A Baseline for 3D Multi-Object Tracking

Xinshuo Weng
Robotics Institute
Carnegie Mellon University
xinshuow@cs.cmu.edu

Kris Kitani
Robotics Institute
Carnegie Mellon University
kkitani@cs.cmu.edu

Abstract: 3D multi-object tracking (MOT) is an essential component technology for many real-time applications such as autonomous driving or assistive robotics. However, recent works for 3D MOT tend to focus more on developing accurate systems giving less regard to computational cost and system complexity. In contrast, this work proposes a simple yet accurate real-time baseline 3D MOT system. We use an off-the-shelf 3D object detector to obtain oriented 3D bounding boxes from the LiDAR point cloud. Then, a combination of 3D Kalman filter and Hungarian algorithm is used for state estimation and data association. Although our baseline system is a straightforward combination of standard methods, we obtain the state-of-the-art results. To evaluate our baseline system, we propose a new 3D MOT extension to the official KITTI 2D MOT evaluation along with two new metrics. Our proposed baseline method for 3D MOT establishes new state-of-the-art performance on 3D MOT for KITTI, improving the 3D MOTA from 72.23 of prior art to 76.47. Surprisingly, by projecting our 3D tracking results to the 2D image plane and compare against published 2D MOT methods, our system places 2nd on the official KITTI leaderboard. Also, our proposed 3D MOT method runs at a rate of 214.7 FPS, 65× faster than the state-of-the-art 2D MOT system. Our code is publicly available at https://github.com/xinshuoweng/AB3DMOT

Keywords: 3D multi-object tracking, real-time, evaluation metrics

1 Introduction

Multi-object tracking (MOT) is an essential component technology for many vision applications such as autonomous driving [1, 2, 3], robot collision prediction [4, 5] and video face alignment [6, 7, 8]. Due to the significant advance in object detection [9, 10, 11, 12, 13, 14, 15, 16, 17], there has been much progress on MOT. For example, for the car class on the KITTI [18] MOT benchmark, the MOTA (multi-object tracking accuracy) has improved from 57.03 [19] to 84.24 [20] in two years.

Although the accuracy has been significantly improved, it has come at the cost of increasing system complexity and computational cost. Complex systems make modular analysis challenging and it is not always clear which part of the system contributes the most to performance. For example, leading works [21, 22, 23, 24] have substantial different system pipelines but only minor differences in performance. Also, the adverse effect of increased computational cost is obvious in [20, 25, 22, 21]. Despite having excellent accuracy, real-time tracking is out of reach.

In contrast to prior work which tends to focus more on accuracy over system complexity and computational cost, this work aims to develop an accurate, simple and real-time 3D MOT system. We show that our proposed system which combines the minimal components for 3D MOT works extremely well. On the KITTI dataset, our system establishes new state-of-the-art performance on 3D MOT. Surprisingly, if we project our 3D tracking results to the 2D image plane and compare against all published 2D MOT methods, our system places 2nd on the official KITTI leaderboard as shown in Figure 1. In addition, due to the simplicity of our system, it can run at a rate of 214.7 FPS on KITTI test set, 65 times faster than the state-of-the-art MOT system BeyondPixels [20]. When comparing against other real-time MOT systems such as Complexer-YOLO [26], LP-SSVM [27], 3D-CNN/PMBM [23], and MCMOT-CPD [28], our system is at least twice as fast and achieves much higher accuracy. We hope that our system will serve as a simple yet strong baseline on which others can easily build to advance the state-of-the-art in 3D MOT.

Technically, we use an off-the-shelf 3D object detector to obtain oriented 3D bounding boxes from the LiDAR point cloud. Then, a combination of 3D Kalman filter (with a constant velocity model) and Hungarian algorithm is used for state estimation and data association. While the combination of modules is standard, we are able to obtain state of the art results. Also, unlike previous 3D MOT systems which often define the state space of the Kalman filter in 2D image space [29] or bird's eye view [30], we extend the state space of the Kalman filter to full 3D domain, including 3D location, size, velocity and orientation of the objects.

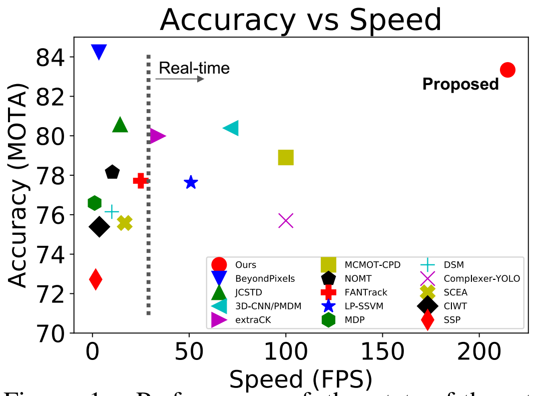


Figure 1: Performance of the state-of-the-art MOT systems and our proposed one on KITTI 2D MOT test set. The higher and more right is better.

In addition, we observe two drawbacks for current 3D MOT evaluation: *(1) Standard MOT benchmark such as the KITTI dataset only supports for 2D MOT evaluation*, *i.e.*, evaluation on the image plane. A tool for evaluating 3D MOT systems directly in 3D space is not currently available. The current convention for 3D MOT evaluation is to project the 3D trajectory outputs to the 2D image plane and evaluate on the KITTI 2D MOT benchmark. However, we believe that this will hamper the future progress of 3D MOT systems because evaluating on the image plane cannot demonstrate the full strength of 3D localization and tracking. To overcome the issue, we offer a 3D extension of the KITTI 2D MOT evaluation, which we call the KITTI-3DMOT evaluation tool; *(2) Common MOT metrics such as the MOTA and MOTP do not consider the confidence of trajectory,* which thus leads to a single confidence threshold selection in order to filter out false positives. Therefore, it cannot robustly reflect the full spectrum of accuracy and precision across different thresholds. To address this issue, we propose two new metrics – AMOTA and AMOTP (average MOTA and MOTP) – to summarize performance across all thresholds. The code for our proposed KITTI-3DMOT evaluation tool along with the new metrics is released and we hope that future 3D MOT systems will use it as a standard evaluation tool.

Our contributions are summarized as follows: (1) we offer a simple yet accurate 3D MOT baseline system for online and real-time applications; (2) a 3D extension of the official KITTI 2D MOT evaluation is implemented for 3D MOT evaluation; (3) two new metrics are proposed for robust MOT evaluation; (4) our system establishes new state-of-the-art performance on the proposed KITTI-3DMOT evaluation tool and places 2nd among all published works on the official KITTI leaderboard while achieving the fastest speed. We emphasize here that we do not claim that our 3D MOT system has significant algorithmic novelty over prior works, in spite of better results and higher speed. As stated above, we hope that our system can serve as a simple and solid baseline on which others can easily build on to advance the state-of-the-art in 3D MOT.

2 Related Works

2D Multi-Object Tracking. Recent 2D MOT systems can be mostly split into two categories based on the data association: batch and online methods. The batch methods attempt to find the global optimal solution from the entire sequence. They often create a network flow graph and can be solved by the min-cost flow algorithms [31, 32]. On the other hand, the online methods consider only the detection at current frame and are usually efficient for real-time application. These methods often formulate the data association as a bipartite graph matching problem and solve it using the Hungarian algorithm [29, 33, 34]. Beyond using the Hungarian algorithm in a post-processing step, modern online methods design the deep association networks [35, 36] that are able to construct the association using neural networks. Our MOT system also belongs to online methods. For simplicity and real-time efficiency, we adopt the original Hungarian algorithm without using neural networks.

Independent of the data association, designing a proper cost function for affinity measure is also crucial to the MOT system. Early works [37, 31] employ hand-crafted features such as spatial distance and color histograms as the cost function. Instead, modern methods apply the motion model [29, 38, 34, 39] and learn the appearance features [29, 38, 40]. In contrast to prior works which combine both appearance and motion models in a complicated way, we choose to employ only the simplest motion model, *i.e.*, constant velocity, without using appearance model.

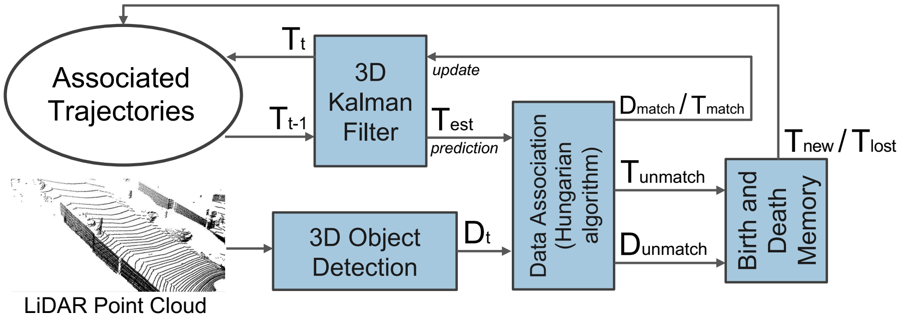


Figure 2: **System Pipeline**: (1) 3D detection module provides the bounding boxes D_t from the LiDAR point cloud; (2) 3D Kalman filter predicts the state of trajectories T_{t-1} to current frame t as T_{est} during the prediction step; (3) the detections D_t and trajectories T_{est} are associated using the Hungarian algorithm; (4) the state of matched trajectories T_{match} is updated based on the corresponding measurement D_{match} to obtain the T_t; (5) the unmatched detections D_{unmatch} and trajectories T_{unmatch} are used to create new trajectories T_{new} and delete disappeared trajectories T_{lost}, respectively.

3D Multi-Object Tracking. Most 3D MOT systems share the same components with the 2D MOT systems. The only distinction lies in that the detection boxes are in 3D space instead of the image plane. Therefore, it has the potential to design the motion and appearance models in 3D space without perspective distortion. [23] proposes an image-based method which estimates the location of objects in image space and also their distance to camera in 3D. Then a Poisson multi-Bernoulli mixture filter is used to estimate the 3D velocity of the objects. [30] applies an unscented Kalman filter in the bird's eye view to estimate not only the 3D velocity but also the angular velocity. [41] proposes a 2D-3D Kalman filter to jointly utilize the observation from the image and 3D world. Instead of using the hand-crafted filters, [21, 22] design Siamese networks to learn the filters from data. Unlike previous works which use complicated filters, our proposed system employs only the original Kalman filter [42] for simplicity, but extends its state space to full 3D domain, including not only 3D velocity but also the 3D size, 3D location and heading angle of the objects.

3D Object Detection. As an indispensable component of the 3D MOT, the quality of the detected 3D bounding box matters. Prior works mainly focus on processing the LiDAR point cloud inputs. [1, 14] divide the point cloud into equally-spaced 3D voxels and apply 3D CNNs for 3D bounding box prediction. [15] converts the point cloud to a bird's eye view representation for efficiency and exploit 2D convolutions. Other works such as [13] directly process the point cloud inputs using PointNet++ [43] for 3D detection. In addition, instead of using the point cloud, [16, 17] achieve the 3D detection from only a single image.

3 Approach

The goal of 3D MOT is to associate the detected 3D bounding boxes in a sequence. As our system is an online method, at every frame, we require only the detection at the current frame and associated trajectories from the previous frame. Our system pipeline is illustrated in Figure 2, which is composed of: (1) 3D detection module provides the bounding boxes from the LiDAR point cloud; (2) 3D Kalman filter predicts the object state to the current frame; (3) data association module matches the detection with predicted trajectories; (4) 3D Kalman filter updates the object state based on the measurement; (5) birth and death memory controls the newly appeared and disappeared trajectories.

Following [13, 17, 21], we parameterize the 3D bounding box as a set of eight parameters, including the 3D coordinate of the object center (x, y, z), object's size (l, w, h), heading angle θ and its confidence s. Except for the 3D object detection module, our 3D MOT system does not need any training process and can be directly applied for inference.

3.1 3D Object Detection

Thanks to the recent advance in 3D object detection, we can take advantage of high-quality detection from many successful detectors. Here, we experiment with two state-of-the-art 3D detectors [13, 17] on the KITTI dataset. We directly adopt their models pre-trained on the training set of the KITTI 3D object detection benchmark. At frame t, the output of the 3D detection module is a set of detections $D_t = \{D_t^1, D_t^2, \cdots, D_t^{n_t}\}$ (n_t is the number of detections which can vary across frames). Each detection D_t^i is represented as a tuple $(x, y, z, l, w, h, \theta, s)$ as mentioned above. We will show how different 3D detection modules affect the performance of our 3D MOT system in the ablation study.

3.2 3D Kalman Filter —– State Prediction

To predict the object state in the next frame, we approximate the inter-frame displacement of objects using the constant velocity model, independent of camera ego-motion. In detail, we formulate the state of object trajectory as a 10-dimensional vector $T = (x, y, z, \theta, l, w, h, v_x, v_y, v_z)$, where the additional variables v_x, v_y, v_z represent the velocity of objects in 3D space. Here, we do not include the angular velocity v_θ in the state space as we observe empirically that including angular velocity will lead to inferior performance.

At every frame, all associated trajectories from previous frame $T_{t-1} = \{T_{t-1}^1, T_{t-1}^2, \cdots, T_{t-1}^{m_{t-1}}\}$ (m_{t-1} is the number of existing trajectories at frame t-1) will be propagated to frame t, named as T_{est}, based on the constant velocity model:

$$x_{\text{est}} = x + v_x \quad y_{\text{est}} = y + v_y \quad z_{\text{est}} = z + v_z \tag{1}$$

As a result, for every trajectory T_{t-1}^j in T_{t-1}, the predicted state after propagation to frame t is $T_{\text{est}}^j = (x_{\text{est}}, y_{\text{est}}, z_{\text{est}}, \theta, l, w, h, v_x, v_y, v_z)$ in T_{est}, which will be fed to the data association module.

3.3 Data Association

To match the detections D_t with predicted trajectories T_{est}, we apply the Hungarian algorithm [33]. The affinity matrix with dimension of $n_t \times m_{t-1}$ is computed using the 3D IoU between every pair of detection D_t^i and T_{est}^j. Then, the bipartite graph matching problem can be solved in polynomial time with the Hungarian algorithm. In addition, we reject the matching when the 3D IoU is less than IoU$_{\text{min}}$. The outputs of the data association module are a set of detections D_{match}=$\{D_{\text{match}}^1, D_{\text{match}}^2, \cdots, D_{\text{match}}^{w_t}\}$ matched with trajectories T_{match}=$\{T_{\text{match}}^1, T_{\text{match}}^2, \cdots, T_{\text{match}}^{w_t}\}$, along with the unmatched trajectories T_{unmatch}=$\{T_{\text{unmatch}}^1, T_{\text{unmatch}}^2, \cdots, T_{\text{unmatch}}^{m_t - w_t}\}$ and unmatched detections D_{unmatch}=$\{D_{\text{match}}^1, D_{\text{match}}^2, \cdots, D_{\text{match}}^{n_t - w_t}\}$, where w_t is the number of matches.

3.4 3D Kalman Filter —– State Update

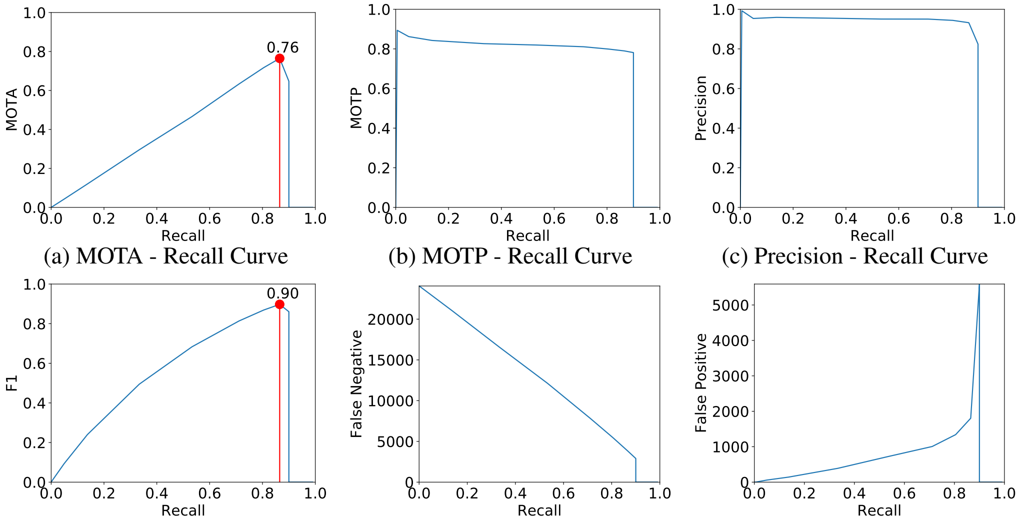
To account for the uncertainty in T_{match}, we update the entire state space of each trajectory in T_{match} based on its corresponding measurement, *i.e.*, matched detection in D_{match}, and obtain the final associated trajectories $T_t = \{T_t^1, T_t^2, \cdots, T_t^{w_t}\}$ at frame t. Following the Bayes rule, the updated state of each trajectory $T_t^i = (x', y', z', \theta', l', w', h', v_x', v_y', v_z')$ is the weighted average between the state space of T_{match}^i and D_{match}^i where the weights are determined by the uncertainty of both the matched trajectory T_{match}^i and detection D_{match}^i (please refer to the Kalman filter [42] for details).

In addition, we observe that the naive weighted average does not work well for orientation. For a matched object i, the orientation of its detection D_{match}^i can be nearly opposite to the orientation of its trajectory T_{match}^i, *i.e.*, differ by π. However, this is impossible based on the assumption that objects should move smoothly and cannot change the orientation by π in one frame (*i.e.*, 0.1s in KITTI). Therefore, the orientation in D_{match}^i or T_{match}^i must be wrong. As a result, the averaged trajectory T_t^i of this object will have an orientation between D_{match}^i and T_{match}^i which leads to a low 3D IoU with the ground truth. To prevent this issue, we propose an orientation correction technique. When the difference of orientation in D_{match}^i and T_{match}^i is greater than $\frac{\pi}{2}$, we add a π to the orientation in T_{match}^i so that its orientation can be roughly consistent with D_{match}^i. We will show the effect of the orientation correction in the ablation study.

3.5 Birth and Death Memory

As the existing objects might disappear and new objects might enter, a module to manage the birth and death of the trajectories is necessary. On one hand, we consider all unmatched detections D_{unmatch} as potential objects entering the image. To avoid tracking of false positives, a new trajectory T_{new}^i will not be created for D_{unmatch}^i until it has been continually detected in the next F$_{\text{min}}$ frames. Once the new trajectory is successfully created, we initialize the state of the trajectory T_{new}^i same as its most recent measurement D_{unmatch}^i with zero velocity for v_x, v_y and v_z.

On the other hand, we consider all unmatched trajectories T_{unmatch} as potential objects leaving the image. To avoid deleting true positive trajectories which have missing detection at certain frames, we keep tracking each unmatched trajectory T_{unmatch}^i for Age$_{\text{max}}$ frames before deleting it from the



(a) MOTA - Recall Curve (b) MOTP - Recall Curve (c) Precision - Recall Curve

(d) F1 - Recall Curve (e) False Negative - Recall Curve (f) False Positive - Recall Curve

Figure 3: **The effect of the confidence threshold $s_{\textbf{thres}}$** on MOT metrics: MOTA, MOTP, Precision, F1, FN and FP. We evaluate our 3D MOT system (with IoU_{min}=0.1, F_{min}=3, Age_{max}=2) on the KITTI validation set using KITTI-3DMOT evaluation tool.

associated trajectories. Ideally, true positive trajectories with missing detection can be maintained and interpolated by our 3D MOT system, and only the trajectories that leave the image are deleted.

4 New MOT Evaluation Tool

4.1 KITTI-3DMOT Evaluation Tool

As one of the most important MOT benchmarks, KITTI [18] dataset is crucial to the progress of MOT systems. Although the KITTI dataset supports for extensive 2D MOT evaluation, *i.e.*, evaluation on the image plane. A tool for evaluating 3D MOT systems directly in 3D space is not currently available. As a result, the current convention for 3D MOT evaluation is to project the 3D trajectory outputs to the image plane and match the projected 2D trajectory outputs with the 2D ground truth trajectories using 2D IoU as the cost function. However, we believe that evaluating on the 2D image plane cannot demonstrate the full strength of 3D localization and tracking.

To better evaluate 3D MOT systems, we break the convention and implement a 3D extension of the official KITTI 2D MOT evaluation, which we call the KITTI-3DMOT evaluation tool. Specifically, we modify the pairwise cost function from 2D IoU to 3D IoU and match the 3D trajectory outputs directly with the 3D ground truth trajectories. Therefore, no projection is needed anymore for 3D MOT evaluation. For every trajectory, a minimum 3D IoU of 0.25 with the ground truth is required in order to be considered as a successful match. We hope that future 3D MOT systems will use the proposed KITTI-3DMOT evaluation tool as a standard.

4.2 New MOT Evaluation Metrics: Average MOTA/MOTP (AMOTA/AMOTP)

Conventional MOT evaluation reports results on extensive metrics such as MOTA (see Section 5.1 for details), MOTP, FP, FN, Precision, F1 score, IDS, FRAG, *etc*. However, none of these metrics consider the confidence[1] of trajectory. As a result, one has to choose a single confidence threshold prior to the evaluation in order to filter out tracking of false positives. Here, we show results of our 3D MOT system on six MOT metrics using different thresholds in Figure 3. We can see that, in Figure 3 (f), the number of false positives increases drastically when the recall is beyond 0.8, *i.e.*, a very low threshold is used. Also, the threshold cannot be set too high either, as it will result in a very low recall and a huge number of false negatives, shown in Figure 3 (e).

As a result, due to the fact that current metrics do not consider the confidence, a MOT system must select a single and proper threshold, *e.g.*, the red dot in Figure 3 (a), in order to achieve its highest MOTA[2] and ranking. Although fixing the system with a single threshold is sufficient for deployment,

[1] We define the confidence of a trajectory as the average confidence of its detection at all frames it appears.

[2] $\text{MOTA} = 1 - \frac{\text{FP+FN+IDS}}{\text{num}_{\text{gt}}}$ is the metric used for ranking methods on most MOT benchmarks.

using a single threshold for evaluation will hamper the future progress of MOT systems as the full spectrum of accuracy and precision cannot be reflected at a single threshold. In other words, a MOT system that achieves high MOTA only at a single threshold but extremely low MOTA at other thresholds can still stand out using current metrics. But ideally, researchers should move forward with developing a MOT system achieving as high MOTA as possible across all thresholds.

To mitigate the above issues, we propose two new metrics, which we call the AMOTA and AMOTP (average MOTA and MOTP), to summarize the MOTA and MOTP across all thresholds instead of using a single threshold. Similar to the average precision in object detection, the AMOTA and AMOTP are computed by integrating the MOTA and MOTP over recall curve. Following the KITTI object detection benchmark, we use the 11-point interpolation method to approximate the integration. We hope the proposed metric AMOTA, that can summarize the MOTA-recall curve across all thresholds and is thus more robust, will serve as the metric to rank MOT systems in the future.

5 Experiment

5.1 Settings

Dataset. We evaluate on the KITTI MOT benchmark, which is composed of 21 training and 29 testing video sequences. For each sequence, the LiDAR point cloud, RGB images along with the calibration file are provided. The number of frames is 8008 and 11095 for training and testing. For the testing split, KITTI does not provide any annotation to users but reserve the annotation on the server for 2D MOT evaluation. For the training split, the number of the annotated objects and trajectories are 30601 and 636 respectively, including car, pedestrian, cyclist and *etc*. As our tracking system does not require training, we use all 21 training sequences for validation. Also, in this work, we only evaluate on the car subset, which has the most number of instances among all object types.

Evaluation Metric. In addition to the proposed metrics AMOTA and AMOTP, we also evaluate on the conventional MOT metrics in order to compare with existing MOT systems, including the MOTA (multi-object tracking accuracy), MOTP (multi-object tracking precision), MT (number of mostly tracked trajectories), ML (number of mostly lost trajectories), IDS (number of identity switches), FRAG (number of trajectory fragmentation interrupted by the false negatives), FPS (frame per second) and FP (number of false positives) and FN (number of false negatives).

Baselines. For 3D MOT, since state-of-the-art methods such as the FANTrack [21], DSM [22], FaF [1] and Complexer-YOLO [26] have not released the code yet, we choose to reproduce the results of two representative 3D MOT systems: FANTrack [21] (the most accurate 3D MOT system on KITTI) and Complexer-YOLO [26] (the fastest 3D MOT system on KITTI), for comparison.

For 2D MOT, we compare with eight best-performed methods on the KITTI dataset, including BeyondPixels [20], JCSTD [25], 3D-CNN/PMBM [23], extraCK [24], MCMOT-CPD [28], NOMT [38], LP-SSVM [27] and MDP [44].

Implementation Details. For our final system shown in Table 1, 2 and 3, we use [13] pre-trained on the training set of the KITTI 3D object detection benchmark as our 3D object detection module, (x, y, z, θ, l, w, h, v_x, v_y, v_z) as the state space without including the angular velocity v_θ, IoU$_{\text{min}}$=0.1 in the data association module, F_{min}=3 and Age$_{\text{min}}$=2 in the birth and death memory module.

5.2 Experimental Results

Quantitative Comparison on KITTI-3DMOT Benchmark. We summarize the results of state-of-the-art 3D MOT systems and our proposed system in Table 1. The results are evaluated using the proposed KITTI-3DMOT evaluation tool. As the KITTI dataset does not release the annotation for the test set, we can only evaluate on the validation set when using the proposed KITTI-3DMOT evaluation tool. Our 3D MOT system consistently outperforms others (except for the MT and ML metrics where we are the second best), establishing new state-of-the-art performance on KITTI 3D MOT and improving the 3D MOTA from 72.23 of prior art to 76.47. Also, we achieve an impressive zero identity switch. In addition, our system is faster than other 3D MOT systems and does not require any GPU. We notice that, although the 3D MOTA of [21] is slightly worse than [26], [21] has a significantly higher 3D AMOTA than [26], suggesting that [21] has much higher overall performance across all thresholds than [26].

Quantitative Comparison on KITTI-2DMOT Benchmark. In addition to evaluate our 3D MOT system using the proposed KITTI-3DMOT evaluation tool, we can also project the 3D trajectory outputs of our 3D MOT system to the 2D image plane and report the 2D MOT results on the test set.

Table 1: Quantitative comparison on KITTI **validation** set for car subset using the proposed **KITTI-3DMOT** evaluation tool. We **bold** the best results for each metric.

Method	**AMOTA** (%) ↑	AMOTP (%) ↑	MOTA (%) ↑	MOTP (%) ↑	MT (%) ↑	ML (%) ↓	IDS ↓	FRAG ↓	FPS ↑
Complexer-YOLO [26]	34.76	68.20	72.23	73.42	61.09	**4.96**	985	1583	93.6
FANTrack [21]	37.79	73.02	72.18	75.62	**72.78**	8.86	13	71	23.1 (GPU)
Ours	**39.44**	**74.60**	**76.47**	**78.98**	69.86	7.27	**0**	**58**	**207.4**

Table 2: Quantitative comparison on KITTI **test** set for car subset using the official KITTI 2D MOT evaluation. We color the best results in red and the second best in blue for each metric.

Method	Type	**MOTA** (%) ↑	MOTP (%) ↑	MT (%) ↑	ML (%) ↓	IDS ↓	FRAG ↓	FPS ↑
Complexer-YOLO [26]	3D	75.70	78.46	58.00	**5.08**	1186	2092	**100.0**
DSM [22]	3D	76.15	83.42	60.00	8.31	296	868	10.0 (GPU)
MDP [44]	2D	76.59	82.10	52.15	13.38	130	387	1.1
LP-SSVM [27]	2D	77.63	77.80	56.31	8.46	62	539	50.9
FANTrack [21]	3D	77.72	82.32	62.61	8.76	150	812	25.0 (GPU)
NOMT [38]	2D	78.15	79.46	57.23	13.23	**31**	**207**	10.3
MCMOT-CPD [28]	2D	78.90	82.13	52.31	11.69	228	536	**100.0**
extraCK [24]	2D	79.99	82.46	62.15	5.54	343	938	33.9
3D-CNN/PMBM [23]	2.5D	80.39	81.26	62.77	6.15	121	613	71.4
JCSTD [25]	2D	80.57	81.81	56.77	7.38	61	643	14.3
BeyondPixels [20]	2D	**84.24**	**85.73**	**73.23**	**2.77**	468	944	3.3
Ours	3D	**83.34**	**85.23**	**65.85**	11.54	**10**	**222**	**214.7**

As the KITTI dataset does not release the annotation for the test set, we cannot compute the results of the proposed AMOTA and AMOTP metrics on the test set. Therefore, only the conventional MOT metrics are shown in Table 2. Surprisingly, among all existing MOT systems on the KITTI dataset, we place 2nd and are only behind the state-of-the-art 2D MOT system BeyondPixels [20] by 0.9 MOTA. Also, our 3D MOT system runs at a rate of 214.7 FPS, which is 65 times faster than BeyondPixels [20]. When comparing with other real-time MOT systems such as MCMOT-CPD [28], LP-SSVM [27], Complexer-YOLO [26] and 3D-CNN/PMBM [23], our system is at least twice as fast and achieves much higher accuracy.

Qualitative Comparison. We visualize the results of our system and previous state-of-the-art 3D MOT system FANTrack [21] on one example sequence of the KITTI test set in Figure 4. We can see that the results of FANTrack (left) contain a few identity switch (*i.e.*, color changing) and missing detections for faraway objects while our system (right) does not have these issues. In the video demo of the supplementary material, we provide detailed qualitative comparison on more sequences and show that (1) our system, which does not require training, does not have the issue of over-fitting while the deep learning based method FANTrack [21] clearly over-fits on several test sequences and (2) our system has much less identity switch, box jittering and flickering than FANTrack [21].

5.3 Ablation Study

Effect of the 3D Detection Quality. In Table 3 (a), we switch the 3D detection module to [17] instead of using [13] in (k). The distinction lies in that [13] requires the LiDAR point cloud while [17] requires only a single image. As a result, [13] provides much higher 3D detection quality than [17] (see [13, 17] for details). We can see that the tracking performance in (k) is significantly better than (a) in all metrics, meaning that the 3D detection quality is crucial to a 3D MOT system.

3D v.s. 2D Kalman Filter. In Table 3, we replace the 3D Kalman filter in the proposed system (k) with a 2D Kalman filter (b), *i.e.*, 3D trajectory outputs are produced by associating 3D detection boxes in 2D space. Specifically, we define the state space of a trajectory T=$(x, y, s, r, v_x, v_y, v_s)$, where ($x$, y) is the object's 2D location, s is the 2D box area, r is the aspect ratio and (v_x, v_y, v_s) is the velocity in the 2D image plane. In Table 3 (b)(k), we observe that using 3D Kalman filter in (k) reduces the IDS from 27 to 0 and FRAG from 122 to 58, which we believe that it is because association in 3D space with depth information can help resolve the depth ambiguity existing in association in 2D space. Overall, the AMOTA is improved from 38.06 in (b) to 39.44 in (k).

Effect of Including the Angular Velocity v_θ in the State Space. We add one more variable v_θ to the state space so that the state space of a trajectory $T = (x, y, z, \theta, l, w, h, v_x, v_y, v_z, v_\theta)$ in (c). In Table 3, we observe that adding v_θ in (c) leads to significant drop in AMOTP and AMOTP, compared to (k). We understand that this might be counter-intuitive. But we found that, most instances of car do not have angular velocity in the KITTI dataset, *i.e.*, they are either static or moving straight. As a result, adding angular velocity introducing unnecessary noise to the orientation in practice.

Effect of the Orientation Correction. As mentioned in Section 3.4, we use the orientation correction in our final model, shown in Table 3 (k). Here, we experiment a variant without using the

Table 3: Ablative analysis on KITTI **validation** set for car evaluated by the proposed **KITTI-3DMOT** evaluation tool. We show individual effect of using different 3D detection, 3D v.s. 2D Kalman Filter, including angular velocity in state space, hyper-parameters IoU_{min}, F_{min}, Age_{max} and orientation correction. We **bold** the best results for each metric.

Method	**AMOTA** (%) ↑	AMOTP (%) ↑	MOTA (%) ↑	MOTP (%) ↑	IDS ↓	FRAG ↓	FN ↓	FP ↓
(a) with detection from [17]	28.84	51.28	56.80	68.43	3	73	2788	7713
(b) with 2D Kalman Filter	38.06	**74.99**	75.92	79.48	27	122	2007	3762
(c) with angular velocity v_θ	33.52	67.26	76.40	78.58	**0**	59	1815	3865
(d) no orientation correction	33.25	66.96	75.75	77.85	**0**	129	1913	3925
(e) $\text{IoU}_{\text{min}} = 0.01$	39.31	74.52	**77.22**	78.79	**0**	53	1777	3706
(f) $\text{IoU}_{\text{min}} = 0.25$	33.83	67.72	73.11	79.37	20	79	1941	4512
(g) $F_{\text{min}} = 1$	32.56	74.44	76.17	78.81	98	165	2310	**3328**
(h) $F_{\text{min}} = 5$	34.38	67.55	75.55	79.05	2	58	**1624**	4258
(i) $\text{Age}_{\text{max}} = 1$	36.11	67.71	76.76	**79.53**	**0**	101	1250	4344
(j) $\text{Age}_{\text{max}} = 3$	36.82	74.49	74.98	78.78	**0**	**48**	2355	3668
(k) **Ours**	**39.44**	74.60	76.47	78.98	**0**	58	1804	3859

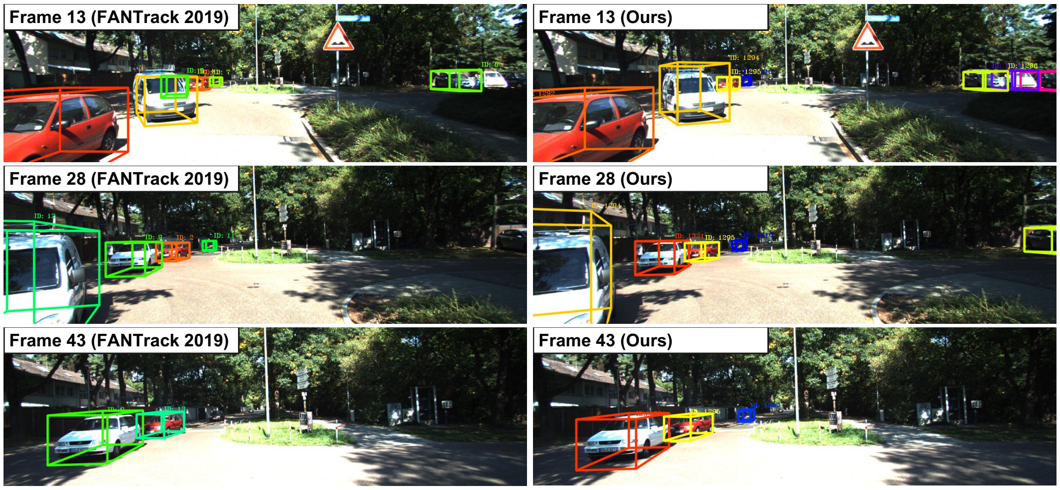


Figure 4: Qualitative comparison of the FANTrack [21] (**left**) and our system (**right**) on sequence 3 of the KITTI **test** set. Our system does not have identity switch or missing detections.

orientation correction in Table 3 (d). We observe that the orientation correction help improve the performance in all metrics, suggesting that this trick should be applied to all future 3D MOT systems.

Effect of the Threshold IoU_{min} in the Data Association. In Table 3, we change the IoU_{min}=0.1 in (k) to IoU_{min}=0.01 in (e) and IoU_{min}=0.25 in (f). We observe that decreasing the IoU_{min} to 0.01 slightly increases the MOTA from 76.47 in (k) to 77.22 in (e). However, both (e) and (f) perform worse than (k) in terms of the AMOTA and AMOTP.

Effect of the Minimum Frames F_{min} for New Trajectory. We adjust the F_{min}=3 in (k) to F_{min}=1 in (g) and F_{min}=5 in (h). The results are shown in Table 3. We can see that using either F_{min}=1 (*i.e.*, creating a new trajectory immediately for an unmatched detection) or F_{min}=5 (*i.e.*, creating a new trajectory after an unmatched detection is continually detected in following five frames) leads to inferior performance in AMOTA and AMOTP, suggesting that F_{min}=3 is appropriate.

Effect of the Age_{max} for Lost Trajectory. We verify the effect of the hyper-parameter Age_{max} by decreasing it to Age_{max}=1 in (i) and increasing it to Age_{max}=3 in (j). We show that both (i) and (j) result in a drop in AMOTA and AMOTP, and it seems like that Age_{max}=2 (*i.e.* keep tracking the unmatched trajectories T_{unmatch} for following two frames) in our final model (k) is the best choice.

6 Conclusion

In this paper, we propose an accurate, simple and real-time baseline system for online 3D MOT. Also, a new evaluation tool along with the new metrics AMOTA and AMOTP is proposed for better 3D MOT evaluation. Through extensive experiments on the KITTI dataset, our system establishes new state-of-the-art performance on 3D MOT. Also, we places 2nd on the official KITTI 2D MOT leaderboard among all published works while achieving the fastest speed, suggesting that simple and accurate 3D MOT can lead to very good results in 2D. We hope that our system will serve as a simple yet solid baseline on which others can easily build to advance the state-of-the-art in 3D MOT.

References

[1] W. Luo, B. Yang, and R. Urtasun. Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net. *CVPR*, 2018.

[2] S. Wang, D. Jia, and X. Weng. Deep Reinforcement Learning for Autonomous Driving. *arXiv:1811.11329*, 2018. URL `http://arxiv.org/abs/1811.11329`.

[3] S. Casas, W. Luo, and R. Urtasun. IntentNet: Learning to Predict Intention from Raw Sensor Data. *CoRL*, 2018.

[4] S. Kayukawa and K. Kitani. BBeep: A Sonic Collision Avoidance System for Blind Travellers and Nearby Pedestrians. *CHI*, 2019.

[5] A. Manglik, X. Weng, E. Ohn-bar, and K. M. Kitani. Future Near-Collision Prediction from Monocular Video: Feasibility, Dataset , and Challenges. *arXiv:1903.09102*, 2019. URL `http://arxiv.org/abs/1903.09102`.

[6] X. Weng and W. Han. CyLKs: Unsupervised Cycle Lucas-Kanade Network for Landmark Tracking. *arXiv:1811.11325*, 2018. URL `http://arxiv.org/abs/1811.11325`.

[7] X. Dong, S.-i. Yu, X. Weng, S.-e. Wei, Y. Yang, and Y. Sheikh. Supervision-by-Registration: An Unsupervised Approach to Improve the Precision of Facial Landmark Detectors. *CVPR*, 2018.

[8] J. S. Yoon and S.-i. Yu. Self-Supervised Adaptation of High-Fidelity Face Models for Monocular Performance Tracking. *CVPR*, 2019.

[9] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *NIPS*, 2015.

[10] T.-y. Lin, F. Ai, and P. Doll. Focal Loss for Dense Object Detection. *ICCV*, 2017.

[11] N. Lee, X. Weng, V. N. Boddeti, Y. Zhang, F. Beainy, K. Kitani, and T. Kanade. Visual Compiler: Synthesizing a Scene-Specific Pedestrian Detector and Pose Estimator. *arXiv:1612.05234*, 2016. URL `http://arxiv.org/abs/1612.05234`.

[12] X. Weng, S. Wu, F. Beainy, and K. Kitani. Rotational Rectification Network: Enabling Pedestrian Detection for Mobile Vision. *WACV*, 2018.

[13] S. Shi, X. Wang, and H. Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. *CVPR*, 2019.

[14] Y. Zhou and O. Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *CVPR*, 2018.

[15] B. Yang, M. Liang, and R. Urtasun. HDNET: Exploiting HD Maps for 3D Object Detection. *CoRL*, 2018.

[16] J. Ku, A. D. Pon, and S. L. Waslander. Monocular 3D Object Detection Leveraging Accurate Proposals and Shape Reconstruction. *CVPR*, 2019.

[17] X. Weng and K. Kitani. Monocular 3D Object Detection with Pseudo-LiDAR Point Cloud. *arXiv:1903.09847*, 2019. URL `http://arxiv.org/abs/1903.09847`.

[18] A. Geiger, P. Lenz, and R. Urtasun. Are We Ready for Autonomous Driving? the KITTI Vision Benchmark Suite. *CVPR*, 2012.

[19] J. H. Yoon, C. R. Lee, M. H. Yang, and K. J. Yoon. Online Multi-Object Tracking via Structural Constraint Event Aggregation. *CVPR*, 2016.

[20] S. Sharma, J. A. Ansari, J. K. Murthy, and K. M. Krishna. Beyond Pixels: Leveraging Geometry and Shape Cues for Online Multi-Object Tracking. *ICRA*, 2018.

[21] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki. FANTrack: 3D Multi-Object Tracking with Feature Association Network. *arXiv:1905.02843*, 2019.

[22] D. Frossard and R. Urtasun. End-to-End Learning of Multi-Sensor 3D Tracking by Detection. *ICRA*, 2018.

[23] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granstr. Mono-Camera 3D Multi-Object Tracking Using Deep Learning Detections and PMBM Filtering. *IV*, 2018.

[24] G. Gunduz and T. Acarman. A Lightweight Online Multiple Object Vehicle Tracking Method. *IV*, 2018.

[25] W. Tian, M. Lauer, and L. Chen. Online Multi-Object Tracking Using Joint Domain Information in Traffic Scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[26] M. Simon, K. Amende, A. Kraus, J. Honer, T. Sämann, H. Kaulbersch, S. Milz, and H. M. Gross. Complexer-YOLO: Real-Time 3D Object Detection and Tracking on Semantic Point Clouds. *CVPRW*, 2019.

[27] S. Wang and C. C. Fowlkes. Learning Optimal Parameters for Multi-Target Tracking with Contextual Interactions. *IJCV*, 2016. doi:10.1007/s11263-016-0960-z.

[28] B. Lee, E. Erdenee, S. Jin, and P. K. Rhee. Multi-Class Multi-Object Tracking Using Changing Point Detection. *ECCVW*, 2016.

[29] N. Wojke, A. Bewley, and D. Paulus. Simple Online and Realtime Tracking with a Deep Association Metric. *ICIP*, 2017.

[30] A. Patil, S. Malla, H. Gang, and Y.-T. Chen. The H3D Dataset for Full-Surround 3D Multi-Object Detection and Tracking in Crowded Urban Scenes. *ICRA*, 2019.

[31] L. Zhang, Y. Li, and R. Nevatia. Global Data Association for Multi-Object Tracking Using Network Flows. *CVPR*, 2008.

[32] S. Schulter, P. Vernaza, W. Choi, and M. Chandraker. Deep Network Flow for Multi-Object Tracking. *CVPR*, 2017.

[33] H. W Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 1955.

[34] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple Online and Realtime Tracking. *ICIP*, 2016.

[35] Y. Xu, Y. Ban, X. Alameda-Pineda, and R. Horaud. DeepMOT: A Differentiable Framework for Training Multiple Object Trackers. *arXiv:1906.06618*, 2019.

[36] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe. MOTS: Multi-Object Tracking and Segmentation. *CVPR*, 2019.

[37] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. *CVPR*, 2015.

[38] W. Choi. Near-Online Multi-Target Tracking with Aggregated Local Flow Descriptor. *ICCV*, 2015.

[39] C. Dicle, O. I. Camps, and M. Sznaier. The Way They Move: Tracking Multiple Targets with Similar Appearance. *ICCV*, 2013.

[40] S. H. Bae and K. J. Yoon. Robust Online Multi-Object Tracking Based on Tracklet Confidence and Online Discriminative Appearance Learning. *CVPR*, 2014.

[41] A. Osep, W. Mehner, M. Mathias, and B. Leibe. Combined Image- and World-Space Tracking in Traffic Scenes. *ICRA*, 2017.

[42] R. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 1960.

[43] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *NIPS*, 2017.

[44] Y. Xiang, A. Alahi, and S. Savarese. Learning to Track: Online Multi-Object Tracking by Decision Making. *ICCV*, 2015.