

# 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 Rezaatofghi, Roberto Martín-Martín, Silvio Savarese

**Abstract**—Robots navigating autonomously need to perceive and track the motion of objects and other agents in its surroundings. This information enables planning and executing robust and safe trajectories. To facilitate these processes, the motion should be perceived in 3D Cartesian space. However, most recent multi-object tracking (MOT) research has focused on tracking people and moving objects in 2D RGB video sequences. In this work we present JRMOT, a novel 3D MOT system that integrates information from RGB images and 3D point clouds to achieve real-time, state-of-the-art tracking performance. Our system is built with recent neural networks for re-identification, 2D and 3D detection and track description, combined into a joint probabilistic data-association framework within a multi-modal recursive Kalman architecture. 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. *JRDB* contains over 60 minutes of data including 360° cylindrical RGB video and 3D pointclouds in social settings that we use to develop, train and evaluate JRMOT. The presented 3D MOT system demonstrates state-of-the-art performance against competing methods on the popular 2D tracking KITTI benchmark and serves as first 3D tracking solution for our benchmark. Real-robot tests on our social robot JackRabbit indicate that the system is capable of tracking multiple pedestrians fast and reliably. We provide the ROS code of our tracker at <https://sites.google.com/view/jrmot>

## I. INTRODUCTION

An autonomous agent such as a mobile robot needs to move between two locations in a safe and robust manner. To navigate safely, the robot 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 the agent to predict the possible future trajectories of the other agents and to plan and execute motion strategies that take them into account.

To facilitate navigation, the motion of the other agents needs to be perceived and represented in the same space the navigation takes place, the 3D Cartesian space. However, most efforts from the robotics and computer vision communities have been dedicated to the development of multi-object tracking (MOT) systems that perceive 2D motion from RGB video streams. The reason for this is two fold. First, detecting and tracking objects in 3D is computationally more expensive than in 2D due to the *curse of dimensionality* in this search problem. And second, there is a lack of adequate

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

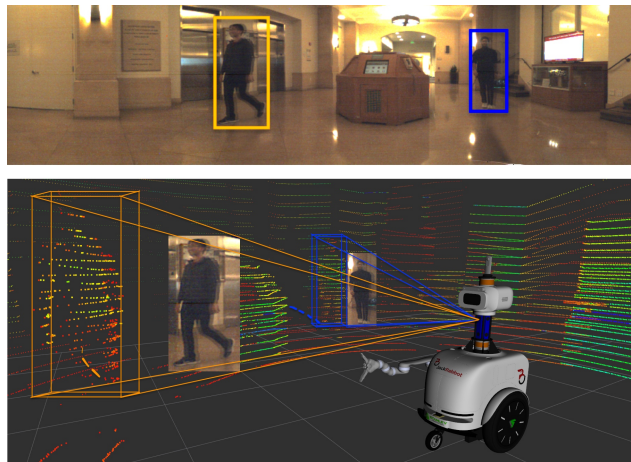


Fig. 1: Robots navigating in human environments need to detect and track humans and other moving targets using their sensor information; JRMOT integrates information from 2D RGB images (top), where appearance is more easily discernible, and 3D point clouds (bottom), where objects are well separated, in a tightly coupled manner to provide real-time 3D multi-object tracking information; JRMOT is developed on the *JRDB* dataset, a novel annotated dataset captured with our social mobile manipulator JackRabbit (bottom)

large-scale curated datasets of 3D data with annotations of moving agents from the perspective of navigating robots in human environments, impeding the application of successful deep learning techniques to 3D tracking.

In this paper we present JRMOT, a novel real-time multi-object detection and tracking framework in 3D Cartesian space. JRMOT detects and tracks multiple targets around the agent by constraining the 3D search with 2D cues, effectively combining information from RGB cameras and LiDAR sensors. RGB images and 3D pointclouds carry complementary information. On the one hand, RGB images are **dense**, which allows us to discern appearances of objects to effectively detect, identify and classify them even at large distances. It is also structured in the form of a **pixel grid**, well suited to be processed with effective tools such as CNNs. On the other hand, 3D point-cloud data is **sparse** but the depth information allows us to separate objects that might overlap in the 2D image space. However, the unordered structure of the pointclouds do not allow for the use of efficient algorithmic architectures such as CNNs. JRMOT leverages the information of each modality (appearance in RGB, geometry in point clouds) to address the shortcomings of the other by i) sequentially processing them to guide the

3D search in regions indicated by the RGB image, ii) fusing their information into a multi-modal descriptor to facilitate tracking and data-association, and iii) updating the tracking state with a novel multi-modal measurement model.

At its core, JRMOT applies state-of-the-art deep neural network architectures to detect objects of interest in RGB images and 3D point clouds, and to characterize tracks with novel multi-modal descriptors, improving the performance of well-established data-association and filtering techniques. Training such networks requires a large amount of 2D RGB images and 3D pointclouds annotated with ground truth labels of the location of objects of interest. The annotated data should be acquired from the perspective of the agent that will execute JRMOT, i.e. a mobile robot. JRMOT is trained with annotated data from a novel dataset of multimodal data, the JRDB dataset. The dataset is captured from the perspective of our social autonomous agent, JackRabbit, leading to a first-of-its-kind dataset that includes **indoor and outdoor scenes**, with over **4.2 million annotated bounding boxes** in 2D RGB images and 3D pointclouds. This dataset enabled us to leverage the complementarity of 2D RGB and 3D pointcloud data with JRMOT.

To summarize, our contributions are:

- 1) We present JRMOT, a novel real-time online 3D MOT system that fuses 2D and 3D information based on latest deep-learning architectures.
- 2) We release the JRDB dataset and benchmark, a first of its kind 2D+3D dataset for the development and evaluation of 2D-3D MOT frameworks and 2D-3D people detection. JRMOT is developed and evaluated on JRDB and serves as first competitive baseline.
- 3) We show that JRMOT achieves **state-of-the-art performance** in the competitive KITTI 2D tracking benchmark. Our tests also indicate that our method can detect and track effectively in real time, running onboard a mobile robot, with only few ID switches and a single missed track in over 100s of experiments. We provide JRMOT as ROS code for other researchers to test and build upon.

## II. RELATED WORK

In recent years, there have been impressive advances in MOT, mostly focused on 2D tracking (in images) with some exceptions of new 3D MOT systems (from images and/or 3D data). We now review previous work in the areas of 2D MOT from 2D RGB videos, 3D MOT from 2D RGB and/or 3D sensors, real-time 3D MOT systems and existing datasets for MOT with 3D data.

**2D MOT with 2D Data:** Tracking in 2D is the task of perceiving continuously the motion of objects in video sequences. There exists a large body of literature for 2D MOT. Works such as [1], [2], [3], [4] leverage the success of deep learning architectures for re-identification, and utilise appearance cues for track-detection association. Other works such as [5], [6], use motion and continuity cues to do the same. JRMOT builds on top of this body of literature - we use 2D RGB to obtain appearance descriptors using deep learning based feature extractors, and use 3D pointcloud data

to circumvent two problems faced by 2D MOT works. First, the problem of occlusion in 2D is largely reduced due to the large separation of objects in 3D space. Second, motion in 3D is often much more simple than the corresponding projected motion in the 2D RGB image and hence can be used as a much better cue for association. Further, the resulting tracks are in 3D space, which is required by most applications involving autonomous agents.

**3D MOT with 2D and/or 3D Data:** With the advent of self driving cars, access to large scale datasets containing LiDAR data [7], [8], [9], [10] has reinvigorated interest in the use of 3D sensors. Some methods use exclusively 3D detections and pointcloud data to perform 3D tracking [11], [12], not using any information from RGB images of the scene. JRMOT uses pointcloud data combined with RGB information to improve on tracking of far away objects; this is explored in Fig 4, which shows that as the distance of an object from the robot increases, our method largely mitigates the drop in performance seen in the baseline.

Baser *et al.* [13] aggregate both 2D RGB appearance descriptor and the bounding box coordinates to learn a similarity function to perform 3D tracking. It independently detects in the 2D and 3D domains, and also only utilises 3D measurements to perform filtering, distinct from our work which tightly couples both 2D and 3D measurements. Luiten *et al.* [14] utilise RGB and depth information to reconstruct a 4D spatio-temporal scene, but unlike our method do so in an offline setting. The effectiveness of these techniques has **not been quantitatively tested** because of the lack of a large scale 3D tracking benchmark. They are often evaluated via the proxy of 2D tracking, or on custom generated small-scale datasets.

**Real-Time 3D MOT Systems** Whereas there is a plethora of real-time 2D MOT systems such as [15], [16], [3], among many others, the community has developed just a handful of real-time 3D MOT systems. Koide *et al.* [17], [18] both propose real time 3D MOT systems based exclusively on 3D LiDAR data without incorporating 2D data. This loose (or complete lack of) coupling of 2D and 3D information is sensitive to cases where 3D detection intermittently fails, whereas the 2D detection is robust, a key advantage of our method as shown in Sec V. Linder *et al.* [19] and Dondrup *et al.* [20] both utilise 2D and 3D data, but do not leverage recent advances in deep learning based detectors and feature descriptors.

**3D Datasets:** 3D sensory systems are becoming increasingly commonplace in sensor suites of autonomous agents. Datasets with this multi-modal data such as KITTI [8], Apolloscape [9], NuScenes [7] and Oxford's Robotic Car [21] have widely driven research in the 3D community. Nonetheless, their targeted domain of application is autonomous driving; the data they provide is captured from sensor suites on top of cars and only depicts streets, roads and highways. Frossard *et al.* [22] specifically mention a lack of available 3D tracking benchmarks.

In this paper, we target a unique visual domain tailored to the perceptual tasks related to navigation in human en-

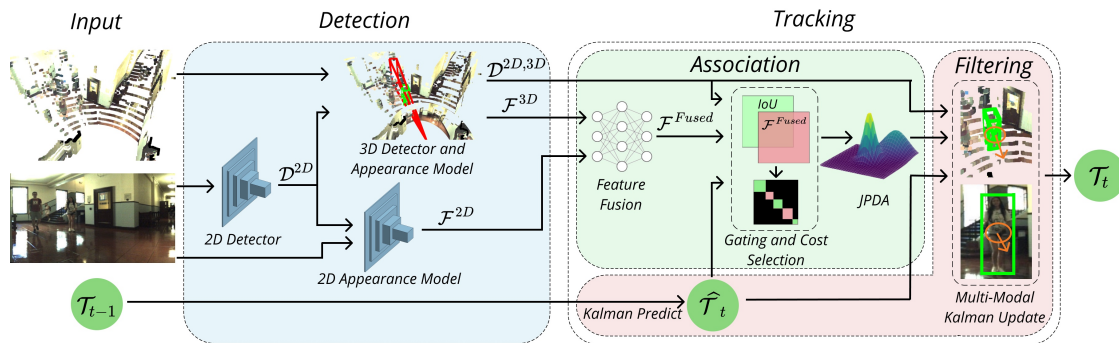


Fig. 2: JRMOT: Our proposed 3D MOT system is composed of a Detection block, that includes the 2D detector (Sec III-A), 2D appearance model (Sec III-B), 3D detection and feature extractor (Sec III-C), and a Tracking block containing data association (Sec III-D and Sec III-E) and filtering (Sec III-F) and track management (Sec III-G) components;  $\mathcal{T}$ ,  $\mathcal{D}$ ,  $\mathcal{F}$  refer to tracks, detections, features respectively with the superscript indicating the space; The system integrates information from 2D RGB images and 3D pointclouds into a single 3D multi-object recursive estimation tracker with real-time performance

vironments, both indoors and outdoors, in crowded scenes. We hope that this new domain provides the community an opportunity to develop visual perception frameworks, limited not only to self-driving cars but also various other types of autonomous navigation agents. Furthermore, we hope this dataset and benchmark will support and drive research in a variety of domains related to social robotics, including but not limited to human detection and tracking.

### 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. 2. JRMOT performs tracking by detection. The detector block contains a 2D detector, a 2D appearance feature extractor and a 3D detector (which also generates a 3D feature descriptor). The detector block takes as input a 2D RGB image and the corresponding pointcloud and produces 2D and 3D detections of all objects of interest, along with their 2D and 3D feature descriptors. This is then passed to the tracking block, which performs data association, as well as multi-modal Bayesian filtering. The output of our system is the location in 3D space of all tracked objects, each uniquely identified over time by a track ID. We assume the extrinsic calibration between the RGB camera and the depth sensor to be known. We now explain each component in detail.

#### A. 2D Detection

First, our system needs to detect all moving instances of objects of interest in the environment. Although we are interested on 3D locations, 2D detectors are faster, more robust and accurate than 3D detectors [23]. Therefore, we exploit state-of-the-art image segmentation (Mask R-CNN [24]) or object detector (YOLO [25] modified for real time) architectures as our detector. The input to this module is a 2D RGB image at time  $t$  and the output is a set of  $N$  detections in 2D,  $\mathcal{D}_t^{2D} = \{(u, v, w, h)_0, \dots, (u, v, w, h)_{N-1}\}_t$ , 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. The available pretrained models have been trained with

different types of images than the ones our robot encounters during navigation. We make use of our JRDB dataset to finetune the networks and adapt them to the special data distribution of the social navigation setup<sup>1</sup>.

#### B. 2D Appearance

The detections from the previous step need to be associated to existing tracks in JRMOT. To this end, we featurize the 2D appearance (appearance in the RGB image) of both detections and tracks in order to compare features and associate them later (Section III-E). We compute Aligned-ReID [26] features when the objects of interest are people, and features from Wu et al. [27] when they are vehicles. The choice of these features is based on their high discriminative capabilities and fast computation time. Both features are trained on JRDB. The input to this module is the 2D RGB image at time  $t$  and the  $N$  detections from the previous step, and the output are their 2D appearance features,  $\mathcal{F}_t^{2D} = \{f_0^{2D}, \dots, f_{N-1}^{2D}\}_t$ .

#### 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 integrate 2D RGB and 3D data provided by a depth sensor, which is a common part of most autonomous navigating systems. We utilise F-PointNet [28], a state-of-the-art algorithm to obtain 3D detections in the form of an oriented cuboid around the object instance for every 2D bounding box. F-PointNet estimates a 3D bounding box for that object within the frustum starting at the RGB camera center and passing through the 2D bounding box as illustrated in Fig 1. We choose F-PointNet because it explicitly gives us an association between every 2D and 3D bounding box, it leverages the robustness of 2D detectors, it has a relatively fast inference time, and it has been shown to be one of the

<sup>1</sup>Our detections are publicly released as part of the JRDB dataset and benchmark for others to use in their MOT systems.

top performing 3D detectors on the KITTI benchmark. 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 time closest to  $t$ . The output is a set of  $M$  detections in 3D for the class of interest at time  $t$ ,  $\mathcal{D}_t^{3D} = \{(x, y, z, w, h, l, \theta)_0, \dots, (x, y, z, w, h, l, \theta)_{M-1}\}_t$ , 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  $\theta_j$  is the rotation of the box around the normal to the floor plane.

Additionally, we exploit the F-PointNet architecture to generate feature descriptions of the shape of the detected objects,  $\mathcal{F}_t^{3D} = \{f_0^{3D}, \dots, f_{M-1}^{3D}\}_t$ . The feature from the penultimate layer of F-PointNet is used to regress the 3D bounding box, and thus, it contains information about the 3D shape of the object. We use this feature as a 3D appearance (shape) descriptor. Note that not every 3D detection has an associated 2D detection. It is possible that F-PointNet does not find a reasonable bounding box within every frustum. Our system accounts for this case, as explained in Sec. III-G.

#### D. Feature Fusion

Due to the coupling of 2D and 3D detections, each object of interest now has a 2D feature descriptor, and a 3D feature descriptor. Depending on the conditions (distance, visibility, occlusion) both 2D and 3D appearance can contain valuable information to associate detections and previous tracks. Therefore, we fuse the 2D and 3D features with a 3-layered fully connected network that receives as input  $\mathcal{F}_t^{cat}$ , given by  $\{[f_0^{2D}, f_0^{3D}], \dots, [f_{M-1}^{2D}, f_{M-1}^{3D}]\}_t$  where  $\square$  denotes concatenation. We train this fusion network via metric learning based on a triplet loss and semi-hard negative mining as in Schroff *et al.* [29], resulting in a robust feature for association between new detections and previous tracks.

#### E. Data Association

Given a set of detections at time  $t$ , we need to associate them to tracks at  $t - 1$  to update the tracks' locations and appearances. To do so we utilize JPDA [30] as it has been shown by [6] to be robust to clutter and reduce the occurrence of ID switches. JPDA requires a cost matrix,  $C \in \mathbb{R}^{K \times N}$  in which element  $c_{ij}$  represents the cost associated with matching track  $i$  to detection  $j$ . We utilize both appearance and 3D spatial location to associate objects. We first compute appearance similarity by calculating the pairwise  $\ell_2$  distance between the  $N$  features of detections and the  $K$  features of tracks and build an appearance cost matrix,  $C_{app} \in \mathbb{R}^{K \times N}$ . Then, we compute the location similarity by calculating the pairwise 3D bounding box intersection over union (IoU) assuming that both 3D bounding boxes have the same orientation (same  $\theta$ ), an approximation that generates fairly good results in much shorter computation time. The result is an IoU cost matrix,  $C_{IoU} \in \mathbb{R}^{K \times N}$ . To simplify the association, we perform *gating* with the Mahalanobis distance (M-distance) with a fixed threshold (0.95 quantile from the  $\chi^2$  distribution)

As the size of the cost matrices scales with the square of the number of objects in the scene, association with the entire cost matrix can lead to slow computation. We therefore construct an undirected graph, where every track and detection is a node, and 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. We perform further processing on a per cluster basis leading to a much lower computation time.

Since JPDA requires a single cost matrix representing the association costs, we perform a cost matrix selection (IOU vs. appearance) based on an **entropy measure**. We select the cost matrix that has a lower entropy per track. A lower entropy cost matrix implies that the cost matrix is more 'peaked', and hence more discriminative.

Given the selected cost matrix, we perform JPDA. To maintain the speed of our tracker, we employ the m-best solution approximation [31] for large clusters. For smaller clusters, complete enumeration is used to obtain the exact solution of JPDA.

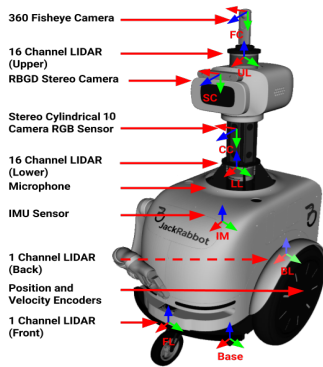
To deal with the case where an object has a 2D detection, but not a corresponding 3D detection, we utilise a two step process. In the first step, all measurements with both 2D and 3D detections are associated with tracks using the procedure above. We then do a second round of cost matrix selection, gating, and JPDA with the appearance cost matrix now only based on the 2D feature descriptor, and the IoU cost calculated with 2D IoU.

#### F. Filtering

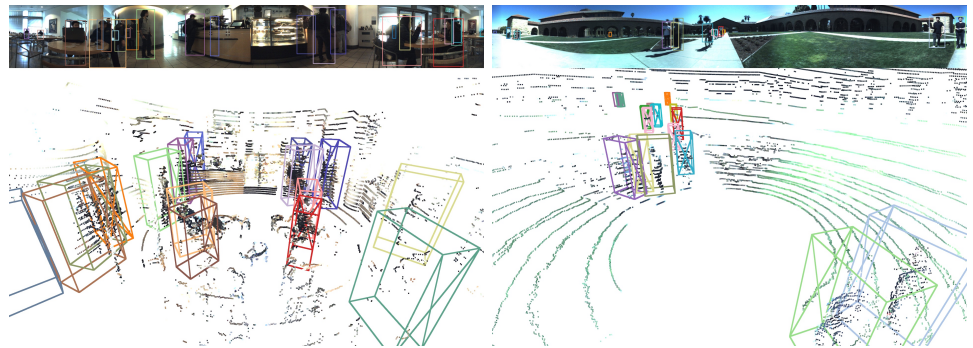
2D and 3D detections are often noisy. Therefore, we filter them over time with a Kalman filter [32] to estimate smooth 3D tracks. The Kalman filter is an optimal estimator (assuming Gaussian noise and linearity in motion) and an online, computationally efficient process that allows JRMOT to be accurate and real time.

The state to estimate per object includes its 3D location,  $x, y, z$ , its dimensions approximated as a 3D bounding box,  $l, w, h$  and the rotation of the box about the vertical axis,  $\theta$ . Since objects in most scenes move along the horizontal axes,  $X$  and  $Z$ , and with very small variation on their orientation, we only track the velocities along  $X$  and  $Y$  axes,  $v_x$  and  $v_y$  respectively. Hence, the state of each object  $O$  is  $\mathbf{x}^O = \{x, y, z, l, w, h, \theta, v_x, v_y\}^O$ . We apply an independent Kalman filter for each object with a constant velocity motion model for predictions.

To leverage the joint nature of the detections and the multi-modal (2D and 3D) sensor source, we use a **dual measurement update**. Each track has two measurement sources, the 2D bounding boxes, as well as the 3D bounding boxes that we assume to be independent, although this is not strictly the case. We combine a first PDA [33] Kalman filter update based on 3D measurements, with a second PDA Extended Kalman Filter (EKF) update, with the 2D measurements. The first linear measurement update serves as the primary component and carries most information, with



(a) JackRabbit, 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; 2D and 3D annotations have same IDs (indicated by similar box color)

the 2D measurement acting as a fine tuning measurement correction.

For those 2D detections without a corresponding 3D detection, we perform a PDA update of the tracks with the 2D measurement only.

### G. Track Management

*Creating and terminating tracks:* When a new object enters the scene, a new track is initiated only if it is outside the gate of all existing tracks. In that case, we create a temporary track (not part of the JRMOT output) and only after  $n_{init}$  number of consecutive matches, we promote it to full track. This process reduces noise and avoids false positives. Further, we terminate a track if there has been no matching detection for  $n_{term}$  consecutive frames, to account for objects leaving the scene.

*Updating tracks' appearance:* At each step, we update the appearance of the tracks with the latest RGB and pointcloud information to facilitate association in the next step. To do so, we need to associate each last detection to only one track. However, JPDA provides a full probability distribution of association between tracks and detections. Therefore, we perform a linear sum assignment on the JPDA output using the Hungarian algorithm [34] with  $p_{assn}$  as the minimum probability for a match to be considered. This process provides one-to-one associations that allow us to update the feature descriptor of each track with the assigned best detection. If for a track no detection is assigned, its features are not updated.

## IV. DATASET

As reviewed in Sec. II, datasets with the type of 3D data and annotations necessary to develop and train 3D MOT systems are scarce and focused on autonomous driving scenarios: there is a need for novel datasets with 3D annotations in social environments from the perspective of navigating robots. We present the **JRDB dataset**, a novel dataset focused on human social environments. Our dataset contains **64 minutes** of sensor data acquired from our mobile robot JackRabbit comprising **54 sequences** indoors and outdoors in a university campus environment. In this section,

we summarize the data collection and labeling process of the dataset.

### A. The JackRabbit Social Robot

The JRDB is a multimodal dataset collected with the sensors on-board of our mobile manipulator JackRabbit. JackRabbit is a custom-design robot platform tailored to navigate and interact in human environments. It is equipped with a state-of-the-art sensor suite including stereo RGB 360°cylindrical video streams (resulting from composing images from two rows of five aligned cameras each), 3D point clouds from two 16 lines LiDAR sensors, front and back single line LiDAR pointclouds, RGB-D and 360°spherical RGB images from the cameras on the head, audio, IMU and GPS sensing. Fig 3a depicts JackRabbit and its on-board sensors. Our goal is to investigate and develop novel solutions for perception and high-level social interactions between humans and robots through JackRabbit.

### B. Data Collection and Annotation

To generate JRDB, we collected data in 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. 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.

A crucial component in the development of social autonomous navigating agents is to perceive and understand the location and motion of humans surrounding the robot. Therefore, in this first round of annotation we focus on detecting and tracking humans. We include the following ground truth labels in JRDB: a) over 2.4 million 2D bounding boxes for human/pedestrian class in both the ten separate RGB images and the two composed cylindrical 360°images, b) over 1.8 million 3D oriented bounding boxes for human/pedestrian class in pointclouds from the two 16-lines LiDAR sensors, c) spatial ID association between corresponding 2D and 3D

bounding boxes (all 3D boxes have an associated 2D box but not vice versa), and d) temporal ID association with time consistent identities for all annotated pedestrians in both 2D and 3D. Fig 3b depicts examples of JRDB and the annotated ground truth labels on both an RGB 360° cylindrical image and a 3D LiDAR pointcloud, colored with the information from the RGB image. With this unique dataset, we hope to facilitate and enable novel research in social navigating autonomous agents. We will augment JRDB in the future with additional annotations related to social understanding in human environments such as 2D human skeleton posture and individual, group and social activity.

## V. EXPERIMENTAL EVALUATION

We adopt the standard Clear-MOT metrics [35] in our evaluation, including accuracy (MOTA), precision (MOTP), and number of ID switches (IDS) along with runtime, as we aim at developing an online real-time MOT system. However, Clear-MOT metrics were developed for 2D tracking, *e.g.* tracks are designated true of false positives based on IoU between estimated and ground truth 2D bounding box in the RGB images. We extend these definitions to 3D based on 3D IoU computed combining the Sutherland-Hodgman algorithm [36] and Gauss’s area formula to determine the volume of the intersection.

Our goal is to develop a real time online MOT system for navigating robots in human environments. Therefore, we evaluate JRMOT on our novel JRDB dataset (3D) and the well-established KITTI dataset [37]. The KITTI dataset contains 2D RGB images and 3D pointclouds, but the benchmark only reports 2D tracking results with 2D Clear-MOT metrics. Though JRMOT is a 3D MOT system, evaluating on KITTI allows us to compare to existing tracking methodologies. To be able to evaluate JRMOT on KITTI, we modify the system presented in Sec. 2 by changing the state in our filter architecture to  $\{x, y, w, h, v_x, v_y\}$ , where  $x, y, w, h$  parameterize the 2D bounding box and  $v_x, v_y$  give the velocity in the 2D image. The JRDB dataset and benchmark contains both RGB and pointcloud inputs, groundtruth 3D bounding boxes of pedestrians and an evaluation script for 3D tracking, which we use. We compare the results of JRMOT to a state-of-the-art baseline, AB3DMOT [12], on people tracking. We choose AB3DMOT as baseline due to it being a real time, online tracker, and the availability of the open-source code. At the time of submission, no other open-source online 3D MOT systems were available.

In order to provide comparable results, we aim to use identical detection inputs for all methods. For the KITTI dataset, we only use publicly available detections for the car and pedestrian challenges. For JRDB, we use the same set of Mask-RCNN detections for all methods. The publicly available detections for KITTI we chose were RRC [38] detections for cars and SubCNN [39] detections for pedestrians. For our evaluation of AB3DMOT on JRDB, which requires 3D detections as input, we used the 3D detections from F-PointNet which were generated as a by product from our tracking system.

	MOTA $\uparrow$	MOTP $\uparrow$	IDS $\downarrow$	Runtime $\downarrow$
MASS [40]	85.04%	85.53%	301	0.01s
mmMOT* [14]	84.77%	85.21%	284	0.01s
MOTBP* [41]	84.24%	<b>85.73%</b>	468	0.3s
IMMDP [42]	83.04%	82.74%	172	0.19s
JCSTD [43]	80.57%	81.81%	<b>61</b>	0.01s
Ours*	<b>85.70%</b>	85.48%	98	0.07s

TABLE I: Results on online KITTI car tracking benchmark. \* indicates that the method used the same public detections as our method

For experiments on KITTI we use the following parameter settings:  $p_{assn} = 0.65, n_{init} = 2, n_{term} = 2$ . For experiments on JRDB, we use:  $p_{assn} = 0.6, n_{init} = 3, n_{term} = 5$ .

## Results

**KITTI Dataset:** Table I shows our results in the car tracking challenge. We achieve state-of-the-art performance (highest MOTA) among all online published 2D MOT methods. Our MOTP is within 0.5% of the leader and we are second in terms of ID switches and beat all other top submissions by sizable margins.

Table II shows our results in the pedestrian tracking challenge. Amongst competing real-time methods (computation time less than 0.1s), our tracker ranks second.

Only one other method uses the same detections as our method. We remain within 1.5% MOTA, while running in only  $\frac{1}{15}^{th}$  the time. The performance gains in our method are a consequence of fusing and fully leveraging complementary information in 2D RGB and 3D pointcloud information. One point to note is the higher IDS of JRMOT. We found that optimizing MOTA decreases FN’s, at the expense of higher IDS. This hyperparameter optimization is only specific to KITTI pedestrian tracking as evidenced by Table I, where our method achieves the 2<sup>nd</sup> lowest IDS. Even though our method was developed for 3D MOT, JRMOT ranks among the state-of-the-art 2D MOT systems in KITTI benchmark, indicating the benefits of our proposed approach, and validating the effectiveness of the system.

	MOTA $\uparrow$	MOTP $\uparrow$	IDS $\downarrow$	Runtime $\downarrow$
CAT [44]	<b>52.35%</b>	71.57%	206	<i>Not Reported</i>
Be-Track [45]	51.29%	<b>72.71%</b>	118	0.02s
MDP* [42]	47.22%	70.36%	87	0.9s
JCSTD [43]	44.20%	72.09%	<b>53</b>	0.07s
RMOT [46]	43.77%	71.02%	153	0.02s
AB3DMOT [12]	36.36%	64.86%	142	0.0047s
Ours*	45.98%	72.63%	395	0.06s

TABLE II: Results on Online KITTI Pedestrian Tracking. \* indicates that the method used the same public detections

**JRDB Dataset:** JRMOT outperforms the baseline, AB3MOT, on the JRDB benchmark with **20.2%** MOTA at **25 fps** (compared to 19.3% MOTA of AB3MOT). These MOTA values indicate that the scenes in our dataset are extremely challenging and will guide new research in the field. Based on the 765,907 false negatives of our method on the test set, we infer that 3D detections are the limiting factor in our tracking system.

The benefits of the JRMOT approach to combine 2D and 3D information are clearer for tracks relatively further away from the sensor, where 3D pointcloud data is sparse, but 2D RGB is a rich source of information. To verify this, we analyze the results as a function of the distance between tracks and robot. Although the MOTA remains fairly similar across all distances, with our method outperforming the baseline, we make the following observations. First, we observe that our hypothesis that 2D data is useful to improve orientation of 3D bounding boxes and make fine adjustments to position is validated in Fig. 4. It can be seen that as the distance from the robot increases, the MOTP of AB3DMOT degrades considerably, whereas our method is consistent across all distance ranges. Further, our method has 30% fewer ID switches. This shows that our method is able to assign a consistent track ID to individual people, far better than AB3DMOT, across all distances.

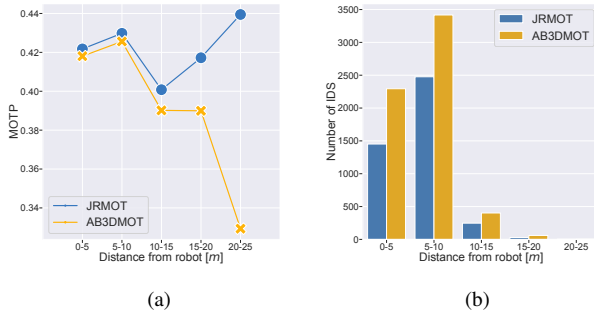


Fig. 4: Comparison of JRMOT and AB3DMOT as a function of distance. a) JRMOT obtains higher MOTP due to a more accurate estimation of the orientation of boxes and fine grained position information through all distances b) Our method also has less IDS (lower is better) than AB3DMOT, indicating more robust and stable tracking at all distances

Additionally, we analyse the contribution of the individual components in the overall performance of JRMOT on a set of ablation studies on the JRDB dataset. First, we conduct an experiment where we update the tracks only with 2D measurements. As expected, we observe that the 3D information is the most crucial for 3D tracking: without 3D data we obtain -20.1% MOTA on the train set. We also analyse the contribution of the 2D RGB appearance feature by using only 3D IoU as association metric. In this case, we see a small degradation in performance of 0.1% MOTA. This indicates that the **3D IoU is the most informative association metric**, but it is slightly improved in some cases with 2D appearance. Our last ablation is to verify that 2D inputs without corresponding 3D bounding boxes are indeed helpful measurements in our MOT system. We observed that if we do not use these only-2D updates, the MOTA remains constant at 42.9% but the MOTP drops 0.6%. The overall contribution of 2D is therefore 0.1% MOTA and 0.6% MOTP. However, this is misleading, due to the large number of objects that are close to the sensor, where 2D information is not expected to help much. **In the 15-20m range, the increase by using 2D information (both appearance and**

**measurements) is 1.3% MOTA.** This confirms our intuition that the 2D measurement can be used to make fine updates on the tracked orientation and location, especially further away from the robot.

## VI. REAL ROBOT EVALUATION

Finally, we evaluate the performance of JRMOT when running on-board of a real robot platform. We test on our social robot JackRabbit, which was used to collect the JRDB dataset. We chose not to run JRMOT at the same time as we collected all data for the JRDB dataset as it is not possible due to computing limitations (recording images and pointclouds considerably slows down tracking performance). Therefore, we cannot compute MOTA and MOTP on annotated data while running in real-time on the robot; we instead analyze the number (ID switches), as well as the number of lost tracks.

We test our solution in three different physical environments, with different lighting conditions (daylight and indoor lighting), with stationary and moving robot, and a different number, distance, and trajectory of moving people. Visualisations of the experimental setup can be seen in Fig. 5 We evaluate on a total of 110s of data with 14 unique identities across all scenes. On the on-board computer JRMOT runs between 9-11 fps and we measure only 4 ID switches and 1 lost track. These preliminary results, together with the extensive positive results on KITTI and JRDB, indicate that our tracker provides information to support autonomous navigation in human environments. We make our code publicly available as ROS packages for the community.



Fig. 5: We conduct on robot experiments in 3 different scenes, shown above, with a varying number (1 - 7) of people, at different distances (1 - 10m), with different types of human trajectories (moving and stationary), and with JackRabbit both moving and stationary. We aimed to conduct experiments in diverse, real-world conditions. The above images depict our experimental setup.

## VII. CONCLUSION AND FUTURE WORK

We presented JRMOT, a novel 3D MOT system that fuses the information contained in 2D RGB images and 3D pointclouds in an efficient manner to provide robust tracking performance even in adversarial and highly crowded environments, all while running in real time. As part of our project we release the JRDB dataset, a novel dataset for 2D and 3D MOT evaluation and development containing multimodal data acquired in human environments, including

inside university buildings and pedestrian areas on campus, as well as scenes where the robot navigates among humans. The dataset has been annotated with ground truth 2D bounding boxes and associated 3D cuboids of all persons in the scenes, which will help future research in 2D and 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 existing 3D MOT systems in our provided JRDB dataset. We also have preliminary on-robot experiments which validate the effectiveness of JRMOT in a real world setting. JRMOT serves as a competitive baseline to encourage further research within the paradigm of leveraging multi-modal sensor measurements to better perform 3D MOT.

## REFERENCES

- [1] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 941–951.
- [2] 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.
- [3] S. Sun, N. Akhtar, H. Song, A. S. Mian, and M. Shah, "Deep affinity network for multiple object tracking," *IEEE TPAMI*, 2019.
- [4] 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.
- [5] 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*, 2019.
- [6] H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, and I. R. A. Dick, "Joint probabilistic data association revisited," *ICCV*, 2015.
- [7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nusenes: A multimodal dataset for autonomous driving," in *IEEE/CVF CVPR*, 2020.
- [8] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.
- [9] 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 CVPR Workshops*, 2018, pp. 954–960.
- [10] 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.
- [11] 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 CVPR*.
- [12] X. Weng and K. Kitani, "A Baseline for 3D Multi-Object Tracking," *arXiv:1907.03961*, 2019.
- [13] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki, "Fantrack: 3d multi-object tracking with feature association network," in *2019 IEEE Intelligent Vehicles Symposium (IV)*.
- [14] J. Luiten, T. Fischer, and B. Leibe, "Track to reconstruct and reconstruct to track," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1803–1810, 2020.
- [15] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.
- [16] Z. Wang, L. Zheng, Y. Liu, and S. Wang, "Towards real-time multi-object tracking," *arXiv preprint arXiv:1909.12605*, 2019.
- [17] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, 2019.
- [18] L. Spinello, M. Luber, and K. O. Arras, "Tracking people in 3d using a bottom-up top-down detector," in *2011 IEEE ICRA*.
- [19] T. Linder and K. O. Arras, "People detection, tracking and visualization using ros on a mobile service robot," in *Robot Operating System (ROS)*. Springer, 2016, pp. 187–213.
- [20] C. Dondrup, N. Bellotto, F. Jovan, M. Hanheide *et al.*, "Real-time multisensor people tracking for human-robot spatial interaction," 2015.
- [21] 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, 2017.
- [22] D. Frossard and R. Urtasun, "End-to-end learning of multi-sensor 3d tracking by detection," in *2018 IEEE/ICRA (ICRA)*. IEEE, 2018, pp. 635–642.
- [23] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.
- [24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE ICCV*, 2017.
- [25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE CVPR*.
- [26] 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.
- [27] 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 CVPR Workshops*, 2018.
- [28] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustrum pointnets for 3d object detection from RGB-D data," *CoRR*, vol. abs/1711.08488, 2017. [Online]. Available: <http://arxiv.org/abs/1711.08488>
- [29] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE CVPR*.
- [30] 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.
- [31] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, "Joint probabilistic matching using m-best solutions," in *CVPR*, 2016.
- [32] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.
- [33] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Systems Magazine*, 2009.
- [34] H. W. Kuhn and B. Yaw, "The hungarian method for the assignment problem," *Naval Res. Logist. Quart.*, pp. 83–97, 1955.
- [35] 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.
- [36] I. E. Sutherland and G. W. Hodgman, "Reentrant polygon clipping," *Communications of the ACM*, vol. 17, no. 1, pp. 32–42, 1974.
- [37] 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.
- [38] 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>
- [39] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *2017 IEEE WACV (WACV)*, March 2017, pp. 924–933.
- [40] H. Karunasekera, H. Wang, and H. Zhang, "Multiple object tracking with attention to appearance, structure, motion and size," *IEEE*, 2019.
- [41] 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 ICRA*.
- [42] 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.
- [43] 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.
- [44] 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*.
- [45] P. W. Dimitrievski M, Veelaert P, "Behavioral pedestrian tracking using a camera and lidar sensors on a moving vehicle," *Sensors*, vol. 19, 2019.
- [46] J. H. Yoon, M. Yang, J. Lim, and K. Yoon, "Bayesian multi-object tracking using motion context from multiple objects," in *2015 IEEE WACV*, Jan 2015, pp. 33–40.