

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 Martín-Martín, Silvio Savarese

Abstract—A robotic agent navigating autonomously needs to perceive and track the motion of objects and other agents in its surroundings to be able to plan and execute robust and safe trajectories. To be useful for planning and execution, the motion should be perceived in 3D Cartesian space. However, most recent multi-object tracking (MOT) research have focused on tracking people and moving objects in 2D RGB video sequences. This is because searching in 3D is costly, there is a lack of annotated datasets with 3D labels of moving agents and a relative scarcity of data with 3D sensor modalities. 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 incorporates recent advancements in neural network based re-identification as well as 2D and 3D detectors and descriptors. We integrate them 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. This dataset, that we use to train and test our model, contains over 60 minutes of data including 360° cylindrical RGB video and 3D pointclouds, in social settings. 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. Moreover, our tests on our social robot JackRabbit indicate that the system is capable of tracking multiple pedestrians fast and reliably. The ROS code of our tracker is publicly available at <https://sites.google.com/view/jrmot>

I. INTRODUCTION

The goal of an autonomous agent such as a self-driving car or a mobile robot is to move between two locations in a safe and robust manner. To navigate safely, 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 the agent to predict the possible future trajectories of the other agents and to plan and execute motion strategies that avoid colliding with them.

To this end, the motion of the other agents needs to be perceived 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

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

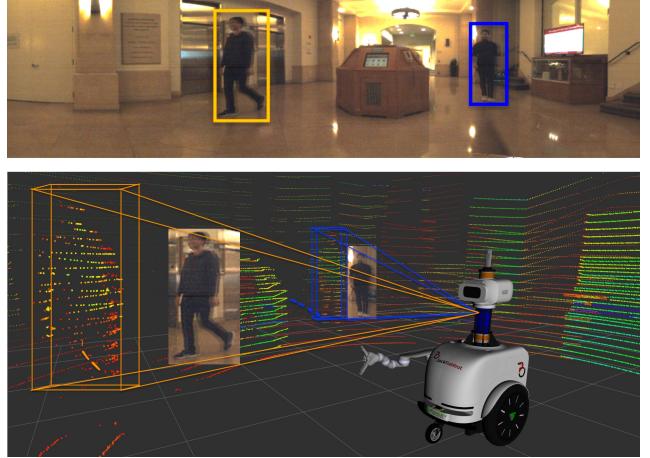


Fig. 1: Robots navigating in human environments need to detect and track humans and other moving targets using their sensor information; JRMOT integrates information from 2D RGB images (top), where appearance is more easily discernible, and 3D point clouds (bottom), where objects are well separated,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

for this is two fold. First, detecting and tracking objects in 3D is computationally expensive, and second, there is a lack of adequate large-scale curated datasets of 3D data with annotations of moving agents. This impedes the application of successful deep learning techniques - such as neural network architectures - to 3D tracking. In this paper we propose a novel method that leverages state-of-the-art neural network detectors and descriptors to perform real-time 3D MOT, and a novel 2D-3D annotated dataset which was used to train and test our method.

Integral to this work is the notion that 2D RGB sensor signals, and 3D LiDAR sensor signals carry complementary information. On one hand, 2D RGB images are **dense** containing on 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 depth information it allows to discern objects which might overlap in a 2D RGB image, but are well separated in 3D space. At the same time, the unordered structure of the

pointclouds do not allow for the use of efficient algorithmic architectures such as CNNs on them.

In this work we propose a method that leverages the information of each modality (appearance in RGB, geometry in point clouds) to address the shortcomings of the other.

We present JRMOT, 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 detection in point clouds, uses novel multi-modal descriptors, and improves upon well-established data-association and filtering techniques. We demonstrate the effectiveness of our framework with extensive experiments in the standard KITTI dataset [1] achieving state of the art performance in the 2D tracking benchmark among all published online real time methods, and additional robot experiments that demonstrate that JRMOT is capable of running and tracking in real time on an autonomous agent.

We also present the JRDB dataset, a novel dataset recorded with our robot platform JackRabbit. The dataset is captured from the perspective of a social autonomous agent, leading to a first of its kind dataset that includes **indoor and outdoor scenes**, with over **4.2 million annotated bounding boxes** in 2D and 3D. This dataset enabled us to leverage the complementarity of 2D RGB and 3D pointcloud data with JRMOT. We hope our new dataset will serve as a solid benchmark and help accelerate future research in 2D/3D multi-object tracking as well as detection, trajectory forecasting and social navigation.

To summarise, our contributions are:

- 1) We fuse 2D and 3D information using recent deep learning architectures and propose a novel real-time, online, 3D MOT system.
- 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. Our proposed 3D MOT system serves as a competitive baseline in our benchmark.
- 3) We demonstrate that our proposed 3D MOT system achieves state-of-the-art performance in the KITTI 2D tracking benchmark and show preliminary experimental evidence that our method can perform effectively in real time on JackRabbit, with only 4 ID switches and 1 missed track in over 100s of captured data.

II. RELATED WORK

In recent years, there have been many advances in the task of multi-object tracking (MOT). Many previous studies have mainly focused on 2D tracking. The number of 3D MOT works is comparatively little, and most systems share the same components with 2D MOT systems with some minor modifications. 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 sensors, real-time 3D MOT systems 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. [2], [3], [4], [5] exploit appearance to accurately re-identify objects from frame to frame. [6], [7] use motion and continuity cues. However, both approaches rely strictly on 2D information, which is susceptible to occlusion. Further, motion in the projected 2D image space can be highly irregular. To alleviate these problems, several approaches such as [8] 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 indoor environments.

3D MOT with 3D Data: With the advent of self driving cars, access to large scale datasets containing LiDAR data [9], [1], [10], [11] has reinvigorated interest in the use of 3D sensors. [12], [13] 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 - a task which can be challenging 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 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. [14] aggregates 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. [15] 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 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 2D MOT has a plethora of real-time MOT systems, 3D MOT systems are fewer in number, and relatively older. Of the recent works, [16] is a real time 3D MOT system which utilises exclusively 3D LiDAR data. This system does not utilise any RGB information. [17] utilises 2D laser scan data and an RGBD sensor to detect people using a random forest classifier, and then performs tracking by using a RGB based cost to perform association. Similarly [18] uses 3D detections from two independent 3D sensors. [19] also utilises RGBD input to obtain 3D detections as the only measurements for the tracking system. This loose coupling of 2D and

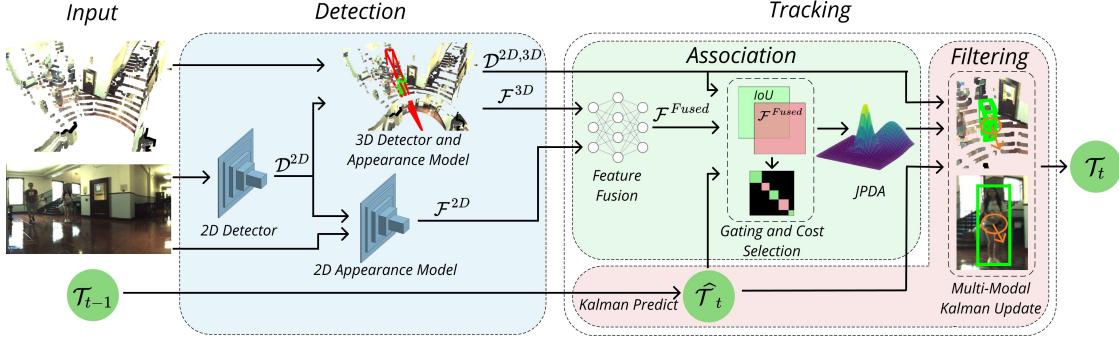


Fig. 2: JRMOT: Our proposed 3D MOT system is composed by detection and tracking (data association and filtering) components; \mathcal{T} , \mathcal{D} , \mathcal{F} refer to tracks, detections, features respectively with the superscript indicating the space; The system integrates information from 2D RGB images and 3D pointclouds into a single 3D multi-object recursive estimation tracker with real-time performance

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.

3D Datasets: 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 [1], Apolloscape [10], NuScenes [9] and Oxford’s Robotic Car [20] have widely driven research in the 3D community. Nonetheless, their targeted domain of application is autonomous driving; the data they provide is captured exclusively from sensor suites on top of cars and only depicts streets, roads and highways. [19] includes a dataset with all scenes exclusively in a $3m \times 5m$ room, with at most 5 people.

In this paper, we target a unique visual domain tailored to the perceptual tasks related to navigation in human environments, both indoors and outdoors, with some scenes containing over 20 people. 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 such as human detection and tracking, trajectory prediction, social navigation and planning, and others not currently supported by existing datasets.

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. Each component leverages the complementary nature of 2D and 3D sensor modalities, as shown in Sec. V. The input to our system is a 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, each uniquely identified over time by a track ID. Details of our method follow.

A. 2D Detection

Detection is at the core of any MOT system. 2D detectors are more robust and accurate than 3D detectors. Therefore, we use a pretrained Mask R-CNN [21] network, the state-of-the-art in image segmentation, but discard the part of the network that performs segmentation and use it as a detector only. We fine-tune the network on our JRDB dataset to adapt to the special data distribution¹. Note that the tracking framework is agnostic to the detector used. 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, $\mathcal{D}_t^{2D} = \{(u, v, w, h)_0, \dots, (u, v, w, h)_{N-1}\}_t$, 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. A modified YOLO [22] is used for real time applications.

B. 2D Appearance

When tracking it is common to have occluded objects wherein we are unaware of the position of an object for a brief period of time, only for it to reappear far away from the point of disappearance. 2D appearance can be a useful cue to re-identify the object as explored in [23] among many other 2D tracking works. To leverage the appearance information present in RGB images we utilise Aligned-ReID [24] when the objects of interest are people, and [25] for cars. The choice of these networks is due to their high performance without a large run-time. The 2D appearance features are given by $\mathcal{F}_t^{2D} = \{f_0^{2D}, \dots, f_{M-1}^{2D}\}_t$, for each of the N detections.

C. 3D Detection and Appearance

We utilise F-PointNet [26], 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. Based on projective geometry, the 3D object instance can be anywhere within the frustum starting at the RGB camera center and passing through the 2D bounding box. F-PointNet estimates a 3D bounding box for that object within the

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

frustum. 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 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 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, $\mathcal{F}_t^{3D} = \{f_0^{3D}, \dots, f_{M-1}^{3D}\}_t$. Since the feature corresponding to the penultimate layer of F-PointNet is used to regress the 3D bounding box, it must have some correlation to the 3D shape of an object. Hence, we use this feature as a 3D appearance (shape) descriptor.

Note that not every 2D detection has an associated 3D 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

Both 2D and 3D appearance can contain valuable information to associate detections and previous tracks and so 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 $[\cdot]$ denotes concatenation. We train this fusion network via metric learning based on the triplet loss and semi-hard negative mining as in Schroff *et al.* [27], resulting in a feature useful for 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 tracks at $t - 1$ to update the tracks' locations and appearances. To do so, we estimate the similarity 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.

The 3D similarity in location between the detected objects and the predicted track locations can also be a useful cue for re-identification. 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, giving a $K \times N$ IoU cost matrix.

To further simplify the association, 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 uses the Mahalanobis distance (M-distance). We assume that the detections whose M-distances to a predicted track are greater than a threshold (0.95 quantile from the χ^2 distribution) are improbable to correspond to that track, *i.e.* outside of the track's gate, and so we set the corresponding values in both IoU and appearance cost matrices to be infinite.

As the size of the cost matrix scales with the square of the number of objects in the scene, using the entire cost matrix can lead to extremely 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. On a per cluster basis, we then 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, which is a measure of how well this cost matrix can separate the detections within a track's gate. We then perform the matching on each cluster separately.

We choose to use Joint Probabilistic Data Association [28] to perform the matching, as it has been shown by [7] to be robust to clutter and reduces the occurrence of ID switches. To maintain the speed of our tracker, we employ the m-best solution approximation [29] for large clusters. For smaller clusters, we do complete enumeration and obtain the exact solution of JPDA.

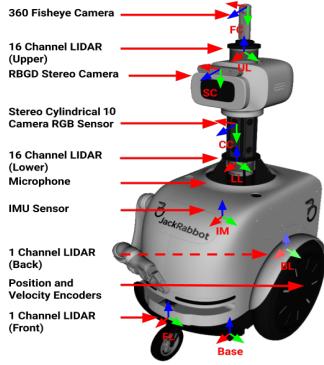
F. Filtering

2D and 3D detections are often noisy. We choose to use a Kalman filter [30] to obtain smooth 3D tracks because a) it is not computationally intensive b) it does not suffer from the curse of dimensionality, allowing for our method to be competitive and real time.

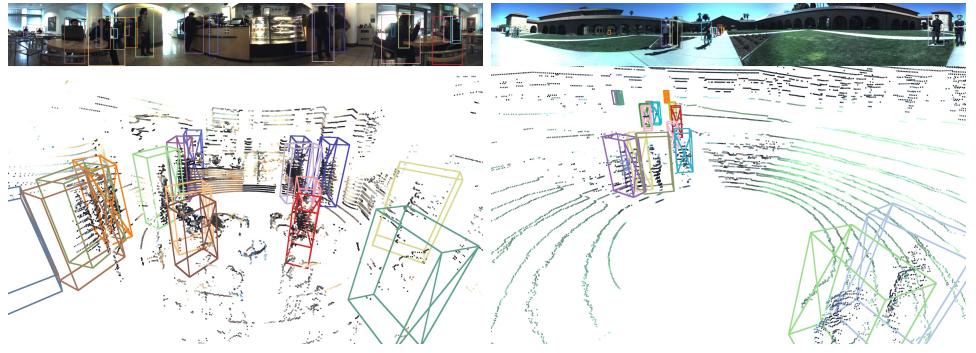
To perform 3D tracking, we require the coordinates of the object in the real world, x, y, z , its dimensions (approximated as a 3D bounding box) l, w, h and the orientation of the box about the vertical axis, θ . Since objects in scenes naturally move along the horizontal axes, X and Z , and it is also unlikely that an object would have a constant angular velocity about the vertical axis, we only model the velocities v_x and v_y along X and Z axes respectively. Hence, the state representation is $\vec{t} = \{x, y, z, l, h, w, \theta, v_x, v_z\}$.

The Kalman filtering is done independently for each track, assuming that each object moves independently. We use a constant velocity motion model to perform the prediction step, since sensor data is captured at 15fps.

To leverage the joint nature of the detections, we use a dual measurement update. Each track has two measurement sources, the 2D bounding boxes, as well as the 3D bounding boxes. We perform joint filtering assuming that the 2D and 3D measurement sources are independent, although this is not strictly the case. The 3D measurements are generated by F-PointNet, conditioned on the 2D measurements. However, based on the results in Sec. V, we believe that this assumption is justified. The update step involves consecutive



(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)

PDA [31] updates of the Kalman Filter first using 3D measurements, then, using an Extended Kalman Filter (EKF) 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.

G. Track Management

When performing the data association step, we require that each 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 [32]. 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 .

When a new object enters the scene, a new track is initiated only if it is a) outside the gate of all existing tracks and b) after n_{init} number of consecutive matches. This is done to avoid false positives. Further, we terminate a track if there has been no matching detection for n_{term} frames, to account for objects leaving the scene.

To deal with the case where an objects has a 2D detection, but not a corresponding 3D detection, we utilise a two step process. In the first step, all measurements with 2D-3D joint 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). 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 used to perform a PDA update of the tracks with the 2D measurement only.

IV. DATASET

As mentioned in Sec. II, datasets with 3D data that can be used to develop and train 3D MOT systems are scarce and are focused on autonomous driving scenarios. Yet, human-centric autonomous agent is another crucial domain where

high level 3D scene understanding is essential. Thus, 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 JackRabbit comprising **54 sequences** indoors and outdoors in a university campus environment. In this section, we describe the data collection and labeling process of the dataset.

A. JackRabbit

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

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. 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 consis-

tent trajectories (tracks) for all annotated persons in both 2D and 3D. In Fig 3b, we visualize the dataset alongside with the annotated ground truth labels on both 2D 360° cylindrical image and 3D LiDAR pointclouds. The dataset has over 2.4 million 2D human bounding boxes and 1.8 million 3D bounding boxes with associated 2D bounding boxes. With this unique domain of data, we are hoping to encourage the research of human-centric social autonomous agents. 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.

V. EXPERIMENTAL EVALUATION

For our evaluation, we will use the standard Clear-MOT metrics [33], including accuracy (MOTA), precision (MOTP), and number of ID switches (IDS) along with runtime. These metrics are defined for 2D tracking, *e.g.* a track is marked as a true positive in a frame if its IoU with respect to the ground truth 2D bounding box is greater than a threshold t_m . We extend the definitions of the metrics to 3D by using 3D IoU. 3D IoU is estimated using a combination of the Sutherland-Hodgman algorithm [34] and the shoelace formula (surveyor’s formula) to determine the volume of intersection. To evaluate our MOT system, we run experiments on our novel JRDB dataset and the well-established KITTI dataset [35].

The KITTI dataset contains 2D RGB images and 3D pointclouds, but the benchmark only reports 2D tracking results with 2D Clear-MOT metrics. While the goal of our method is 3D MOT, 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. Hence, we directly evaluate 3D tracking, using the 3D tracking evaluation. We compare the results of JRMOT to a state-of-the-art baseline, AB3DMOT [13], 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.

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 best performing publicly available detections for KITTI were RRC [36] detections for cars and SubCNN [37] 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.

Results

KITTI Dataset: Table I shows our results in the car tracking challenge. We achieve state-of-the-art performance

	MOTA \uparrow	MOTP \uparrow	IDS \downarrow	Runtime \downarrow
MASS [38]	85.04%	85.53%	301	0.01s
mmMOT* [15]	84.77%	85.21%	284	0.01s
MOTBP* [39]	84.24%	85.73%	468	0.3s
IMMDP [40]	83.04%	82.74%	172	0.19s
JCSTD [41]	80.57%	81.81%	61	0.01s
Ours*	85.70%	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

(highest MOTA) among all online published 2D MOT methods. Further, our MOTP is within 0.5% of the leader, indicating the small differences in performance between them. Additionally, 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. 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 [42]	52.35%	71.57%	206	<i>Not Reported</i>
Be-Track [43]	51.29%	72.71%	118	0.02s
MDP* [40]	47.22%	70.36%	87	0.9s
JCSTD [41]	44.20%	72.09%	53	0.07s
RMOT [44]	43.77%	71.02%	153	0.02s
AB3DMOT [13]	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

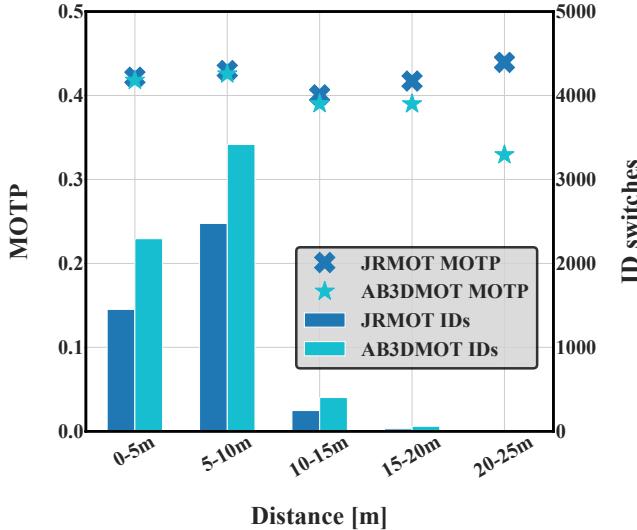


Fig. 4: Comparison of JRMOT and AB3DMOT including MOTP (lines; higher is better) and ID switches (bars; lower is better); our method obtains higher MOTP due to a more accurate estimation of the orientation of tracks and fine grained position information through all distances thanks to the combination of RGB and 3D information; our method also has less ID switches than AB3DMOT, indicating more robust and stable tracking at all distances

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.

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 different number, distance, and trajectory of moving people. Some 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 the information to support autonomous navigation in human environments. We make our code publicly available as ROS packages for the community.

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



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.

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.

REFERENCES

- [1] 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.
- [2] P. Bergmann, T. Meinhart, and L. Leal-Taixe, “Tracking without bells and whistles,” *arXiv preprint arXiv:1903.05625*, 2019.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, and I. R. A. Dick, “Joint probabilistic data association revisited,” *ICCV*, 2015.
- [8] 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.
- [9] 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.
- [10] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, “The apolloscapes dataset for autonomous driving,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 954–960.
- [11] 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.
- [12] 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.
- [13] X. Weng and K. Kitani, “A Baseline for 3D Multi-Object Tracking,” *arXiv:1907.03961*, 2019.
- [14] E. Baser, V. Balasubramanian, P. Bhattacharyya, and K. Czarnecki, “Fantrack: 3d multi-object tracking with feature association network,” *arXiv preprint arXiv:1905.02843*, 2019.
- [15] J. Luiten, T. Fischer, and B. Leibe, “Track to reconstruct and reconstruct to track,” 2019.
- [16] 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.
- [17] 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.
- [18] C. Dondrup, N. Bellotto, F. Jovan, M. Hanheide *et al.*, “Real-time multisensor people tracking for human-robot spatial interaction,” 2015.
- [19] M. Munaro and E. Menegatti, “Fast rgb-d people tracking for service robots,” *Autonomous Robots*, vol. 37, no. 3, pp. 227–242, 2014.
- [20] 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.
- [21] K. He, G. Gkioxari, P. Dollr, and R. Girshick, “Mask r-cnn,” 2017.
- [22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [23] 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>
- [24] 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.
- [25] 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.
- [26] 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>
- [27] 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.
- [28] 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.
- [29] S. H. Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid, “Joint probabilistic matching using m-best solutions,” in *CVPR*, 2016.
- [30] R. E. Kalman, “A new approach to linear filtering and prediction problems,” 1960.
- [31] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” *IEEE Control Systems Magazine*, 2009.
- [32] H. W. Kuhn and B. Yaw, “The hungarian method for the assignment problem,” *Naval Res. Logist. Quart.*, pp. 83–97, 1955.
- [33] 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.
- [34] I. E. Sutherland and G. W. Hodgman, “Reentrant polygon clipping,” *Communications of the ACM*, vol. 17, no. 1, pp. 32–42, 1974.
- [35] 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.
- [36] 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>
- [37] 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.
- [38] H. Karunasekera, H. Wang, and H. Zhang, “Multiple object tracking with attention to appearance, structure, motion and size,” *IEEE*, 2019.
- [39] 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.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] P. W. Dimitrievski M, Veelaert P, “Behavioral pedestrian tracking using a camera and lidar sensors on a moving vehicle,” *Sensors*, vol. 19, 2019.
- [44] 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.