Aggregated Euclidean Distances for a Fast and Robust Real-Time 3D-MOT

Rahmad Sadli, Mohamed Afkir, Abdenour Hadid, Atika Rivenq, Abdelmalik Taleb-Ahmed

Aggregated Euclidean Distances for a Fast and Robust Real-Time 3D-MOT

Rahmad Sadli, Mohamed Afkir, Abdenour Hadid, Atika Rivenq, and Abdelmalik Taleb-Ahmed

Abstract —Autonomous driving systems must have the ability to monitor the kinematic behaviour of multiple obsta-cles. Therefore, 3D multi-object tracking (3D-MOT) is one of the crucial modules in autonomous driving to detect the presence of potential hazard movements such as human operated vehicles and pedestrians. In this work, we present a novel online 3D multi-tracking system that uses the Aggre-gated Euclidean Distances (AED) in data association module instead of using Intersection over Union (IoU) as a new metric. AED is used in order to obtain the relationship between predicted tracks and current object detections. There are several benefits from using AED in data association module. Firstly, it can reduce the system's complexity so that the execution time can be significantly reduced (as calculating Euclidean distances is much faster than obtaining 3D-IoU). Secondly, AED can provide distance measurement even when there is no overlaps between the predicted tracks and the current detections, while 3D-IoU produces zeros for non-overlapping cases. To demonstrate the validity of our proposed method, we performed extensive experiments on KITTI multi-tracking benchmark and nuScenes validation datasets. The experimental results are compared against the open-sourced state of the art 3D MOTs such as AB3DMOT, FANTrack, and mmMOT. Our method clearly outperforms the AB3DMOT baseline method and other methods in terms of accuracy and/or processing speed.

***Index Terms*—Aggregated Euclidean distances, AED, 3D MOT, real-time multi-tracking.**

I. Introduction

AUTONOMOUS driving or driverless car system is a promising technology for future transportation that potentially has the capacity to improve road safety and to have a better mobility. Self-driving cars promise to bring a number of benefits to society, including prevention of road accidents, optimal fuel usage, comfort and convenience [1].

In order to integrate driverless cars in urban traffics, their safe operation must be ensured from the presence of potential hazards such as human operated vehicles and pedestrians [2]. Besides having good perception systems, they must also have the ability to monitor the kinematic behaviour of the multiple obstacles. Monitoring the kinematic behavior of multiple obstacles is commonly known as Multiple-Object Tracking (MOT). MOT involves a stage where newly detected obstacles are evaluated for association with already known obstacle tracks. This stage, called the Data Association stage, is responsible for partitioning sensor reports into tracks and false alarms [3].

In this work, we propose an approach for fast and robust 3D-MOT for real-time applications. The objective is to obtain the highest possible accuracy in the least possible time in order to have a feasible multi-object tracking system that can be used in real-time applications. Our present article is basically inspired by the work on 3D-MOT called AB3DMOT [7]. We introduce several extensions. Firstly, while AB3DMOT [7] usually uses identity matrices multiplied by a chosen scalar for covariance matrices used in kalman filter process, we propose to estimate the covariance matrices to have a better performance (details in Section III). Secondly, instead of using 3D-IoU as in AB3DMOT [7], we propose to use *Aggregated Euclidean Distances* (AED) to obtain a robust data association and to speed-up the data association process. AED is calculated based upon the aggregate of the euclidean distances between corners

This work was supported in part by the European project InDiD: *Infrastructure Digitale de Demain* and in part by*la chaire d'excellence RIVA de la région Hauts de France*. The associate editor coordinating the review of this article and approving it for publication was Prof. Pierluigi Salvo Rossi.*(Corresponding author: Rahmad Sadli.)*

Rahmad Sadli, Atika Rivenq, and Abdelmalik Taleb-Ahmed are with the Polytechnic University of Hauts-de-France, 59313 Valenciennes, France (e-mail: rahmad.sadli@uphf.fr; atika.menhaj@uphf.fr; abdelmalik.taleb-ahmed@uphf.fr).

Mohamed Afkir is with Transalley Technopôle, 59300 Famars, France (e-mail: mohamed.afkir@transalley.com).

Abdenour Hadid was with the University of Oulu, 90570 Oulu, Finland. He is now with the Polytechnic University of Hauts-de-France, 59313 Valenciennes, France (e-mail: abdenour.hadid@ieee.org).

boxes and between centroids of the predicted tracks and the new detection objects. Finally, in order to recover the lost track caused by miss detections, we propose to include the maximum skipped frames module in Birth/Death controller. The unmatched tracks will be preserved for a certain number of frames and will be combined with the updated matched tracks and the newly created tracks to predict new tracks positions.

To demonstrate the validity of our proposed method, we perform extensive experiments by evaluating our framework on the 3D MOT benchmarks: KITTI [8] and nuScenes [9] validation datasets.

To summarize, our main contributions are as follows:

- We propose to estimate the covariance matrices used in Kalman prediction states in order to have a better accuracy.
- We propose to use Aggregated Euclidean Distance instead of using 3D-IoU as a metric for obtaining the relationship between the predicted tracks and the current detected objects in the data association module to accelerate the tracking process.
- We propose to include maximum skipped frames in Birth/Death controller to recover tracks from lost detections.
- We perform extensive experimental and comparative analysis on two publicly available datasets (KITTI [8] and nuScenes [9]).

The rest of the paper is structured as follows. Related works are discussed in Section II. Section III presents our proposed approach. The framework of the proposed pipeline is presented in this section. Section IV describes data association including the proposed AED and maximum skipped frames. Section V provides the experiments and discusses the obtained results. The last section, Section VI, draws some conclusions and future directions.

II. Related Works

Multi-object tracking can be divided into two categories, batch-based methods, and filter-based methods. Batch-based methods also known as off-line methods use the entire sequences in order to find the global optimal solution. A common solution used in batch-based methods is the minimum cost flow algorithm to find the optimum solution of the network flow graphs [10], [15]. Graph-based clustering [30], Network flow optimization [29] and Bayesian filtering-based tracking [33] are some of the popular methods of the off-line MOT methods in the recent past years. Another off-line MOT proposed a submodular optimization for multi-object visual tracking [19]. This method adopts a strategy to reduce the search complexity by finding low-level tracklets crossing several adjacent frames, and then combines them into complete trajectories. To find low-level tracklets, the method formulates data association in each segment as a network flow optimization problem and then uses a network flow algorithm to find the solution.

Online methods (like [5]–[7], [16], [17], and [18]) use only the past and the current observations and associate the current incoming observations to existing trajectories using state spaces models like Kalman filter or Particle Filters. This association is often formulated as a bipartite graph matching problem and traditionally solved by using the Hungarian algorithm [6], [10], [11], [21].

Data association is an important component in a multi-object tracking systems. Karunasekera *et al*. [26] tried to resolve the uncertainty of targets states by proposing a dissimilarity measure based on object motion, appearance, structure, and size. They used four distances, including appearance-based distance, structured-based distance, motion-based distance, and size-based distance. These dissimilarity values are then used in Hungarian algorithm, in the data association module for track identity assignment. This method showed good results, but applied to 2D images only.

Many other approaches have also been proposed (such as [31], [32], [34], and [35]). In these works, the authors use Recurrent Neural Networks (RNNs) to construct the affinity model. Even though they have demonstrated the effectiveness of their methods, these approaches may have a potential vulnerability because the model is trained on a different distribution from the test scenario, which can both diminish the discriminability and result in error accumulation during inference [12].

Systems based on tracklet association have been also proposed in [12], [13], and [14]. Such systems employ deep-learning models and require a high computational cost that makes real-time performance a challenge. The concern of real-time MOT application is not merely on the accuracy, but also on the computational efficiency and system simplicity.

In order to improve the computational efficiency, a unified motion and affinity model into a single framework has been proposed in [20]. However, even though they stated that this design has improved the computational efficiency with low memory requirement and simplified training procedure, this system still struggles with the processing speed that is only successfully run at 5 FPS on a 1080 Ti NVidia GPU.

Several 3D multi-object tracking methods are extended from 2D multi-object tracking methods and most of them use 3D-IoU as a metric to obtain the relationship between the predicted tracks and the new incoming detections. Weng
textit *et al.* [7] have resulted a state of the art in 3D multi-object tracking. Their simple method resulted in significant achievement in 3D-MOT. For real-time efficiency and to keep a simple design, they did not use neural networks but only employed the Hungarian algorithm and Kalman Filter. Therefore, in this work, we consider their method as a baseline reference for developing our 3D-MOT system.

III. Our Approach

In this work, we propose to extend the 3D MOT baseline method called AB3DMOT [7]. We provide several extensions as follows:

- *Firstly*, AB3DMOT used a default covariance matrices for Kalman filter process, while in our method, we propose to estimate the covariance matrices. This approach will be discussed in the next subsection.
- *Secondly*, instead of using 3D-IoU, we propose to use Aggregated Euclidean Distances (AED) to have a robust

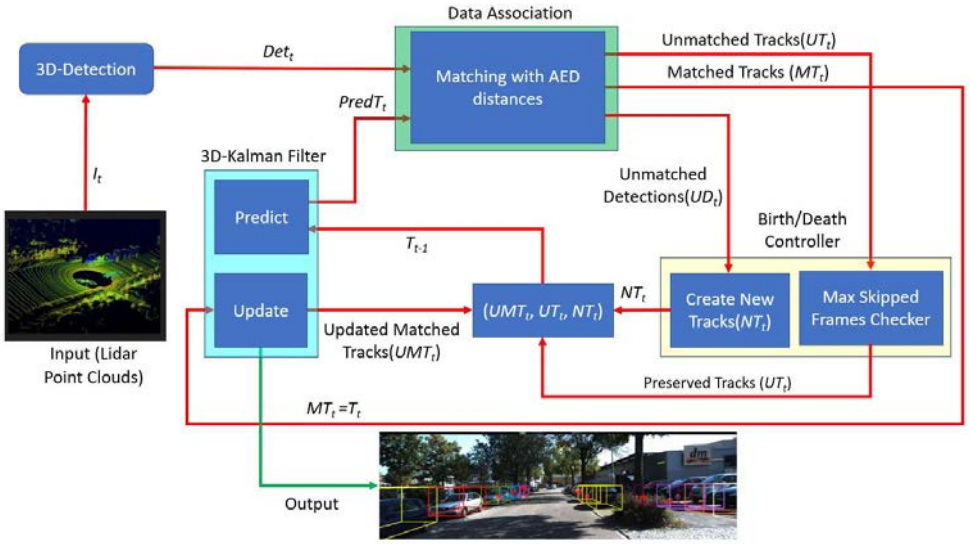

Fig. 1. Proposed System Pipeline. In this work, we have added three contributions. Firstly, we estimate the covariance matrices Q and R in 3D Kalman filter module (blue color box). Secondly, we replace the 3D-IoU method in Data Association Module (green color box) used in AB3DMOT method by the Aggregated Euclidean Distance (AED). And thirdly, we add the maximum skipped frames parameter in Birth/Death module (yellow color box) to prevent the loss of many potential positive trajectories caused by false negative detections.

data association and to speed up the data association process.

- *Thirdly*, we propose to include maximum skipped frames in Birth/Death controller to recover the lost tracks caused by false negative detections.

AED is the aggregate of the euclidean distances between corners boxes and between centroids of the predicted tracks and the newly detected objects. Calculating euclidean distances takes a low computational cost compared to 3D-IoU's calculation. Moreover, the Aggregated Euclidean Distances can provide distance measurement even when there is no overlaps between the predicted tracks and the current detections, while the 3D-IoU only results in zeros. We believe that using AED in the data association module can reduce the system complexity and can perform the tracking process much faster than the use of 3D-IoU. In addition, we have validated that our proposed 3D MOT system using the AED performs very well and gives a very good performance for the tracking accuracy and the processing speed.

The framework of the proposed algorithm is presented in Figure 1. It includes four main parts: Moving 3D-target detection as input to the system, target state prediction using 3D Kalman Filter, associated matrix reasoning or data association module using Hungarian algorithm and Bird/Death controller.

For 3D detections, we use the 3D detection obtained by PointRCNN [27] and MEGVII [28] for evaluating KITTI and nuScences datasets, respectively as used in AB3DMOT [7]. The classical 3D Kalman Filter algorithm is employed in the target state prediction, and Hungarian algorithm is used to associate the target states produced by 3D Kalman Filter and the current detections that takes an input from the 3D detection module.

A. 3D Multi-Object Tracking

We adopt the AB3DMOT [7] baseline in our work. For every frame t, the output of 3D detection is a set of detection $\mathbf{D}_{(t)} = \{D_0^{(t)}, D_1^{(t)}, ..\}$. Every object is described in ego-vehicle coordinates by its detection score s and box coordinates, which are represented by its 3D box center coordinate (x, y, z), 3D size (w, l, h), and box orientation θ. We model object's state $\mathbf{s}_t$ that has 11 variables as follows:

$$\mathbf{s}_t = \left(x, y, z, \theta, w, l, h, \dot{x}, \dot{y}, \dot{z}, \dot{\theta}\right) \tag{1}$$

where: $\dot{x}$, $\dot{y}$, $\dot{z}$, and $\dot{\theta}$ are the velocities in x-,y-,z-directions and the angular velocity, respectively.

AB3DMOT [7] is developed from SORT [11] algorithm and it uses default covariance matrices which is a chosen scalar multiplied by identity matrices. In our approach, we estimate covariance matrices using an approximation of noise covariance as explained in the following subsection. Moreover, AB3DMOT [7] uses the 3D-IoU as the affinity function. Meanwhile, we propose to use a new distance metric, which is called the Aggregated Euclidean Distances (AED). However, we still use the same data association method as used in AB3DMOT, which is the Hungarian algorithm.

At time t, we are given an image of the current frame $F_{(t)} \in \mathbb{R}^{WxHx3}$, and the previous frame $F_{(t-1)} \in \mathbb{R}^{WxHx3}$, as well as the tracked object information $\mathbf{T}_{(t-1)} = \{T_0^{(t-1)}), T_1^{(t-1)}), ..\}$. Every track is described by its track identity (*track id*), box coordinates in ego-vehicle coordinates represented by its 3D box center coordinate (x, y, z), 3D size (w, l, h) and box orientation θ, and box detection score s. Our aim is to track the current detected objects $\mathbf{D}_{(t)} = \{D_0^{(t)}, D_1^{(t)}, ..\}$ in the current frame $F_{(t)}$ by assigning the same objects appearing in the previous frame $F_{(t-1)}$ with a consistent track identities, which are the same track identities as in the previous frame $F_{(t-1)}$, and assigning the newly detected objects with new tracks identities. Algorithm 1 explains the details of this implementation.

B. Kalman Filter

Kalman filtering [22] makes a forward projection state or predicts the next state using prior knowledge of the state and

the current observation or measurement. It estimates the state of a system at time k using the linear stochastic difference. The state of a system at a time k ($\mathbf{s}_k$) that evolves from the prior state at time $k-1$ is written in the following form:

$$\mathbf{s}_k = A\mathbf{s}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{2}$$

and the measurement model z_k describing a relation between the state and measurement at the current step k is written as:

$$\mathbf{z}_k = H\mathbf{s}_k + \mathbf{v}_k \tag{3}$$

where A, a matrix nxn, is the state transition matrix relating the previous time step $k-1$ to the current state k. B, a matrix nxl, is a control input matrix applied to the optional control input $\mathbf{u}_{k-1}$. H, a matrix mxn, is a transformation matrix that transforms the state into the measurement domain. $\mathbf{w}_k$ and $\mathbf{v}_k$ represent the process noise vector with the covariance Q and the measurement noise vector with the covariance R, respectively. They are assumed statistically independence Gaussian noise with the normal probability distribution.

$$\begin{aligned} p\,(w) &\sim N(0, Q) \\ p\,(v) &\sim N(0, R) \end{aligned} \tag{4}$$

In real-world applications, Q and R are difficult to tune, and the performance of Kalman filter is strongly dependent upon the results of tuning the Q and R [23].

Kalman filter [22] uses a feedback control form to estimate the state by firstly predicts the state in a particular time and then calculates the feedback using a noisy measurement. Therefore, the equation of Kalman Filter can be divided into two groups: the time update equations and measurement update equations. In time update equations, the *a priori* state estimate $\hat{\mathbf{s}}_k^-$ is predicted by using the state dynamic equation model that projects forward one step in time as follows:

$$\hat{\mathbf{s}}_k^- = A\hat{\mathbf{s}}_{k-1} + B\mathbf{u}_{k-1} \tag{5}$$

where: $\hat{\mathbf{s}}_{k-1}$ is the previously *a posteriori* estimated state. Then, the error covariance matrix $\mathbf{P}_k^-$ is predicted by:

$$\mathbf{P}_k^- = A\mathbf{P}_{k-1}A^T + \mathbf{Q} \tag{6}$$

where $\mathbf{P}_{k-1}$ is the previously estimated error covariance matrix and $\mathbf{Q}$ is the process noise covariance.

In measurement update equations, we start by computing the Kalman gain $\mathbf{K}_k$ as follows:

$$\mathbf{K}_k = \mathbf{P}_k^- H^T (H\mathbf{P}_k^- H^T + \mathbf{R})^{-1} \tag{7}$$

where $\mathbf{R}$ is the measurement noise covariance. After that, we perform the actual measurement $\mathbf{z}_k$.

A posteriori state estimate $\hat{\mathbf{s}}_k$ can be computed as a linear combination of an *a priori* state estimate $\hat{\mathbf{s}}_k^-$ and a product of Kalman gain $\mathbf{K}_k$ and the measurement residual, which is the difference between an actual measurement $\mathbf{z}_k$ and a measurement prediction $H\hat{\mathbf{s}}_k^-$.

$$\hat{\mathbf{s}}_k = \hat{\mathbf{s}}_k^- + \mathbf{K}_k(\mathbf{z}_k - H\hat{\mathbf{s}}_k^-) \tag{8}$$

After obtaining the updated (*a posteriori*) state estimate, the filter calculates the updated error covariance $\mathbf{P}_k$, which will be used in the next time step.

$$\mathbf{P}_k = (I - \mathbf{K}_k H)\mathbf{P}_k^- \tag{9}$$

Algorithm 1 Proposed 3D MOT Algorithm

input : Set of Detections $\mathbf{D} = \{D_0, D_1, ..\}$,
Maximum Age Age_{max},
Max Skipped Frames $SkippedFrames_{max}$
output: Set of Valid Tracks $\hat{\mathbf{T}} = \{\hat{T}_0, \hat{T}_1, ..\}$
Init set of Valid Tracks $\hat{\mathbf{T}} \leftarrow \emptyset$
Init set of Tracks $\mathbf{T} \leftarrow \emptyset$
Init set of Track Predictions **PredT** $\leftarrow \emptyset$
Init set of New Tracks **NT** $\leftarrow \emptyset$
Init Matched Tracks Indices $\mathbf{MT}_{indices} \leftarrow \emptyset$
Init Unmatched Tracks Indices $\mathbf{UT}_{indices} \leftarrow \emptyset$
Init Unmatched Detections Indices $\mathbf{UD}_{indices} \leftarrow \emptyset$
while *True* **do**
  Get new detections **D**
  # Use 3D-KF to predict the previous tracks
  PredT $\leftarrow$ 3DKalmanFilter.*predict*(**T**)
  T.$Age \leftarrow$ **T**.Age +1
  # Associate tracks with new detections using Hungarian and AED method
  $MT_{indices}, UD_{indices}, UT_{indices} \leftarrow$ AssociateTracking(**PredT**, **D**)
  # Update matched tracks with assigned detections

  for $i, trk \in enumerate(\boldsymbol{T})$ **do**
    if $i \notin UT_{indice}$ **then**
      $T_i \leftarrow$ 3DKalmanFilter.*update*(trk)
      $T_i.Age \leftarrow 0$
    end
  end
  # Initialize new tracks for unmatched detections
  for $i \in$ $\boldsymbol{UD}_{indice}$ **do**
    $NT_i \leftarrow$ 3DKalmanFilter.*initTracks*(D_i)
    $NT_i.Age \leftarrow 0$
    APPEND(**T**, NT_i.)
  end
  # Tracks validation and deletion
  for $i \leftarrow 0$ **to** $length(\boldsymbol{T})$ **do**
    if $T_i.Age < Age_{max}$ **then**
      APPEND($\hat{\mathbf{T}}$,T_i)
    end
    if $T_i.Age > SkippedFrames_{max}$ **then**
      Delete track T_i
    end
  end
  return *valid tracks* $\hat{\mathbf{T}}$
end

where I is an identity matrix.

For more detail about the principle of the Kalman filter, we encourage the readers to refer to the [22].

State Prediction

We formulate the state of dynamic model using constant velocity model as the following: The state in time k, denoted by $\mathbf{s}_k$, can be predicted by the previous state in time $k-1$, $\mathbf{s}_{k-1}$. Let x, y, and z be the positions in the x-, y- and z-directions, respectively, and let θ be the orientation. Also let $\dot{x}$ and $\dot{y}$ be

the velocities in x-, y- and z-directions, respectively, and the $\dot{\theta}$ be the angular velocity. Then, the 3D Kinematic equation for state $\mathbf{s}_k$ can be written as follows:

$$\begin{bmatrix} x_k \\ y_k \\ z_k \\ \theta_k \\ w_k \\ l_k \\ h_k \\ \dot{x}_k \\ \dot{y}_k \\ \dot{z}_k \\ \dot{\theta}_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1}\Delta t + \ddot{x}_{k-1}\frac{\Delta t^2}{2} \\ y_{k-1} + \dot{y}_{k-1}\Delta t + \ddot{y}_{k-1}\frac{\Delta t^2}{2} \\ z_{k-1} + \dot{z}_{k-1}\Delta t + \ddot{z}_{k-1}\frac{\Delta t^2}{2} \\ \theta_{k-1} + \dot{\theta}_{k-1}\Delta t + \ddot{\theta}_{k-1}\frac{\Delta t^2}{2} \\ w_{k-1} \\ l_{k-1} \\ h_{k-1} \\ \dot{x}_{k-1} + \ddot{x}_{k-1}\Delta t \\ \dot{y}_{k-1} + \ddot{y}_{k-1}\Delta t \\ \dot{z}_{k-1} + \ddot{z}_{k-1}\Delta t \\ \dot{\theta}_{k-1} + \ddot{\theta}_{k-1}\Delta t \end{bmatrix} \tag{10}$$

Process Noise Covariance Matrix Q

As discussed in the above subsection that the performance of Kalman filter is strongly dependent on the tuning of Q and R. In some real-world applications, the diagonal parameterization of the covariance matrices works quite well, however, it yields indeed a sub-optimal solution of the original problem [23]. Therefore, in this work we present our approach to address this problem in order to obtain the Q and R as natural as possible.

The process noise covariance matrix $\mathbf{Q}$ or error in the state process can be written as in eq.11, as shown at the bottom of the next page, where $(\sigma_x, \sigma_y, \sigma_z)$ and $(\sigma_{\dot{x}}, \sigma_{\dot{y}}, \sigma_{\dot{z}})$ are the standard deviations of the central position (x, y, z) and their velocities, respectively. The σ_θ is the standard deviation for the orientation θ, and the $\sigma_{\dot{\theta}}$ is the standard deviation of its angular velocity.

Since we have the noise coming from the accelerometer output [24], then the process noise Q can be regarded as the uncertain product of error in acceleration [25]. Therefore, we can find a relation between the position and the acceleration, as well as the relation between the velocity and the acceleration. These relations can be obtained from the equation eq.10 that the position is affected by $\frac{\Delta t^2}{2}$ multiplied by the acceleration, and the velocity is affected by Δt multiplied by the acceleration. This means that if we have the error in the acceleration, it will automatically affect the position and the velocity. Therefore, since we have error in the acceleration, we can define the standard deviation of position as the standard deviation of acceleration σ_a multiplied by $\frac{\Delta t^2}{2}$. Likewise, Δt is the effect on the velocity caused by the acceleration, we can define the standard deviation of the velocity as the standard deviation of acceleration σ_a multiplied by Δt.

Then, the process covariance noise $\mathbf{Q}$ can be written as in eq.12, as shown at the bottom of the next page, where σ_{a_x}, σ_{a_y}, and σ_{a_z}, are the standard deviations of the acceleration in x-, y-, and z-directions, respectively, and σ_{a_θ} is the standard deviation of the angular acceleration of the orientation θ.

Measurement Noise Covariance Matrix R

By supposing the measurement positions x, y, z and θ are independent from one another, we can discard any interaction between them so that the covariances between these elements are 0. Then, we can only focus on the variance for each element. Based on this assumption, the measurement noise covariance R can be written as in eq.13.

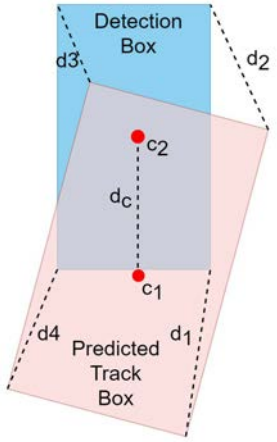

Fig. 2. Description of Aggregated Euclidean Distances used in our method.

$$\mathbf{R} = \begin{array}{c} \\ x \\ y \\ z \\ \theta \\ w \\ l \\ k \end{array} \begin{array}{c} \begin{array}{ccccccc} x & y & z & \theta & w & l & h \end{array} \\ \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\theta}}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \tag{13}$$

IV. Data Association

A. Aggregated Euclidean Distance

The costs are the fundamental parameters in track association. Having a cost function which dependent with some parameters make the track association more robust. Regarding data association between the predicted tracks and current detections, we approach our 3D-MOT system by using the Aggregated Euclidean Distances (AED) instead of using 3D-IoU as applied in the AB3DMOT [7] baseline and other previous works. To the best of our knowledge, this is the first new metric proposition applied to this domain. There are two intuitive reasons using this technique, firstly, compared to 3D-IoU, calculating Euclidean distance is much faster than obtaining the 3D-IoU. Therefore, using this technique can reduce the system complexity and the execution time significantly. Secondly, calculating the Aggregated Euclidean Distances in rotated boxes can achieve a robust distance between two boxes which have a strong correlation to each other. AED can provide distance measurement even when there is no overlaps between the predicted tracks and the current detections, while 3D-IoU only produces zeros for non overlapping cases. Figure 2 illustrates the AED used in the

proposed method. The AED can be calculated as follows:

$$AED = \frac{1}{2}(\sum_{i=1}^{n} d_i + d_c) \tag{14}$$

where d_i is the distance between the corners of the bounding boxes of the predicted tracks and the current detection objects, and i varies from 1 to $n = 4$. We only use four bottom box corners because the four top box corners have the same coordinates as the four bottom box corners regarded from z axis. The d_c is the distance of the bounding boxes centers of the predicted track and the current detection object.

The figure 3 shows two sets of illustration of measuring similarity between the predicted track and the current detection using IoU and AED for the overlapping case and non-overlapping case.

B. Initiation and Deletion of Track Identities

Also known as Death and Birth Memory in most of MOT papers, it is a task to record the matched or unmatched tracks with the new detections. It initiates new tracks or deletes existing tracks when needed. The observations that are not assigned to the existing tracks can initiate new tentative tracks. A tentative track is confirmed when the observation quality included in the track satisfies the confirmation criteria. Similarly, low-quality tracks, as usually determined by the update history, are deleted. Track quality may be defined as a criterion for initiating a new track or deleting an existing one.

In this work, we propose to preserve the tracks for the number of frames in order to recover the lost tracks caused by false negative detections. We include the *maximum skipped frames* parameter in Birth/Death controller. The *unmatched tracks* will be preserved for a certain number of frames and will be combined with the *updated matched tracks* and the new created tracks in order to predict new tracks positions.

Intuitively, if there is a lost track, it will be recovered if the module finds the shortest AED between the lost track and the new incoming object. If this new incoming object passes the threshold criteria, it is considered as a lost track and will be assigned the same *track id* as the last known *track id* for this object and then will be marked as a *matched track*. The figure 4 shows an illustration of recovering the lost track ($Tid = 2$) after occurring false negative detection for 3 consecutive frames. In this case, the position of the lost object is predicted based on the last known of its estimated position. If this object is still missing in detection process, the system continues to predict its position until the maximum value of the skipped frames allowed as illustrated in the figure. The lost track ($Tid = 2$) can be recovered at frame 5 after the same object is re-detected by the object detector.

The difference between our proposed method and the method used in AB3DMOT [7] is that, in AB3DMOT [7], they use the same parameter for both tracks validation and

$$\mathbf{Q} = \begin{array}{c} \\ x \\ y \\ z \\ \theta \\ w \\ l \\ k \\ \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \end{array} \begin{array}{c} \begin{array}{cccccccccccc} x & y & z & \theta & w & l & h & \dot{x} & \dot{y} & \dot{z} & \dot{\theta} \end{array} \\ \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_x\sigma_{\dot{x}} & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_y\sigma_{\dot{y}} & 0 & 0 \\ 0 & 0 & \sigma_z^2 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_z\sigma_{\dot{z}} & 0 \\ 0 & 0 & 0 & \sigma_\theta^2 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\theta\sigma_{\dot{\theta}} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma_{\dot{x}}\sigma_x & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\dot{x}}^2 & 0 & 0 & 0 \\ 0 & \sigma_y\sigma_{\dot{y}} & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\dot{y}}^2 & 0 & 0 \\ 0 & 0 & \sigma_{\dot{z}}\sigma_z & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\dot{z}}^2 & 0 \\ 0 & 0 & 0 & \sigma_{\dot{\theta}}\sigma_\theta & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{\dot{\theta}}^2 \end{bmatrix} \end{array} \tag{11}$$

$$\mathbf{Q} = \begin{bmatrix} \frac{\Delta t^4}{4}\sigma_{a_x}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\Delta t^3}{2}\sigma_{a_x}^2 & 0 & 0 & 0 \\ 0 & \frac{\Delta t^4}{4}\sigma_{a_y}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\Delta t^3}{2}\sigma_{a_y}^2 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^4}{4}\sigma_{a_z}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\Delta t^3}{2}\sigma_{a_z}^2 & 0 \\ 0 & 0 & 0 & \frac{\Delta t^4}{4}\sigma_{a_\theta}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\Delta t^3}{2}\sigma_{a_\theta}^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\Delta t^3}{2}\sigma_{a_x}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t^2\sigma_{a_x}^2 & 0 & 0 & 0 \\ 0 & \frac{\Delta t^3}{2}\sigma_{a_y}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t^2\sigma_{a_y}^2 & 0 & 0 \\ 0 & 0 & \frac{\Delta t^3}{2}\sigma_{a_z}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t^2\sigma_{a_z}^2 & 0 \\ 0 & 0 & 0 & \frac{\Delta t^3}{2}\sigma_{a_\theta}^2 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t^2\sigma_{a_\theta}^2 \end{bmatrix} \tag{12}$$

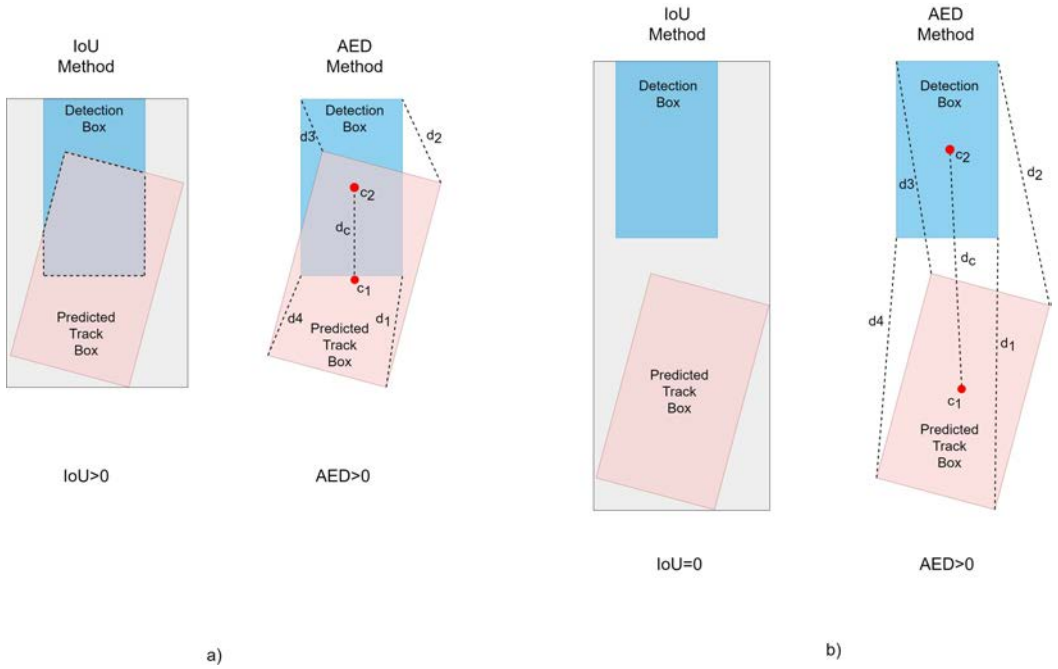

Fig. 3. Two sets of examples for obtaining the relation between the predicted track and the current detection using IoU and AED. In overlaping case (a), both IoU and AED methods produce the positive values, but for the non-overlapping case (b), the result of the IoU method is always zero, while the result of AED method is always a positive value.

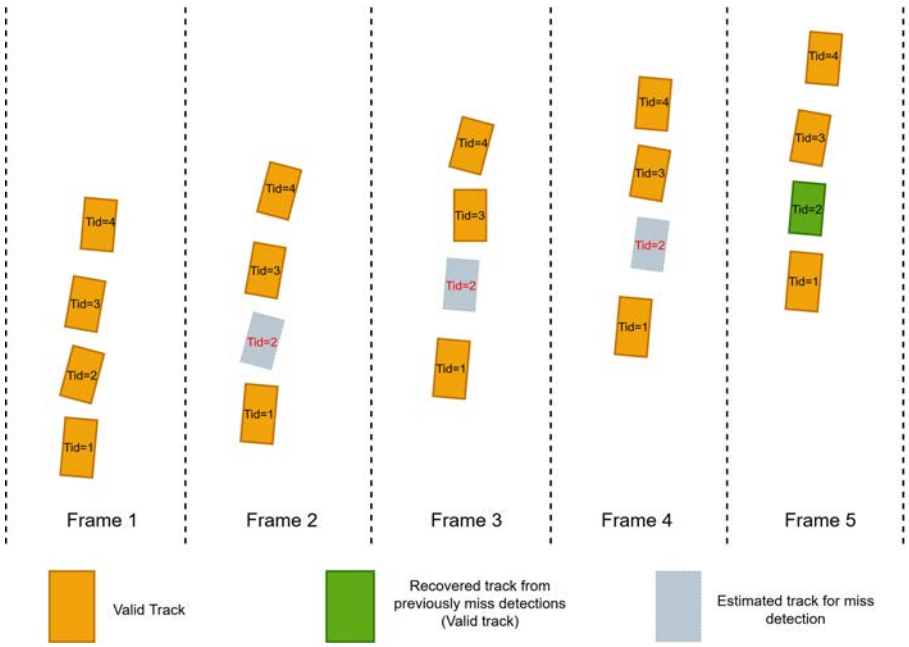

Fig. 4. Illustration of using the maximum skipped frames. This figure illustrates an example of a lost track ($Tid = 2$) caused by false negative detection object at frames 2, 3 and 4. At frame 5, the same object is re-detected and successfully re-assigned the same track id ($Tid = 2$).

tracks deletion, which is the Age_{max}. A track is considered as a valid track if its age is less than the value of the Age_{max} and the one whose the age is greater or equal to the value of the Age_{max} will be discarded from the tracklets. Based on their ablation study, the optimal performance is obtained at the Age_{max} value is set to 2. This means that, they consider every unmatched track after 2 successive frames will be deleted. As a consequence, it causes the possibility deletion of many potential positive trajectories that might still exist in the scene that cannot find the matched objects due to false negative detections. Meanwhile, in our method, we still adopt the Age_{max} parameter but only for validating of the matched tracks. However, for removing the unmatched track, we propose to use a different parameter that we call as the maximum skipped frames instead of using the same value of the Age_{max} as proposed in AB3DMOT. As the result,

by using two different parameters for validating and removing tracks, we can set a better range to minimize the loss of potential positive trajectories. Based on our ablation study, the maximum skipped frames that yields a better performance is set to 10 frames.

V. Experiments and Analysis

A. Experimental Data

We evaluated our proposed 3D-MOT method on the KITTI [8] 3D MOT benchmark and nuScenes [9] datasets. Both datasets provide LiDAR point cloud and 3D bounding box trajectories. The KITTI 3D MOT dataset consists of 21 training/validation sequences and 29 test sequences. Since KITTI provides the ground truth labels publicly only for training/validation split, we evaluated our system using the training/validation split only as in AB3DMOT [7]. Meanwhile, the nuScenes [9] dataset consists of 1000 driving scenes. Each scene is approximately 20s long and fully annotated with 3D bounding boxes for 23 classes and 8 attributes. The nuScenes dataset is divided into 3 parts, mini, trainval, and test parts. The mini-part is a subset of the trainval-part containing 10 scenes used to explore data without downloading the whole dataset. The trainval-part contains 850 scenes, which are 700 scenes for training dataset and 150 scenes for validation with fully annotated. The test-part contains only 150 scenes with no annotations. For nuScenes dataset, we also evaluated the same partition as used in AB3DMOT [7], that is the trainval-part.

B. Evaluation Metrics

Conventional metrics applied to MOT systems are based on CLEAR MOT metrics [36], such as MOTA (Multi Object Tracking Accuracy), MOTP (Multi-Object Tracking Precision), IDS (Number of identity switches), FRAG (Number of fragmentations generated by false negatives), FN/FP (Number of false negatives/positives), and ML/MT (Number of Mostly Lost/Tracked trajectories). However, these metrics do not take into account the confidence's score explicitly, which means that the CLEAR metrics consider all object trajectories having the perfect confidence's score ($s = 1$). This assumption is not optimal because there could be many false positive trajectories with low confidence scores [7].

In order to tackle the conventional MOT metrics, in which evaluation metrics do not consider the confidence and only evaluate at a single threshold, the authors of the AB3DMOT [7] introduced two integral metrics – AMOTA and AMOTP (average MOTA and MOTP) in order to summarize the performance of MOTA and MOTP across many thresholds, as shown in eq. 15.

$$AMOTA = \frac{1}{L} \sum_{r \in \{\frac{1}{L}, \frac{2}{L}, ..., 1\}} (1 - \frac{FP_r + FN_r + IDS_r}{num_{gt}}) \tag{15}$$

where FP_r, FN_r, and IDS_r are the number if false positives, false negatives and identity switches at a specific recall value r. L is the number of recall value and num_{gt} is the number of ground-truths. Likewise, AMOTP can be derived by integrating MOTP across all recall values.

TABLE I
Setting Parameters Used for Estimating Noise Covariance Matrices

Parameters	KITTI Dataset	nuScenes Dataset
Δt	20s	5s
σ_x	0.5m	3m
σ_y	0.5m	3m
σ_z	0.5m	3m
σ_θ	0.5rad	0.1rad
σ_{a_x}	$0.5m/s^2$	$15m/s^2$
σ_{a_y}	$0.5m/s^2$	$15m/s^2$
σ_{a_z}	$0.5m/s^2$	$15m/s^2$
σ_{a_θ}	$0.5rad/s^2$	$0.1rad/s^2$

To normalize the value of the integral metric AMOTA to range between 0% to 100%, the authors of the AB3DMOT [7] scale the range of the MOTA at the specific recall value r by introducing two new metrics called sMOTA and sAMOTA, which are formulated as follows:

$$sMOTA = max(0, 1 - \frac{FP_r + FN_r + IDS_r - (1-r) \times num_{gt}}{r \times num_{gt}}) \tag{16}$$

$$sAMOTA = \frac{1}{L} \sum_{r \in \{\frac{1}{L}, \frac{2}{L}, ..., 1\}} sMOTA_r \tag{17}$$

For nuScenes dataset, we follow nuScenes evaluation metric that uses AMOTA, which penalizes ID switches, false positive, and false negatives and is averaged among various recall thresholds.

C. Experimental Results

1) Optimal Parameters Setting: For our optimal results presented in tables II, III, IV and V, we use the parameters setting as the following:

- *Parameters setting for estimating noise covariance matrices:* For simplicity, we assume that the standard deviations in the positions $(\sigma_x, \sigma_y, \sigma_z)$ are all the same. We also consider the standard deviations in the acceleration $(\sigma_{a_x}, \sigma_{a_y}, \sigma_{a_z})$ are the same. Empirically, we found the optimum values that give the best of our results are as shown in Table I.

- *Parameters setting used in data association module:* Based on our ablation study, we found that setting the maximum skipped frames to 10 frames in the birth and death memory module can give the best performance in terms of accuracy. We also found empirically for AED's threshold values used to reject none matching criteria in the data association module are as the following: For KITTI dataset, we set $AED_{threshold}$ to 4 meters, 2 meters, and 1 meter, respectively for car, cyclist, and pedestrian can achieve the best performance as presented in the tables II, III, and IV. For the nuScenes dataset, we set $AED_{threshold}$ to 4 meters for all categories for our optimal results.

TABLE II

MOT PERFORMANCE OF **CAR** EVALUATION ON THE **KITTI VALIDATION SET** USING THE PROPOSED METHOD. IN EACH COLUMN, THE BEST OBTAINED RESULTS ARE TYPESET IN **BOLD** FONTS AND THE SECOND BEST RESULT ARE IN BLUE VIOLET COLOR. (* WE RE-EXECUTED THE OPEN SOURCE CODE OF AB3DMOT [7] USING OUR COMPUTER IN ORDER TO HAVE A FAIR COMPARISON)

Method	Matching Criteria	sAMOTA↑	AMOTA↑	AMOTP↑	MOTA↑	MOTP↑	IDS↓	FRAG↓	FPS↑
mmMOT [5] (ICCV019)	$\text{IoU}_{th} = 0.25$	70.61	33.08	72.45	74.07	78.16	10	55	4.8 (GPU)
	$\text{IoU}_{th} = 0.5$	69.14	32.81	72.22	73.53	78.51	10	64	
	$\text{IoU}_{th} = 0.7$	62.72	24.71	66.06	49.19	79.01	38	406	
FANTrack [6] (IV020)	$\text{IoU}_{th} = 0.25$	82.97	40.03	75.01	74.30	75.24	35	202	25.0 (GPU)
	$\text{IoU}_{th} = 0.5$	80.14	38.16	73.62	72.71	74.91	36	211	
	$\text{IoU}_{th} = 0.7$	63.91	24.91	67.32	51.91	80.71	24	141	
AB3DMOT [7] (IROS020)	$\text{IoU}_{th} = 0.25$	93.28	45.43	77.41	86.24	78.43	**0**	**15**	*117.2 (CPU)
	$\text{IoU}_{th} = 0.5$	90.38	42.79	75.65	84.02	78.97	**0**	**51**	
	$\text{IoU}_{th} = 0.7$	68.81	27.26	67.00	57.06	**82.43**	**0**	**157**	
OURS	$\text{IoU}_{th} = 0.25$	**94.66**	**47.66**	**79.84**	**86.86**	**78.85**	*7*	37	**214.0 (CPU)**
	$\text{IoU}_{th} = 0.5$	**91.90**	**44.98**	**78.13**	**84.21**	**79.48**	5	88	
	$\text{IoU}_{th} = 0.7$	**74.01**	**30.38**	**69.13**	**61.00**	82.41	3	235	

TABLE III

MOT PERFORMANCE OF **CYCLIST** EVALUATION ON THE **KITTI VALIDATION SET** USING THE PROPOSED METHOD. IN EACH COLUMN, THE BEST OBTAINED RESULTS ARE TYPESET IN **BOLD** FONTS AND THE SECOND BEST RESULT ARE IN BLUE VIOLET COLOR. (* WE RE-EXECUTED THE OPEN SOURCE CODE OF AB3DMOT [7] USING OUR COMPUTER IN ORDER TO HAVE A FAIR COMPARISON)

Method	Matching Criteria	sAMOTA↑	AMOTA↑	AMOTP↑	MOTA↑	FPS↑
AB3DMOT [7] (IROS020)	$\text{IoU}_{th} = 0.25$	91.36	44.34	79.18	84.87	*772.7 (CPU)
	$\text{IoU}_{th} = 0.5$	89.27	42.39	77.56	79.82	
OURS	$\text{IoU}_{th} = 0.25$	**95.94**	**50.91**	**80.72**	**87.39**	**857.7 (CPU)**
	$\text{IoU}_{th} = 0.5$	**92.86**	**48.09**	**79.33**	**85.24**	

TABLE IV

MOT PERFORMANCE OF **PEDESTRIAN** EVALUATION ON THE **KITTI VALIDATION SET** USING THE PROPOSED METHOD. IN EACH COLUMN, THE BEST OBTAINED RESULTS ARE TYPESET IN **BOLD** FONTS AND THE SECOND BEST RESULT ARE IN BLUE VIOLET COLOR. (* WE RE-EXECUTED THE OPEN SOURCE CODE OF AB3DMOT [7] USING OUR COMPUTER IN ORDER TO HAVE A FAIR COMPARISON)

Method	Matching Criteria	sAMOTA↑	AMOTA↑	AMOTP↑	MOTA↑	FPS↑
AB3DMOT [7] (IROS020)	$\text{IoU}_{th} = 0.25$	75.85	31.04	55.53	70.90	*250.8 (CPU)
	$\text{IoU}_{th} = 0.5$	70.95	27.31	52.45	65.06	
OURS	$\text{IoU}_{th} = 0.25$	**79.19**	**33.28**	**56.85**	**71.21**	**345.3 (CPU)**
	$\text{IoU}_{th} = 0.5$	**73.35**	**28.82**	**53.74**	**67.22**	

2) Quantitative Results: We compared our method against open-sourced state of the art 3-D MOTs such as AB3DMOT [7], FANTrack [6] and mmMOT [5]. We used 3-D detections obtained by PointRCNN [27] and MEGVII [28] for evaluating KITTI benchmark 3D-MOT and NuScences datasets, respectively as used in AB3DMOT [7]. Tables II to V summarize the quantitative evaluation results.

Table II shows the MOT performance of **car** evaluation on the KITTI validation set using the proposed method. It shows that our proposed 3D MOT system consistently outperforms other modern 3D MOT systems in almost all evaluated metrics for different matching criteria (e.g., 3D IoU_{thresh} = 0.25, 0.5, and 0.7) except for ID switch (IDS) and Fragmentation (FRAG). For IDS and FRAG, our results achieve the second best results after the AB3DMOT [7]. The performance on the KITTI validation set for car evaluation demonstrates a very significant result in terms of the processing speed. We achieved an impressive result, almost twice as fast as the AB3DMOT [7] thanks to the use of Aggregated Euclidean Distance.

The interesting results occur when evaluating the cyclists and pedestrians. These two objects are very challenging compared to cars due to the small objects size and they can be very close to each other. However, our proposed 3D MOT again shows its superiority against other methods as shown in tables III and IV for the MOT performances of **cyclist** and **pedestrian** evaluation, respectively on the KITTI validation set using the proposed method. The asterisk (*) signs marked in the tables II, III and IV indicate that we re-executed the open-source code of AB3DMOT [7] baseline in order to have a fair FPS comparison.

In addition to KITTI dataset, we also conducted evaluation on the nuScenes dataset as performed in AB3DMOT [7].

Fig. 5. Qualitative comparison between AB3DMOT [7] (a) and our proposed system (b) on the sequence 0 of the KITTI test set.

TABLE V
MOT PERFORMANCE FOR **ALL** CATEGORIES ON **NUSCENES** VALIDATION SET USING THE PROPOSED METHOD. IN EACH COLUMN, THE BEST OBTAINED RESULTS ARE TYPESET IN **BOLDFACE**

Method	AMOTA↑	AMOTP↑	MOTA↑
AB3DMOT [7] (IROS020)	8.94	29.67	31.40
OURS	**40.30**	**89.95**	**33.34**

Our obtained results first confirmed the conclusions stated in AB3DMOT in the sense that nuScenes dataset is more challenging than KITTI dataset due to sparse lidar points cloud inputs, complex scenes, and low frame rates that impact to 3D detections on nuScenes significantly lower quality than 3D detections on KITTI [7]. However, compared to the result of the AB3DMOT [7], our proposed 3D MOT demonstrates an impressive improvement of the performance as shown in Table V.

3) Qualitative Results: We provide examples of the qualitative results of the comparison for car evaluation between AB3DMOT [7] and our proposed 3D MOT system as shown in figure 5a) and figure 5b), respectively for AB3DMOT [7] and ours. For this comparison, we took the sequence 0 of the KITTI test dataset from frame 1 to frame 7. The results are visualized with different colors representing the different track identities. We can see that the results of AB3DMOT contain identity switches as marked by green boxes and yellow circles for different frames as shown in figure 5a). The changes of these identities are caused by miss detection of the objects. Meanwhile, our proposed system presented in Figure 5b) shows different results. Thanks to the preservation of the maximum skipped frames, it does not have these identity switches issues on the example sequences. We can see that the results show that the lost tracks caused by false negative detections are successfully recovered. The proposed system produces stable results and has fewer identity switches.

D. Ablation Study

We performed an ablation study for cars on the KITTI validation set by modifying different parameters.

1) Effect of Preserving Maximum Skipped Frames: We first studied the effect of preserving of the maximum skipped frames to the tracking performance and speed. Table VI shows the effect of preserving maximum skipped frames to the performance accuracy and tracking speed. We found that the optimal result is achieved at the number of maximum skipped frames is set to **10**. Setting the number of maximum skipped frames to below **10** impacts to degrading the performance accuracy. As well, when the number of the maximum skipped

TABLE VI

ABLATION STUDY FOR THE EFFECT OF PRESERVING MAXIMUM SKIPPED FRAMES FOR CAR ON THE KITTI VALIDATION SET USING THE PROPOSED 3D MOT

Variants	sAMOTA↑	AMOTA↑	AMOTP↑	MOTA↑	MOTP↑	IDS↓	FRAG↓	FPS↑
MaxSkippedFrames = 4	92.05	44.85	77.95	86.05	78.84	13	45	**285.3**
MaxSkippedFrames = 6	92.57	45.39	77.90	86.48	78.71	14	50	235.4
MaxSkippedFrames = 8	92.10	44.98	77.91	**87.33**	78.81	12	42	221.0
MaxSkippedFrames = 10	**94.66**	**47.66**	**79.84**	86.86	**78.85**	7	37	214.0
MaxSkippedFrames = 12	94.49	47.43	79.80	86.57	78.84	7	35	190.0
MaxSkippedFrames = 14	94.21	47.36	79.84	83.88	78.81	**2**	**28**	179.8

TABLE VII

ABLATION STUDY FOR THE EFFECT OF DIFFERENT AED'S THRESHOLD APPLIED (IN *meters*) FOR CAR EVALUATION ON THE KITTI VALIDATION SET USING THE PROPOSED 3D MOT. MAXIMUM SKIPPED FRAME WAS SET TO **10**, WHICH IS THE OPTIMAL VALUE OBTAINED FROM THE TABLE VI

Variants	sAMOTA↑	AMOTA↑	AMOTP↑	MOTA↑	MOTP↑	IDS↓	FRAG↓	FP↓	FN↓
$\text{AED}_{Thresh} = 3$	92.64	45.71	77.90	84.94	78.92	10	**31**	**327**	925
$\text{AED}_{Thresh} = 4$	**94.66**	**47.66**	**79.84**	**86.86**	**78.85**	**7**	37	415	679
$\text{AED}_{Thresh} = 5$	94.19	47.38	79.82	86.78	78.81	9	41	422	**677**
$\text{AED}_{Thresh} = 6$	94.48	47.30	79.83	86.57	78.85	**7**	37	430	688
$\text{AED}_{Thresh} = 7$	94.14	47.04	79.76	84.28	78.81	12	44	443	862

TABLE VIII

ABLATION STUDY FOR THE EFFECT OF DIFFERENT $1t$ FOR CAR EVALUATION ON THE KITTI VALIDATION SET USING THE PROPOSED 3D MOT. WE EMPIRICALLY SET THE STANDARD DEVIATIONS PARAMETERS AS FOLLOWS: $\sigma_x = \sigma_y = \sigma_z = 0.5$ M, $\sigma_\theta = 0.5$ RAD, $\sigma_{a_x} = \sigma_{a_y} = \sigma_{a_z} = 0.5$ M/s^2, $\sigma_{a_\theta} = 0.5$ RAD/s^2

Variants	sAMOTA↑	AMOTA↑	AMOTP↑	MOTA↑	MOTP↑	IDS↓	FRAG↓	FP↓	FN↓
$\Delta t = 5s$	93.97	46.94	79.81	86.68	78.81	10	38	410	702
$\Delta t = 10s$	94.07	46.97	79.76	86.54	78.78	8	38	412	739
$\Delta t = 15s$	94.11	46.90	79.81	86.73	78.83	8	39	405	683
$\Delta t = 18s$	94.15	47.01	79.84	86.54	78.68	**7**	**35**	411	706
$\Delta t = 20s$	**94.66**	**47.66**	**79.84**	**86.86**	**78.85**	**7**	37	415	697
$\Delta t = 21s$	94.30	47.01	79.79	86.82	78.83	8	39	408	688
$\Delta t = 22s$	94.05	46.86	79.82	86.75	78.83	**7**	36	408	695
$\Delta t = 25s$	93.07	45.90	77.79	86.13	78.63	**7**	37	**401**	**665**

TABLE IX

ABLATION STUDY FOR THE EFFECT OF DIFFERENT SETTING OF PARAMETERS FOR ESTIMATION OF THE PROCESS NOISE COVARIANCE MATRIX Q AND THE MEASUREMENT NOISE R FOR CAR EVALUATION ON THE KITTI VALIDATION SET USING THE PROPOSED 3D MOT. WE USE OPTIMAL $1t$ = 20S OBTAINED FROM TABLE VIII

Variants	sAMOTA↑	AMOTA↑	AMOTP↑	MOTA↑	MOTP↑	IDS↓	FRAG↓	FP↓	FN↓
Using default covariance matrices Q and R	94.01	46.98	78.84	86.29	77.75	8	37	402	706
$\sigma_x = \sigma_y = \sigma_z = 0.1m$, $\sigma_\theta = 0.1\,rad$ $\sigma_{a_x} = \sigma_{a_y} = \sigma_{a_z} = 0.1\,m/s^2$, $\sigma_{a_\theta} = 0.1\,rad/s^2$	94.23	47.21	79.80	86.68	78.81	8	**35**	402	706
$\sigma_x = \sigma_y = \sigma_z = 0.5m$, $\sigma_\theta = 0.5\,rad$ $\sigma_{a_x} = \sigma_{a_y} = \sigma_{a_z} = 0.5\,m/s^2$, $\sigma_{a_\theta} = 0.5\,rad/s^2$	**94.66**	**47.66**	**79.84**	**86.86**	**78.85**	**7**	37	415	679
$\sigma_x = \sigma_y = \sigma_z = 1m$, $\sigma_\theta = 1\,rad$ $\sigma_{a_x} = \sigma_{a_y} = \sigma_{a_z} = 1\,m/s^2$, $\sigma_{a_\theta} = 1\,rad/s^2$	92.87	45.86	77.82	87.16	78.63	7	39	**396**	**673**

frames is set to above **10** also impacts to degrading performance accuracy. However, in terms of speed, the longer of skipped frames preserved, the longer of the execution time required that impacts to the lower FPS.

2) Effect of Different AED's Threshold Applied: In this study, we varied the distance threshold of AED in order to obtain the best performance. Table VII shows the results of study on the effect of applying different AED's threshold for car

evaluation on the KITTI validation set. The results show that the best AED's threshold setting is at 4 $meters$, which results in the optimal performance accuracy.

3) Effect of Different Parameters Setting Used for Estimating the Covariance Matrices Q and R: In order to have a better performance, we conducted a study by empirically varying the standard deviation parameters (σ_x, σ_y, σ_z, σ_θ, σ_{a_x}, σ_{a_y}, σ_{a_z}, σ_{a_θ}) and Δt used for estimating the covariance matrices Q and R. The tables VIII and IX show the results of this study that the best performance is achieved when Δt is set to 20s, and the values of the standard deviation parameters are set to $\sigma_x = \sigma_y = \sigma_z = 0.5m$, $\sigma_\theta = 0.5\,rad$, $\sigma_{a_x} = \sigma_{a_y} = \sigma_{a_z} = 0.5\,m/s^2$, and $\sigma_{a_\theta} = 0.5\,rad/s^2$.

In addition, we have also investigated to the performance of our proposed 3D MOT in this study using the default covariance matrices for Q and R as used in AB3DMOT [7] baseline method. As we can see in Table IX, the performance of our 3D MOT using the default covariance matrices is not better than that of using the estimated covariance matrices that we proposed. However, this performance is still better than that of the AB3DMOT [7] baseline method thanks to our proposed AED and maximum skipped frames methods.

VI. Conclusion

In this work, we proposed a simple yet accurate, fast and robust 3D MOT system for real-time applications. We performed extensive experiments on the KITTI and nuScenes 3D MOT datasets. Our proposed system has shown competitive results against state of the art 3D MOT such as AB3DMOT [7], FANTrack [6] and mmMOT [5]. The proposed method also showed a very impressive processing speed. Additionally, we explored the impact of the preserving maximum skipped frames to recover the lost track affected by false negative detection. In future work, it will be interesting to investigate the evolutionary algorithm such as genetic algorithm in order to estimate automatically the covariance noise matrices and to estimate data association module instead of using classical Hungarian method.

Acknowledgment

The authors gratefully acknowledge the support of *Infrastructure Digitale de Demain* and *la chaire d'excellence RIVA de la région Hauts de France*. Also, they thank the authors of AB3DMOT [7] for inspiration.

References

[1] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Auto. Robots*, vol. 26, nos. 2–3, pp. 123–139, Apr. 2009, doi: 10.1007/s10514-009-9115-1.

[2] O. Shorinwa, J. Yu, T. Halsted, A. Koufos, and M. Schwager, "Distributed multi-target tracking for autonomous vehicle fleets," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 3495–3501, doi: 10.1109/ICRA40945.2020.9197241.

[3] J. Khan, S. Niar, A. Menhaj, Y. Elhillali, and J. L. Dekeyser, "An MPSoC architecture for the multiple target tracking application in driver assistant system," in *Proc. Int. Conf. Appl.-Specific Syst., Archit. Processors*, Leuven, Belgium, 2008, pp. 126–131, doi: 10.1109/ASAP.2008.4580166.

[4] M. Delavarian, O. Reza Marouzi, H. Hassanpour, R. M. Parizi, and M. S. Khan, "Multi-camera multiple vehicle tracking in urban intersections based on multilayer graphs," *IET Intell. Transp. Syst.*, vol. 14, no. 12, pp. 1673–1690, Dec. 2020, doi: 10.1049/iet-its.2020.0086.

[5] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy, "Robust multi-modality multi-object tracking," in *Proc. ICCV*, Oct. 2019, pp. 2365–2374.

[6] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki, "FANTrack: 3D multi-object tracking with feature association network," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1426–1433.

[7] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Oct. 2020, pp. 10359–10366, doi: 10.1109/IROS45743.2020.9341164.

[8] 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.*, Jun. 2012, pp. 3354–3361.

[9] H. Caesar *et al.*, "NuScenes: A multimodal dataset for autonomous driving," 2019, *arXiv:1903.11027*. [Online]. Available: http://arxiv.org/abs/1903.11027

[10] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker, "Deep network flow for multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6951–6960.

[11] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[12] T. Hu, L. Huang, and H. Shen, "Multi-object tracking via end-to-end tracklet searching and ranking," 2020, *arXiv:2003.02795*. [Online]. Available: http://arxiv.org/abs/2003.02795

[13] T. Gao, H. Pan, Z. Wang, and H. Gao, "A CRF-based framework for tracklet inactivation in online multi-object tracking," *IEEE Trans. Multimedia*, early access, Mar. 1, 2021, doi: 10.1109/TMM.2021.3062489.

[14] H. Shen, L. Huang, C. Huang, and W. Xu, "Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking," 2018, *arXiv:1808.01562*. [Online]. Available: http://arxiv.org/abs/1808.01562

[15] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[16] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1820–1833, Sep. 2011.

[17] J. Xiao, R. Stolkin, M. Oussalah, and A. Leonardis, "Continuously adaptive data fusion and model relearning for particle filter tracking with multiple features," *IEEE Sensors J.*, vol. 16, no. 8, pp. 2639–2649, Apr. 2016, doi: 10.1109/JSEN.2016.2514704.

[18] P. G. Bhat, B. N. Subudhi, T. Veerakumar, V. Laxmi, and M. S. Gaur, "Multi-feature fusion in particle filter framework for visual tracking," *IEEE Sensors J.*, vol. 20, no. 5, pp. 2405–2415, Mar. 2020, doi: 10.1109/JSEN.2019.2954331.

[19] 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.

[20] J. Yin, W. Wang, Q. Meng, R. Yang, and J. Shen, "A unified object motion and affinity model for online multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6768–6777.

[21] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "GNN3DMOT: Graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6499–6508.

[22] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep. NC 27599-317524, Jul. 2006. [Online]. Available: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

[23] S. Formentin and S. Bittanti, "An insight into noise covariance estimation for Kalman filter design," *IFAC Proc. Volumes*, vol. 47, no. 3, pp. 2358–2363, 2014.

[24] A. Nez, L. Fradet, F. Marin, T. Monnet, and P. Lacouture, "Identification of noise covariance matrices to improve orientation estimation by Kalman filter," *Sensors*, vol. 18, no. 10, p. 3490, 2018.

[25] Y. Kim and H. Bang, "Introduction to Kalman filter and its applications," in *Introduction Implementations Kalman Filter*. Rijeka, Croatia: IntechOpen, 2019, doi: 10.5772/intechopen.80600.

[26] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE Access*, vol. 7, pp. 104423–104434, 2019, doi: 10.1109/ACCESS.2019.2932301.
[27] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.
[28] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3D object detection," in *Proc. CVPR*, 2019, pp. 1–8.
[29] S. Tang, M. Andriluka, B. Andres, and B. Schiele, "Multiple people tracking by lifted multicut and person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3539–3548.
[30] R. Henschel, L. Leal-Taixé, D. Cremers, and B. Rosenhahn, "Fusion of head and full-body detectors for multi-object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 1428–1437.
[31] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 300–311.
[32] C. Ma *et al.*, "Trajectory factory: Tracklet cleaving and re-connection by deep Siamese Bi-GRU for multiple object tracking," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, San Diego, CA, USA, Jul. 2018, pp. 1–6.
[33] B. Lee, E. Erdenee, S. Jin, M. Y. Nam, Y. G. Jung, and P. K. Rhee, "Multi-class multi-object tracking using changing point detection," in *Computer Vision*, G. Hua and H. Jégou, Eds. Cham, Switzerland: Springer, 2016, pp. 68–83.
[34] C. Kim, F. Li, and M. James Rehg, "Multi-object tracking with neural gating using bilinear LSTM," in *15th Eur. Conf. Computer Vis.*, Munich, Germany, Sep. 2018, pp. 208–224.
[35] G. P. Mauroy and E. W. Kamen, "Multiple target tracking using recurrent neural networks," in *Proc. Int. Conf. Neural Netw.*, San Francisco, CA, USA, 2017, pp. 4225–4232.
[36] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Dec. 2008.