本文设计的方案，基本上就是最最简单的极其常见的C+L异步用KalmanFilter实现det level fusion tracking的方案，只不过用在了C+L detection level fusion上。除此之外，这里的detection level fusion实际上不是我们通常说的detection level fusion for tracking，而是detection level fusion for object detection，先得到fused detection之后再用fused detection做tracking。

EagerMOT: 3D Multi-Object Tracking via Sensor Fusion

Aleksandr Kim, Aljoša Ošep and Laura Leal-Taixé

***Abstract*— Multi-object tracking (MOT) enables mobile robots to perform well-informed motion planning and navigation by localizing surrounding objects in 3D space and time. Existing methods rely on depth sensors (*e.g.*, LiDAR) to detect and track targets in 3D space, but only up to a limited sensing range due to the sparsity of the signal. On the other hand, cameras provide a dense and rich visual signal that helps to localize even distant objects, but only in the image domain. In this paper, we propose EagerMOT, a simple tracking formulation that eagerly integrates all available object observations from both sensor modalities to obtain a well-informed interpretation of the scene dynamics. Using images, we can identify distant incoming objects, while depth estimates allow for precise trajectory localization as soon as objects are within the depth-sensing range. With EagerMOT, we achieve state-of-the-art results across several MOT tasks on the KITTI and NuScenes datasets. Our code is available at `https://github.com/aleksandrkim61/EagerMOT`**

I. Introduction

For safe robot navigation and motion planning, mobile agents need to be aware of surrounding objects and foresee their future states. To this end, they need to detect, segment, and – especially critical in close proximity of the vehicle – precisely localize objects in 3D space across time.

As shown by Weng and Kitani [35], even a simple method that relies on linear motion models and 3D overlap-driven two-frame data association yields a competitive tracking performance when using a strong LiDAR-based 3D object detector [30]. However, compared to their image-based counterparts, methods that rely on depth sensors are more sensitive to reflective and low-albedo surfaces, and can operate only within a limited sensing range due to the sparsity of the input signal. On the other hand, image-based methods leverage a rich visual signal to gain robustness to partial occlusions and localize objects with pixel-precision in the image domain, even when objects are too far away to be localized reliably in 3D space [34], [28]. However, 3D localization of the surrounding objects is vital in mobile robot scenarios.

In this paper, we present EagerMOT, a simple tracking framework that fuses all available object observations originating from 3D and 2D object detectors, to obtain a well-informed interpretation of the scene dynamics. Using cameras, our method identifies and maintains tracks in the image domain, while 3D detections allow for precise 3D trajectory localization as soon as objects enter the LiDAR sensing area. We achieve this via the two-stage association procedure. First, we associate object detections originating from different sensor modalities. Then, we employ a tracking formulation that allows us to update track states even when only partial (either image-based or LiDAR-based) object evidence is available. This way, our EagerMOT is robust to false negatives originating from different sensor modalities and can initialize object tracks before objects enter the depth-sensing range.

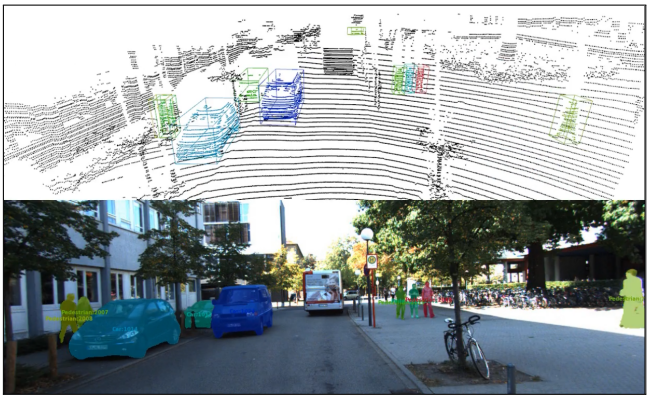

Fig. 1: Our method eagerly associates different sources of object detection/segmentation information (2D/3D detections, instance segmentation) when available to obtain an, as complete as possible, interpretation of the scene dynamics.

Our method is versatile enough to be applied to several different sensory configurations, such as LiDAR combined with a front-facing camera (as used in KITTI [13]), or combined with multiple cameras with non-overlapping view frustums (as employed in NuScenes [5]). With EagerMOT, we establish a new state-of-the-art on the large-scale NuScenes 3D MOT benchmark [5] and KITTI tracking benchmark [13] for 2D multi-object tracking and segmentation.

Our method merely assumes a mobile platform with a calibrated sensory setup equipped with a LiDAR and (possibly multiple) camera sensors. Given a pre-trained object detector for both sensor modalities, our method can be easily deployed on any mobile platform without additional training and imposes a minimal additional computational cost at the inference time.

In summary, **our contributions** are the following: (i) we propose a simple yet effective multi-stage data association approach that can leverage a variety of different object detectors, originating from potentially different modalities; (ii) we show that our approach can be applied to a variety of MOT tasks (2D/3D MOT and MOTS) and on different sensor configurations; and finally (iii), we perform a thorough analysis of our method, demonstrating through ablation studies the effectiveness of the proposed approach to data association and state-of-the-art results on three different benchmarks.

All authors are with the Technical University of Munich. E-mail: aleksandr.kim@tum.de, aljosa.osep@tum.de, leal.taixe@tum.de

II. Related Work

2D MOT. The majority of the existing vision-based tracking methods rely on recent advances in the field of object detection [27], [15] to detect and track objects in the image domain. TrackR-CNN [34] extends Mask R-CNN [15] with 3D convolutional networks to improve temporal consistency of the detector and uses object re-identification as a cue for the association. Tracktor [2] re-purposes the regression head of Faster R-CNN [27] to follow the targets. Similarly, CenterTrack [42] augments the object detector [43] with an offset-regression head used for cross-frame association. Recent trends are going in the direction of end-to-end learning [37], [12] and learning to associate using graph neural networks [4], [36].

3D MOT. Early methods for LiDAR-based multi-object tracking first perform bottom-up segmentation of LiDAR scans, followed by segment association and track classification [33], [21]. Due to recent advances in point cloud representation learning [24], [25] and 3D object detection [9], [31], [30], LiDAR and stereo-based tracking-by-detection has recently been gaining popularity [23], [12]. The recent method by Weng *et al*. [35] proposes a simple yet well-performing 3D MOT method; however, due to its strong reliance on 3D-based detections, it is susceptible to false positives and struggles with bridging longer occlusion gaps. A follow-up method [10] replaces the intersection-over-union with a Mahalanobis distance-based association measure. The recently proposed CenterPoint [39] method detects 3D centers of objects and associates them across frames using the predicted velocity vectors. In contrast, we propose a method that combines complementary 3D LiDAR object detectors that precisely localize objects in 3D space, and 2D object detectors, that are less susceptible to partial occlusions and remain reliable even when objects are far away from the sensor.

Fusion-based methods. Fusing object evidence from 2D and 3D during tracking is an under-explored area. Ošep *et al*. [23] propose a stereo vision-based approach. At its core, their method uses a tracking state filter that maintains each track's position jointly, in the 3D and the image domain, and can update them using only partial object evidence. In contrast, our method treats different sensor modalities independently. We track targets in both domains simultaneously, but we do not explicitly couple their 2D-3D states. Alternatively, BeyondPixels [28] leverages monocular SLAM to localize tracked objects in 3D space. MOTSFusion [19] fuses optical flow, scene flow, stereo-depth, and 2D object detections to track objects in 3D space. Different from that, our method relies only on bounding box object detections obtained from two complementary sensor modalities and scales well across different sensory environments (*e.g*., single LiDAR and multiple cameras [5]). The recently proposed GNN3DMOT [36] learns to fuse appearance and motion models, independently trained for both images and LiDAR sequences. We compare their method to ours in Sec. IV.

III. Method

Our EagerMOT framework combines complementary 2D, and 3D (*e.g*., LiDAR) object evidence obtained from pre-trained object detectors. We provide a general overview of our method in Fig. 2. As input at each frame, our method takes a set of 3D bounding box detections ${}^{3d}D_t$ and a set of 2D detections ${}^{2d}D_t$. Then, the observation fusion module (i) associates 2D and 3D detections originating from the same objects, (ii) the two-stage data association module associates detections across time, and, based on the available detection information (full 2D+3D, or partial) we update the track states and (iv) we employ a simple track management mechanism to initialize or terminate the tracks.

This formulation allows all detected objects to be associated to tracks, even if they are not detected either in the image domain or by a 3D sensor. This way, our method can recover from short occlusions and maintain approximate 3D location when one of the detectors fails, and, importantly, we can track far-away objects in the image domain before objects enter the 3D sensing range. Once objects enter the sensing range, we can smoothly initialize a 3D motion model for each track.

A. Fusion

We obtain two sets of object detections at the input, extracted from the input video (2D) and LiDAR (3D) streams. LiDAR-based object detections ${}^{3d}D_t$ are parametrized as 3D object-oriented bounding boxes, while image-based object detections ${}^{2d}D_t$ are defined by a rectangular 2D bounding box in the image domain. First, we establish a matching between the two sets.

The fusion module performs this task by greedily associating detections in ${}^{3d}D_t$ to detections in ${}^{2d}D_t$ based on their 2D overlap in the image domain and produces a set of fused object instances I_t=$\{I_t^0, ..., I_t^i\}$. We define 2D overlap for a pair ${}^{3d}D_t^i$ and ${}^{2d}D_t^i$ as the intersection over union (IoU) between the 2D projection of ${}^{3d}D_t^i$ in the camera image plane and ${}^{2d}D_t^i$. We note that while different associations criteria could be additionally taken into account (as in, *e.g*., [23]), this simple approach was empirically proven to be robust.

During the greedy association, we sort all possible detection pairings by their overlap in descending order. Pairs are considered one-by-one and are combined to form a single fused instance ${}^{both}I_t^i$ when (i) their overlap is above a threshold θ_{fusion} and (ii) neither 2D, or 3D detection has been matched yet. Fused instances ${}^{both}I_t \subseteq I_t$ contain information from both modalities: a precise 3D location of the object and its 2D bounding box. Instances may also store additional available information, *e.g*., a 2D segmentation mask [34]. We refer to the remaining (unmatched) detections, which form instances ${}^{3d}I_t^i \subseteq I_t$ and ${}^{2d}I_t^i \subseteq I_t$, as *partial observations*, containing information about only one of the two modalities. Note that ${}^{both}I_t \subseteq {}^{3d}I_t$ and ${}^{both}I_t \subseteq {}^{2d}I_t$.

这里的greedy association应该指的就是Nearest Neigbor(NN).

Multi-camera setup. For scenarios where multiple cameras are available (*e.g*., in the NuScenes dataset [5]), we adapt our fusion algorithm as follows. In each camera, we perform

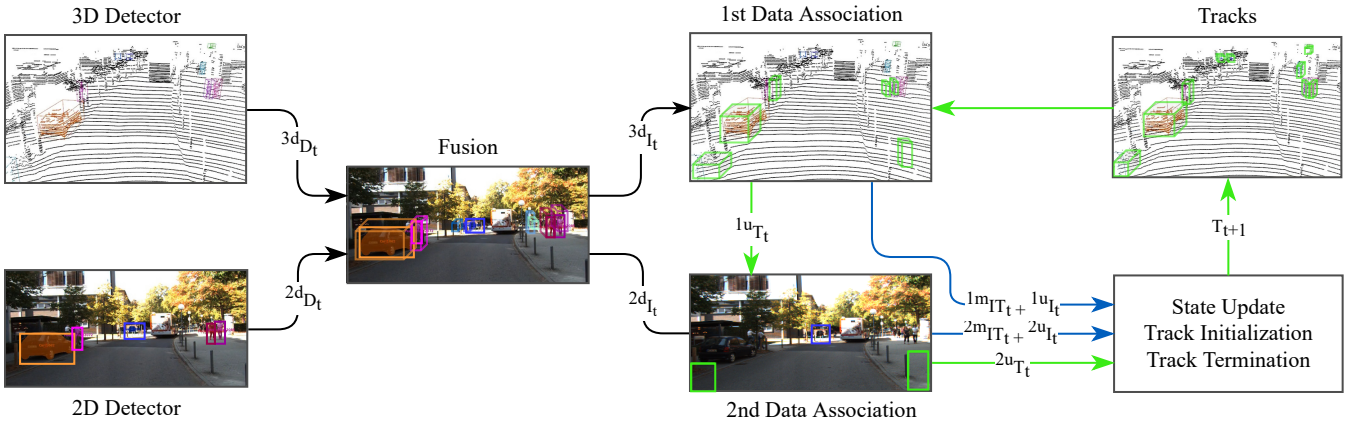

Fig. 2: A high-level overview of our tracking framework: at the input, we obtain object detections from different sensor modalities, *e.g.*, an image-based detector/segmentation model and a LiDAR/stereo-based 3D object detector. We then fuse these detections into fused object instances, parameterized jointly in 3D and/or 2D space. We then pass them through a two-stage association procedure that allows us to update object tracks, even if detections originating only from one sensor modality are available. In the *first stage*, instances with 3D information (with/without 2D information) are matched to existing tracks. In the *second association stage*, unmatched tracks from the previous step $1u_t^T$ are matched with instances, localized only in 2D.

fusion as explained above; in case a 3D detection is not visible in a particular camera, we consider its overlap with 2D detections in that image plane to be empty. After we perform fusion in each 2D plane individually, 3D detections visible through more than one camera might have multiple potential matches. We always associate only one 2D detection with a track and heuristically pick a detection from the view in which the projected 3D bounding box covers the largest area. Other potential pairings from other views are discarded.

B. Matching

During each frame t, fused instances I_t enter a two-stage matching process to update existing tracks T_t with new 3D and/or 2D information.

Track parameterization. As in [23], we maintain 2D and 3D state of tracks T_t in parallel. However, we treat them independently. We represent the 3D state of a track by a 3D object-oriented bounding box and a positional velocity vector (excluding angular velocity, as in [35]), while a 2D bounding box represents its 2D state. Since we track objects primarily in 3D, a track's confidence score is equal to its 3D state's confidence. Note that these states do not have to be fully observed for each frame, tracks might be updated using only 3D information ${}^{3d}T_t \subseteq T_t$, only 2D information ${}^{2d}T_t \subseteq T_t$, or both ${}^{both}T_t \subseteq T_t$, ${}^{both}T_t \subseteq {}^{3d}T_t$, ${}^{both}T_t \subseteq {}^{2d}T_t$.

For tracks ${}^{3d}T_t$ we additionally maintain a constant-velocity motion model, modeled by a linear Kalman filter. For each new frame $t+1$, existing tracks ${}^{3d}T_t$ predict their location (an oriented 3D bounding box) in the current frame based on previous observations and velocity estimates.

First stage data association. In the first association stage, we match instances detected in 3D with existing tracks using track 3D state information. In particular, we greedily pair detected instances ${}^{3d}I_t$ with tracks ${}^{3d}T_t$ based on the scaled distance between instances' oriented bounding boxes and tracks' predicted oriented boxes. We define the scaled distance for a pair of oriented 3D bounding boxes as the Euclidean distance between them, multiplied by the normalized cosine distance between their orientation vectors:

$$d(B^i, B^j) = \|B^i_\rho - B^j_\rho\| * \alpha(B^i, B^j), \quad (1)$$

$$\alpha(B^i, B^j) = 2 - \cos\langle B^i_\gamma, B^j_\gamma\rangle, \in [1, 2], \quad (2)$$

where $B^i_\rho = [x, y, z, h, w, l]$ is a vector containing the 3D location and dimensions of the bounding box and B^i_γ represents the orientation of the box around the vertical axis.

Compared to planar Euclidean distance, this approach takes into account orientation similarity, which can be informative for non-omnidirectional objects such as vehicles or pedestrians. Experimentally, we found this association criterion to be more robust compared to 3D IoU [35] and Mahalanobis distance that takes the predictive and observational uncertainty into account [10]), especially in low frame rate scenarios (*e.g.*, in NuScenes dataset [5]).

Similar to subsection III-A, best-matching instance-track pairs (below a maximum threshold θ_{3d}) form successful matching tuples ${}^{1m}IT_t = \{(I^i_t, T^j_t), ...\}$. We label the rest as unmatched, *i.e.*, instances ${}^{1u}I_t$ and tracks ${}^{1u}T_t$. After this first matching stage, all object instances detected in 3D should be successfully associated with existing tracks or be labeled as unmatched and will not participate in further matching.

Second stage data association. In the second stage, we match detected instances to tracks in the 2D image domain. We greedily associate instances ${}^{2d}I_t \setminus {}^{both}I_t$ to remaining tracks ${}^{1u}T_t \cup {}^{2d}T_t$ based on the 2D IoU criterion. For each instance-track pair, we evaluate the overlap between the instance's 2D bounding box in the current frame and the 2D

projection of the track's predicted 3D bounding box or the last observed 2D bounding box in case a 3D prediction is not available (for ${}^{2d}T_t$). Note that instances that were detected in 3D do not participate in this matching stage even if they were also detected in 2D, *i.e.*, ${}^{both}I_t$.

This association stage is identical to the first one, except we use here 2D box IoU as the association metric, together with its threshold θ_{2d}. Similarly, the output of this stage are a set of matches ${}^{2m}IT_t = \{(I_t^i, T_t^j), ...\}$, a set of unmatched instances ${}^{2u}I_t$, and unmatched tracks ${}^{2u}T_t$. In the case of multiple cameras being available, we modify the algorithm as described earlier in Part III-A.

We use a 3D motion model to obtain 2D bounding box predictions in the image domain using a camera projection operation. There is not enough 3D evidence to initialize the motion model reliably for certain tracks – this usually happens for objects observed outside of the LiDAR sensing range. In such scenarios, the apparent bounding box motion is usually negligible, and association can be made purely based on observed 2D boxes. Adding a prediction model for the 2D state (as in [23], [36]) or a (learned) appearance model [18], [34] could be used to improve the second association stage further and remains our future work.

State update. We use matched detected instances to update corresponding tracks with new 3D and/or 2D state information. We simply update the 2D state (top-left and bottom-right bounding box corners) by over-writing the previous state with the newly-detected 2D bounding box. We model the 3D state of a track (*i.e.*, object-oriented bounding box parameters) as a multi-variate Gaussian and filter its parameters using a constant-velocity linear Kalman filter (exactly as in [35]). When 3D object detection information is not available (*e.g.*, we have a partial observation providing only a 2D bounding box or a segmentation mask in the image domain, we only perform the Kalman filter *prediction step* to extrapolate the state.

C. Track lifecycle

Following AB3DMOT [35], we employ a simple set of rules to manage object trajectories and their lifecycle. A track is discarded if it has not been updated with any instance (either 3D or 2D) in the last Age_{max} frames. As 3D object detectors are usually not as reliable as image-based detectors in terms of precision, a track is considered confirmed if it was associated with an instance in the current frame and has been updated with 2D information in the last Age_{2d} frames. Finally, all detected instances ${}^{2u}I_t$ that were never matched start new tracks.

IV. Experimental Evaluation

We evaluate our method using two datasets, KITTI [13] and NuScenes [5] using four different multi-object tracking benchmarks: (i) NuScenes 3D MOT, (ii) KITTI 3D MOT, (iii) KITTI 2D MOT, and (iv) KITTI MOTS [34]. For NuScenes 3D MOT, KITTI 2D MOT, and KITTI MOTS, we use the official benchmarks and compare our method to published and peer-reviewed state-of-the-art methods.

Method	AMOTA	MOTA	Recall	IDs
Ours	**0.68**	**0.57**	**0.73**	1156
CenterPoint [39]	0.65	0.54	0.68	**684**
StanfordIPRL-TRI [10]	0.55	0.46	0.60	950
AB3DMOT [35]	0.15	0.15	0.28	9027

TABLE I: Results on the NuScenes 3D MOT benchmark. Methods marked in gray are not yet peer-reviewed.

	Method	Input	sAMOTA	MOTA	MOTP	IDs
car	**Ours**	2D+3D	94.94	**96.61**	**80.00**	2
	Ours†	2D+3D	**96.93**	95.29	76.97	1
	GNN3DMOT [36]	2D+3D	93.68	84.70	79.03	10
	mmMOT [40]	2D+3D	70.61	74.07	78.16	125
	FANTrack [1]	2D+3D	82.97	74.30	75.24	202
	AB3DMOT† [35]	3D	91.78	83.35	78.43	**0**
ped.	**Ours**	2D+3D	**92.92**	**93.14**	**73.22**	36
	Ours†	2D+3D	80.97	81.85	66.16	**0**
	AB3DMOT† [35]	3D	73.18	66.98	67.77	1

TABLE II: 3D MOT evaluation on the KITTI val set (following evaluation protocol by [35]). Methods marked with † use the Point R-CNN [30] 3D object detector. Baseline results taken from [36]. Note that several methods are only reported results for the *car* class.

Evaluation measures. We discuss the results using standard CLEAR-MOT evaluation measures [3] and focus the discussion on the *multi-object tracking accuracy* (*MOTA*) metric. For KITTI 3D MOT, we follow the evaluation setting of [35] and report averaged variants of CLEAR-MOT evaluation measures (*AMOTA* and *AMOTP* stand for averaged *MOTA* and *MOTP*). For MOTS, we follow the evaluation protocol of [34] and report *multi-object tracking and segmentation accuracy* (*MOTSA*) and *precision* (*MOTSP*). On KITTI 2D MOT and MOTS benchmarks we additionally report the recently introduced *higher-order tracking accuracy* (*HOTA*) metric [20][1]. HOTA dis-entangles detection and tracking aspects of the task by separately measuring *detection accuracy* (*DetA*) that evaluates detection performance, and *association accuracy* (*AssA*) that evaluates detection association.

3D detections. For our final model on NuScenes, we use detections provided by CenterPoint [39]. On KITTI 3D MOT, we report and compare results obtained using state-of-the-art Point-GNN [31] and Point R-CNN [30] (as used by [35]) 3D object detectors. For our model, submitted to the KITTI benchmark, we used Point-GNN [31]. We do not pre-filter 3D object detections and take all of them as input to our tracking pipeline.

2D detections. On NuScenes, we use the Cascade R-CNN [6], [8] object detector, trained on the NuImages [5] dataset. On KITTI, we follow MOTSFusion [19] and use 2D detections from RRC [26] for *cars* and TrackR-CNN [34] for *pedestrians*. We use thresholds of 0.6 and 0.9 for RRC and TrackR-CNN detections, respectively

[1]The official KITTI 2D MOT benchmark switched to HOTA-based evaluation shortly before releasing this paper, therefore we only report benchmark results using this metric.

Method	Inputs	car					pedestrian				
		HOTA	DetA	AssA	MOTA	IDs	HOTA	DetA	AssA	MOTA	IDs
Ours	2D+3D (LiDAR)	**74.39**	75.27	74.16	87.82	239	39.38	40.60	38.72	49.82	496
mono3DT [16]	2D+GPS	73.16	72.73	**74.18**	84.28	379	–	–	–	–	–
CenterTrack [42]	2D	73.02	**75.62**	71.20	**88.83**	254	40.35	**44.48**	36.93	**53.84**	425
SMAT [14]	2D	71.88	72.13	72.13	83.64	198	–	–	–	–	–
3D-TLSR [22]	2D+3D (stereo)	–	–	–	–	–	**46.34**	42.03	**51.32**	53.58	**175**
Be-Track [11]	2D+3D (LiDAR)	–	–	–	–	–	43.36	39.99	47.23	50.85	199
AB3DMOT [35]	3D (LiDAR)	69.81	71.06	69.06	83.49	**126**	35.57	32.99	38.58	38.93	259
JRMOT [29]	2D+3D (RGB-D)	69.61	73.05	66.89	85.10	271	34.24	38.79	30.55	45.31	631
MOTSFusion [19]	2D+3D (stereo)	68.74	72.19	66.16	84.24	415	–	–	–	–	–
MASS [17]	2D	68.25	72.92	64.46	84.64	353	–	–	–	–	–
BeyondPixels [28]	2D+3D (mono SLAM)	63.75	72.87	56.40	82.68	934	–	–	–	–	–
mmMOT [41]	2D+3D	62.05	72.29	54.02	83.23	733	–	–	–	–	–

TABLE III: Results on the 2D MOT KITTI benchmark. Note: reported methods use different object detectors, *e.g.* our method uses the RRC [26] detector for the *car* class, same as MOTSFusion [19] and BeyondPixels [28].

A. Ablation studies

Data association. In Table IV, we compare different variants of our method, evaluated on the NuScenes validation set. The significant difference between *"Full"* (0.712 AMOTA) and *"No 2D info"* (0.651 AMOTA) highlights the impact of leveraging 2D object detections on the overall performance. We note that as we improve the recall ($+0.054$) with our full model, we observe a decrease in AMOTP (-0.018), which measures localization precision, averaged over all trajectories. This is because our method can leverage 2D object detections and update track states even when 3D detections are not available. In this case, we cannot update the track state using 3D evidence. However, we can still localize objects by performing Kalman filter predictions at the loss of overall 3D localization precision.

Next, we ablate the impact of our data association function. The configuration *"No 2D info; 2D distance"* highlights the performance of a variant that does not use 2D detection information and performs association by simply computing Euclidean distance (on the estimated 2D ground-plane) between the track prediction and detections as an association criterion for the (only) matching stage. The variant *"No 2D info; 3D IoU"* is the variant that uses 3D IoU (as in [35]) as the association metric. As can be seen, our association function is more robust compared to 2D distance ($+0.004$ AMOTA) and 3D IoU ($+0.036$ AMOTA). We conclude that 3D IoU is not suitable for NuScenes due to a significantly lower scan-rate compared to KITTI.

Detection sources. In Table V, we show the impact of detection quality on overall performance. One of the advantages of our method is its flexibility. Unlike other trackers, our framework does not need expensive training and can be easily applied to off-the-shelf detectors. As expected, better detectors lead to better tracking performance.

B. Benchmark results

NuScenes. We report the results obtained using the official NuScenes large-scale tracking benchmark in Table I. In addition to published methods, we include in our analysis the highest-ranking unpublished method [39], marked with gray. The test set includes 150 scenes, 80 seconds each. This is a challenging benchmark due to a wide variety of object classes and a low frame rate of 2FPS.

Method	AMOTA	AMOTP	Recall	IDs
Full	**0.712**	0.569	**0.752**	899
No 2D info	0.651	0.587	0.698	864
No 2D; 2D distance	0.647	0.595	0.689	**783**
No 2D; 3D IoU	0.615	**0.658**	0.692	2749

TABLE IV: Data association ablation study, performed on the NuScenes 3D MOT val set.

3D source	2D source	MOTA car	MOTA ped
Point-GNN [30]	RRC [26] + Track-RCNN [34]	92.5	**72.4**
Point R-CNN [35]	RRC [26] + Track-RCNN [34]	**92.7**	65.6
Point-GNN [30]	Cascade R-CNN [6]	89.0	69.5

TABLE V: Ablation on the effect of using different object detection sources (KITTI 2D MOT val set).

For a fair comparison, we use the same 3D detections as CenterPoint [39]. However, we only use 3D bounding box information (and not the predicted velocity vectors). The difference in recall supports our assumption that fusing 2D detections helps to bridge occlusions and recover tracks that would otherwise be lost.

KITTI 3D MOT. In Table II, we compare our 3D MOT performance to several baselines, as reported in [36]. Methods marked with † use the Point R-CNN [30] 3D object detector. As our method uses the same 3D detections as AB3DMOT, we can conclude that the improvements ($+5.15$ and $+7.79$ sAMOTA for *car* and *pedestrian* classes, respectively) show the merit of our two-stage association procedure that leverages 2D information to improve 3D MOT performance.

KITTI 2D MOT. In Table III, we report 2D MOT results we obtain on the KITTI test set. Even though we track objects in 3D space, we can report 2D tracking results by projecting 3D bounding boxes to the image plane using camera intrinsics and report minimal axis-aligned 2D bounding boxes that fully enclose those projections as tracks' 2D positions. Even though we track objects only in 3D, use 2D detections only as a secondary cue and report approximate 2D locations, we achieve state-of-the-art results in terms of the HOTA metric. In Fig. 3, we highlight examples where 3D detector fails due to signal sparsity or occlusions; however, we obtain 2D object detections, which we use to update the

track states. This example demonstrates that 3D LiDAR and image-based detections are complementary cues for tracking. We note that our method performs especially well in terms of the association accuracy (74.16 *AssA* on the *car* class), confirming that the secondary association stage does improve not only the detection aspect of the task but also the temporal association. For the *pedestrian* class, we obtain lower performance compared to current top entries, as we are using weaker object detectors (TrackR-CNN [34], trained on the non-amodal MOTS dataset) and Point-GNN [31].

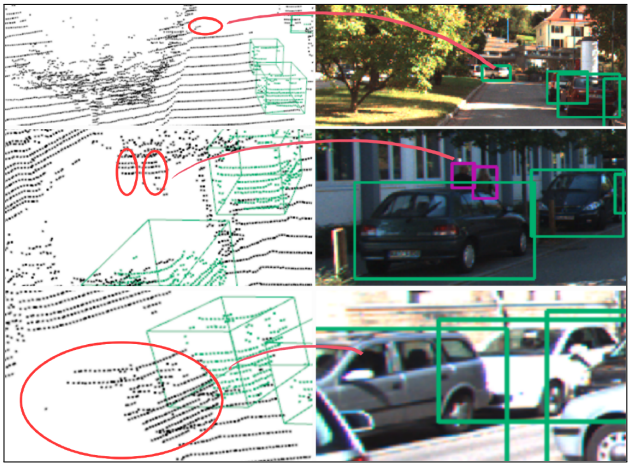

Fig. 3: Examples of objects overlooked by the 3D detector but recognized by the image-based detector. From the top: out of range, partially occluded, detector failure.

KITTI MOTS. Multi-object tracking and segmentation (MOTS) extends MOT with pixel-precise localization of object tracks. We can easily adapt our MOTS approach by additionally passing segmentation masks from instances to tracks after the data association.

In Table VI, we report our MOTS performance on the KITTI test set and compare it to other published methods. As can be seen, we obtain better results compared to MOTSFusion on both classes ($+1.03$ for *car* and $+3.61$ for *pedestrian* class) despite using the same set of 2D segmentation masks. We note, however, that EagerMOT additionally used 3D object detections obtained from the LiDAR stream, while MOTFusion relies on stereo cameras. Our method is applicable to a wide variety of LiDAR-centric sensory configurations, often employed in modern automotive datasets, *e.g.*, NuScenes [5], Waymo Open Dataset [32] and Argoverse [7]. Moreover, our method runs at 90 FPS on KITTI (LiDAR + single camera) compared to MOTSFusion at 2 FPS (stereo cameras).[2] Finally, our method establishes new state-of-the-art results for both, *car* (74.66) and *pedestrian* (57.65) classes in terms of HOTA. Furthermore, our method performs especially well in terms of association accuracy (*AssA*), again confirming that using additional sensor observations helps to maintain track consistency.

[2]Both exclude the time spent on object detection and ego-motion estimation.

	Method	HOTA	DetA	AssA	sMOTSA	IDs	FPS
car	**Ours**	**74.66**	76.11	**73.75**	74.53	458	**90**
	MOTSFusion [19]	73.63	75.44	72.39	74.98	**201**	2
	PointTrack [38]	61.95	**79.38**	48.83	**78.50**	346	22
	TrackR-CNN [34]	56.63	69.90	46.53	66.97	692	2
ped.	**Ours**	**57.65**	60.30	**56.19**	58.08	270	**90**
	MOTSFusion	54.04	60.83	49.45	58.75	279	2
	PointTrack	54.44	**62.29**	48.08	**61.47**	**176**	22
	TrackR-CNN	41.93	53.75	33.84	47.31	482	2

TABLE VI: Results on the 2D KITTI MOTS benchmark (for *car* and *pedestrian* classes). Note: our method uses the same set of object detections and segmentation masks as MOTSFusion.

C. Runtime discussion

Excluding the time spent on object detection and ego-motion estimation, our Python implementation runs at 4 FPS on NuScenes. It is slower (but more accurate) compared to StanfordIPRL-TRI [10] and AB3DMOT [35] that only use LiDAR data and run at 10 FPS.

On KITTI, our method runs at 90 FPS because we only have a single camera and do not need to perform multi-camera association. We report higher frame rates compared to several 3D MOT methods reported on the KITTI benchmark, including GNN3DMOT (5 FPS), mmMOT (4 FPS), and FANTrack (25 FPS), that also leverage both 2D and 3D input.

Implementation details. On KITTI, we use $\theta_{fusion} = 0.01$, $\theta_{3d} = 0.01$, $\theta_{2d} = 0.3$, $Age_{max} = 3$, and $Age_{2d} = 3$ for both classes. For 2D MOT evaluation, we report 2D projections of estimated 3D bounding boxes only for confirmed tracks.

On NuScenes, we use $\theta_{fusion} = 0.3$, $\theta_{2d} = 0.5$, and $Age_{max} = 3$ for all seven classes. Other parameters are class-specific: $\theta_{3d} = (7.5, 1.8, 4.4, 8.15, 7.5, 4.9, 7.5)$, $Age_{2d} = (2, 3, 1, 3, 3, 2, 2)$ for *car, pedestrian, bicycle, bus, motorcycle, trailer, truck*. Additionally, $\theta_{fusion} = 0.01$ for *trailer* and $\theta_{fusion} = 0.3$ for all other classes.

For 3D MOT evaluation on KITTI and NuScenes, we report estimated 3D boxes for confirmed tracks with their original confidence scores. Estimates for unconfirmed tracks are also reported. However, we halve their scores for each frame for which we do not perform 2D updates.

V. Conclusion

We presented a tracking framework that can leverage different sources of object detections originating from varying sensor modalities through a two-stage association procedure. Our experimental evaluation reveals that our method performs consistently well across different datasets and tracking tasks and can be used in combination with a variety of different object detectors – without requiring any additional detector-specific fine-tuning. We hope that our framework will serve as a baseline for future research in sensor-fusion-based multi-object tracking.

Acknowledgements: This project was funded by the Humboldt Foundation through the Sofja Kovalevskaja Award. We thank Paul Voigtlaender for his feedback.

REFERENCES

[1] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki. Fantrack: 3d multi-object tracking with feature association network. In *Intel. Vehicles Symp.*, 2019.

[2] P. Bergmann, T. Meinhardt, and L. Leal-Taixé. Tracking without bells and whistles. In *ICCV*, 2019.

[3] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *JIVP*, 2008:1:1–1:10, 2008.

[4] G. Brasó and L. Leal-Taixé. Learning a neural solver for multiple object tracking. In *CVPR*, 2020.

[5] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[6] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018.

[7] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, 2019.

[8] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[9] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun. 3D object proposals for accurate object class detection. In *NIPS*, 2015.

[10] H.-k. Chiu, A. Prioletti, J. Li, and J. Bohg. Probabilistic 3d multi-object tracking for autonomous driving. *arXiv preprint arXiv:2001.05673*, 2020.

[11] M. Dimitrievski, P. Veelaert, and W. Philips. Behavioral pedestrian tracking using a camera and lidar sensors on a moving vehicle. *Sensors*, 19(2), 2019.

[12] D. Frossard and R. Urtasun. End-to-end learning of multi-sensor 3d tracking by detection. *ICRA*, 2018.

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

[14] N. F. Gonzalez, A. Ospina, and P. Calvez. Smat: Smart multiple affinity metrics for multiple object tracking. In *ICIAR*, 2020.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, 2017.

[16] H.-N. Hu, Q.-Z. Cai, D. Wang, J. Lin, M. Sun, P. Krahenbuhl, T. Darrell, and F. Yu. Joint monocular 3d vehicle detection and tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[17] H. Karunasekera, H. Wang, and H. Zhang. Multiple object tracking with attention to appearance, structure, motion and size. *IEEE Access*, 7:104423–104434, 2019.

[18] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler. Learning by tracking: Siamese cnn for robust target association. *CVPR Workshops*, 2016.

[19] J. Luiten, T. Fischer, and B. Leibe. Track to reconstruct and reconstruct to track. *IEEE RAL*, 5(2):1803–1810, 2020.

[20] J. Luiten, A. Ošep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe. Hota: A higher order metric for evaluating multi-object tracking. *IJCV*, 2020.

[21] F. Moosmann and C. Stiller. Joint self-localization and tracking of generic objects in 3d range data. In *ICRA*, 2013.

[22] U. Nguyen and C. Heipke. 3d pedestrian tracking using local structure constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:347–358, 2020.

[23] A. Ošep, W. Mehner, M. Mathias, and B. Leibe. Combined image- and world-space tracking in traffic scenes. In *ICRA*, 2017.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.

[25] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

[26] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. In *CVPR*, 2017.

[27] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[28] S. Sharma, J. A. Ansari, J. Krishna Murthy, and K. Madhava Krishna. Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In *ICRA*, 2018.

[29] A. Shenoi, M. Patel, J. Gwak, P. Goebel, A. Sadeghian, H. Rezatofighi, R. Martín-Martín, and S. Savarese. Jrmot: A real-time 3d multi-object tracker and a new large-scale dataset. In *IROS*, 2020.

[30] S. Shi, X. Wang, and H. Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019.

[31] W. Shi and R. R. Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *CVPR*, 2020.

[32] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

[33] A. Teichman, J. Levinson, and S. Thrun. Towards 3D object recognition via classification of arbitrary object tracks. In *ICRA*, 2011.

[34] P. Voigtlaender, M. Krause, A. Ošep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe. MOTS: Multi-object tracking and segmentation. In *CVPR*, 2019.

[35] X. Weng, J. Wang, D. Held, and K. Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. *IROS*, 2020.

[36] 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 *CVPR*, 2020.

[37] Y. Xu, A. Ošep, Y. Ban, R. Horaud, L. Leal-Taixé, and X. Alameda-Pineda. How to train your deep multi-object tracker. In *CVPR*, 2020.

[38] Z. Xu, W. Zhang, X. Tan, W. Yang, H. Huang, S. Wen, E. Ding, and L. Huang. Segment as points for efficient online multi-object tracking and segmentation. In *ECCV*, 2020.

[39] T. Yin, X. Zhou, and P. Krähenbühl. Center-based 3d object detection and tracking. *arXiv:2006.11275*, 2020.

[40] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy. Robust multi-modality multi-object tracking. In *ICCV*, 2019.

[41] W. Zhang, H. Zhou, S. Sun, Z. Wang, J. Shi, and C. C. Loy. Robust multi-modality multi-object tracking. In *ICCV*, 2019.

[42] X. Zhou, V. Koltun, and P. Krähenbühl. Tracking objects as points. In *ECCV*, 2020.

[43] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.