MAF) model. Those key modules of parameters are simply tuned through adjusting the values in 0.0 to 1.0 ranges. Moreover, to show our final method's efficiency, we compare four MAF_HDA settings with a conventional KCF, a modified KCF for MOT (KCF2), and two state-of-the-art Siamese network based SOT methods: SiamRPN [24] and DaSiamRPN [25]. As a result, the four MAF_HDA methods show competitive performance in two popular KITTI-MOTS [2] and MOTS20 [3] datasets against state-of-the-art MOTS methods [12]–[20] and KCF2 shows the best performance among the four SOTs.

III. PROPOSED METHOD

In this section, we introduce the proposed online multi-object tracking and segmentation (MOTS) framework in terms of input/output interfaces (I/O) and key modules in detail. Following the tracking-by-segmentation paradigm, the MOTS method receives image sequences and instance segmentation results as inputs and gives MOTS results as outputs, which are shown in Figures 1 and 3. Each instance has an object type, pixelwise segment, and confidence score but does not include time series information. Through the MOTS method, we can assign tracking IDs to the object segments and turn them into time series information, i.e., MOTS results.

The proposed MOTS framework is not only built based on a HDA strategy consisting of segment-to-track association (S2TA) and track-to-track association (T2TA) but is also implemented as a fully online process using only information at the present time t and the past times 0 to $t-1$. In each observation-to-state association step, affinities between states and observations are calculated considering position, motion, and appearance. The ''position and motion'' and ''appearance'' affinities are computed by using a GMPHD filter [22] and a single object tracking (SOT) method such as KCF [23], SiamRPN [24], and DaSiamRPN [25], respectively. Since these two types of affinities have different filtering domains, one affinity can be of a much higher magnitude than the other affinity. To appropriately consider position, motion, and appearance information in HDA, we devise a MAF method. Additionally, to reduce false positive segments, we adopt the mask intersection-over-union (IoU)-based merging technique between S2TA and T2TA.

In summary, the proposed MOTS framework follows the order of (1) S2TA, (2) mask merging, and then (3) T2TA, in which the affinities of each association are computed by exploiting the GMPHD filter and KCF, are fused by using MAF. In what follows, we use MAF_HDA as the abbreviation for the proposed framework (see Figure 3).

A. GMPHD FILTERING THEORY

The main steps of the GMPHD filtering-based tracking includes initialization, prediction, and update. The set of

states (segment tracks) and the set of observations (instance segmentations) at time t are X_t and Z_t represented as follows:

$$X_t = \{\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{N_t}\}, \tag{1}$$

$$Z_t = \{\mathbf{z}_t^1, \ldots, \mathbf{z}_t^{M_t}\}, \tag{2}$$

where a state vector $\mathbf{x}_t$ is composed of $\{c_x, c_y, v_x, v_y\}$ with a tracking ID, and segment mask. c_x, c_y, and v_x, v_y indicate the center coordinates of the mask's 2D box, and the velocities of the x and y directions of the object, respectively. An observation vector $\mathbf{z_t}$ is composed of center point $\{c_x, c_y\}$ of a segment mask $\mathbf{s_t^k}$ with a confidence score δ_t^k. The Gaussian model $\mathcal{N}$ representing $\mathbf{x_t}$ is initialized by $\mathbf{z_t}$, predicted to $\mathbf{x_{t+1|t}}$, and updated to $\mathbf{x_{t+1}}$ by $\mathbf{z_{t+1}}$.

1) INITIALIZATION

The Gaussian mixture model g_t are initialized by using the initial observations from the detection responses. In addition, when an observation fails to find the association pair, i.e., to update the target state, the observation initializes a new Gaussian model. We call this *birth* (a kind of initialization). Each Gaussian $\mathcal{N}$ represents a state model with weight w, mean vector $\mathbf{x}$, input observation vector $\mathbf{z}$, and covariance matrix P, which are as follows:

$$g_t(\mathbf{z}) = \sum_{i=1}^{N_t} w_t^i \mathcal{N}(\mathbf{z}; \mathbf{x}_t^i, P_t^i), \tag{3}$$

where N_t is the number of Gaussian models. At this step, we set the initial velocities of the mean vector to zero. Each weight is set to the normalized confidence value of the corresponding detection response: confidence score δ given by instance segmentation module. Additionally, the method of setting covariance matrix P is shown in Section IV-B2.

2) PREDICTION

We assume that there already has been the Gaussian mixture g_{t-1} of the target states at the previous frame $t-1$, as shown in (4). Then, we can predict the state at time t using Kalman filtering. In (5), $\mathbf{x}_{t|t-1}^i$ is derived by using the velocity at time $t-1$ and the covariance P is also predicted by the Kalman filtering method in (6) as:

$$g_{t-1}(\mathbf{z}) = \sum_{i=1}^{N_{t-1}} w_{t-1}^i \mathcal{N}(\mathbf{z}; \mathbf{x}_{t-1}^i, P_{t-1}^i), \tag{4}$$

$$\mathbf{x}_{t|t-1}^i = F\mathbf{x}_{t-1}^i, \tag{5}$$

$$P_{t|t-1}^i = Q + FP_{t-1}^i(F)^T, \tag{6}$$

where F is the state transition matrix, and Q is the process noise covariance matrix. Those two matrices are constant in our tracker.

3) UPDATE

The goal of the update step is to derive (7). First, we should find an optimal observation $\mathbf{z}$ at time t to update the Gaussian model. The optimal $\mathbf{z}$ in the observation set Z makes q_t the maximum value in (8) as:

$$g_{t|t}(\mathbf{z}) = \sum_{i=1}^{N_{t|t}} w_t^i(\mathbf{z}) \mathcal{N}(\mathbf{z}; \mathbf{x}_{t|t}^i, P_{t|t}^i), \tag{7}$$

$$q_t^i(\mathbf{z}) = \mathcal{N}(\mathbf{z}; H\mathbf{x}_{t|t-1}^i, R + HP_{t|t-1}^i(H)^T). \tag{8}$$

From the perspective of application, the update step involves data association. Finding the optimal observations and updating the state models is equivalent to finding the association pairs. R is the observation noise covariance. H is the observation matrix uesd to transform a state vector into an observation vector. Both matrices are constant in our application. After finding the optimal $\mathbf{z}$, the Gaussian mixture is updated in the order of (9), (10), (11), and (12) as:

$$w_t^i(\mathbf{z}) = \frac{w_{t|t-1}^i q_t^i(\mathbf{z})}{\sum_{l=1}^{N_{t|t-1}} w_{t|t-1}^l q_t^l(\mathbf{z})}, \tag{9}$$

$$\mathbf{x}_{t|t}^i(\mathbf{z}) = \mathbf{x}_{t|t-1}^i + K_t^i(\mathbf{z} - H\mathbf{x}_{t|t-1}^i), \tag{10}$$

$$P_{t|t}^i = [I - K_t{}^i H]P_{t|t-1}^i, \tag{11}$$

$$K_t^i = P_{t|t-1}^i H^T(HP_{t|t-1}^i H^T + R)^{-1}, \tag{12}$$

where the set of $w_{t|t-1}$ includes w_{t-1} (weights from the targets at the previous frame) and w_{t-1} (weights of newly born targets). Likewise, $N_{t|t-1}$ is the sum of N_{t-1} and the number of the newly born targets.

B. HIERARCHICAL DATA ASSOCIATION (HDA)

To compensate for the imperfection of the framewise one-step online propagation of the GMPHD filtering process, we extend the GMPHD filter-based online MOT with a hierarchical data association (HDA) strategy that has two-step association steps: S2TA and T2TA (see Figure 4). Each association has different states and observations as inputs, which are used to compute position and motion affinity$_{pm}$ and appearance affinity$_{appr}$ (see Figure 5). Song *et al.* [28] proposed a GMPHD filter based hierarchical data association strategy. They adjust the minimum consecutive frames for initialization in detection-to-track association and the minimum track length for track-to-track association since they use only GMPHD filter based position and motion affinity for realtime speed. So false associations can be prevented between just close tracks each other by using reliable tracks. However, in our work, a track state is initialized in a single frame as soon as S2TA succeeds and the minimum track length for T2TA is 1 for fully online process, and the false associations can be prevented by using the appearance affinity.

To build the proposed HDA strategy, we define some online MOT's processing units at the present time t. S_t indicates the instance segmentation results and the k^{th} segment is denoted by $\mathbf{s_t^k}$. T indicates a set of tracks. These units are defined in detail as:

$$S_t = \{\mathbf{s_t^1}, \ldots, \mathbf{s_t^k}\}, \tag{13}$$

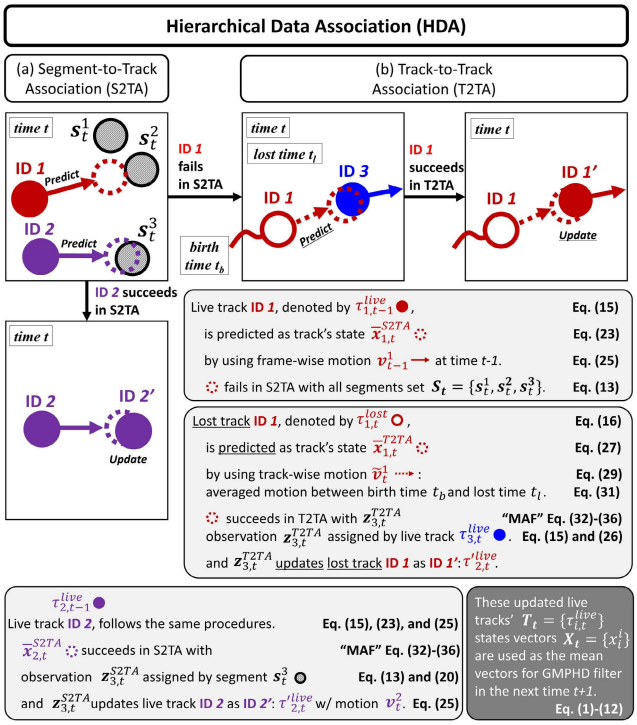


FIGURE 4. Demonstration of the hierarchical data association (HDA) process with (a) S2TA and (b) T2TA. Track $\tau^{live}_{1,t-1}$ can be associated with an observation $z^{S2TA}_{k,t}$ from segment k. If S2TA fails, τ^{live}_{1,t_1} becomes $\tau^{lost}_{1,t}$, and it can succeed in T2TA with an observation $z^{T2TA}_{3,t}$ from the live track $\tau^{live}_{3,t}$.

$$T^{live}_t = \{\tau^{live}_{1,t}, \ldots, \tau^{live}_{i,t}\}, \tag{14}$$
$$\tau^{live}_{i,t} = \{\mathbf{x}^i_{t_b}, .., \mathbf{x}^i_{t_l}\}, \quad 0 \le t_b < t_l,\ t_l = t, \tag{15}$$
$$T^{lost}_t = \{\tau^{lost}_{1,t}, \ldots, \tau^{lost}_{j,t}\}, \tag{16}$$
$$\tau^{lost}_{j,t} = \{\mathbf{x}^j_{t_b}, .., \mathbf{x}^j_{t_l}\}, \quad 0 \le t_b < t_l < t, \tag{17}$$
$$\mathbf{x}_{t_b} = \{c_{x,t_b}, c_{y,t_b}, v_{x,t_b}, v_{y,t_b}\}^T, \tag{18}$$
$$\mathbf{x}_{t_l} = \{c_{x,t_l}, c_{y,t_l}, v_{x,t_l}, v_{y,t_l}\}^T, \tag{19}$$

where two attributes: *"live"* and *"lost"*, indicate success and failure in tracking at time t, respectively, which are not compatible, and thus $T^{lost}_t \cup T^{live}_t = T^{all}_t$ and $T^{lost}_t \cap T^{live}_t = \phi$ are satisfied. T_t is composed of a track $\tau_{i,t}$ with identity i which is also a set of state vectors from the birth time t_b to the last tracking time t_l. In the case of $\tau^{live}_{i,t}$, t_l is identical to the present time t, in the case of $\tau^{lost}_{j,t}$, t_l is less than time t. Regardless of when time t is, state vector $\mathbf{x}$ has the center point $\{c_x, c_y\}$ in the segment bounding box, velocities $\{v_x, v_y\}$ in the x and y axis directions, an identity (ID), and a segment mask (see (18) and (19)).

1) SEGMENT-TO-TRACK ASSOCIATION (S2TA)

In S2TA, the observations denoted by Z^{S2TA}_t are frame-by-frame instance segmentation results S_t. If there are no track states, the states X^{S2TA}_t are initialized from Z^{S2TA}_t, and otherwise, $\bar{X}^{S2TA}_t$ is predicted from T^{live}_{t-1} and updated by using the GMPHD filter with the processing units as follows:

$$\mathbf{z}^{S2TA}_{k,t} = \{c^k_{x,t}, c^k_{y,t}\}^T \text{ from } \mathbf{s}^{\mathbf{k}}_{\mathbf{t}}, \tag{20}$$
$$\mathbf{x}^{S2TA}_{i,t-1} = \{c^i_{x,t-1}, c^i_{y,t-1}, v^i_{x,t-1}, v^i_{y,t-1}\}^T \text{ from } \tau^{live}_{i,t-1}, \tag{21}$$
$$F^{S2TA} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{22}$$
$$\bar{\mathbf{x}}^{S2TA}_{i,t} = F^{S2TA}\mathbf{x}^{S2TA}_{i,t-1}, \tag{23}$$
$$\bar{\mathbf{x}}^{S2TA}_{i,t} = \{\bar{c}^i_{x,t}, \bar{c}^i_{y,t}, v^i_{x,t-1}, v^i_{y,t-1}\}^T, \tag{24}$$

where matrix F^{S2TA} (22) makes (21) be (24) after multiplication (23) that is identical to (5) from the *Prediction* step and $\bar{c}^i_t$ is equal to $c^i_{t-1} + v^i_{t-1}$. In (22), 1 at 1^{st} row and 3^{rd} column and 1 at 2^{nd} row and 4^{th} column indicate the frame difference between $t-1$ and t. An example is *ID*2 in Figure 4(a).

General Kalman filter was designed for predicting a single object in a space so it can be easily drifted by initial velocity to find the one object. However, in MOT, the drifting can cause false associations with other observations. Depending on whether the state x finds that an observation z is associated, is born, or neither, the framewise motion v is updated as follows:

$$\mathbf{v}^i_t = \begin{cases} \beta * \mathbf{v}^i_{t-1} + (1.0-\beta) \\ \quad * \begin{Bmatrix} c^k_{x,t} - \bar{c}^i_{x,t} \\ c^k_{y,t} - \bar{c}^i_{y,t} \end{Bmatrix}, & \text{if } \mathbf{z}^k_t \text{ is assigned to } \bar{\mathbf{x}}^{S2TA}_{i,t} \\ \{0,0\}^T, & \text{else if } \mathbf{x}^{S2TA}_{i,t} \text{ is born} \\ \mathbf{v}^i_{t-1}, & \text{otherwise,} \end{cases} \tag{25}$$

where β can be differently set according to the scene context and frame rate. The impact of β is presented in Figure 9 and 8 in detail.

2) TRACK-TO-TRACK ASSOCIATION (T2TA)

In T2TA, observations Z^{T2TA}_t and states X^{T2TA}_t (inputs) are built from the live track set T^{live}_t and lost track set T^{lost}_t, respectively. Each of T^{live}_t and T^{lost}_t consists of the track vectors of $\tau^{live}_{i,t}$ and $\tau^{lost}_{j,t}$ with their identities (see (14) and (16)). The track vectors have temporal information with the birth time t_b and loss time t_l. The live track's t_l is identical to the current time t, which means that the track is not yet lost, while the lost track's t_l is less than t, which means the track was lost before the time t (see (15) and (17)).

Since general Kalman filter only considers frame-by-frame prediction, unlike that the *Prediction* step of S2TA uses the framewise motion from time $t-1$ to t, we devise a simple trackwise motion model considering temporal information. The trackwise motion analysis is used in T2TA as follows:

$$\mathbf{z}^{T2TA}_{i,t} = \{c^i_{x,t}, c^i_{y,t}\}^T \text{ from } \tau^{live}_{i,t}, \tag{26}$$
$$\mathbf{x}^{T2TA}_{j,t-1} = \{c^j_{x,t_l}, c^j_{y,t_l}, \tilde{v}^j_{x,t}, \tilde{v}^j_{y,t}\}^T \text{ from } \tau^{lost}_{j,t}, \tag{27}$$
$$F^{T2TA} = \begin{pmatrix} 1 & 0 & d_f & 0 \\ 0 & 1 & 0 & d_f \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{28}$$

$$\bar{\mathbf{x}}_{j,t}^{T2TA} = F^{T2TA}\mathbf{x}_{j,t-1}^{T2TA}, \tag{29}$$

$$\bar{\mathbf{x}}_{j,t}^{T2TA} = \{\bar{c}_{x,t}^{j}, \bar{c}_{y,t}^{j}, v_{x,t}^{j}, v_{y,t}^{j}\}^{T}, \tag{30}$$

where $d_f(i,j)$ (28) is the frame difference between $\tau_{i,t}^{live}$'s first element $\mathbf{x}_{t_b}^{i}$ (15) and $\tau_{j,t}^{lost}$'s last element $\mathbf{x}_{t_l}^{j}$ (17). The trackwise motion vector $\tilde{\mathbf{v}}_t^j$ of (27) has two linearly averaged velocities $\tilde{v}_{x,t}^{j}$ and $\tilde{v}_{y,t}^{j}$ of a track, in the directions of the x-axis and y-axis, respectively, as follows:

$$\tilde{\mathbf{v}}_t^j = \{\tilde{v}_{x,t}^{j}, \tilde{v}_{y,t}^{j}\}^{T} = \{\frac{c_{x,t_l}^{j} - c_{x,t_b}^{j}}{t_l - t_b}, \frac{c_{y,t_l}^{j} - c_{y,t_b}^{j}}{t_l - t_b}\}^{T}, \tag{31}$$

where the velocities $\frac{c_{x,t_l}^{j} - c_{x,t_b}^{j}}{t_l - t_b}$ and $\frac{c_{y,t_l}^{j} - c_{y,t_b}^{j}}{t_l - t_b}$ are calculated by subtracting the center position of the first object state $\mathbf{x}_{t_b}^{j}$ from that of the last state $\mathbf{x}_{t_l}^{j}$ and dividing it by $t_l - t_b$, which is the frame difference and is equivalent to the length of the track $\tau_{j,t}^{lost}$. A related example is shown in Figure 4(b) *ID*1.

In terms of temporal motion analysis, S2TA has the same time interval *"1"* between states and observations in transition matrix F, whereas T2TA has a different time interval (frame difference) between states and observations. The variable d_f depends on which state of the lost track and observation of the live track are paired. (20)-(25) of S2TA are the prediction step with framewise motion analysis and update, but (26)-(31) of T2TA contain the prediction step with trackwise linear motion analysis. Detailed examples are shown in Figure 4.

Recent researches [16], [33] exploit both Kalman filter and optical flow in their motion models, we exclude the optical flow since its intensive computation. Instead, we devise framewise motion (25) and trackwise motion (31) in HDA. Following the proposed HDA strategy, for S2TA and T2TA, two cost matrices can be filled by using the affinities between the differently defined states and observations. In the next subsection, we present an efficient mask-based affinity calculation method considering position, motion, and appearance for multi-object tracking and segmentation.

C. MASK-BASED AFFINITY FUSION (MAF)

We adopt a simple score-level fusion method to adequately consider position, motion, and appearance between states and observations. Fusing affinities obtained from different domains requires a normalization step that can balance the different affinities and avoid bias toward one affinity, which may have a much higher magnitude than the others.

1) POSITION AND MOTION AFFINITY

The GMPHD filter includes Kalman filtering in its *Prediction* step, (4)-(6), designed with a linear motion model with noise Q. Additionally, we present two different linear motion models for the hierarchical data association with two steps, S2TA and T2TA, as described in (25) and (31). Therefore, the position and motion affinity between the i^{th} state and j^{th} observation gives the probabilistic value $w \cdot q(\mathbf{z})$ obtained by the GMPHD filter as follows:

$$A_{pm}^{(i,j)} = w^{i} \cdot q^{i}(\mathbf{z}^{j}), \tag{32}$$

which is acquired from (8) and (9) of the *Update* step.

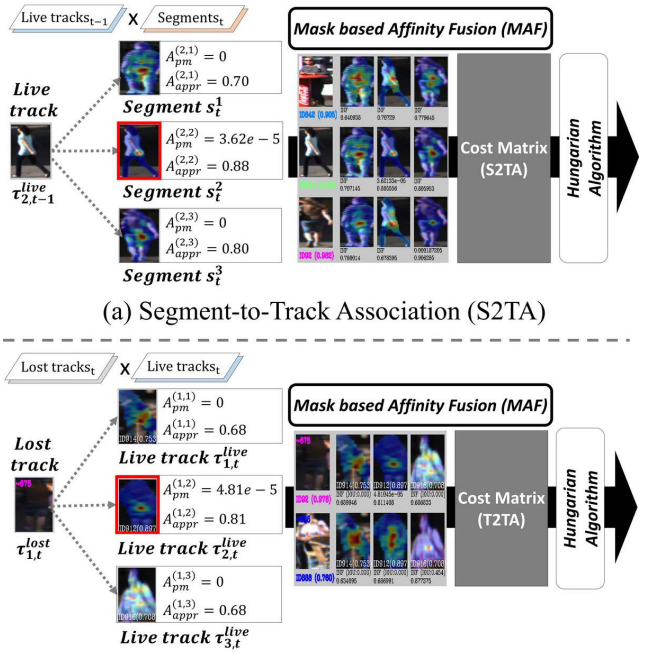


FIGURE 5. Detailed examples of the proposed mask-based affinity fusion (MAF) method with the hierarchical data association (HDA): (a) S2TA and (b) T2TA.

2) APPEARANCE AFFINITY

We exploit single object tracking (SOT) methods [23]–[25] to compute the appearance affinity between the i^{th} state and j^{th} observation since instance segmentation results does not provide appearance features to discriminate the objects belonging to a single class, pedestrian or cars. The SOT does not have class dependency because it was originally designed for single-object tracking challenges such as the VOT benchmark [43]. So it can be applied multi-class tracking and we utilize it for calculating the appearance similarity by matching object templates in this paper. Before applying the SOT method, the state and observation image templates are preprocessed by setting the backgrounds pixels to zero in the RGB channel's 0 to 255 ranges. This preprocessing step ensures that he appearance affinity pays attention to the foreground pixels based on the segment mask. The SOT-based affinity can be derived as follows:

$$A_{appr}^{(i,j)} = 1 - \frac{\sum_{c=x_j}^{width_j}\sum_{r=y_j}^{height_j}\bar{s}_{SOT}^{(i,j)}(r,c)}{width_j \cdot height_j}, \tag{33}$$

where $\bar{s}(\cdot)$ indicates the normalized SOT similarity value, which varies from 0.0 to 1.0 at each pixel. To verify that single-object tracking does efficiently work in computing this appearance affinity, one conventional method: KCF [23],

and two state-of-the-art methods: SiamRPN [24] and DaSiamRPN [25], are adopted in our work.

3) MIN-MAX NORMALIZATION

In our experiments, A_{pm} and A_{appr} have quite different magnitudes, e.g., $A_{pm} = \{10^{-9}, \ldots, 10^{-3}\}$ and $A_{appr} = \{0.4, \ldots, 1.0\}$ (see Figure 5). To fuse two affinities, we apply min-max normalization to them as follows:

$$\bar{A}^{(i,j)} = \frac{A^{(i,j)} - \min_{\substack{1\le i\le N \\ 1\le j\le M}} A^{(i,j)}}{\max_{\substack{1\le i\le N \\ 1\le j\le M}} A^{(i,j)} - \min_{\substack{1\le i\le N \\ 1\le j\le M}} A^{(i,j)}}, \tag{34}$$

where A_{appr} and A_{pm} are normalized into $\bar{A}_{appr}$ and $\bar{A}_{pm}$, respectively. A_{appr} and A_{pm} have quite different magnitudes but are normalized 0.0 to 1.0 ranges in (34). First after subtracting the minimum value from the original affinity A value, and it is divided by the difference between the maximum value and the minimum value. Then, we finally propose a MAF model represented by:

$$A^{(i,j)}_{maf} = \bar{A}^{(i,j)}_{pm} \bar{A}^{(i,j)}_{appr}. \tag{35}$$

Figures 6 and 7 show the probabilistic distributions before MAF and after MAF. From this fused affinity, we can compute the final cost between states and observations as follows:

$$Cost(\mathbf{x}^{\mathbf{i}}_{\mathbf{t}|\mathbf{t-1}}, \mathbf{z}^{\mathbf{j}}_{\mathbf{t}}) = -\alpha \cdot \ln A^{(i,j)}_{maf}, \tag{36}$$

where α is a scale factor empirically set to 100. If one of the affinities is close to zero, such as 10^{-39}, the cost is set to 10000 to prevent the final cost from becoming an infinite value. Then, the final cost ranges from 0 to 10000.

From the different states and observations (inputs) in S2TA and T2TA, two cost matrices are computed in every frame and we utilize the Hungarian algorithm [44] to solve the cost matrices, which has $O(n^3)$ time complexity, as shown in Figure 5. Then, observations succeeding in S2TA or T2TA are assigned to the associated states for *Update*, and other observations failing in S2TA and T2TA initialize new states.

4) ANALYSIS OF AFFINITY DATA

Figures 6(a)-(c) and 7(a)-(c) show that the position and motion affinity A_{gmphd} and appearance affinity A_{kcf} have quite different data magnitudes and distributions. In our experiments, $A_{pm} = \{10^{-9}, \ldots, 10^{-3}\}$ and $A_{appr} = \{0.4, \ldots, 1.0\}$ are observed. Figure 6(a) shows that the cars have more concentrated distributions, with mean $m_{kcf} \approx 0.944$ for appearance affinity than the pedestrians, with $m_{kcf} \approx 0.905$ in Figure 7(a). On the other hand, for the GMPHD affinity, pedestrians have more concentrated distributions as seen in Figures 6(b) and Figure 7(b). These facts are interpreted as follows: cars can be well discriminated by position and motion while pedestrians can be well discriminated by appearance. To considering these two characteristics, we propose MAF; from the distributions of normalized affinities $\bar{A}_{gmphd}$ and $\bar{A}_{kcf}$ in Figures 6(d) and 7(d), the gaps are much closer than before, and the two affinities are fused into A_{maf} by using MAF.

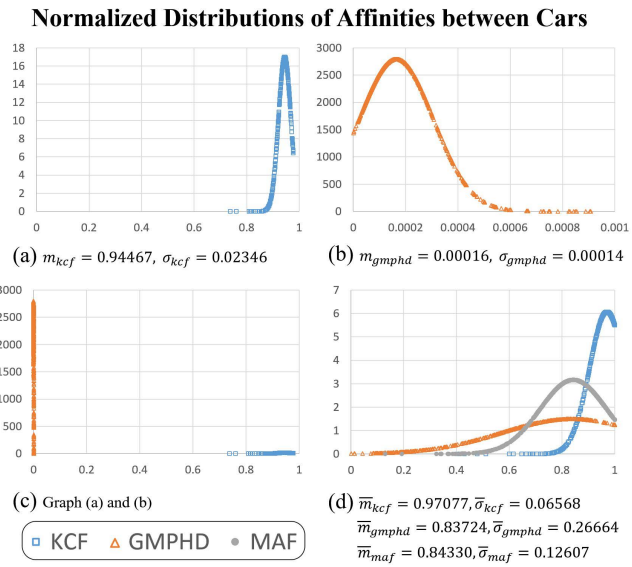


FIGURE 6. Normalized distributions of the affinities between cars in KITTI-MOTS training sequence 0019. KCF and GMPHD represent "appearance affinity" and "position and motion affinity", respectively. (a) and (b) show the distributions with each average m and standard deviation σ, and (c) shows that (a) and (b) are very different from each other. (d) The proposed mask-based affinity fusion (MAF) method can determine the scale difference between the KCF and GMPHD affinities and then normalize the two affinities and fuse (multiply) them. $\bar{m}$ and $\bar{\sigma}$ denote the normalized values in (34).

D. MASK MERGING

As shown in Figure 3, for mask merging, i.e., track merging, we can utilize bounding box-based IoU or segment mask-based IoU (mask IoU) measures that calculate boxwise or pixel-wise overlapping ratios between two objects, respectively. The two measures are represented by:

$$\text{IoU}(A, B) = \frac{\text{bbox}(A) \cap \text{bbox}(B)}{\text{bbox}(A) \cup \text{bbox}(B)}, \tag{37}$$

$$\text{Mask IoU}(A, B) = \frac{\text{mask}(A) \cap \text{mask}(B)}{\text{mask}(A) \cup \text{mask}(B)}. \tag{38}$$

If the value of a selected measure is greater than or equal to the threshold t_m, the two objects are merged into one object. Mask merging is applied only between tracking objects, i.e., states, that are not observations, after S2TA.

E. PARALLEL PROCESSING

We assume that data association runs only between the same class of objects. For example, if the instance segmentation module provides two or more object classes, e.g., car and pedestrian classes, our proposed framework is easily expansible (see Figure 1). In this paper, we implement the MOTS module with two parallel processes because the datasets used for our experiments produce car and pedestrian segments. Then, the time complexity $O((|car| + |pedestrian|)^3)$ decreases to the slower of $O(|car|^3)$ and $O(|pedestrian|^3)$.

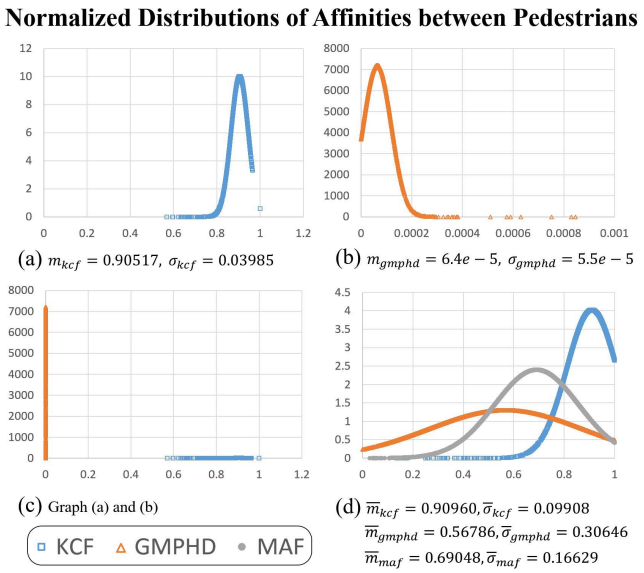


FIGURE 7. Normalized distributions of the affinities between pedestrians in KITTI-MOTS training sequence 0019. KCF and GMPHD represent "appearance affinity" and "position and motion affinity", respectively. (a) and (b) show the distributions with each average m and standard deviation σ, and (c) shows that (a) and (b) are very different from each other. (d) The proposed mask-based affinity fusion (MAF) method can determine the scale difference between the KCF and GMPHD affinities and then normalize the two affinities and fuse (multiply) them. $\bar{m}$ and $\bar{\sigma}$ denote the normalized values in (34).

TABLE 1. Dataset specifications for MOTS20 and KITTI-MOTS.

Dataset	Set	Sequence	FPS	Resolution	Frame
MOTS20	Train	MOTS20-02	30	1920x1080	600
		MOTS20-05	14	640x480	837
		MOTS20-09	30	1920x1080	525
		MOTS20-11	30	1920x1080	900
	Total Frames				**2,862**
	Test	MOTS20-01	30	1920x1080	450
		MOTS20-06	14	640x480	1,194
		MOTS20-07	30	1920x1080	500
		MOTS20-12	30	1920x1080	900
	Total Frames				**3,044**
KITTI-MOTS	Train	kitti-tracking-train: 00∼20	10	1224x370, 1238x374, , or 1242x375	**8,008**
	Test	kitti-tracking-test: 00∼28	10	1224x370, 1238x374, , or 1242x375	**11,095**

IV. EXPERIMENTS

In this section, we present experimental studies for the proposed MOTS method, named MAF_HDA, in detail. In IV-A, we note that MAF_HDA is studied with state-of-the-art MOTS20 [12] and KITTI-MOTS [2] datasets and new evaluation measures. In IV-B, the implementation details of our method are addressed in terms of development environments and parameter settings. In IV-C, experimental studies on the key parameters are addressed. In IV-D, we determine the effectiveness of key modules through ablation studies in the dataset training subset. The ablation studies show that the proposed key modules comprehensively improve the baseline model $p1$ remarkably in terms of IDS. In particular, we compare the effectiveness of KCF, SiamRPN, and DaSiamRPN as "appearance" affinity by using correlation with "position and motion" affinity (GMPHD) as shown in Figure 10. Finally, in IV-E, we show that the final proposed model $p6$ achieves competitive performances on the test sequences of the datasets in terms of the sMOTSA, MOTSP, and IDS measures.

TABLE 2. Evaluation measures. sMOTSA is mainly used for measuring the tracking performance as a key measure.

Measure	Better	Perfect	Description
MOTSA	↑	100%	Multi-Object Tracking and Segmentation Accuracy [12]. This measure is the mask-based MOTS accuracy which combines four sources: TP, FN, FP, and IDS.
sMOTSA	↑	100%	Soft Multi-Object Tracking and Segmentation Accuracy [12]. This measure is the soft mask-based MOTS accuracy which combines three sources: TP, FP, and IDS.
TP	↓	0	Total number of true positive masks.
FP	↓	0	Total number of false positive masks.
FN	↓	0	Total number of false negative masks (missed targets).
IDS	↓	0	Total number of identity switches. Please note that we follow the stricter definition of identity switches as described in [46].
FPS	↑	∞	Processing speed in frames per second on the benchmark. Infinity symbol means that higher is better.

A. DATASETS AND MEASURES

MAF_HDA is evaluated on MOTS20 [12] and KITTI-MOTS [2], which are the most popular datasets for MOTS. Voigtlaender *et al.* [12] proposed new MOTS measures and two MOTS datasets that were extended from image sequences of MOT16 [3] and KITTI [2]. They have been widely used for multi-object tracking with 2D bounding box-based detection results but instance segmentation results with the same image sequences were provided for MOTS, created by Mask RCNN [11] X152 of Detectron2 [45]. Table 1 describes the MOTS20 and KITTI-MOTS benchmark datasets in terms of training and test sequences, frames per second (FPS), resolution, and the number of frames (Frame). MOTS20 provides six high resolution 1920 × 1080 images with 30 FPS and two low resolution 640 × 480 image sequences with 14 FPS containing only pedestrians. MOTS02-01, MOTS20-02, and MOTS20-09 are taken by static CCTV, and the rest of sequences are taken in human holding moving cam. MOTS20 set are divided into 4 training sequences with 2,862 images and 4 test sequences with 3,044 images. On the other hand, KITTI-MOTS provides image sequences taken in the camera on a vehicle with 1224 × 370, 1238 × 374, and 1242 × 375 resolutions and 10 FPS, which are divided into 21 training sequences with 8,008 images and 29 test sequences with 11,095 images. Pedestrians and cars appear in the KITTI-MOTS scenes. The ablation studies and experimental results using these datasets are presented in Tables 5, 4, 7, and 6 of Section IV. For evaluation, sMOTSA and IDS are mainly used in this paper.

These measures are mask-based variants of the original CLEAR MOT measures [47] as follows:

$$\text{MOTSA} = \frac{|\text{TP}| - |\text{FP}| - |\text{IDS}|}{|\text{M}|}, \tag{39}$$

TABLE 3. Threshold settings for "Mask Merging" and "Mask-Based Affinity Fusion (MAF)."

Symbol	Description	Value
β	framewise motion update ratio in S2TA	ped: 0.4 (MOTS20) car: 0.4, ped: 0.5 (KITTI)
t_m	upper threshold for mask merging	ped: 0.3 (MOTS20) car: 0.3, ped: 0.4 (KITTI)
f_{pm}	upper threshold for A_{pm} in MAF for preventing A_{maf} to be zero in (35)	ped: 10^{-39} (MOTS20) car & ped: 10^{-39} (KITTI)
f_{appr}	upper threshold for A_{appr} in MAF	ped: 0.5 (MOTS20) car & ped: 0.85 (KITTI)

$$\widetilde{\text{TP}} = \sum_{h \in \text{TP}} \text{Mask IoU}(h, gt(h)), \tag{40}$$

$$\text{sMOTSA} = \frac{\widetilde{\text{TP}} - |\text{FP}| - |\text{IDS}|}{|\text{M}|}, \tag{41}$$

where M is a set of ground truth (GT) pixel masks, h is a track hypothesis mask, and $gt(h)$ is the most overlapping mask among all GTs. In multi-object tracking and segmentation accuracy (MOTSA), a mask-based variant of the original multi-object tracking accuracy (MOTA), a case is only counted as a true positive (TP) when the mask IoU value, between h and $gt(h)$, is greater than or equal to 0.5, but in soft multi-object tracking and segmentation accuracy (MOTSA), $\widetilde{\text{TP}}$ is used, which is a soft version of TP. Other details of the measures are displayed in Table 2.

B. IMPLEMENTATION DETAILS

1) DEVELOPMENT ENVIRONMENTS

All experiments are conducted on an Intel i7-7700K CPU @ 4.20GHz, DDR4 32.0GB RAM, and Nvidia GTX 1080 Ti. We implement MAF_HDA by using OpenCV image processing libraries written in Visual C++. The official code implementation is available at the Github repository .[1]

2) PARAMETER SETTINGS

The matrices F, Q, P, R, and H are used in *Initialization*, *Prediction*, and *Update* for the GMPHD filter's tracking process. Experimentally, the parameter matrices are set as:

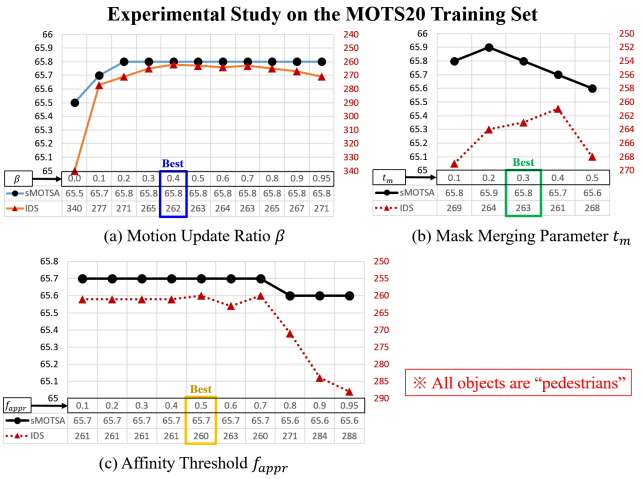
$$F = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$Q = \frac{1}{2}\begin{pmatrix} 5^2 & 0 & 0 & 0 \\ 0 & 10^2 & 0 & 0 \\ 0 & 0 & 5^2 & 0 \\ 0 & 0 & 0 & 10^2 \end{pmatrix},$$

$$P = \begin{pmatrix} 5^2 & 0 & 0 & 0 \\ 0 & 10^2 & 0 & 0 \\ 0 & 0 & 5^2 & 0 \\ 0 & 0 & 0 & 10^2 \end{pmatrix}, \quad R = \begin{pmatrix} 5^2 & 0 \\ 0 & 10^2 \end{pmatrix},$$

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

[1]https://github.com/SonginCV/MAF_HDA



(a) Motion Update Ratio β (b) Mask Merging Parameter t_m

(c) Affinity Threshold f_{appr}

FIGURE 8. Experimental studies for the parameters in the MOTS20 training set. The best sMOTSA and IDS scores are shown when β, t_m, and f_{appr} are (a) 0.4, (b) 0.3, and (c) 0.5 for pedestrians. The same values are set for the test which are presented in Tables 4 and 6.

TABLE 4. Evaluation results on the MOTS20 training set. *p*1 is the baseline method and *p*6 is selected as a final model.

Proposed Trackers	MOTS20 Training Set			
	Pedestrians			
	sMOTSA↑	MOTSA↑	IDS↓	FPS↑
$p1$	64.5	75.9	686	**139**
$p2$	64.5	75.9	535	7.3
$p3$	64.6	75.9	565	7.2
$p4$	65.0	76.3	539	7.0
$p5$	65.6	**77.1**	265	4.2
***p*6**	**65.8**	**77.1**	**234**	5.2
***p*7**	**65.8**	**77.1**	**234**	2.2
***p*8**	**65.8**	**77.1**	244	1.9

We uniformly truncate the segmentation results under threshold values, which are 0.6 for cars and 0.7 for pedestrians.

C. EXPERIMENTAL STUDIES ON KEY PARAMETERS

Figures 9 and 8 address experimental studies on key parameters of our method and show that the parameters such as β (framewise motion update ratio in S2TA), t_m (upper threshold for mask merging with mask IoU), and f_{appr} (upper threshold for A_{appr} in MAF) can be tuned by simple numerical studies. In Figure 9, comparing (c) to (e) and (d) to (f), mask IoU is less sensitive to parameter settings and shows better sMOTSA and IDS than IoU. The final parameter settings are summarized in Table 3 whos values are learned in MOTS20 and KITTI-MOTS training sets and are identically set for evaluation in the training and test sets as shown in Tables 5, 4, 7, and 6.

D. ABLATION STUDIES

For the ablation studies, MAF_HDA is evaluated on the training sequences of MOTS20 and KITTI-MOTS.

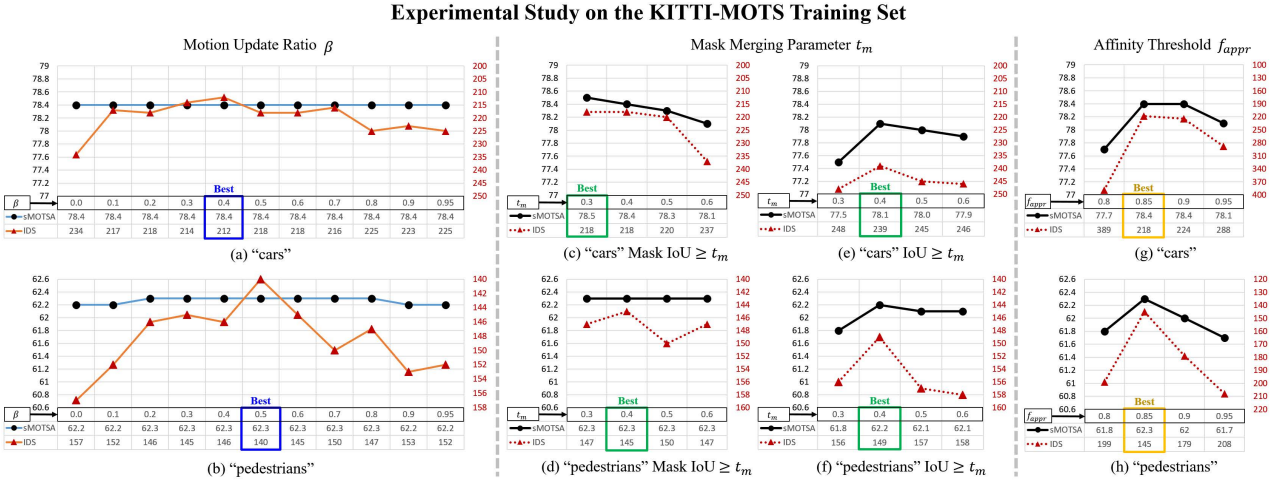


FIGURE 9. Experimental studies for parameters β, t_m, and f_{appr} in the KITTI-MOTS training set. The best sMOTSA and IDS scores are shown when β is (a) 0.4 and (b) 0.5 for cars and pedestrians, respectively. The best scores are observed setting t_m to (c) 0.3 and (d) 0.4 with the Mask IoU measure and setting f_{appr} to (g) 0.85 and (h) 0.85. The same values are set for the test which are presented in Tables 5 and 7.

TABLE 5. Evaluation results on the KITTI-MOTS training set. $p1$ is the baseline method without any proposed modules. KCF2 indicates a simplified version for MOT which uses fixed-size window instead of multi-scale windows used in the referenced version of KCF. We select $p6$ as a final model.

Proposed Trackers	Modules				KITTI-MOTS Training Set						
	S2TA	Mask Merging		T2TA	Cars			Pedestrians			FPS↑
	MAF	IoU	Mask IoU	MAF	sMOTSA↑	MOTSA↑	IDS↓	sMOTSA↑	MOTSA↑	IDS↓	
$p1$: baseline					73.7	84.0	1322	56.4	71.2	800	**233**
$p2$: MAF$_{KCF}$	✓				76.3	86.6	642	59.6	74.5	428	6.1
$p3$: MAF$_{KCF+IoU}$	✓	✓			76.8	86.5	598	59.5	74.3	429	6.2
$p4$: MAF$_{KCF+mIoU}$	✓		✓		77.0	86.7	581	59.6	74.4	423	6.1
$p5$: MAF_HDA$_{KCF}$	✓		✓	✓	77.8	87.2	362	61.2	75.5	245	4.6
$p6$: MAF_HDA$_{KCF2}$	✓		✓	✓	**78.5**	**88.1**	**212**	**62.3**	**77.0**	**140**	9.5
$p7$: MAF_HDA$_{Siam}$	✓		✓	✓	77.4	87.1	480	61.7	76.3	211	1.2
$p8$: MAF_HDA$_{DaSiam}$	✓		✓	✓	77.1	86.8	562	61.6	76.3	218	1.2

1) KEY MODULES

As discussed in Section III, our method includes three key modules: HDA, mask merging, and MAF. HDA consists of S2TA and T2TA in order. Then, we can rearrange these modules with "MAF in S2TA", "Mask Merging", and "MAF in T2TA" considering serial processes as described in Table 5. Additionally, either IoU (37) or Mask IoU (38) for "Mask Merging" can be selected.

2) EFFECTIVENESS OF THE KEY MODULES

As seen in Tables 5 and 4, when the key modules "MAF in S2TA", "Mask Merging", and "MAF in T2TA" are added to the baseline method $p1$ one by one, our method shows incremental and remarkable improvements. Comparing $p1$ and $p2$, $p1$ exploits one-step GMPHD filtering in computing only position and motion affinity, but $p2$ considers the position-motion affinity with the GMPHD filter and appearance affinity by the KCF in "MAF in S2TA". The remarkable improvements in IDS and FM indicate that the proposed affinity fusion method works effectively. Comparing $p2$ and $p3$ in both Tables, because the results are advanced only in KITTI-MOTS, "Mask Merging" may merge more than two segments of one object into one segment or not. However, we can see that at least Mask IoU works better than IoU in the merging of the results of $p3$ and $p4$. In $p5$ and $p6$, "MAF in T2TA" is applied to our method, where KCF extracts the appearance affinities by using multi-scale windows, but KCF2 uses fix-size window to rely on the object sizes from instance segmentation responses. In addition, we apply two more state-of-the-art SOT methods: SiamRPN [24] and DaSiamRPN [25], in the proposed appearance affinity model. In MOTS20, Table 4, $p6$, $p7$, and $p8$ show comparative performances, but in KITTI-MOTS, Table 5, $p7$ and $p8$ show worse than $p6$ and even worse than $p5$ and $p6$ for cars. Figure 10 shows that reason. SiamRPN shows a biased correlation so DaSiamRPN shows too wide correlation in appearance affinity space for cars. We think that is because cars are hard to be discriminated especially in relatively low-resolution of KITTI-MOTS images and tiny-size of objects. On the other hand, KCF's moderate correlation can be appropriate to be fuses with GMPHD filter seeing the better performance. Moreover, since [24], [25] exploit the siamese network which requires GPU processing, those methods cannot extract appearance affinities for dozens of objects in data association steps in parallel with one single GPU. Thus, even if they presented 100 FPS in SOT, $p6$ and $p7$ run at 1.0-2.0 FPS. Comparing the settings without T2TA, from

TABLE 6. Evaluation results on the MOTS20 test set. Proposed methods are denoted by MAF_HDA. The red and blue results indicate the first and second best scores among online processing (proc.) approaches. The bold results indicate the best scores among offline proc. approaches. "not available (n/a)" FPS indicates the case that only total FPS is provided. '-' denotes unpublished results in their paper and the MOTS20 leaderboard at https://motchallenge.net/results/MOTS/.

Trackers	Proc.	Stage	Instance Segmentation	Seg. FPS↑	MOTS20 Test Set				Total FPS
					Pedestrians			Trck. FPS↑	
					sMOTSA↑	MOTSP↑	IDS↓		
MAF_HDA$_{KCF2}$	online	≥2	MaskRCNN X152 [45]	2.5	69.9	84.1	401	4.6	1.6
MAF_HDA$_{KCF}$		≥2	MaskRCNN X152	2.5	69.4	84.2	484	2.6	1.3
MAF_HDA$_{Siam}$		≥2	MaskRCNN X152	2.5	69.4	84.2	453	1.1	0.8
MAF_HDA$_{DaSiam}$		≥2	MaskRCNN X152	2.5	69.4	84.2	442	1.0	0.7
CPPNet [19]		1	CPPNet	n/a	59.3	80.9	484	n/a	7.0
PointTrack [13]		≥2	PointTrack	5.4	58.0	-	-	22.2	4.3
SORTS+RReID [15]		≥2	MaskRCNN X152	2.5	55.8	81.9	304	36.4	2.3
TraDeS [20]		1	TraDeS	n/a	50.8	79.5	492	n/a	11.5
ReMOTS [14]	offline	≥2	MaskRCNN X152	2.5	**70.4**	84.0	**231**	0.30	0.27
UniTrack [18]		≥2	COSTA [48]	2.1	68.9	**84.2**	622	**7.6**	1.6
TrackRCNN [12]		1	TrackRCNN	n/a	40.6	76.1	576	n/a	**2.0**

TABLE 7. Evaluation results on the KITTI-MOTS test set. Proposed methods are denoted by MAF_HDA. The red and blue results indicate the first and second best scores among online processing (proc.) approaches. The bold results indicate the best scores among offline proc. approaches. "not available (n/a)" FPS indicates the cases that only total FPS is provided. All entries are available at http://www.cvlibs.net/datasets/kitti/old_eval_mots.php.

Trackers	Proc.	Stage	Detection & Instance Segmentation	Det.& Seg. FPS↑	KITTI-MOTS Test Set							Total FPS
					Pedestrians			Cars			Trck. FPS↑	
					sMOTSA↑	MOTSP↑	IDS↓	sMOTSA↑	MOTSP↑	IDS↓		
CPPNet [19]	online	1	CPPNet	n/a	70.5	82.4	254	84.5	89.7	202	n/a	7.0
MAF_HDA$_{KCF2}$		≥2	MaskRCNN X152 [45]	3.8	65.0	82.3	301	77.2	88.4	415	10.9	2.8
MAF_HDA$_{KCF}$		≥2	MaskRCNN X152	3.8	64.9	82.3	348	76.5	88.4	475	6.3	2.4
MAF_HDA$_{Siam}$		≥2	MaskRCNN X152	3.8	64.6	82.3	381	75.8	88.4	892	1.5	1.1
MAF_HDA$_{DaSiam}$		≥2	MaskRCNN X152	3.8	64.5	82.3	402	75.6	88.4	968	1.4	1.0
PointTrack [13]		≥2	PointTrack	5.4	61.5	82.4	632	78.5	87.1	114	22.2	4.3
MOTSFusion [16]		≥2	car: [40] & BB2Seg [41] + ped: [12] & [41]	n/a	58.7	81.5	279	75.0	89.3	201	n/a	2.3
EagerMOT [17]		≥2	2D: RRC [40] & [12] + 3D: PointGNN [10]	2.0	58.1	81.5	270	74.5	89.6	457	90.9	1.96
ReMOTS [14]	offline	≥2	MaskRCNN X152	3.8	**66.0**	**82.0**	**391**	**75.9**	**88.2**	716	0.30	0.28
TrackRCNN [12]		1	TrackRCNN [12]	n/a	47.3	74.6	481	67.0	85.1	**692**	n/a	**2.0**

$p2$ to $p4$, and with T2TA, $p5$, $p6$, $p7$, and $p8$, the results show that HDA with MAF reduces IDS very effectively in both datasets and KCF2, the simplified version of KCF, shows faster FPS and better performance in terms of sMOTSA, MOTSA, IDS than the conventional KCF and the state-of-the-art SOT methods [24], [25].

Numerically, when adding the key modules "MAF in S2TA", "Mask Merging", and "MAF in T2TA" one by one, as shown in Tables 5 and 4, and Figure 12, our MOTS method shows incremental improvements from $p1$ to $p6$. The baseline method $p1$ is numerically improved as follows: for the KITTI-MOTS Cars training set, sMOTSA changes from 73.7 to 78.5 and IDS changes from 1,322 to 212; for the KITTI-MOTS Pedestrians training set, sMOTSA changes from 56.4 to 62.3 and IDS changes from 800 to 140; and for the MOTS20 training set, sMOTSA changes from 64.5 to 65.8 and IDS changes from 686 to 234. Thus, $p6$:MAF_HDA$_{KCF2}$ are selected as our final model.

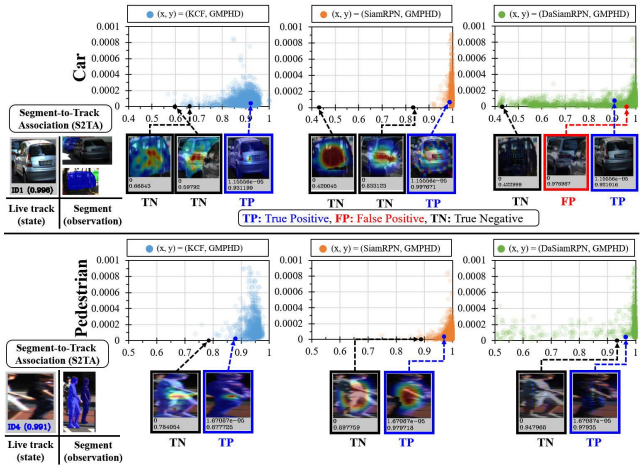


FIGURE 10. Correlation maps between appearance affinities: KCF, SiamRPN, and DaSiamRPN, and position-motion affinity: GMPHD, in KITTI-MOTS.

E. TEST RESULTS

We evaluate the proposed MOTS method against state-of-the-art MOTS methods [12]–[20] in the test set of the MOTS20 and KITTI-MOTS benchmarks. Tables 6 and 7 show the evaluation results and Figure 11 describes comparisons of speed (FPS) vs. MOTS accuracy (sMOTSA and IDS) where our method is denoted by MAF_HDA.

1) SPEED COMPARISON W/SEGMENTATION

For fair comparison of one-stage MOTS methods [12], [19], [20] and multi-stage methods [13]–[18], we present not

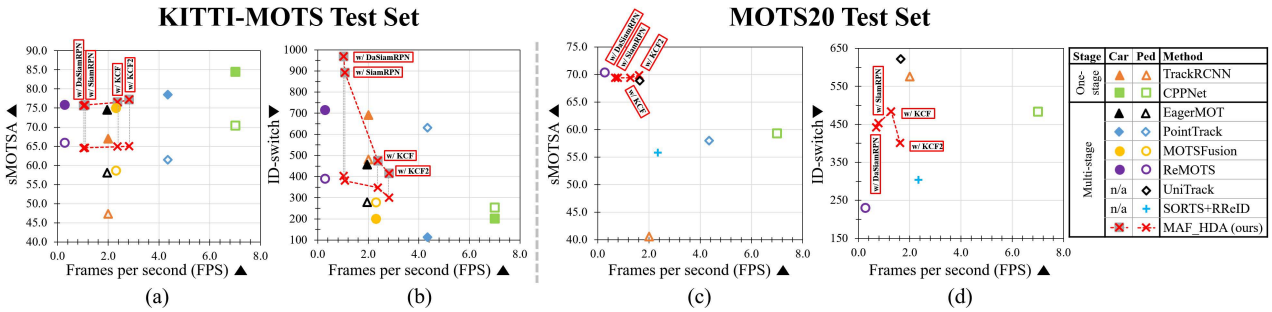


FIGURE 11. Comparisons of speed (FPS) vs. MOTS accuracy (sMOTSA and IDS) against state-of-the-art methods in KITTI-MOTS and MOTS test sets.

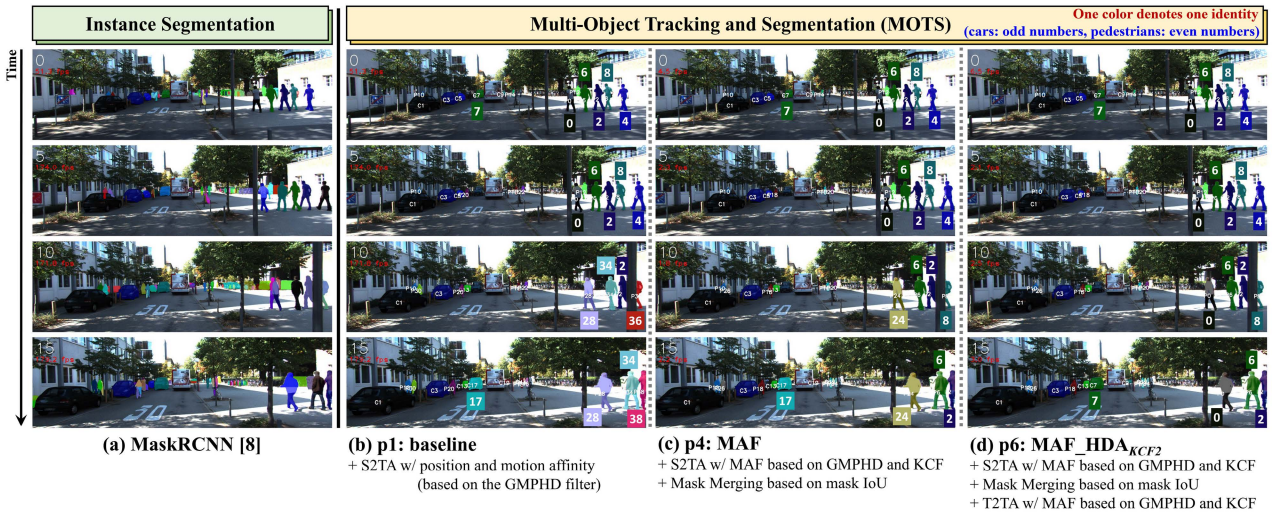


FIGURE 12. Visualization of the segmentation and MOTS results on KITTI-MOTS test sequence 0018. (b), (c), and (d) are the results of the three different settings of the proposed method, which are based on the same segmentation results (a) from MaskRCNN [11]. Comparing (b) the baseline model *p*1 and (c) the model *p*4 with S2TA and mask merging (without T2TA), in (b), the IDs, "0, 2, 4, 6, 8", of the five pedestrians at the right side of the scene are switched except the person w/ ID 2, but, in (c), only the pedestrian w/ ID 0 gets switched to ID 24. In (d) the final model *p*6, the five IDs are preserved since T2TA can find the IDs after occlusion with trees at the right side. In addition, the car w/ ID 7 at frame 0 are recovered at frame 15, while (b) and (c) do not recover the car w/ ID 7 that are switched to ID 17.

only tracking speed but also detection and segmentation speed in Tables 6 and 7. Speed of the public segmentation, Mask RCNN X152, and speed of private segmentation, PointTrack, are measured in our environment presented in Subsection IV-B. Other speeds are referenced from their paper. PointTrack [13], SORTS+RReID [15], and EagerMOT [17] introduce fast tracking speeds 22.2, 36.4, and 90.9 FPS, respectively, but including detection and segmentation, the speeds drop to 4.3, 2.3, and 1.96 FPS. Likewise multi-detector fusion based methods EagerMOT and MOTSFusion [16] show similar speed 1.96 and 2.3 FPS. Among those state-of-the-art methods, one-stage methods CPPNet [19] and TraDeS [20] show faster speeds 7.0 FPS and 11.5 FPS respectively, but still not enough to achieve realtime speeds which are 30 FPS for MOTS20 and 10 FPS for KITTI-MOTS. Among the proposed methods, MAF_HDA$_{KCF2}$ achieves 4.6 FPS in MOTS20 and 10.9 FPS in KITTI-MOTS with tracking only, and 1.6 FPS in MOTS20 and 2.8 FPS in KITTI-MOTS with segmentation and tracking. Compared to others, MAF_HDA$_{KCF2}$ shows moderate speeds (see Figure 11(a)-(b)). Therefore, efficiency of MOTS method in terms of speed versus accuracy is still a challenging issue.

2) ACCURACY COMPARISON W/SOTA METHODS

Some state-of-the-art methods [13], [16], [17], [19] have tackled raising the detection and segmentation quality, EagerMOT and MOTSFusion utilized fusion of multi-detector from multi-domain, and PointTrack and CPPNet focus on learning segmentation model from scratch of MOTS20 and KITTI-MOTS training sets. The former has promising performance since that multi-source detectors can complement each other. However it inevitably needs heavy computing resources. The latter can achieve fine performance as seen in Table 7 if fine data such as over 8,000 images with uniformed resolutions of the KITTI-MOTS training set is given as seen in Table 1. However, in 2,862 images of MOTS20 training set with various resolutions like 1920 × 1080 and 640 × 480, their MOTS accuracy drops sharply contrast to Mask RCNN based methods such as [14] and MAF_HDA (see Figure 11(a) and (c)).

To summarize numerically, we refer to Tables 6 and 7. First, in particular, comparing the variants of MAF_HDA with KCF, KCF2, SiamRPN, and DaSiamRPN, MAF_HDA$_{KCF2}$ shows the best performance in terms of speed and accuracy. MAF_HDA$_{Siam}$ and MAF_HDA$_{DaSiam}$

show drastic speed drop compared to MAF_HDA$_{KCF}$ and MAF_HDA$_{KCF2}$ Those results follow the evaluation results in the training sets (see Tables 5 and 4). Against state-of-the-art MOTS methods [12]–[20], our proposed method named MAF_HDA$_{KCF2}$ ranks 2^{nd} sMOTSA score (1^{st} among the online approaches), 69.9, in the MOTS20 test set. In addition, MAF_HDA$_{KCF2}$ ranks 3^{rd} sMOTSA score, 65.0, for pedestrians and 3^{rd} sMOTSA score, 77.2, for cars in the KITTI-MOTS test set.

V. CONCLUSION

In this paper, we propose a highly feasible MOTS method named MAF_HDA, which is an easily reproducible reassembly of four key modules: a GMPHD filter, HDA, mask merging, and MAF. These key modules can operate in the proposed fully online MOTS framework which tracks cars and pedestrians in parallel CPU-only processes. In addition, the key parameters can be simply tuned through experimental studies adjusting the values in 0.0 to 1.0 ranges, and these modules show remarkable improvements in evaluation on the training sets of MOTS20 and KITTI-MOTS in terms of MOTS measures such as sMOTSA and IDS. In the test sets of the two popular datasets, MAF_HDA achieves very competitive performance against the state-of-the-art MOTS methods. In future work, we expect that the proposed MOTS method will be reproduced and extended in research community with a more precise and simpler position and motion filtering model and more rapid and sophisticated appearance feature extractors such as deep neural network-based re-identification techniques.

ACKNOWLEDGMENT

This work was performed based on the cooperation with GIST-LIG Nex1 Cooperation and was supported by Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2014-3-00077-008, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis).

REFERENCES

[1] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a benchmark for multi-target tracking," 2015, *arXiv:1504.01942*.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Sep. 2012, pp. 3354–3361.

[3] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," vol. 2016, *arXiv:1603.00831*.

[4] L. Chen, S. Lin, X. Lu, D. Cao, H. Wu, C. Guo, C. Liu, and F.-Y. Wang, "Deep neural network based vehicle and pedestrian detection for autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3234–3246, Jun. 2021.

[5] D. Feng, C. Haase-Schutz, L. Rosenbaum, H. Hertlein, C. Glaser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[6] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2015, pp. 91–99.

[9] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D Object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.

[10] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1711–1719.

[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[12] P. Voigtlaender, M. Krause, A. Ošep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "MOTS: Multi-object tracking and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7942–7951.

[13] Z. Xu, W. Zhang, Z. Tan, W. Yang, H. Huang, S. Wen, E. Ding, and L. Huang, "Segment as points for efficient online multi-object tracking and segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 264–281.

[14] F. Yang, X. Chang, C. Dang, Z. Zheng, S. Sakti, S. Nakamura, and T. Wu, "ReMOTS: Self-supervised refining multi-object tracking and segmentation," 2020, *arXiv:2007.03200*.

[15] M. Ahrnbom, M. Nilsson, and H. Ardö, "Real-time and online segmentation multi-target tracking with track revival re-identification," in *Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, Feb. 2021, pp. 777–784.

[16] J. Luiten, T. Fischer, and B. Leibe, "Track to reconstruct and reconstruct to track," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1803–1810, Apr. 2020.

[17] A. Kim, A. Osep, and L. Leal-Taixe, "EagerMOT: 3D multi-object tracking via sensor fusion," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11315–11321.

[18] Z. Wang, H. Zhao, Y.-L. Li, S. Wang, P. H. Torr, and L. Bertinetto, "Do different tracking tasks require different appearance models?" in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2021, pp. 15323–15332.

[19] Z. Xu, A. Meng, Z. Shi, W. Yang, Z. Chen, and L. Huang, "Continuous copy-paste for one-stage multi-object tracking and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15323–15332.

[20] J. Wu, J. Cao, L. Song, Y. Wang, M. Yang, and J. Yuan, "Track to detect and segment: An online multi-object tracker," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12352–12361.

[21] A. Hu, A. Kendall, and R. Cipolla, "Learning a spatio-temporal embedding for video instance segmentation," 2019, *arXiv:1912.08969*.

[22] B. N. Vo and W. K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4091–4104, Nov. 2006.

[23] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[24] B. Li, W. Wu, Z. Zhu, and J. Yan, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8971–8980.

[25] Z. Zhu, Q. Wang, and B. Li, "Distractor-aware Siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Dec. 2018, pp. 101–117.

[26] R. P. S. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

[27] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo implementation of the PHD filter for multi-target tracking," in *Proc. 6th Int. Conf. Inf. Fusion (ICIF)*, Jul. 2003, pp. 792–799.

[28] Y.-M. Song, K. Yoon, Y.-C. Yoon, K. C. Yow, and M. Jeon, "Online multi-object tracking with GMPHD filter and occlusion group management," *IEEE Access*, vol. 7, pp. 165103–165121, 2019.

[29] R. Sanchez-Matilla, F. Poiesi, and A. Cavallaro, "Multi-target tracking with strong and weak detections," in *Proc. Eur. Conf. Comput. Vis. Workshops (ECCVW)*, Oct. 2016, pp. 84–99.

[30] R. Sanchez-Matilla and A. Cavallaro, "A predictor of moving objects for first-person vision," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2189–2193.
[31] T. Kutschbach, E. Bochinski, V. Eiselein, and T. Sikora, "Sequential sensor fusion combining probability hypothesis density and kernelized correlation filters for multi-object tracking in video data," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.
[32] Z. Fu, P. Feng, F. Angelini, J. Chambers, and S. M. Naqvi, "Particle PHD filter based multiple human tracking using online group-structured dictionary learning," *IEEE Access*, vol. 6, pp. 14764–14778, 2018.
[33] M. P. Muresan and S. Nedevschi, "Multi-object tracking of 3D cuboids using aggregated features," in *Proc. IEEE 15th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2019, pp. 11–18.
[34] W. Li, J. Mu, and G. Liu, "Multiple object tracking with motion and appearance cues," *IEEE Access*, vol. 7, pp. 104423–104434, 2019.
[35] Y.-C. Yoon, D. Y. Kim, Y.-M. Song, K. Yoon, and M. Jeon, "Online multiple pedestrians tracking using deep temporal appearance matching association," *Inf. Sci.*, vol. 561, pp. 326–351, Jun. 2021.
[36] K. Yoon, D. Y. Kim, Y.-C. Yoon, and M. Jeon, "Data association for multi-object tracking via deep neural networks," *Sensors*, vol. 19, pp. 1–15, Jan. 2019.
[37] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
[38] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2129–2137.
[39] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2071–2084, Oct. 2015.
[40] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y. Tai, and L. Xu, "Accurate single stage detector using recurrent rolling convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5420–5428.
[41] J. Luiten, P. Voigtlaender, and B. Leibe, "PReMVOS: Proposal-generation, refinement and merging for video object segmentation," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, Dec. 2018, pp. 565–580.
[42] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.
[43] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Dec. 2015, pp. 1–23.
[44] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, Nov. 1987.
[45] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. (2019). *Detectron2*. [Online]. Available: https://github.com/facebookresearch/detectron2
[46] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: HybridBoosted multi-target tracker for crowded scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2953–2960.
[47] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, May 2008.
[48] (2020). *Costa St Tracker*. [Online]. Available: https://motchallenge.net/method/MOTS=87&chl=17

YOUNG-MIN SONG (Graduate Student Member, IEEE) received the B.S. degree in computer science and engineering from Chungnam National University, Daejeon, South Korea, in 2013, and the M.S. degree in information and communications from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2015, where he is currently pursuing the Ph.D. degree in electrical engineering and computer science. His research interests are multi-object tracking and data fusion.

YOUNG-CHUL YOON received the B.S. degree in electronics and communications engineering from Kwangwoon University, Seoul, South Korea, and the M.S. degree in electrical engineering and computer science from the Gwangju Institute of Science and Technology, Gwangju, South Korea, in 2019. He is currently a Software Engineer at Robotics Laboratory, Hyundai Motor Company. His research interests include multi-object tracking and video analysis.

KWANGJIN YOON received the Ph.D. degree in electrical engineering and computer science from the Gwangju Institute of Science and Technology, in 2019. He is currently a Research Scientist with SI Analytics Company Ltd. His research interests include multi-object tracking, computer vision, and deep learning.

HYUNSUNG JANG is currently pursuing the Ph.D. degree in electrical and electronic engineering with Yonsei University, Seoul, South Korea. He is also a Chief Research Engineer with the Department of EO/IR Systems Research and Development, LIG Nex1. He was officially certified as a Software Architect at LG Electronics, in 2020, and completed the Software Architect Course at Carnegie Mellon University. His current research interests include computer vision, deep learning, and quantum computing.

NAMKOO HA received the Ph.D. degree in computer engineering from Kyungpook National University, South Korea, in 2008. He is currently a Chief Research Engineer with the Department of EO/IR Systems Research and Development, LIG Nex1. His current research interest includes developing intelligent EO/IR systems through the use of deep learning.

MOONGU JEON (Senior Member, IEEE) received the B.S. degree in architectural engineering from Korea University, Seoul, South Korea, in 1988, and the M.S. and Ph.D. degrees in computer science and scientific computation from the University of Minnesota, Minneapolis, MN, USA, in 1999 and 2001, respectively.

As a Postgraduate Researcher, he worked on optimal control problems at the University of California at Santa Barbara, Santa Barbara, CA, USA, from 2001 to 2003, and then moved to the National Research Council of Canada, where he worked on the sparse representation of high-dimensional data and the image processing, until 2005. In 2005, he joined the Gwangju Institute of Science and Technology, Gwangju, South Korea, where he is currently a Full Professor with the School of Electrical Engineering and Computer Science. His current research interests include machine learning, computer vision, and artificial intelligence.