JRMOT: A Real-Time 3D Multi-Object Tracker and a New Large-Scale Dataset

Abhijeet Shenoi, Mihir Patel, JunYoung Gwak, Patrick Goebel,
Amir Sadeghian, Hamid Rezatofighi, Roberto Martin-Martin, Silvio Savarese

***Abstract*— An autonomous navigating agent needs to perceive and track the motion of objects and other agents in its surroundings to achieve robust and safe motion planning and execution. While autonomous navigation requires a multi-object tracking (MOT) system to provide 3D information, most research has been done in 2D MOT from RGB videos. In this work we present JRMOT, a novel 3D MOT system that integrates information from 2D RGB images and 3D point clouds into a real-time performing framework. Our system leverages advancements in neural-network based re-identification as well as 2D and 3D detection and descriptors. We incorporate this into a joint probabilistic data-association framework within a multi-modal recursive Kalman architecture to achieve online, real-time 3D MOT. As part of our work, we release the *JRDB* dataset, a novel large scale 2D+3D dataset and benchmark annotated with over 2 million boxes and 3500 time consistent 2D+3D trajectories across 54 indoor and outdoor scenes. The dataset contains over 60 minutes of data including 360°cylindrical RGB video and 3D pointclouds. The presented 3D MOT system demonstrates state-of-the-art performance against competing methods on the popular 2D tracking KITTI benchmark and serves as a competitive 3D tracking baseline for our dataset and benchmark.**

I. Introduction

The goal of an autonomous navigating agent such as a self-driving car or a mobile robot is to move between two points in 3D space in a safe and robust manner. In order to achieve that, the agent needs to perceive the motion of the multiple dynamic objects and other agents, *e.g.* people and cars, in its vicinity. This perceived motion allows to predict the possible future trajectories of the other agents and to plan and execute motion strategies that avoid colliding with them. To achieve this safe navigation, the motion of the other agents needs to be perceived in the same space the navigation takes place, the 3D space. However, so far, the majority of efforts from the computer vision community have been dedicated to the development of multi-object tracking (MOT) systems that perceive 2D motion from RGB video streams.

There have been some methods that proposed to tackle the 3D MOT problem based only on 2D RGB videos [1], [2] or only on 3D pointcloud data [3], [4]. However, most current autonomous cars and robots are equipped with sensors for both modalities, and their signals carry complementary information. On one hand, 2D RGB images are **dense** containing on the order of millions of pixels, which allows us to discern appearances of objects to effectively identify and classify them even at large distances. Moreover, RGB data is well structured in the form of a **pixel grid** suited to be processed with effective tools such as convolutional neural networks. On the other hand, 3D pointcloud data is **sparse** but due to the additional dimension it allows to discern objects which might overlap in a 2D RGB image, but are well separated in 3D space. However, the unordered structure of the pointclouds do not enable to apply CNNs on them. This dichotomy of appearance information, available in RGB images, and richer geometric information present in pointclouds is what we aim to exploit in this work.

In this paper, we present a novel, real-time, multi-object tracking framework that leverages the best of both sensor domains, 2D RGB images and 3D pointclouds, in order to generate time consistent tracks in 3D space. To obtain real-time 3D tracks, our framework optimally integrates recent state-of-the-art solutions for 2D detection in RGB images and 3D point clouds with novel multi-modal descriptors and well-established data-association and filtering techniques. First, we integrate 2D detections from Mask R-CNN [5] with 3D information through a frustrum segmentation from F-PointNet [6]. This segmentation step allows us to discern overlapping objects in the RGB image through their different depths while generating also a descriptor of their 3D shape. This procedure also ensure a one-to-one bundle between the detections in these two sensor modalities. Second, we fuse the 3D shape descriptor with a 2D RGB descriptor into a multi-modal feature to perform data-association through the recent re-identification module, Aligned-ReID [7]. The fused feature is more robust than the unimodal counterparts (see Sec. V). Third, we tackle the data-association problem between detection and tracks in an online manner using an optimal joint probabilistic data association step [8]. Finally, we propose a novel multi-modal recursive Kalman filter with dual measurement spaces, one from cylindrical 360°RGB images and the other based on 3D observations. We demonstrate the effectiveness of our framework with experiments in the standard KITTI dataset [9] achieving state of the art performance in the 2D tracking benchmark among all published online real time methods.

One of the reasons 2D MOT systems have developed further than than their 3D counterparts is that they have been supported by multiple 2D annotated video datasets [10], [11], [12], [13] and boosted by popular 2D tracking challenges, *e.g.* MOTchallenge [13] and KITTI [9]. In 3D, such datasets and benchmarks are lacking. We present, as part of this project, the JRDB dataset, a novel dataset recorded with

All the authors are with the Stanford Vision and Learning Laboratory,Stanford University, USA.
E-mail: [ashenoi,mihirp,jgwak,pgoebel,amirabs, hamidrt,robertom,ssilvio]@cs.stanford.edu

our robot platform JackRabbot. The dataset contains over 60 minutes of sensor data recorded at over 25 locations, both indoor and outdoor. The sensor data includes stereo 360°RGB cylindrical video streams, continuous 3D point clouds from two LiDAR sensors, audio, GPS sensing, RGB-D images and a video stream captured from a fisheye camera. To enable the development of MOT systems, the dataset has been annotated with over 2.4 million 2D human bounding boxes and 1.8 million 3D bounding boxes with associated 2D bounding boxes. This allows us to leverage the complementarity of 2D RGB and 3D pointcloud data. We hope our new dataset will serve as a solid benchmark and help accelerate future research in 2D/3D multi-object tracking. The presented JRMOT 3D MOT framework serves as a competitive baseline for 3D tracking in our novel dataset and outperforms the only other 3D MOT solution with publicly available implementation.

To summarise, our contributions are:

1) We fuse 2D and 3D descriptors through a deep-learning architecture for re-identification with joint probabilistic data-association.
2) We propose a novel multi-modal Kalman filter measurement update based on 2D measurements from the RGB image space and direct 3D measurements from the pointcloud signals for 3D MOT.
3) We release the JRDB dataset, a novel 2D+3D dataset for the development and evaluation of 3D MOT frameworks. Our proposed 3D MOT system serves as competitive baseline in our benchmark.
4) We demonstrate that our proposed 3D MOT system achieves state-of-the-art performance in the KITTI 2D tracking benchmark even when our focus has been on 3D tracking.

II. Related Work

In recent years, there have been many advances in the task of multi-object tracking (MOT). Many of the previous studies have mainly focused on 2D tracking. Most 3D MOT systems share the same components with the 2D MOT systems, and change the detection boxes to 3D boxes instead of the image plane with some minor changes in motion and appearance feature calculation. In this work, we present a 3D MOT system that fuses information from 2D RGB videos and 3D point clouds for tracking multiple objects in 2D/3D space. Moreover, we introduce a novel large-scale dataset to the tracking community for further development and testing of 3D multi-object trackers. This dataset is targeted to a unique visual domain tailored to the perceptual tasks related to navigation in human environments, both indoors and outdoors, and is complementary to previous tracking datasets and not limited only to self-driving cars but also other types of agents like social mobile robots. In the rest of this section we will review previous works in the areas of 2D MOT from 2D RGB videos, 3D MOT from 2D RGB and/or 3D point clouds, and other existing datasets for MOT.

2D MOT with 2D Data: Tracking in 2D is the task of perceiving continuously the motion of objects in video sequences. Several different approaches belong to this category of exclusively using 2D RGB sensor input. One category relies on exploiting re-identification modules [14], [15], [16], [17], [18] to accurately re-identify objects from frame to frame. Another category employs motion and continuity cues [19], [20], [21], [22], [23], [24]. However, both approaches rely strictly on 2D information, which has certain pitfalls. Occlusion can prevent reliable detection, and can cause noisy motion estimates and unreliable re-identification descriptors. Further, motion in the projected 2D image space can be highly irregular. To alleviate these problems, several approaches [2], [25] attempt to infer 3D properties such as shape and approximate depth from only RGB input. However, inferring 3D properties from 2D images is inherently ill posed, and state of the art methods are largely data driven. Therefore, due to the lack of datasets containing indoor scenes, these approaches are unlikely to work well for autonomous agents that navigate in environments that are different from those encountered by self driving cars.

3D MOT with 3D Data: With the advent of self driving cars, access to large scale datasets containing LiDAR data [26], [27], [28], [29] has reinvigorated interest in the use of 3D sensors. [3], [30] both work exclusively with 3D detections and pointcloud data respectively to perform 3D tracking. However, these methods do not use any cues from the RGB image of the scene. The RGB sensor contains useful information, which can especially be used to accurately estimate the orientation of the 3D bounding box (which can be challenging to do in 3D). Further, objects very far away may have little to no points on them, but can be identified in 2D, which can be an important cue to update the location of the object in 3D. 3D MOT is a relatively unexplored field, with there being no open benchmark for 3D tracking at the time of writing of this work.

3D MOT with 2D and/or 3D Data: As discussed above, 3D and 2D sensor modalities have complementary information that can be combined to yield better results. Works such as [31] fuse both RGB and LiDAR information to perform tracking. However, this does so in a single object setting. [32] aggregates both 2D RGB appearance descriptor and the bounding box coordinates to learn a similarity function to perform 3D tracking. However, it independently detects in the 2D and 3D domains, and also only utilises 3D measurements to perform filtering. [33] also utilises RGB and depth information to reconstruct a 4D spatio-temporal scene. However, this is done in an offline setting. The effectiveness of these techniques has not been quantitatively tested because of the lack of a 3D tracking benchmark. They are often evaluated via the proxy of 2D tracking.

Dataset: As mentioned above, 3D sensory systems are becoming increasingly commonplace in sensor suites of autonomous agents. Datasets with this multi-modal data such as KITTI [9], Apolloscape [28] and Oxford's Robotic Car [34] have widely driven research in the 3D community in. Nonetheless, their targeted domain application is autonomous driving: the data they provide is captured exclusively from sensor suites on top of cars and the data only depicts streets,

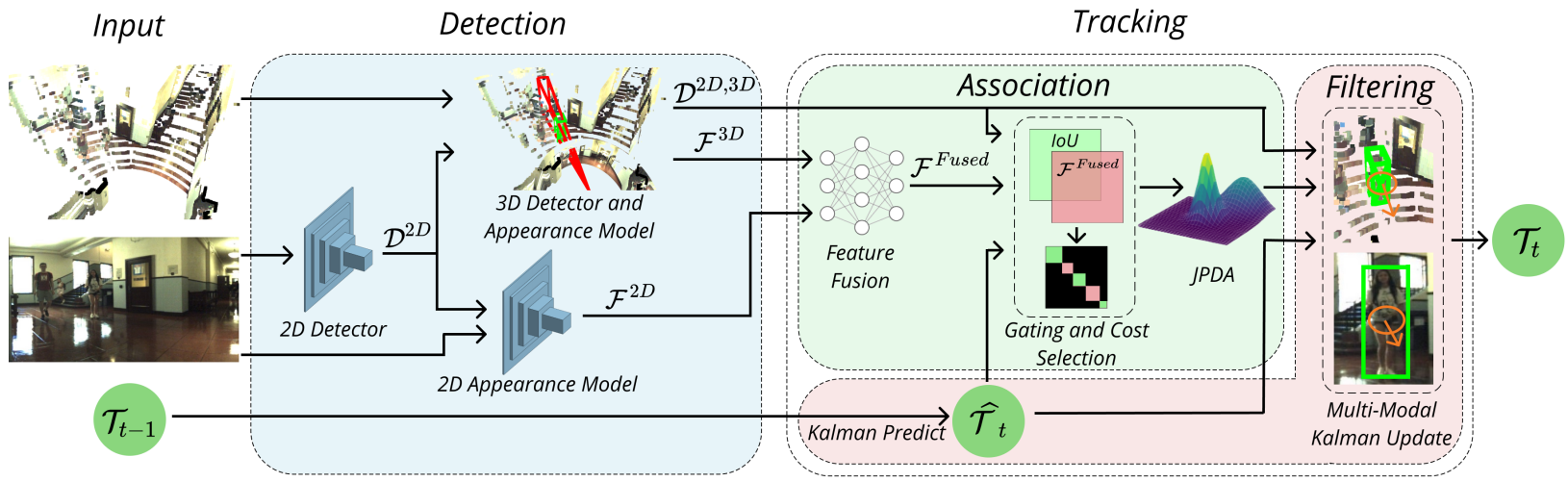


Fig. 1: JRMOT: Our proposed 3D MOT system is composed by detection and tracking (data association and filtering) components. $\mathcal{T}, \mathcal{D}, \mathcal{F}$ refer to tracks, detections, features respectively with the superscript indicating the space.

roads and highways. Additionally, at the time of submission, they have no active 3D tracking challenges.

In this paper, we target a unique visual domain tailored to the perceptual tasks related to navigation in human environments, both indoors and outdoors. We hope that this new domain provides the community an opportunity to develop visual perception frameworks for various types of autonomous navigation agents, limited not only to self-driving cars but also other types of agents like social mobile robots.

III. JRMOT: 3D Multi-Object Tracking from 2D and 3D Data

Our proposed 3D MOT system fusing 2D and 3D data is depicted in Fig. 1. Each component leverages the complementary nature of 2D and 3D sensor modalities, as shown in Sec. V. The input to our system is an 2D RGB video stream and a loosely time synchronized stream of 3D pointclouds. The extrinsic calibration between the RGB camera and the depth sensor is known. The output of our system is a time sequence of tracks: the location in 3D space of all instances of the tracked object class around the sensors, each one uniquely identified over time by a label ID. In the following we explain the details of each step of our system from the sensor input to the tracking output.

A. 2D Detection

Detection is at the core of any MOT system. So far, detection results have been consistently better (more robust and accurate) when the process is performed in 2D RGB images because 1) searching in 2D space is easier than in 3D, and 2) the density of data in RGB images is higher than in most 3D point clouds (more pixels per object instance than 3D points). Therefore, we use a pretrained Mask R-CNN [5] network, the state of the art in image segmentation. We discard the part of the network that performs segmentation and use it as a detector only. We finally fine-tune the network on our JRDB dataset to adapt to the special data distribution [1]. Note that, while the final results of our 3D MOT system depend on the detector used, the tracking framework is agnostic to it and can readily be replaced by others. The input to this module is thus a 2D RGB image at time t and the output is a set of N detections in 2D for the class of interest, $\mathcal{D}_t^{2D} = \{(u, v, w, h)_0, \ldots, (u, v, w, h)_{N-1}\}_t$, where $(u, v)_i$ is the upper-left corner of the detected bounding box around the instance i and $(w, h)_i$ are the width and height of that box.

B. 2D Appearance

When tracking objects, it is common to have occlusions wherein we are unaware of the position of an object for a brief period of time. The object might reappear far away from the point of disappearance. In these cases, the 2D appearance can be a useful cue to re-identify the object as explored in [17] among many other 2D tracking works. While tracking in 3D, the sparse nature of the pointcloud can make it difficult to discern heading direction. The lack of color information can make objects of the same shape look very similar to each other.

To leverage the appearance information present in RGB images we utilise Aligned-ReID [7] when the objects of interest are people, and [35] for cars. These networks are chosen because they involve a straightforward inference step, which does not significantly increase tracking time. Specifically, Aligned-ReID is chosen because during inference a global feature obtained by pooling over all spatial dimensions is used as an embedding for re-identification which allows for fast inference. For metrics on JRDB dataset, Aligned ReID was retrained on the train set.

C. 3D Detection and Appearance

As mentioned before (Sec. II), it is possible to obtain a noisy estimate of the 3D location of a detected object from its 2D detected box. However, in this work we propose to

[1]Our detections are publicly released as part of the JRDB dataset and benchmark for others to use in their MOT systems.

integrate 2D RGB and 3D data provided by a depth sensor, which is common part of any autonomous navigating system nowadays. To do that, we need to estimate the 3D location of the object instance enclosed by each bounding box.

Based on projective geometry, the object instance can be anywhere within the frustrum starting at the RGB camera center and passing through the bounding box. We utilise F-PointNet [6], a state-of-the-art algorithm to obtain 3D detection in the form of an oriented cuboid around the object instance for every 2D bounding box. F-PointNet segments out the 3D points within the frustrum that belongs to the target object, and estimates a 3D bounding box that would enclose the entire instance and that it is vertically aligned (the base of the box is parallel to the floor plane). The input to the 3D detection module is the set of detected 2D bounding boxes around instances of interest at time t, D_t^{2D}, and the 3D pointcloud at closest time to t. The output is a set of M detections in 3D for the class of interest at time t, $\mathcal{D}_t^{3D} = \{(x, y, z, w, h, l, \theta)_0, \ldots, (x, y, z, w, h, l, \theta)_{M-1}\}_t$, where $(x, y, z)_j$ is the center of the bottom face of the detected 3D bounding box around the instance j, $(w, h, l)_j$ are the width, height and length of that box and θ_j is the rotation of the box around the normal to the floor plane.

Additionally, the F-PointNet architecture can be used to generate a feature description of the content of the 3D bounding box, $\mathcal{F}_t^{3D} = \{f_0^{3D}, \ldots, f_{M-1}^{3D}\}_t$. We will use the output of the penultimate layer output of F-PointNet as 3D appearance descriptor, fuse it with the 2D appearance descriptor per detection (see next subsection) and use the multi-modal result to improve data association between detections and previous tracked instances.

A benefit of the presented procedure is that we obtain direct correspondences between 2D and 3D detections. However, in some cases some of the 2D bounding boxes do not lead to a corresponding 3D bounding box due to lack of enough 3D points in the frustrum. Therefore, the number of 2D bounding boxes and 3D bounding box may be different ($N \neq M$). We will overcome this limitation with a multi-modal Kalman filter leveraging both 2D and 3D detections and measurements.

D. Feature Fusion

Both 2D and 3D appearance can contain valuable information to associate detections and previous tracks. We fuse the 2D and 3D features with a 3-layered fully connected network that receives as input the concatenation of the 2D and 3D appearance embeddings. We train this fusion network via metric learning based on the triplet loss and semi-hard negative mining as in Schroff *et al.* [36]. As a result, the fused feature is close in the learned space for detections at different time steps on the same instance and further for detections on different instances, allowing a robust association between new detections and previous tracks.

E. Data Association

Given a set of detections at time t, we need to associate them to existing tracks in our filter in order to update the tracks' locations and appearances. To do so, we estimate the similarity in appearance between the new detections and the existing tracks. We calculate the pairwise ℓ_2 distance between the N features describing the detections at time t and the K features describing the tracks at time $t-1$ to build a $K \times N$ appearance cost matrix.

In 3D, the similarity in appearance is sometimes not as descriptive as the similarity in location between the detected objects and the predicted track locations. To estimate this location similarity, we compute an approximation of the pairwise 3D bounding box intersection over union (IoU) assuming that both 3D bounding boxes, the one given by the 3D detector and the other generated by the filter prediction, have the same orientation (same θ). This approximation generates a fairly good result in much shorter computation time because it avoids the costly polygon clipping step. The result of this parwise 3D IoU computation is a $K \times N$ IoU cost matrix.

The distance between the detections and the track predictions can be further used to simplify the data association procedure: we only consider assigning detections to tracks when the detections are "close enough" to the predicted location of the tracks. This process is called *gating* and we use the Mahalanobis distance, the physical distance weighted by the uncertainty of the predicted track location, between each detection to every track to perform it. We assume that the detections whose Mahalanobis distance to a predicted track is greater than a threshold (0.95 quantile from the χ^2 distribution) are considered to be improbable to correspond to that track, *i.e.* outside of the track's gate. To indicate this, we set the corresponding values in both cost matrices, appearance and IoU, to be infinite.

For a crowded scene, the number of tracks and detections can be large. As the size of the cost matrix scales with the square of the number of objects in the scene, it can be computationally expensive to perform the data association on these cost matrices. To help reduce the computation, we decompose the cost matrices into clusters. To do so, we construct a graph within which every track and detection is a node, such that an edge exists between track i and detection j if detection j is within the gate of track i. Every connected component in this graph is a cluster. On a per cluster basis, we then perform a cost matrix selection. Since we would like to select the most informative cost matrix, we select the cost matrix with the least total entropy.

Given this cost matrix, the goal is to then associate detections with tracks, so that we can update the location of the tracks. We can do so using the Hungarian algorithm [37] to minimise the total cost of the association or Joint Probabilistic Data Association [8] for example. We choose to use the latter approach as it has been shown by Rezatofighi *et al.* [22] to be robust to cluster and reduces the occurrence of ID switches. To maintain the speed of our tracker, we employ the m-best solution approximation [38] for clusters with more than six tracks and detections. For all other cases, we do complete enumeration and obtain the exact solution of JPDA, because it is a) faster for smaller matrices and b)

it is not an approximation.

F. Filtering

2D and 3D detections are often noisy. Since they have some amount of noise about the true location of the object (in the 2D image plane or the 3D world), it is commonplace to perform filtering on the detections to obtain the denoised estimate of the location of the object. There are various Bayesian filtering frameworks utilised in tracking, such as particle filters [39], Kalman filters [17]. We chose to use a Kalman filter because 1) It is not computationally intensive 2) It does not suffer from the curse of dimensionality. Since we were aiming to build an real-time 3D tracker, we required that both of these conditions be satisfied.

To perform 3D tracking, we require the coordinates of the object in the real world, X, Y, Z, its dimensions (approximated as a 3D bounding box) l, w, h and the orientation of the box about the vertical axis, θ. Since objects in scenes naturally move along the horizontal axes, x and y, and but not in vertical direction, and that it is also unlikely that an object would rotate about the vertical axis with a constant angular velocity, we only model the velocities along x and z axes (horizontal axes). Hence, the state representation is $\overrightarrow{t} = \{x, y, z, l, h, w, \theta, v_x, v_z\}$.

The Kalman filtering consists of two steps. The prediction, and the update. Each is done per track, independently of other tracks. The assumption is that the movement of one track does not depend on the movements of others. This simplifying assumption is made to avoid modelling interactions between the tracked objects.

Since rapid changes to an object's velocity, when sensor data is captured at 15fps, is not expected, we use a constant velocity motion model to perform the prediction step.

To leverage the joint nature of the detections, we leverage a dual measurement update. Each track has two measurement sources, the 2D bounding boxes, as well as the 3D bounding boxes. The probability of association of each of the measurements is known from JPDA. We perform joint filtering with both these measurement sources. This joint filtering assumes that the 2D and 3D measurement sources are independent, although this is not strictly the case. The 3D measurements are generated by F-PointNet while conditioned on the 2D measurements. However, based on the results in V-A, we believe that this assumption is justified. Therefore, the update step involves first performing a PDA [40] update of the Kalman Filter, followed by a PDA update with the 2D measurements. We choose to update with the 3D measurement first, as it serves as the primary measurement source, with the 2D measurement acting as a fine tuning measurement on top.

The second update using the 2D measurements is done using Extended Kalman Filtering, with a non-linear observation model. For a real world point (X, Y, Z) the Jacobian of the projection to the 2D RGB cylindrical image, $\mathcal{H}_{2D}$ is given by:

$$\mathcal{H}_{2D} = \begin{bmatrix} \frac{c_x z}{x^2+z^2} & 0 & \frac{-c_x x}{x^2+z^2} \\ \frac{c_y xy}{(x^2+z^2)^{\frac{3}{2}}} & \frac{-c_y}{\sqrt{x^2+z^2}} & \frac{c_y zy}{(x^2+z^2)^{\frac{3}{2}}} \end{bmatrix}$$

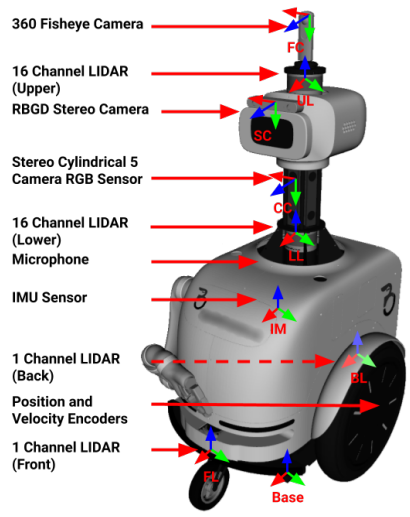
where c_x and c_y are constants related to the camera parameters. Details of the derivation can be found in the supplementary material. We further compute the derivative of the corners of 3D box with respect to the center of the box, the dimensions and the orientation, thus obtaining by the chain rule, the derivative of the transform from the mean to the corners of the 3D bounding box projected onto the 2D image. We then select the appropriate derivative by selecting the corners of the 3D bounding box which form the extremes of the projected 2D bounding box.

G. Track Management

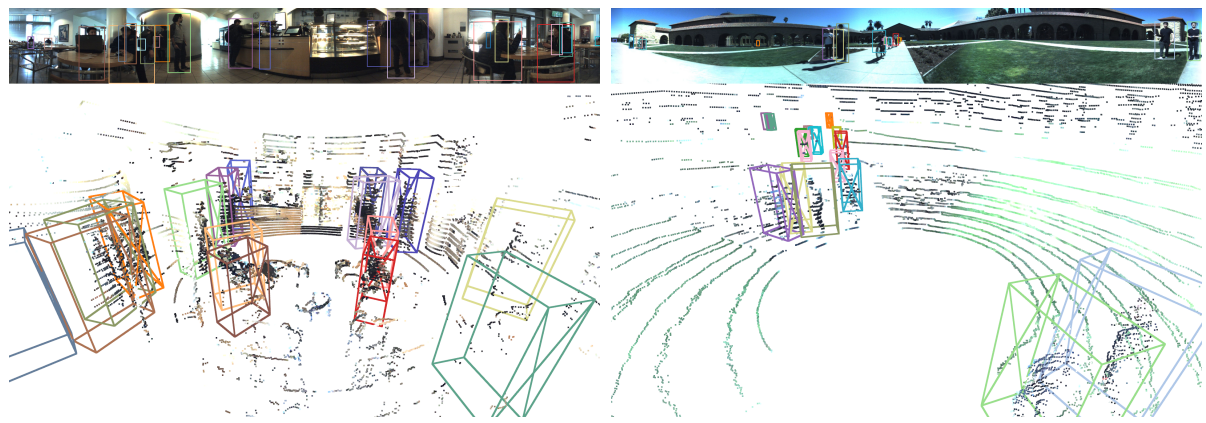
When performing the data association step, we compare detections at time t to the predicted track $\hat{\mathcal{T}}_t$. Thus, we require that the track has an associated feature. However, since we utilise JPDA to perform detection to track association, we only have access to the probability distribution of an association for each track across all detections. To obtain the feature to be associated with the track, we perform a linear sum assignment on the JPDA output using the Hungarian algorithm [37]. To avoid associating a track with an unlikely detection, we set a threshold p_{assn}, as the minimum probability for a match to be considered. This is the criteria for a track $\mathcal{T}$ to have a match at frame t.

The discussion so far has revolved around the assumption that we have tracks at time $t-1$ and after performing association, we update the track state to account for measurements at time t. However, when a new object enters the scene, we must also determine if and when to initiate a new track. Similarly, when an object leaves the scene, the track corresponding to it must be terminated. To perform these operations, we utilise a track management system - the tracker. To prevent false positive detections from initiating tracks, we require that a new track be initiated only if it is outside the gate of all existing tracks. Further, to deal with false positives which are far enough away from existing tracks, we only confirm a track after n_{init} number of consecutive matches. When an object leaves the scene, there are no longer detections corresponding to it. Thus, we terminate a track if there has been no matching detection for n_{term} frames.

Since our system has 2D detections with associated 3D detections, there are unique situations which arise. To deal with the case where an object has a 2D detection, but not a corresponding 3D detection, we utilise a two step process. In the first step, all measurements with 3D detections are associated with tracks through the procedure listed above. To fully utilise the information from the 2D detections which do not have an associated 3D detection, we then do a second round of data association (cost matrix selection, gating, and JPDA). This only operates in the 2D domain. The appearance cost matrix is now only based on the 2D feature descriptor, and the IoU cost is calculated with 2D IoU. This is then



(a) JackRabbot, our data collection platform and its equipped sensors



(b) Sample visualization of the dataset with stationary (left) and moving (right) robot; Top: 2D stitched 360°panorama with human-annotated 2D bounding boxes; Bottom: 3D pointclouds with human-annotated 3D oriented bounding boxes

used to perform a PDA update of the tracks with the 2D measurement only.

IV. Dataset

The development of 2D MOT system has been supported by datasets with annotated ground truth tracks. However, these datasets are scarce for 3D MOT and are focused on autonomous driving scenario. Yet, human-centric autonomous agent is another crucial domain where high level 3D scene understanding is essential. To this extent, we present a novel dataset, the **JRDB dataset**, focused on human environments. Our dataset contains **64** minutes of sensor data acquired from our mobile robot JackRabbot comprising **54** sequences indoors and outdoors in a university campus environment. In this section, we briefly describe the data collection and labeling process of the dataset. For full details, please refer to the supplementary material.

A. *JackRabbot*

We collected our multimodal dataset with the sensors on-board of our mobile-manipulator JackRabbot. JackRabbot is a custom-design robot platform tailored to navigate and interact in human environments. JackRabbot is equipped with various state-of-the-art sensors that are commonly used in autonomous cars such as stereo RGB 360°cylindrical video streams, continuous 3D point clouds from two LiDAR sensors, audio, and GPS sensing. In Fig 2a, we visualize JackRabbot and its on-board sensors. We are hoping to tackle interesting problems from perception to high-level social interaction of the robots through JackRabbot.

B. *Data collection and annotation*

We collect data from 30 different locations indoors and outdoors, all in a university campus environment, with varying and uncontrolled environmental conditions such as illumination and other natural and dynamical elements. We also ensure the recorded data captures a variation of natural human posture, behaviour and social activities in different crowd densities in indoor and outdoor environments. Furthermore, to incorporate a diversity in the robot's ego-motion, we use a combination of static and moving sensor (robot) views to capture the data. Additionally, we stitch all 2D images from 5 cameras into a single 360°panorama in order to perceive the full surroundings in both 2D and 3D domain.

The very first step to build social autonomous mobile agent is in understanding the location and movements of humans surrounding the robot. Therefore, we annotate the following ground truth label for the data we collected: a) 2D bounding boxes in each camera and the stitched 360°panorama images for human/pedestrian class, b) 3D oriented bounding boxes from LiDAR data for human/pedestrian class, c) an association link between 2D and 3D bounding boxes d) time consistent trajectories (tracks) for all annotated persons in both 2D and 3D. In Fig 2b, we visualize the dataset we collected alongside with the ground truth labels we annotated on both 2D 360°cylindrical image and 3D LiDAR pointclouds.

With this unique domain of data, we are hoping to encourage the research of human-centric social autonomous agent. In the near future, we are planning to augment this data with other annotations such as 2D human skeleton pose and individual, group and social activity labels. As a first step of this effort toward social autonomous agent, we evaluate our proposed 3D tracking method as a competitive baseline of this dataset.

V. Experimental Evaluation

Setup: MOT evaluation is most commonly done using the Clear-MOT metrics outlined in [41]. The relevant metrics are Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), Mostly Tracked tracks (MT), Mostly Lost tracks (ML), False Positives (FP), False Negatives (FN), ID Switches (IDS) but in the result tables we only report MOTA, MOTP, MT, and Runtime for conciseness (additional results are included in the supplementary material). However, widespread uses of these metrics in datasets are restricted to 2D tracking evaluation. In this setting, a track is marked as a true positive if it has an IoU greater than t_m with a ground truth track, for a given frame. Since it is our objective to track and evaluate in 3D, we extend this evaluation by replacing the 2D IoU with 3D IoU between

the predicted track's 3D bounding box and the ground truth 3D bounding box. The 3D IoU is calculated using a combination of the Sutherland-Hodgman algorithm [42] and the shoelace formula (surveyor's formula) to determine the area of intersection.

To evaluate our system with the aforementioned metrics, we run experiments on our novel JRDB and the well-established KITTI [27] datasets. We report results on KITTI to provide a comparison to other MOT existing systems, and results on JRDB as first and competitive set of results for 3D people tracking.

The KITTI benchmark has both 2D RGB data and 3D pointcloud data, but only evaluates tracking in 2D using the Clear-MOT metrics. In order to run our system on KITTI, we use the 2D tracking head shown in 1. We utilise 3D information, such as distance of the object from the camera, and the 3D velocity of an object with respect to the car, to heuristically set the filtering parameters. While 2D tracking is not the objective of this framework, strong results would reinforce the belief that 2D and 3D tracking are inherently linked and demonstrate the advantages of our framework. In order to test this hypothesis, we use publicly available detections to get comparable results. Additionally, we use [35] as a vehicle re-identification module instead of AlignedReID because the latter was specially designed for human appearance. The 2D tracking evaluation takes place for cars and pedestrians separately.

The JRDB contains both RGB and pointcloud inputs. On the JRDB benchmark, we exclusively perform 3D tracking using JRMOT, as described in III. The objective is 3D tracking of pedestrians. Evaluation utilises the clear-MOT metrics modified to handle 3D tracking, as mentioned above.

All experiments are executed on a machine equipped with 4 Intel Xeon CPUs at 2.80 GHz and a Nvidia GTX Titan X GPU. The hyperparameters used have been included in the supplementary material.

A. Results

	MOTA ↑	MOTP ↑	MT ↑	Runtime ↓
MASS [43]	85.04%	85.53%	**74.31%**	0.01s
mmMOT [33]	84.77%	85.21%	73.23%	0.01s
3DT [1]	84.52%	85.64%	73.38%	0.03s
MOTBP [25]	84.24%	**85.73%**	73.23%	0.3s
IMMDP [44]	83.04%	82.74%	60.62%	0.19s
aUToTrack [45]	82.25%	80.52%	72.62%	0.01s
JCSTD [46]	80.57%	81.81%	56.77%	0.01s
Ours	**85.70%**	85.48%	71.85%	0.07s

TABLE I: Results on Online KITTI Car Tracking

KITTI Dataset: Table I shows our results in the car tracking challenge. We achieve state of the art (highest MOTA) performance among all online published methods. We show competitive results (top 5) when considering MOTP, for all online real time published methods. Further, our MOTP is within 0.5% of the leader. Table II shows our results in the pedestrian tracking challenge. Here we demonstrate competitive results (high MOTA) especially among real-time (performance over 10 fps) trackers, where our tracker ranks second.

It is important to note we primarily use publicly available detections on KITTI from SubCNN [47] for pedestrians and RRC [48] for cars. This allows us to focus the evaluation on the performance of the tracking and compare to other methods that also use these detections [33], [25], [44]. The performance gains in our method are a consequence of fusing and leveraging both 2D and 3D information as opposed to training a better detector, demonstrating how joint utilization of information can lead to better tracking. These results also indicate that even though our focus is 3D MOT, JRMOT is also able perform competitively on 2D MOT benchmarks compared to state of the art methods. However, using the publicly available detections on KITTI hinders our method since better detections lead to better tracking performance, as suggested by the fact that in the pedestrian benchmark all top methods utilize private detections from more modern detectors.

JRDB Dataset: We compare our results on people tracking on JRDB to the performance of one other method AB3DMOT [30]. This method was selected as it is the only other 3D MOT solution with available code released more than a month prior to submission. AB3DMOT requires as input 3D detections for which we use the detections from F-PointNet. On the test set, our tracker results in better performance. We report performance of **20.2%** MOTA while running at **25 fps**. This shows that our dataset is extremely challenging. It is reasonable to believe, that since there are $765,907$ false negatives on the test set, the limiting factor in our tracking system is the 3D detection. The baseline method, AB3DMOT, has a performance of 19.3%. We note that even though the baseline method uses 3D IoU based association and no descriptor information, its performance is close to that of JRMOT. This seemingly counter intuitive fact can be due to the following: Even though 2D IoU is not a reliable association metric, 3D IoU is extremely good. However, there are a few cases where 3D IoU on its own cannot effectively accomplish the data association. We hypothesise our improved results are due to our fusion of both 2D and 3D measurements as confirmed by our ablation studies.

Finally, to analyse the contribution of the individual components in the overall performance of JRMOT we perform a set of ablation studies. First, we conduct an experiment where we update the filter only with 2D measurements. As expected, we observe that the 3D measurements are critical for good filtering, the lack of which results in -20.1% MOTA on the train set. To analyse the contribution of the 2D appearance feature we use only 3D IoU as association metric. In this case, we see a small degradation in performance of 0.1% MOTA. This is in line with our intuition that the 3D IoU is the most informative association metric, but it can be slightly improved in some corner cases with 2D appearance. Our last ablation is to verify that 2D inputs without their corresponding 3D bounding boxes are indeed helpful measurements in our filter. We notice that although the MOTA stays constant at 42.9%, MOTP drops

0.6% when 2D detections which do not have associated 3D detections are not used. This confirms our intuition that the 2D measurement can be used to make fine updates on the tracked orientation and location. Visualizations of the resulting 3D tracks with our method and the ablated versions compared to ground truth are included in the supplementary material.

	MOTA ↑	MOTP ↑	MT ↑	Runtime ↓
CAT [49]	**52.35%**	71.57%	**34.36%**	*Not Reported*
Be-Track [50]	51.29%	**72.71%**	20.96%	0.02s
MDP [44]	47.22%	70.36%	24.05%	0.9s
JCSTD [46]	44.20%	72.09%	16.49%	0.07s
SCEA [51]	43.91%	71.86%	16.15%	0.06s
RMOT [52]	43.77%	71.02%	19.59%	0.02s
AB3DMOT [30]	36.36%	64.86%	14.09%	0.0047s
Ours	45.98%	72.63%	23.02%	0.06s

TABLE II: Results on Online KITTI Pedestrian Tracking

VI. Conclusion and Future Work

We presented JRMOT, a new 3D MOT system that fuses the information contained in 2D RGB images and 3D point-clouds in an efficient manner and provides robust tracks even in adversarial and highly crowded environments. As part of our project we release the JRDB dataset, a novel dataset for 2D and 3D MOT evaluation and development containing multimodal streams acquired in human environments, university indoor buildings and pedestrian areas on campus, including scenes where the robot navigates among humans. The dataset has been annotated with ground truth 2D bounding boxes and associated 3D cuboids around all persons in the scenes which will help future research in 3D MOT. We establish a strong baseline for 3D MOT with JRMOT. JRMOT achieves state of the art performance in the well-known KITTI 2D MOT benchmark and shows better performance than the existing 3D MOT system in our provided JRDB dataset.

References

[1] 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 International Conference on Computer Vision*, 2019, pp. 5390–5399.

[2] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, "Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 433–440.

[3] 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," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3569–3577, 2018.

[4] V. Vaquero, I. del Pino, F. Moreno-Noguer, J. Sola, A. Sanfeliu, and J. Andrade-Cetto, "Deconvolutional networks for point-cloud vehicle detection and tracking in driving scenarios," in *2017 European Conference on Mobile Robots (ECMR)*. IEEE, 2017, pp. 1–7.

[5] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask r-cnn," 2017.

[6] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from RGB-D data," *CoRR*, vol. abs/1711.08488, 2017. [Online]. Available: http://arxiv.org/abs/1711.08488

[7] X. Zhang, H. Luo, X. Fan, W. Xiang, Y. Sun, Q. Xiao, W. Jiang, C. Zhang, and J. Sun, "Alignedreid: Surpassing human-level performance in person re-identification," *arXiv preprint arXiv:1711.08184*, 2017.

[8] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[10] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixe, "Cvpr19 tracking and detection challenge: How crowded can it get?" *arXiv preprint arXiv:1906.04567*, 2019.

[11] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Joint attention in autonomous driving (jaad)," *arXiv preprint arXiv:1609.04741*, 2016.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[13] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[14] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," *arXiv preprint arXiv:1903.05625*, 2019.

[15] W. Feng, Z. Hu, W. Wu, J. Yan, and W. Ouyang, "Multi-object tracking with multiple cues and switcher-aware classification," *arXiv preprint arXiv:1901.06129*, 2019.

[16] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[17] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *CoRR*, vol. abs/1703.07402, 2017. [Online]. Available: http://arxiv.org/abs/1703.07402

[18] Y.-C. Yoon, D. Y. Kim, K. Yoon, Y.-m. Song, and M. Jeon, "Online multiple pedestrian tracking using deep temporal appearance matching association," *arXiv preprint arXiv:1907.00831*, 2019.

[19] W. Choi, "Near-online multi-target tracking with aggregated local flow descriptor," *ICCV*, 2015.

[20] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4696–4704.

[21] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[22] H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, and I. R. A. Dick, "Joint probabilistic data association revisited," *ICCV*, 2015.

[23] H. Shen, L. Huang, C. Huang, and W. Xu, "Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking," *arXiv preprint arXiv:1808.01562*, 2018.

[24] G. Wang, Y. Wang, H. Zhang, R. Gu, and J.-N. Hwang, "Exploit the connectivity: Multi-object tracking with trackletnet," in *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, 2019, pp. 482–490.

[25] 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 *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 3508–3515.

[26] 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.

[27] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[28] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 954–960.

[29] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[30] X. Weng and K. Kitani, "A Baseline for 3D Multi-Object Tracking," *arXiv:1907.03961*, 2019. [Online]. Available: https://arxiv.org/pdf/1907.03961.pdf

[31] A. Asvadi, P. Girão, P. Peixoto, and U. Nunes, "3d object tracking using rgb and lidar data," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 1255–1260.

[32] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki, "Fantrack: 3d multi-object tracking with feature association network," *arXiv preprint arXiv:1905.02843*, 2019.

[33] J. Luiten, T. Fischer, and B. Leibe, "Track to reconstruct and reconstruct to track," 2019.

[34] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[35] C.-W. Wu, C.-T. Liu, C.-E. Chiang, W.-C. Tu, and S.-Y. Chien, "Vehicle re-identification with the space-time prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 121–128.

[36] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[37] H. W. Kuhn and B. Yaw, "The hungarian method for the assignment problem," *Naval Res. Logist. Quart*, pp. 83–97, 1955.

[38] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic matching using m-best solutions," in *CVPR*, 2016.

[39] R. Jinan and T. Raveendran, "Particle filters for multiple target tracking," *Procedia Technology*, vol. 24, pp. 980–987, 2016.

[40] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009.

[41] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal on Image and Video Processing*, vol. 2008, p. 1, 2008.

[42] I. E. Sutherland and G. W. Hodgman, "Reentrant polygon clipping," *Communications of the ACM*, vol. 17, no. 1, pp. 32–42, 1974.

[43] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE*, vol. 7, pp. 104 423–104 434, 2019.

[44] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[45] K. Burnett, S. Samavi, S. L. Waslander, T. D. Barfoot, and A. P. Schoellig, "autotrack: A lightweight object detection and tracking system for the SAE autodrive challenge," *CoRR*, vol. abs/1905.08758, 2019. [Online]. Available: http://arxiv.org/abs/1905.08758

[46] W. Tian, M. Lauer, and L. Chen, "Online multi-object tracking using joint domain information in traffic scenarios," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2019.

[47] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2017, pp. 924–933.

[48] J. S. J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y. Tai, and L. Xu, "Accurate single stage detector using recurrent rolling convolution," *CoRR*, vol. abs/1704.05776, 2017. [Online]. Available: http://arxiv.org/abs/1704.05776

[49] U. Nguyen, F. Rottensteiner, and C. Heipke, "Confidence-aware pedestrian tracking using a stereo camera," *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2/W5, pp. 53–60, 05 2019.

[50] P. W. Dimitrievski M, Veelaert P, "Behavioral pedestrian tracking using a camera and lidar sensors on a moving vehicle," *Sensors*, vol. 19, 2019.

[51] J. Hong Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon, "Online multi-object tracking via structural constraint event aggregation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[52] J. H. Yoon, M. Yang, J. Lim, and K. Yoon, "Bayesian multi-object tracking using motion context from multiple objects," in *2015 IEEE Winter Conference on Applications of Computer Vision*, Jan 2015, pp. 33–40.