Online Multi-Object Tracking and Segmentation with GMPHD Filter and Mask-based Affinity Fusion

Young-min Song, Young-chul Yoon, Kwangjin Yoon, Moongu Jeon*, Seong-Whan Lee, and Witold Pedrycz

Abstract—In this paper, we propose a highly practical 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 a single object tracker such as a kernalized correlation filter. 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 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. They also released a new dataset extended from KITTI [2] and MOTChallenge [3] image sequences. Our proposed MOTS framework is denoted by GMPHD-MAF paradigm has been the mainstream for MOT. Additionally, breakthroughs in object detection have been achieved by many deep neural network (DNN)-based detectors [4]–[7] from various sensor domains, such as color cameras (2D images) and LiDAR (3D point clouds). According to the input source, the detectors give different outputs, i.e., observations. For instance, the detection responses of [5], [6] are 2D bounding boxes and those of [4], [7] are 3D boxes. In addition, K. He et al. [8] introduced a pixelwise classification and detection method represented by instance segmentation, which has motivated many segmentation-based studies.

Recently, a new MOT method extended with segmentation, named MOTS has been developed for pixelwise intelligent systems beyond 2D bounding boxes and was first introduced in Voigtlaender et al. [9] with new evaluation measures and a new baseline method. They also released a new dataset extended from KITTI [2] and MOTChallenge [3] image sequences. Luiten et al. [10] proposed a MOTS method that uses a fusion of 2D box detection, 3D box detection, and instance segmentation results. Motivated by these MOTS works and other conventional MOT studies, in this paper, we propose a highly practical online MOTS method.

Our contributions are summarized as follows:

1) We propose a highly practical online MOTS method consisting of (a) two-stage hierarchical data association (HDA), (b) mask-based affinity fusion (MAF), and (c) mask merging. These key modules can form the proposed online MOTS method with
CPU-only execution.
2) Particularly among the key modules, (b) MAF effectively fuses “position and motion” affinity with a Gaussian mixture probability density filter (GMPHD) and “appearance” with a kernelized correlation filter (KCF) to improve the MOTS performance compared to a baseline method using only one-stage GMPHD filter association.
3) Additionally, the proposed method can be implemented with CPU-only execution 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], [9]. The evaluation results on the training sets show incremental improvements compared to a 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 KITTI-MOTS and MOTSChallenge websites.

The proposed MOTS method has high applicability due to CPU-only execution and simple parameter tuning unlike many state-of-the-art DNN based tracking methods [9]–[16] that need intensive hyperparameter optimization and heavy computing resources. 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 the proposed ones 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 GMPHD-MAF as the abbreviation for the proposed method.

II. RELATED WORKS

A. Multi-Object Tracking with a PHD Filter

The PHD filter [17]–[19] was originally designed to deal with radar and sonar data-based MOT systems. Mahler et al. [17] proposed recursive Bayes filter equations for the PHD filter that optimize multi-target tracking processes in radar and sonar systems with a random-finite set definition of states and observations. Following this PHD filtering theory, Vo et al. [19] proposed a sequential Monte Carlo implementation of the PHD filter by using particle filtering and clustering, named the SMCPHD filter, and implemented the governing equations by using the Gaussian mixture model as a closed-form recursion method named the Gaussian mixture 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. Song et al. [20] extended GMPHD filter-based tracking with a two-stage hierarchical data association scheme to recover lost tracks IDs. They modeled an affinity function in the second stage association between the tracks by using the tracks’ position and linear motion. 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 stage association between detections and tracks. Sanchez-Matilla et al. [21] 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. More intensive solutions [22]–[24] using appearance learning and motion modeling have been proposed. Kutschbach et al. [22] applied kernelized correlation filters (KCF) [25] to the naive GMPHD filtering process for online appearance updating to discriminate occluded objects. They proposed robust online appearance learning to refine the IDs of lost tracks. However, updating the appearance of every object in every frame requires heavy computing resources and inevitably increases the runtime. Fu et al. [23] add an adaptive gating technique and an online group-structured dictionary for appearance learning into the GMPHD filter. They made the GMPHD filter into a sophisticated tracking process suitable for video-based MOT. Sanchez-Matilla et al. [24] proposed a global motion model based on long short-term memory (LSTM) models for MOT. Some methods [26]–[28] have proposed fusion methods to complement false positive and negative detections. Kutschbach et al. [26] developed a fusion model of a blob detector [29] and head detector [30] in the GMPHD filter-based tracking. Fu et al. [27] used a full-body detector [31] and body-part detector [32] in their tracking-by-detection method. Baisa et al. [28] proposed another type of fusion method that tracks multiple types of objects (cars and pedestrians) simultaneously by using an object recognition method such as a faster region-based convolutional neural network (FRCNN) [6] in parallel processes. These online MOT methods based on the PHD filter have successfully improved the tracking performance by using the detector fusion, appearance learning, and motion models.

B. State-of-the-Art MOTS Methods

Conventional MOT methods [20]–[24], [26]–[28], [33]–[40] have exploited the tracking-by-detection paradigm, where the detectors [5], [6], [31], [41], [42] generate 2D bounding box results and the trackers assign tracking identities (IDs) to the bounding boxes via data association. Unlike MOT, MOTS uses pixelwise instance segmentation results as a tracking input instead of 2D bounding box results. P. Voigtlaender et al. [9] first introduced the MOTS task. They extended the popular MOT datasets such as KITTI [2] and MOTChallenge [3] with instance segmentation results by using a fine-tuned MaskRCNN [8] for the same image sequences, and
SONG et al.: ONLINE MULTI-OBJECT TRACKING AND SEGMENTATION WITH GMPHD FILTER AND MASK BASED AFFINITY FUSION

proposed a new soft multi-object tracking and segmentation accuracy (sMOTS) 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 [10], [14], [15] have been proposed very recently. MOTSNet [14] presents an intensive and semiautomated learning strategy for harvested datasets, e.g., ImageNet [43] and Mapillary Vistas [44] to improve instance segmentation quality. Additionally, these methods use a novel mask-pooling layer for improved object association over time. MOTSFusion [10] proposes a fusion-based MOTS method exploiting bounding box detection [45] and instance segmentation [46]. 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 [15] devises a new feature extractor based on PointNet [47] 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.

Motivated by these state-of-the-art MOTS studies [9], [10], [14], [15], we propose a highly practical MOTS method without intensive or additional learning in [14], [47] for more precise pixelwise segmentation and without requiring multi-domain data such as detection, segmentation, and 2D-to-3D reconstruction to perform fusion [10]. Our proposed method, named GMPHD_MAF, exploits the tracking-by-instance-segmentation paradigm, which uses only 2D images and one instance segmentation result as inputs and performs the MOTS task by using two popular filtering methods, the GMPHD filter [18] and the KCF [25], with simple parameter tuning. We build a two-step hierarchical data association strategy to handle tracklet loss and ID switches. In each association stage, position and motion affinity are calculated by the GM-PHD filter, and appearance affinity is calculated by the KCF. To appropriately consider these two affinities, we propose a mask-based affinity fusion model. GMPHD_MAF shows comparative performance in two popular KITTI-MOTS [2] and MOTSChallenge [3] datasets against state-of-the-art MOTS methods [9], [10], [14], [15].

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 and the past times 0 to t – 1 (see Figure 2). In each observation-to-state association stage, 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 [18] and a KCF [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.

![Fig. 2. Processing pipeline of the proposed online multi-object tracking and segmentation framework, where segment-to-track association and track-to-track association are abbreviated as S2TA and T2TA, respectively. As shown in this figure, the fully online approach uses only past and current information, without any intentional frame latency.](image-url)
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 GMPHD_MAF 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:

\[ X_t = \{ x_1^t, \ldots, x_N^t \}, \]  
\[ Z_t = \{ z_1^t, \ldots, z_M^t \}, \]  

where a state vector $x_i$ 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 $z_i$ is composed of $\{c_x, c_y\}$ with a segment mask. The Gaussian model $N$ representing $x_i$ is initialized by $z_i$, predicted to $x_{i+1|t}$, and updated to $x_{i+1}$ by $z_{i+1}$.

Initialization: The Gaussian mixture model $g_i$ 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 $N$ represents a state model with weight $w$, mean vector $m$, input state vector $x$, and covariance matrix $P$, which are as follows:

\[ g_i(x) = \sum_{i=1}^{N_i} w_i^i N(x; m_i^i, P_i^i), \]  

where $N_i$ 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. Additionally, the method of setting covariance matrix $P$ is shown in Section IV-B.

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), $m_{i|t-1}$ 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:

\[ g_{t-1}(x) = \sum_{i=1}^{N_{t-1}} w_{i-1}^i N(x; m_{i-1}^i, P_{i-1}^i), \]  
\[ m_{i|t-1} = F m_{i|t-1}, \]  
\[ P_{i|t-1} = Q + F P_{i|t-1} F^T, \]  

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

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

\[ g_{i|t}(x) = \sum_{i=1}^{N_{t|t}} w_i^t N(x; m_{i|t}, P_i^t), \]  
\[ q_i^t(z) = N(z; H m_{i|t-1} + R + H P_{i|t-1} H^T), \]  

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 used to transform a state vector into an observation vector. Both matrices are constant in our application. After
finding the optimal \( z \), the Gaussian mixture is updated in the order of (9), (10), (11), and (12) as:

\[
\begin{align*}
    w_i^t(z) &= \frac{u_{i|t-1}^t q_i^t(z)}{\sum_{j=1}^{N_{t-1}} u_{i|t-1}^t q_j^t(x)}, \\
    m_i^t(z) &= m_{i|t-1}^t + K_i^t(z - H m_{i|t-1}^t), \\
    P_i^t &= [I - K_i^t H] P_{i|t-1}^t, \\
    K_i^t &= P_{i|t-1}^t H^T (H P_{i|t-1}^t H^T + R)^{-1},
\end{align*}
\]

where the set of \( u_{i|t-1} \) includes \( u_{t-1} \) (weights from the previous frame) and \( u_t \) (weights of newly born targets). Likewise, \( N_{t-1} \) is the sum of \( \bar{N}_{t-1} \) and the number of the newly born targets.

\[ S = \{ s_1, ..., s_k \}, \]

\[ T_l^\text{live} = \{ t_1^\text{live}, ..., t_d^\text{live} \}, \]

\[ T_{t | t}^\text{live} = \{ x_{t, t}^k, x_{t}^k \}, \quad 0 \leq t < t, \quad t = t, \]

\[ T_{t | t}^\text{lost} = \{ x_{t, t}^k, x_{t}^k \}, \quad 0 \leq t < t, \]

\[ x_{t, t} = \{ c_{x, t}, c_{y, t}, v_{x, t}, v_{y, t} \}, \]

\[ x_{t} = \{ c_{x, t}, c_{y, t}, v_{x, t}, v_{y, t} \}, \]

where “live” indicates that tracking succeeds at time \( t \). “lost” indicates that tracking fails at time \( t \). The two attributes are not compatible, and \( T_t^\text{lost} \cup T_t^\text{live} = T_t^\text{all} \) and \( T_t^\text{lost} \cap T_t^\text{live} = \phi \) are satisfied. \( T_t^\text{lost} \) is composed of a track \( \tau_{i,t} \) with identity \( i \) which is also a set of live track vectors from the birth time \( t_b \) to the last tracking time \( t_l \). In the case of \( t_{i,t}^\text{lost} \), \( t_l \) is identical to the present time \( t \), in the case of \( t_{i,t}^\text{lost} \), \( t_l \) is less than time \( t \). Regardless of when time \( t \) is, state vector \( x \) has the center point \( \{ c_{x, t}, c_{y, t} \} \) in the segment bounding box, velocities \( \{ v_{x, t}, v_{y, t} \} \) in the directions of the \( x \)-axis and \( y \)-axis, an identity (ID), and a segment mask (see (18) and (19)).

**Segment-to-track association (S2TA):** In S2TA, the observations denoted by \( Z_{t}^\text{S2TA} \) are frame-by-frame instance segmentation results \( S_i \). If there are no track states, the states \( X_{t}^\text{S2TA} \) are initialized from \( Z_{t}^\text{S2TA} \), and otherwise, \( X_{t}^\text{S2TA} \) is predicted from \( T_{t}^\text{live} \) and updated by using the GMPHD filter with the processing units as follows:

\[
\begin{align*}
    Z_{k,t}^\text{S2TA} &= \{ x_{k,t}, y_{k,t} \}^T \text{ from } k, \\
    X_{t - 1, t}^\text{S2TA} &= \{ x_{t - 1, t}, y_{t - 1, t}, v_{x,t-1, t}, v_{y,t-1, t} \}^T \text{ from } T_{t}^\text{live}, \quad (21) \\
    F^{\text{S2TA}} &= \begin{bmatrix} 1 & 0 & 0 \\
    0 & 1 & 0 \\
    0 & 0 & 1 \end{bmatrix}, \quad (22) \\
    X_{t | t}^\text{S2TA} &= F^{\text{S2TA}} X_{t - 1, t}^\text{S2TA} + \text{noise}, \quad (23) \\
    v_{t | t}^\text{S2TA} &= \begin{bmatrix} \beta \cdot v_{t - 1} + (1 - \beta) \cdot c_{t - 1, t} - c_{t - 1, t} \\
    0, 0 \end{bmatrix}^T, \quad \text{if } Z_{k,t}^\text{S2TA} \text{ is assigned to } X_{t | t}^\text{S2TA} \\
    &\quad \text{else if } X_{t | t}^\text{S2TA} \text{ is born} \quad (25)
\end{align*}
\]

where \( \beta \) can be set differently set according to the scene context and frame rate.

**Track-to-track association (T2TA):** In T2TA, observations \( Z_{t}^\text{T2TA} \) and states \( X_{t}^\text{T2TA} \) (inputs) are built from the live track set \( T_{t}^\text{live} \) and lost track set \( T_{t}^\text{lost} \) respectively. Each of \( T_{t}^\text{live} \) and \( T_{t}^\text{lost} \) consists of the track vectors of \( z_{t | t}^\text{live} \) and \( z_{t | t}^\text{lost} \) 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)).

Unlike the Prediction step of S2TA, where the framewise motion from time \( t-1 \) to \( t \) is used, a trackwise motion analysis is used in T2TA as follows:

\[
x_{i,t}^{T2TA} = \{x_{i,t}^j, y_{i,t}^j\}^T \quad \text{from} \quad \tau_{i,t}^{live},
\]

(26)

\[
x_{j,t-1}^{T2TA} = \{x_{j,t}^j, y_{j,t}^j\}^T \quad \text{from} \quad \tau_{j,t}^{lost},
\]

(27)

\[
F^{T2TA} = \begin{pmatrix} 1 & 0 & d_f & 0 \\ 0 & 1 & 0 & d_f \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},
\]

(28)

\[
x_{j,t}^{T2TA} = F^{T2TA} x_{j,t-1}^{T2TA},
\]

(29)

\[
x_{j,t}^{T2TA} = \{x_{j,t}^j, y_{j,t}^j\}^T,
\]

(30)

where \( d_f(i,j) \) (28) is the frame difference between \( \tau_{i,t}^{live} \)'s first element \( x_{i,t}^j \) (15) and \( \tau_{j,t}^{lost} \)'s last element \( x_{j,t}^j \) (17). The trackwise motion vector \( \{x_{j,t}^j, y_{j,t}^j\}^T \) of (27) is calculated as follows:

\[
\tilde{v}_t^j = \{x_{j,t}^j, y_{j,t}^j\}^T = \left\{ \frac{\bar{c}_{x,t}^j - \bar{c}_{x,t}^j}{t_l - t_b}, \frac{\bar{c}_{y,t}^j - \bar{c}_{y,t}^j}{t_l - t_b} \right\},
\]

(31)

where \( \bar{c}_{x,t}^j \) and \( \bar{c}_{y,t}^j \) are the averaged velocities in the directions of the x-axis and y-axis, respectively. The velocities are computed by subtracting the center position of the first object state \( x_{i,t}^j \) from that of the last state \( x_{j,t}^j \) and dividing it by the frame difference \( t_l - t_b \), which is equivalent to the length of the track \( \tau_{j,t}^{lost} \).

In terms of temporal motion analysis, S2TA has the same time interval “\( t \)” 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. A detailed example is shown in Figure 4.

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.

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 stages, 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 \( w \cdot q(z) \) obtained by the GMPHD filter as follows:

\[
A_{ipm}^{(i,j)} = w \cdot q(z),
\]

(32)

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

Appearance affinity: We exploit the KCF [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 KCF does not have class dependency because it was originally designed for single-object tracking challenges such as the VOT benchmark [48]. 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 KCF, 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 KCF-based affinity can be derived as follows:

\[
A_{appr}^{(i,j)} = 1 - \sum_{r=x_{x}} \sum_{t=y_{y}} \frac{\tilde{d}^{KCF}(r,c)}{width_{j} \cdot height_{j}},
\]

(33)

where \( \tilde{d}(\cdot) \) indicates the normalized KCF distance value, which varies from 0.0 to 1.0 at each pixel. More intensive and advanced single-object tracking and template matching methods such as SiamRPN [49] and reidentification [50] can
be adopted in computing this affinity but it is beyond the scope of this paper.

Min-max normalization: In our experiments, \(A_{pmn}\) and \(A_{appr}\) have quite different magnitudes, e.g., \(A_{pmn} = \{10^{-9}, \ldots, 10^{-3}\}\) and \(A_{appr} = \{0.4, \ldots, 1.0\}\) (see Figures 5-7). To fuse two affinities, we apply min-max normalization to them as follows:

\[
A^{(i,j)} = \frac{A^{(i,j)} - \min_{1 \leq i \leq N} A^{(i,j)}}{\max_{1 \leq i \leq N} A^{(i,j)} - \min_{1 \leq i \leq N} A^{(i,j)}}.
\]  

(34)

Then, we finally propose a MAF model represented by:

\[
A_{maf}^{(i,j)} = A_{pmn}^{(i,j)} A_{appr}^{(i,j)}.
\]  

(35)

From this fused affinity, we can compute the final cost between states and observations as follows:

\[
Cost(x_t^i, z_{t-1}^j) = -\alpha \cdot \ln A_{maf}^{(i,j)}
\]  

(36)

where \(\alpha\) 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 [51] to solve the cost matrices, 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.

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:

\[
\text{IoU}(A, B) = \frac{\text{bbox}(A) \cap \text{bbox}(B)}{\text{bbox}(A) \cup \text{bbox}(B)},
\]  

(37)

\[
\text{Mask IoU}(A, B) = \frac{\text{mask}(A) \cap \text{mask}(B)}{\text{mask}(A) \cup \text{mask}(B)}.
\]  

(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. Thus, if the instance segmentation module provides two or more object classes, e.g., car and pedestrian classes, our proposed framework is easily expandable (see Figure 1). In this paper, we implement the MOTS module with two parallel MOTS processes because the datasets used for our experiments produce car and pedestrian segments.

IV. EXPERIMENTS

In this section, we present experimental studies for the proposed MOTS method, named GMPHD_MAF, in detail. In IV-A we note that GMPHD_MAF is studied with state-of-the-art KITTI-MOTS [2] and MOTSChallenge [9] 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-D, we determine the effectiveness of key modules through ablation studies in the dataset training sequences. The ablation studies show that the proposed key modules comprehensively improve the baseline model \(p1\) remarkably in terms of IDS. In particular, the proposed MAF technique effectively fuses “position and motion (GMPHD)” and “appearance (KCF)” affinities, which are described in Figures 6 and 7. Finally, in IV-E, we show that the final proposed model \(p6\) achieves comparable performances on the test sequences of the datasets in terms of the smOTS, MOTSA, MOTSP, and IDS measures.

A. Datasets and Measures

GMPHD_MAF is evaluated on KITTI-MOTS [2] and MOTSChallenge [9], which are the most popular datasets for MOTS. P. Voigtlaender et al. [9] proposed new MOTS measures and two MOTS datasets that were extended from two representative MOTSChallenge [3] and KITTI [2] datasets. 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 R-CNN [8] X152 of Detectron2 [53]. For evaluation, smOTS and IDS are mainly used in this paper. These measures are mask-based variants of the original

| Measure | Better | Perfect |
|---------|-------|---------|
| MOTSA   | ↑      | 100%    |
| sMOTS   | ↑      | 100%    |
| MOTSP   | ↑      | 100%    |
| TP      | ↓      | 0       |
| FP      | ↓      | 0       |
| FN      | ↓      | 0       |
| IDS     | ↓      | 0       |
| FPS     | ↑      | ∞       |

TABLE I

EVALUATION MEASURES. sMOTS HAS BEEN MAINLY USED FOR MEASURING THE TRACKING PERFORMANCE AS A KEY MEASURE.
Fig. 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 \( \bar{m} \) and standard deviation \( \bar{\sigma} \), 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).

CLEAR MOT measures [54] as follows:

\[
\text{MOTSA} = \frac{|TP| - |FP| - |IDS|}{|M|},
\]

(39)

\[
\bar{TP} = \sum_{h \in TP} \text{Mask IoU}(h, g(t(h))),
\]

(40)

\[
\text{sMOTSA} = \frac{\bar{TP} - |FP| - |IDS|}{|M|},
\]

(41)

\[
\text{MOTSP} = \frac{\bar{TP}}{|TP|},
\]

(42)

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

B. Implementation Details

Development environments: All experiments are conducted on an Intel i7-7700K CPU @ 4.20GHz and DDR4 32.0GB RAM without GPU acceleration. We implement GMPHD_MAF by using OpenCV image processing libraries written in Visual C++. The code implementation is available at https://github.com/SonginCV/GMPHD_MAF.

Parameter settings: The matrices \( F, Q, P, R, \) and \( H \) are used in Initialization, Prediction, and Update. Experimentally, the parameter matrices for the GMPHD filter’s tracking process are set as follows:

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

(52)

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

\[
P = \begin{pmatrix}
5^2 & 0 & 0 & 0 \\
0 & 10^2 & 0 & 0 \\
0 & 0 & 5^2 & 0 \\
0 & 0 & 0 & 10^2
\end{pmatrix},
\]

\[
R = \begin{pmatrix}
5^2 & 0 \\
0 & 10^2
\end{pmatrix},
\]

(52)

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

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

C. 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, \( A_{gmphd} = \{10^{-9}, \ldots, 10^{-3}\} \) and \( A_{appr} = \{0.4, \ldots, 1.0\} \) are observed. Additionally, 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 consider these two characteristics, we propose MAF; from the distributions of normalized affinities $A_{gmphd}$ and $A_{kcf}$ 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.

D. Ablation Studies

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

**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 II. Additionally, we can select either IoU (37) or Mask IoU (38) for “Mask Merging”.

**Experimental Studies on Key Parameters:** We address experimental studies on key parameters in Appendix C, and the parameter settings are summarized in Table IV.

**Effectiveness of the key modules:** As seen in Tables II and III, 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 p5 computes the appearance affinities for bounding box pixels, but p6 considers pixelwise mask-based foreground by setting the background pixels to zeros. Comparing the settings without T2TA, from p2 to p4, and with T2TA, p5 and p6, the results of p5 and p6 show that HDA with MAF reduces IDS very effectively in both datasets and mask-based affinity fusion works better than bounding box-based affinity fusion.

In summary, when adding the key modules “MAF in S2TA”, “Mask Merging”, and “MAF in T2TA” one by one, as shown in Tables II and III, and Figure 11, 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 299. In addition, state-of-the-art 2D MOT benchmarks.Each dataset has a training set, including a

| Tracks | MOTSChallenge Training Set | KITTI-MOTS Training Set |
|--------|---------------------------|-------------------------|
|        | Pedestrians               | Cars                    |
|        | sMOTSA†                  | sMOTSA†                 | MOTS†                  | MOTS†                  | IDS†                  | IDS†                  | FM†                  | FM†                  |
| p1     | 64.5                      | 75.9                    | 686                    | 604                    |
| p2     | 64.5                      | 75.9                    | 555                    | 487                    |
| p3     | 66.6                      | 75.9                    | 562                    | 525                    |
| p4     | 65.0                      | 76.3                    | 539                    | 497                    |
| p5     | 66.6                      | 77.4                    | 535                    | 509                    |
| p6     | 65.8                      | 77.1                    | 262                    | 465                    |

| Symbol | Description | Value |
|--------|-------------|-------|
| $\beta$ | framewise motion update ratio in S2TA | car: 0.4, ped: 0.5 |
| $t_m$ | upper threshold for mask merging | car: 0.3, ped: 0.4 |
| $t_{pm}$ | upper threshold for $A_{pm}$ before MAF | car & ped: 10$^{-3}$ |
| $t_{appr}$ | upper threshold for $A_{appr}$ in MAF | car & ped: 0.85 |
| Trackers (online methods in bold) | Det. | Seg. | KITTI-MOTS Validation Set | KITTI-MOTS Challenge Test Set |
|----------------------------------|------|------|---------------------------|----------------------------|
|                                 |      |      | Cars                      | Pedestrians                |
|                                 | sMOTA↓ | MOTSP↑ | IDS↓ | FPS↑ | sMOTA↓ | MOTSP↑ | IDS↓ | FPS↑ |
| TrackRCNN [9]                   | 67.0 | 85.1 | 692 | 2.0 | 47.3 | 74.6 | 481 | 2.0 |
| MOTSFusion [10]                 | 75.0 | 89.3 | 201 | 2.3 | 58.7 | 81.5 | 279 | 2.3 |
| ReMOTS [16]                     | 75.9 | 88.2 | 716 | 0.3 | 66.0 | 82.0 | 291 | 0.3 |
| PointTrack [15]                 | 78.5 | 87.1 | 114 | 22.2 | 61.5 | 82.4 | 632 | 22.2 |
| GMPHD_MAF (p6)                  | 76.7 | 88.4 | 430 | 7.7 | 65.2 | 82.3 | 277 | 19.4 |

| Trackers (online methods in bold) | MOTSFChallenge Training Set | MOTSFChallenge Test Set |
|----------------------------------|-----------------------------|-------------------------|
|                                 | Pedestrians                 | sMOTA↑ | MOTSA↑ | MOTSP↑ |
| MHT-DAM [55]                    | 48.0 | 62.7 | 79.8 |
| FWT [56]                        | 49.3 | 64.0 | 79.7 |
| MOTDT [57]                      | 47.8 | 61.1 | 80.0 |
| JC [38]                         | 48.3 | 63.0 | 79.9 |
| TrackRCNN [9]                   | 52.7 | 66.9 | 80.2 |
| MOTSFusion [10]                 | 56.8 | 69.4 | 82.7 |
| PointTrack [15]                 | 58.1 | 70.6 | - |
| GMPHD_MAF (p6)                  | 65.8 | 77.1 | 86.1 |

**TABLE V**

Evaluation results on the KITTI-MOTS Validation set. Our method is denoted as GMPHD_MAF. The 1st and 2nd best scores are highlighted in RED and BLUE, respectively. TrackRCNN [9] and MaskRCNN [8] are the public detection & segmentation methods but the others are privately refined segmentation results.

**TABLE VI**

Evaluation results on the KITTI-MOTS and MOTSFChallenge test sets. Ours is denoted by GMPHD_MAF. All methods except PointTrack [15] exploits instance segmentation results of MaskRCNN [8] (public segmentation) as an input. The 1st and 2nd best scores are highlighted in RED and BLUE, respectively.

**TABLE VII**

Evaluation results on MOTSFChallenge training (= validation) set. Ours is denoted by GMPHD_MAF. All methods except [14], [15] exploit the instance segmentation results of MaskRCNN [8] (public segmentation) as an input. The 1st and 2nd best scores are highlighted in RED and BLUE, respectively.

Kitti-MOTS sequences are addressed in Table VIII. Other methods [9], [10], [15], [16] also provide the evaluation results in the test set. In the validation set, the developed method shows the second best sMOTA and MOTSP scores for cars, and the second best sMOTA and the best MOTSP for pedestrians among all methods but the best scores compared to the public detection and segmentation results: TrackRCNN- and MaskRCNN-based MOTS methods [9]–[12] for cars and pedestrians (see Table V). [13]–[15] adapt refinement techniques to TrackRCNN or MaskRCNN. In the test set, our method shows that the second best sMOTA and MOTSA scores among all methods for cars but the best scores against all online methods for pedestrians (see Table VI). If our method shows a lower sMOTA score, than PointTrack [15], 62.4, for pedestrians in the validation, ours achieves a better sMOTA, 65.2, than PointTrack [15], 61.5, in the test.

MOTSFChallenge: This dataset contains only pedestrians, and from the results of Tables VII and VI, we can see that the proposed method is more competitive for pedestrians against state-of-the-art methods [9], [10], [14]–[16], [55]–[58] in the validation set. In the test sequences, our method achieves the second best sMOTA, 69.4, in MOTSFChallenge, and the best is 70.4, achieved by ReMOTS [16]. However, ReMOTS shows a lower sMOTA than ours for cars in KITTI-MOTS (see Table VI). This means that ours is comparable with offline methods.

In summary, referring to Tables V, VII, and VI, our proposed method, named GMPHD_MAF, not only shows the best
sMOTSA scores among online methods using public segmentation without a refinement process but also achieves competitive performance against state-of-the-art online and offline MOTS methods [9]–[16], [55]–[58]. In particular, comparing our method to these methods, relatively better performances are measured for pedestrians than for cars. We infer that this is due to the difference in the rigidity of each object. We think cars have relatively rigid shapes so the tracking performance may be greatly affected by the accuracy of segmentation for position affinity, while pedestrians have nonrigid shapes so appearance affinity may affect the performance more than segmentation does. We discussed this in IV-C. In the KITTI-MOTS test set, GMPHD_MAF (p6) runs at 7.7 FPS for cars and 19.4 FPS for pedestrians when operating separately. When tracking both cars and pedestrians, our method runs at 5.3 FPS with serial processing but 6.3 FPS with parallel processing, which is approximately 20 percent faster than the former. In the MOTSChallenge test set, our method shows the highest speed even though it runs at 2.6 FPS, since the test set consists of three high-resolution scenes (1920x1080) and one normal-resolution scene (640x480), as described in Table VIII.

V. CONCLUSIONS

In this paper, we propose a highly practical MOTS method named GMPHD_MAF, which is a feasible and easily reproducible combination 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. These modules show incremental improvements in evaluation on the training sets of KITTI-MOTS and MOTSChallenge in terms of MOTS measures such as sMOTSA, MOTSA, IDS, FM, and FPS. In the validation and test sets of the two popular datasets, GMPHD_MAF 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 with a more precise or simpler position and motion filtering model and more rapid or sophisticated appearance feature extractors such as deep neural network-based re-identification techniques.

APPENDIX A

DATASET SPECIFICATIONS

Table VIII describes the KITTI-MOTS and MOTSChallenge benchmark datasets in terms of training, validation (Valid), and test sequences, frames per second (FPS), resolution, and the number of frames (Frame). The ablation studies and experimental results using these datasets are presented in Tables II, III, V, VI, and VII of Section IV.

APPENDIX B

BENCHMARK LEADERBOARDS

The full benchmark results, including published and unpublished methods are available at the online leaderboards below.

KITTI-MOTS: http://www.cvlibs.net/datasets/kitti/old_eval_mots.php
MOTSChallenge: https://motchallenge.net/results/MOTS/

APPENDIX C

EXPERIMENTAL STUDIES ON THE PARAMETERS

Experimental studies of key parameters of our method are presented in Figure 8, 9, and 10. The parameters are addressed in Section III and summarized in Table IV.
Experimental Study on the Mask Merging Parameter $f_m$

(a) “Cars” Mask IoU $\geq f_m$

(b) “Cars” Mask IoU $\geq f_m$

(c) “Pedestrians” Mask IoU $\geq f_m$

(d) “Pedestrians” Mask IoU $\geq f_m$

Fig. 9. The mask merging threshold $f_m$ is presented in Subsection III-D. In the above experiments on the KITTI-MOTS training set, using mask IoU (a)(c), the best sMOTSA and IDS scores are achieved when $\beta$ is 0.4 and 0.4 for cars and pedestrians, respectively. Using IoU (b)(d), the best sMOTSA and IDS scores are achieved when $\beta$ is 0.4 and 0.4 for cars and pedestrians, respectively. Comparing (a)(c) and (b)(d), mask IoU shows not only less sensitivity to the parameter values but also better performance than IoU. Thus, mask IoU is selected for the mask merging measure, the same values are set for the test and the results are presented in Tables V and VI.

Experimental Study on the Affinity Threshold $f_{appr}$

(a) KITTI-MOTS train “cars”

(b) KITTI-MOTS train “pedestrians”

Fig. 10. The appearance affinity threshold $f_{appr}$ is used for high $A_{appr}$ value filtering before MAF (see (35)). When $A_{appr}$ (33) between a state and an observation is greater than or equal to $f_{appr}$, the pair is considered for association even if $A_{pm}$ is less than $f_{appr}$ (see Table IV). In the above experiments on the KITTI-MOTS training set, the best sMOTSA and IDS scores are achieved when $f_{appr}$ is 0.85 for both cars and pedestrians. The same values are set for the test, and the results are presented in Tables V and VI.

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

[4] 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 Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 697–705.

[5] J. Redemon, 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), Jul. 2016, pp. 779–788.

[6] 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.

[7] 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.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2961–2969.

[9] P. Voigtlaender, M. Krause, A. Oelep, 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.

[10] 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.

[11] A. Osep, W. Meiner, P. Voigtlaender, and B. Leibe, “Track, then decide: Category-agnostic vision-based multi-object tracking,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 3494–3501.

[12] A. Osep, W. Meiner, M. Mathias, and B. Leibe, “Combined image- and world-space tracking in traffic scenes,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2017, pp. 1988–1995.

[13] S. Sharma, J. A.Ansari, J. Krishna Murthy, and K. Madhava Krishna, “Beyond Pixels: Leveraging geometry and shape cues for online multi-object tracking,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2018, pp. 3508–3515.

[14] L. Porzi, M. Hofinger, I. Ruiz, J. Serrat, S. R. Bulo, and P. Kontschieder, “Learning multi-object tracking and segmentation from automatic annotations,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6845–6854.

[15] Z. Xu, W. Zhang, Z. Tan, W. Yang, H. Huang, S. Wen, and E. D. nad 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.

[16] F. Yang, X. Chang, C. Deng, Z. Zheng, S. Sakti, S. Nakamura, and T. Wu, “ReMOTS: Self-supervised refining multi-object tracking and segmentation,” 2020, [Online]. Available: arXiv:2007.03200.

[17] R. P. S. Mahler, “Multitarget Bayes filtering via first-order multitarget association even if $A_{pm}$ is less than $f_{appr}$ (see Table IV). In the above experiments on the KITTI-MOTS training set, the best sMOTSA and IDS scores are achieved when $\beta$ is 0.4 and 0.4 for cars and pedestrians, respectively. Comparing (a)(c) and (b)(d), mask IoU shows not only less sensitivity to the parameter values but also better performance than IoU. Thus, mask IoU is selected for the mask merging measure, the same values are set for the test and the results are presented in Tables V and VI.

[18] 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, Oct. 2006.

[19] B. N. Vo, S. Singh, and A. Doucet, “Sequential Monte Carlo implementation of the PHD filter for multi-target tracking,” in Proc. Int. Conf. Information Fusion (ICIF), Jul. 2003, pp. 792–799.

[20] Y. Song, K. Yoon, Y.-C. Yoon, K. C. Yow, and M. Jeon, “Online multi-object tracking with GM-PHD filter and occlusion group management,” IEEE Access, vol. 7, pp. 165 103–165 121, Nov. 2019.

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

[22] 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. IEEE Int. Workshop Traffic Street Surveillance. Safety Secur. (AVSS), Sep. 2017, pp. 1–6.

[23] Z. Fu, F. Feng, F. Angelini, J. Chambers, and S. M. Naqv, “Particle PHD filter based multiple human tracking using online group-structured dictionary learning,” IEEE Access, vol. 6, pp. 17464–17478, Mar. 2018.

[24] R. Sanchez-Matilla and A. Cavallaro, “A predictor of moving objects for first-person vision,” in Proc. IEEE Int. Conf. Image Processing (ICIP), Sep. 2019, pp. 2189–2193.

[25] 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.

[26] V. Eiselein, D. Arp, M. Patzold, and T. Sikora, “Real-time multi-human tracking using a probability hypothesis density filter and multiple detectors,” in Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS), Sep. 2012, pp. 325–330.

[27] Z. Fu, F. Angelini, J. Chambers, and S. M. Naqv, “Multi-level cooperative fusion of GM-PHD filters for online multiple human tracking,” IEEE Trans. Multimedia, vol. 21, pp. 2277–2291, Sep. 2019.

[28] N. L. Baisa and A. Wadhwa, “Development of a N-type GM-PHD filter for multiple target, multiple type visual tracking,” Journal of Visual Communication and Image Representation, vol. 59, pp. 257–271, 2019.

[29] R. H. Evangelio, T. Senst, and T. Sikora, “Detection of static objects and segmentation,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Jan. 2011, pp. 534–540.

[30] M. Patzold and R. H. Evangelio, and T. Sikora, “Correcting people in crowded environments by fusion of shape and motion information,” in Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill. (PETS Workshop), Aug. 2010, pp. 157–164.
the car w/ ID 7 that are switched to ID 17.

Settings of the proposed method, which are based on the same segmentation results (a) from MaskRCNN [8]. Comparing (b) the baseline model p1 and (c) the model p4 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 p6, 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.

Fig. 11. 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 [8]. Comparing (b) the baseline model p1 and (c) the model p4 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 p6, 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.

[S31] 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.

[S32] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, , and B. Schiele, “DeeperCut: A deeper, stronger, and faster multi-person pose estimation model,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Oct. 2016, pp. 34–55.

[S33] Z. Fu, F. Angelini, J. Chambers, and S. M. Naqvi, “Multi-level cooperative fusion of GM-PHD filters for online multiple human tracking,” IEEE Trans. Multimedia, vol. 21, no. 9, pp. 2277–2291, 2019.

[S34] N. L. Balsa and A. Wallace, “Development of a N-type GM-PHD filter for multiple target, multiple type visual tracking,” Journal of Visual Communication and Image Representation, vol. 59, pp. 257–271, 2019.

[S35] Y.-C. Yoon, D. Y. Kim, Y. Song, K. Yoon, and M. Jeon, “Online multiple pedestrians tracking using deep temporal appearance matching association,” Information Sciences, vol. 561, pp. 326–351, Jun. 2021.

[S36] Y. Song, K. Yoon, Y. Yoon, K. C. Yow, and M. Jeon, “Online multi-object tracking with historical appearance matching and scene adaptive detection filtering,” in Proc. IEEE Int. Conf. Adv. Video Signal Based Survell. (AVSS), Nov. 2018, pp. 1–19.

[S37] K. Yoon, Y. Song, and M. Jeon, “Multiple hypothesis tracking algorithm for multi-target multi-camera tracking with disjoint views,” IET Image Processing, vol. 12, no. 7, pp. 1175–1184, Jul. 2018.

[S38] 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.

[S39] J. Shen, Z. Liang, J. Liu, H. Sun, L. Shao, and D. Tao, “Multiobject tracking by submodular optimization,” IEEE Trans. Cybern., vol. 49, no. 6, pp. 1990–2001, Jun. 2019.

[S40] X. Cao, X. Jiang, X. Li, and P. Yan, “Correlation-based tracking of multiple targets with hierarchical layered structure,” IEEE Trans. Cybern., vol. 48, no. 1, pp. 90–102, Jan. 2018.

[S41] 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.

[S42] 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.

[S43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, Z. H. S. Ma, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, Dec. 2015.

[S44] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, “The Mapillary Vistas dataset for semantic understanding of street scenes,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 4990–4999.

[S45] 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.

[S46] 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.

[S47] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: deep learning on point sets for 3D classification and segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, p. 652–660.

[S48] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Čehovin, G. Fernández, T. Vojir, G. Hager, G. Nebayh, and R. Pflugfelder, “The visual object tracking vot2015 challenge results,” in Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW), Dec. 2015, pp. 1–13.

[S49] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High performance visual tracking with siamese region proposal network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 8971–8980.

[S50] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi, “Deep learning for person re-identification: A survey and outlook,” IEEE Trans. Pattern Anal. Mach. Intell., 2021.

[S51] 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, 1987.

[S52] 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), Aug. 2009, pp. 2953–2960.

[S53] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” https://github.com/facebookresearch/detectron2, 2019–.

[S54] 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.

[S55] 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.

[S56] R. Henschel, L. Leal-Taixe, D. Cremers, and B. Rosenhahn, “Fusion of head and full-body detectors for multi-object tracking,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2018, pp. 1541–1550.
[57] L. Chen, H. Ai, Z. Zhuang, and C. Shang, “Real-time multiple people tracking with deeply learned candidate selection and person re-identification,” in *IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.

[58] M. Keuper, S. Tang, B. Andres, T. Brox, and B. Schiele, “Motion segmentation & multiple object tracking by correlation co-clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 140–153, Jan. 2020.