2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

2D vs. 3D LiDAR-based Person Detection on Mobile Robots

Dan Jia Alexander Hermans Bastian Leibe

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981519

***Abstract*— Person detection is a crucial task for mobile robots navigating in human-populated environments. LiDAR sensors are promising for this task, thanks to their accurate depth measurements and large field of view. Two types of LiDAR sensors exist: the 2D LiDAR sensors, which scan a single plane, and the 3D LiDAR sensors, which scan multiple planes, thus forming a volume. How do they compare for the task of person detection? To answer this, we conduct a series of experiments, using the public, large-scale JackRabbot dataset and the state-of-the-art 2D and 3D LiDAR-based person detectors (DR-SPAAM and CenterPoint respectively). Our experiments include multiple aspects, ranging from the basic performance and speed comparison, to more detailed analysis on localization accuracy and robustness against distance and scene clutter. The insights from these experiments highlight the strengths and weaknesses of 2D and 3D LiDAR sensors as sources for person detection, and are especially valuable for designing mobile robots that will operate in close proximity to surrounding humans (e.g. service or social robot).**

I. Introduction

Person detection is an important task in many robotic applications, including safe autonomous navigation in human-populated environments or during human-robot interactions. LiDAR sensors are well-suited for person detection, thanks to their accurate depth measurements, long sensing range, and large field of view. There are many successful object detection methods using 3D LiDAR sensors in driving scenarios [53], [46], [48], [23], [40], [38], [47], [49], where pedestrian is typically one of the classes to be detected. Meanwhile, in mobile robot scenarios persons have successfully been detected using range scans from 2D LiDAR sensors [2], [32], [24], [5], [20], [21].

In this work, we present a comparative study on 2D and 3D LiDAR-based person detection. Properly understanding the performance differences (detection accuracy and runtime) between these two sensor types plays an important role for well-informed robot design [28]. We focus specifically on scenarios that are encountered by *e.g.* social robots or service robots, which are becoming increasingly relevant in recent years.[1] Using the state-of-the-art CenterPoint [49] and DR-SPAAM [20] as representatives for methods based on 3D and 2D LiDAR sensors, we conduct a series of experiments on the large-scale, publicly available JackRabbot dataset [31], comparing the performance differences between these two sensor types.

*All authors are with the Visual Computing Institute, RWTH Aachen {jia, hermans, leibe}@vision.rwth-aachen.de
An earlier version with additional experiments can be found at https://arxiv.org/abs/2106.11239v1.

[1]These scenarios differ significantly from driving scenarios in terms of density and proximity of surrounding persons, sensor height, encountered objects, *etc*.

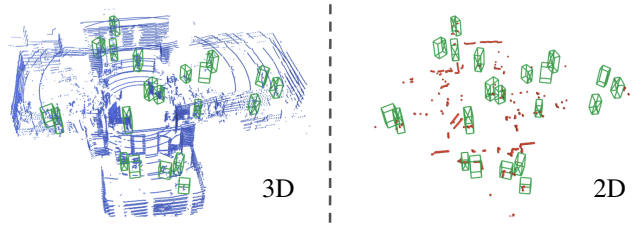


Fig. 1: Scans from a 3D and a 2D LiDAR sensor with person bounding boxes annotations. The vast difference between the scans are apparent, but how do these two sensors compare, when it comes to person detection? Understanding how these sensors affect this task is crucial for designing robots that are intended to operate around humans.

Our experiments reveal that the 3D LiDAR-based CenterPoint provides superior detection accuracy compared to the 2D LiDAR-based DR-SPAAM, but when only considering persons visible in the sensor data, these two methods perform on par. In applications where detecting nearby persons (which are most likely to be visible) is the primary requirement, 2D LiDAR sensors, which are often readily available on many mobile robots for mapping and safety purposes, can be a sufficient detection source. Additionally, 2D LiDAR-based DR-SPAAM has the advantage of higher inference speed, which is beneficial for mobile robots with limited onboard compute or power. General purpose person detection, however, is better accomplished with 3D LiDAR sensors, which scan a 3D sensing volume (not limited to a single scan plane as 2D LiDAR sensor) and are thus more robust towards occlusion. Our experiments also show that both methods can deliver well-localized detections and have similarly robustness against dense crowds. A closer examination on positive and negative predictions show that a non-negligible amount of persons are only detected by one of the detectors, implying that one should opt for an ensemble-based approach when maximum reliability is the primary design objective.

II. Related Work

Person detectors based on 3D LiDAR data estimate 3D bounding boxes of persons in a scene. Most such detectors have been designed for multi-class detection relying on autonomous driving datasets [16], [6], [42], [22], in which the pedestrian class is one of the classes to be detected. Existing deep-learning-based methods can largely be grouped into two categories. Single-stage detectors [14], [53], [46], [23], [52], [29], [44], [9], [47], [50], [49], [51], [18] use backbone

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 18:18:24 UTC from IEEE Xplore. Restrictions apply.

networks, either based on points [34], [43] or voxels [53], [23], [17], [11] to process the scene and generate bounding boxes directly from the extracted features. Two-stage detectors [39], [33], [48], [40], [10], [38] additionally introduce a bounding box refinement stage, which pools features within the box proposals and generates refined predictions. There exist, in addition, person detection methods that use hand-crafted features [45], which typically do not perform as well as the end-to-end learned detectors.

The KITTI dataset [16] is a forerunner in providing a standardized benchmark for testing LiDAR-based 3D object detectors, but it is now succeeded by newer and larger datasets, including the nuScenes dataset [6] and the Waymo Open Dataset [42]. However, these driving datasets differ from the mobile robot scenarios, where the robot moves in close proximity to humans (*e.g*. the JackRabbot dataset [31]).

In this work we use CenterPoint [49] as the representative for 3D LiDAR-based detectors. At the time of writing, it ranks second on the nuScenes benchmark [6] (the source-code for the higher ranking method [18] is not available), and thus represents the current available state-of-the-art in the area of 3D detection.

There are, in addition, 3D methods developed primarily on datasets collected by scanning static indoor scenes with RGB-D cameras [41], [8], [12], [1]. While in theory some of these approaches could be adapted for detecting persons in LiDAR data, we do not experiment with them in this work.

Person detectors based on 2D LiDAR data focus on estimating centroids of persons in the scene, parametrized as x, y coordinates on the LiDAR scan plane. It has long been a relevant task in the robotics community, given it is crucial in order for a mobile agent to autonomously navigate in human-populated environments. Early approaches [15], [36], [37] detect blobs with manually engineered heuristics, and track these blobs in sequential scans, leveraging motion as a clue for detection. Later works [2], [32], [24] improved the detection stage, replacing the heuristics with learned classifiers (*e.g.* AdaBoost) on hand-crafted features, and still rely on motion-based filtering to obtain high quality detections [24], [32].

Most recent developments [4], [5], [20], [21] resorted to deep-learning techniques by applying 1D CNNs to range data, and no longer require motion-based filtering. The first of these approaches is the DROW detector [4]. It was originally designed to detect walking aids, and was later extended to also detect persons [5]. The current state-of-the-art method is the DR-SPAAM detector [20], [21], which augments the DROW detector with a temporal aggregation paradigm, incorporating information over sequential LiDAR scans in order to improve the detection performance.

Most similar to our work, is the recent study of Linder *et al*. [28] who compare a range of different 2D, 3D, and RGB-D-based person detectors. While they compare a large number of older and more recent detectors, we limit our experiments to the two current state-of-the-art (and open-source) approaches and perform our experiments on the larger, public, and more diverse JackRabbot dataset [31]. Their findings with respect to 2D and 3D LiDAR-based person detection are in line with ours, however, we delve deeper into the differences, yielding interesting additional insights in how these two types of sensors compare for the task of person detection.

III. Person Detectors

The experiments in this paper are conducted with CenterPoint [49] and DR-SPAAM [20], state-of-the-art detectors based on 3D and 2D LiDAR sensors respectively. While we here very briefly recap the main ideas of the two detectors, we refer the interested reader to the main publications for a detailed overview.

CenterPoint [49] takes a voxelized 3D point cloud as input and uses either a VoxelNet [53] or PointPillars [23] to extract a 2D feature map on the bird's-eye-view (BEV) plane. From the extracted features, a center head is used to produce heatmaps, corresponding to x, y locations of bounding box centers on the BEV plane. This center head is supervised with 2D Gaussians produced by projecting bounding box centers onto the BEV plane, together with the focal loss [26]. In addition, regression heads are used to obtain x, y center refinement, center elevation, box dimensions, and orientation (encoded with sine and cosine values). These regression heads are supervised at the ground truth centers with an $L1$ loss. In this work, we use the CenterPoint with a VoxelNet backbone for our experiments.

DR-SPAAM [20] takes as input the range component of a 2D LiDAR scan encoded using polar coordinates. A preprocessing step is used to extract points within small angular windows (called *cutouts*), and each window is normalized and passed into a 1D CNN. The network has two branches: a classification branch, which classifies if the window center is in the proximity of a person, and a regression branch, which regresses an offset from the window center to the x, y center location of the person. The classification branch is supervised with a binary cross-entropy loss, whereas the regression branch is supervised only for positive windows, using an $L2$ loss. Finally, the predictions from all windows are post-processed with a distance-based non-maximum-suppression to obtain the final detections.

IV. Experimental Setup

A. Datasets

Our main experiments are conducted with the JackRabbot Dataset and Benchmark (JRDB) [31], however, we additionally use the nuScenes dataset [6] for some pretraining experiments.

JRDB [31] contains 54 sequences collected with a mobile robot (the *JackRabbot*) moving in both indoor and outdoor environments on university campuses. These sequences are split into 27 sequences for training and validation, and 27 for testing. The robot is equipped with two 3D LiDAR sensors (Velodyne 16 Puck LITE), on the upper and lower part of the robot respectively, each producing scans with approximately 18,000 points. Persons in the environment are annotated with 3D bounding boxes, with a total of roughly 1.8M annotations

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 18:18:24 UTC from IEEE Xplore. Restrictions apply.

across the dataset. At the moment of writing, JRDB is the only large-scale dataset that focuses on mobile robot scenarios, while also featuring LiDAR point cloud with 3D person annotations.

In addition, the JackRabbot is equipped with two 2D LiDAR sensors (SICK LMS500-20000), front and rear facing respectively. They are mounted at the height of the lower legs, and their scans are merged to a single 360° scan, having 1091 points. Jia *et al*. [21] used these scans to evaluate 2D LiDAR-based person detectors by synchronizing and aligning them to the 3D annotations. In this work, we use 2D LiDAR scans from JRDB, following the same data preparation procedure. All our reported numbers are based on the JRDB validation set obtained from the standard train-validation split.

The nuScenes dataset [6] contains 1,000 short sequences (approximately 20 seconds each) collected from driving vehicles. These sequences are split into a train, validation, and test set, having 700, 150, 150 sequences respectively. The dataset is captured using a 32 beam LiDAR, producing scans with approximately 30,000 points at 20 Hz. Compared to JRDB, these scans cover a larger area and have significantly sparser point clouds. Every tenth frame (0.5 seconds apart) is annotated with 3D bounding boxes, with a total of 10 classes, one of which being pedestrian. A common practice is to combine the unlabeled scans to obtain a denser point cloud [6], [54], [47], [49] during training and inference.

B. Training Setup

For CenterPoint, we mostly use the same hyperparameters from [49]. We follow the same training procedures and data augmentation scheme, with the AdamW optimizer [30], but train the network for 40 epochs with batch size 32, max learning rate $1e-3$, weight decay 0.01, and momentum 0.85 to 0.95. For the voxelization of the input point cloud, we found that the default settings for outdoor scenarios are not optimal for JRDB. We experiment with two settings: one where we uses the parameters for nuScenes: (0.1m, 0.1m, 0.2m) voxel grids, and limited detection range of [-51.2m, 51.2m] for the x and y axes, and [-5m, 3m] for the z axis. Additionally, we use a more fine-grained voxelization using a (0.05m, 0.05m, 0.2m) grid and a limited detection range of [-25.6m, 25.6m] for the x and y axes, and [-2m, 8m] for the z axis. This reduction in voxel size and detection range is motivated by the fact that scenes in JRDB are typically of a smaller scale compared to those in nuScenes.

When fine-tuning a network pretrained on nuScenes, we reduce the training duration to 10 epochs. This resulted in the same performance as a complete 40 epoch training schedule on top of the pretrained network. Since nuScenes provides additional time increments and reflectance features for every point, we cannot directly fine-tune checkpoints provided by the authors. Instead we retrain the network on nuScenes, using only x, y, z coordinate of the points as input features, with the default training hyperparameters from [49], and fine-tune this network.

To train DR-SPAAM, we follow the procedures from [21], and train the network for 20 epochs with a batch size of 6, using the same pre and postprocessing hyperparameters. The original DR-SPAAM only predicts x, y locations of the person, providing no information related to bounding boxes. In order to compare with 3D-LiDAR-based methods, we experiment with modifying the regression branch, additionally predicting the bounding box width, length, and orientation. The width and length are parameterized as $log(w/\bar{w})$ and $log(l/\bar{l})$, where $\bar{w} = 0.5$ and $\bar{l} = 0.9$, which represent the average box width and length in the JRDB training set. The orientation is parameterized by its sine and cosine as in [49]. Both the size and orientation regression are supervised with an $L2$ loss, with a weighting factor of 0.2 for the orientation. In addition, assuming persons do not vary significantly in height, we experiment with generating 3D bounding boxes, using the predicted BEV boxes and the average height from the training set. We did not experiment with regressing box height from the 2D LiDAR data.

C. Metrics

Following the standard for object detection, we use the Average Precision (AP) as the main evaluation metric. We use three variants, AP_{box}, AP_{BEV}, and $\text{AP}_{\text{centroid}}$, each having different criteria for assigning detections to ground truth boxes. For AP_{box}, a detection can be assigned to a ground truth, if their 3D IoU is above 0.3 (following the JRDB convention [31]). AP_{BEV} relaxes the requirement to a 2D IoU criterion on the bird's-eye-view boxes, discarding the requirement related to box height and elevation. $\text{AP}_{\text{centroid}}$ focuses on the center localization only, assigning a detection to a ground truth if their x, y location difference is smaller than 0.5m [24], [5], [20], dropping the requirements on box size and orientation. In all these three variants, a ground truth can only be matched with one detection.

Since ground truth boxes have different visibility to 3D and 2D LiDAR sensors, we compute all three APs under two settings. In the first setting, we evaluate detections against ground truth boxes that contain at least 10 points from the 3D LiDAR. This corresponds to the default evaluation procedure of JRDB and we thus refer to it as the *default* evaluation. In the second setting, which we term *2D-visible*, we evaluate detections against ground truth boxes which have at least 5 points from 2D LiDAR within a 0.5m radius from the box x, y centroid. These two evaluation settings enable us to examine the detector performance from both types of sensors, independent of factors that cause persons to be completely invisible to the sensor (thus being impossible to detect).

V. Results

A. 2D vs. 3D LiDAR-based Detection Performance

The performance of CenterPoint and DR-SPAAM are compared in Table I. Under the default evaluation setting, DR-SPAAM trails CenterPoint trained on JRDB (with fine-grained voxelization) by a significant 31.9% AP_{box} and 26.3% AP_{BEV} (the gap on AP_{box} is greater since DR-SPAAM

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 18:18:24 UTC from IEEE Xplore. Restrictions apply.

		AP_{box}		AP_{BEV}		$AP_{centroid}$	
		Default	2D-visible	Default	2D-visible	Default	2D-visible
CenterPoint	trained on nuScenes	26.3	26.0	30.9	28.5	35.3	32.1
	+ JRDB fine-tuning	58.2	64.2	61.2	67.1	69.5	75.0
	trained on JRDB	60.0	68.2	62.6	69.9	67.9	75.1
	trained on JRDB (fine-grained voxelization)	66.0	75.0	67.1	76.5	70.1	78.1
	+ nuScenes pretraining	70.0	78.6	71.4	80.7	74.9	82.7
DR-SPAAM	BEV	34.1*	58.2*	40.8	67.3	46.6	74.8
	Centroid only	–	–	–	–	47.6	77.2

TABLE I: Performance of different CenterPoint and DR-SPAAM variants on the JRDB validation set. *:3D Bounding boxes are obtained using predicted BEV boxes and the average box height from the training set.

can not estimate the bounding box height). However, when evaluated against ground truth that is visible to 2D LiDAR sensors, this gap shrinks to 9.2% AP_{BEV}. For the $AP_{centroid}$, DR-SPAAM and CenterPoint differ by 3.3%. A DR-SPAAM that regresses centroid only (as it originally is in [20]) has a further improved performance, leaving a small performance gap of 0.9% $AP_{centroid}$ to CenterPoint trained only with JRDB.

These numbers show that, when only the centroid locations of visible persons are concerned, detectors using 2D or 3D LiDAR sensors have a similar performance, despite the vast information gap between the detector input (a simple planar scan of the legs *vs*. a full-body surface scan). Thus, for tasks like person avoidance or following, a 2D LiDAR sensor is sufficient. General purpose detection, however, is better carried out with 3D LiDAR-based detectors, since it is easy for persons to be fully occluded, and thus impossible to detect, from the scan plane of 2D LiDAR sensors. When designing robots, the choice between 2D and 3D LiDAR sensors should be made with task requirements in view.

The first row of Table I furthermore shows that a quite big domain gap exists between autonomous driving and mobile robot scenarios. There are many pre-trained 3D LiDAR-based detectors for driving scenarios, but it is unlikely that they will perform well on robotic tasks involving close contact with persons. Fortunately, this domain gap can be bridged by fine-tuning and adapting suited voxel sizes. These numbers highlight the importance of training on data similar to that will be encountered during deployment. For 2D LiDAR sensors, estimates about this domain gap are not directly possible, since they are rarely used in outdoor driving scenarios.

B. Detailed Detector Comparisons

The remainder of this section focuses on comparison of CenterPoint and DR-SPAAM in specific aspects. The best variants of each detector (CenterPoint pretrained on nuScenes, fine-tuned on JRDB with fine-grained voxelization, and centroid-only DR-SPAAM) are used for these comparisons with $AP_{centroid}$ as the evaluation metric.

1) Localization Accuracy: The default ground truth association radius of 0.5m used by $AP_{centroid}$ does not capture how well localized the different detections are. To further evaluate the localization accuracy, we run the evaluation with a varying association radius and plot the development of the performance in Fig. 2. Note that for this evaluation the gradient of a curve is interesting and not so much the absolute performance.

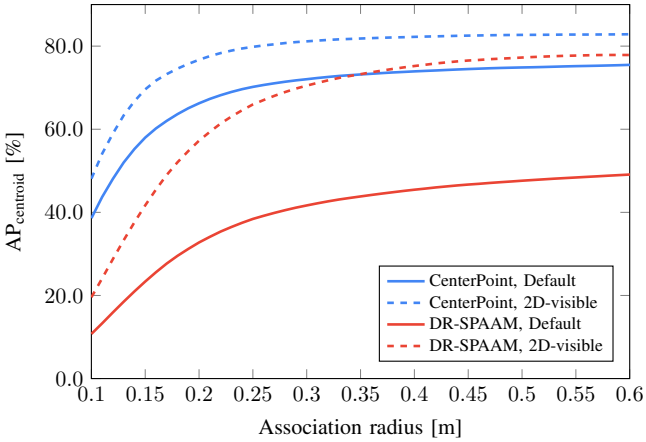


Fig. 2: $AP_{centroid}$ evaluated at different ground truth association radii. The CenterPoint curves are slightly steeper at the start, suggesting a better localization of the detections.

After a quick increase, the performances slowly saturate, suggesting that most detections are made with a reasonable accuracy. The main difference between CenterPoint and DR-SPAAM is the slope of the curves for small association radii. Here the performance for CenterPoint increases faster, suggesting a slightly better localization of its detections. Furthermore, the CenterPoint curves are almost saturated after an association radius of $\sim$0.3m, indicating an upper bound for its localization inaccuracy. Here the DR-SPAAM curves are still slightly increasing, meaning that some of the detections are less well localized. Note that small inaccuracies exist in the annotations. Thus, detection scores obtained with an overly small association radius (*e.g*. lower than 0.1m) do not provide meaningful information.

2) Distance-based Evaluation: The density of LiDAR data varies significantly with the distance to the sensor, which can affect the performance of detectors. Because of that we run an additional sliding-window evaluation, where we only consider ground truth annotations within a 2m window at varying distances from the sensor. For a meaningful evaluation we also constrain the detections to be in or near

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 18:18:24 UTC from IEEE Xplore. Restrictions apply.

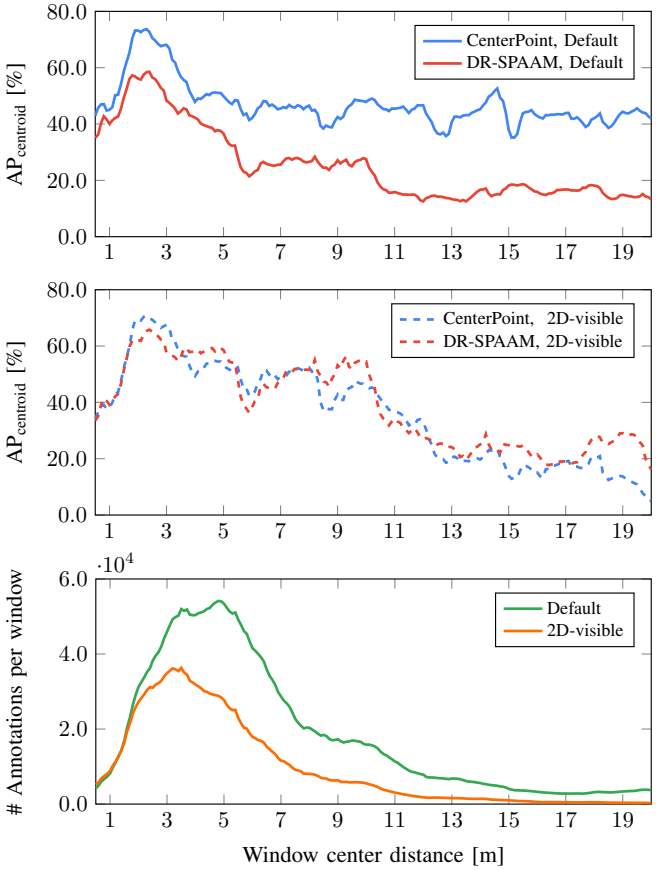


Fig. 3: $AP_{centroid}$ evaluated within a 2m window at different distances from the sensor. The top and middle plot show how the performance develops across the distance for the default and 2D-visible evaluation respectively. The bottom plot shows the number of annotations within the windows for both evaluations.

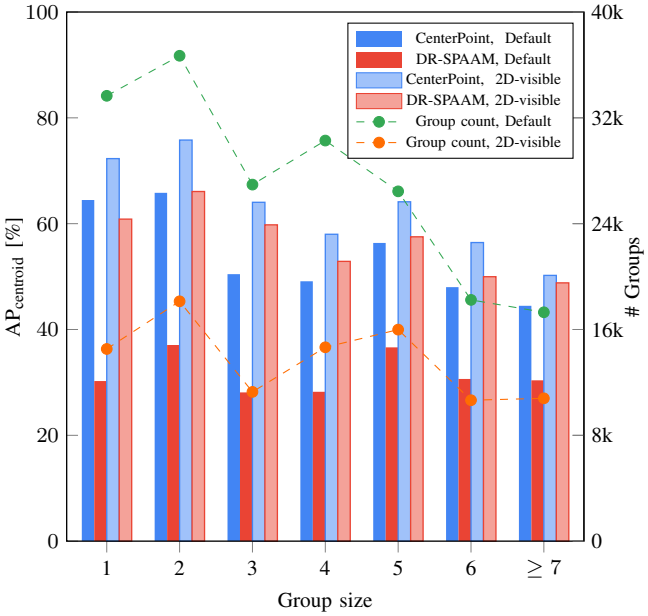


Fig. 4: Evaluation split up by how crowded it is around annotations. While the performance for larger group sizes decreases a little, the effect is less pronounced than might be expected. We additionally plot how many groups of a specific size are present in the data, showing that JRDB is indeed fairly crowded.

the evaluation window, as to not report unnecessary false positives. We do this by extending the window by 0.5m (the association radius) to the front and back and only consider detections within this larger window. This will still introduce additional false positives but also increase the true positive count, effectively being a trade-off between precision and recall. For this reason the overall performance is lower than reported in Table I, but here the relative performance between the two detectors is the interesting part.

The top and middle plots in Fig. 3 shows how the detector performances behave for this experiment. In all evaluations the performance is best within the first few meters and drops at higher distances. For the default evaluation, CenterPoint performs significantly more stable across the whole range, whereas the DR-SPAAM performance decreases more at higher ranges. This means that CenterPoint is better equipped to deal with the large variation of point densities, potentially due to the fact that objects are still visible to multiple LiDAR beams, whereas the 2D LiDAR only uses a single beam. For the 2D-visible evaluation, the performance of both detectors is surprisingly similar across the whole range, suggesting that for visible persons, both detectors have similar range-robustness.

The bottom plot in Fig. 3 shows how many ground truth annotations are present within the windows for both types of evaluations. Here it becomes apparent that, from 2m and onward, a significant number of the persons are invisible to 2D LiDAR scans, and thus impossible to be detected. The largest chunk of annotations is found within the first 12m. After that only few annotations remain, for both the default and 2D-visible evaluation, and the evaluation becomes less meaningful. This highlights another difference between JRDB and autonomous driving datasets, where persons are regularly perceived at much greater distances.

3) The Influence of Crowds: People are often perceived while interacting amongst each other, typically resulting in them being quite close together. This results in many occlusions and could potentially lead to a more difficult task of detecting the separate persons properly. To investigate how the detection performance changes with the object density of a scene, we look at a radius of 1.5m around every ground truth annotation and count how many other annotations can be found. We categorize all annotations by this "group size" and perform a separate evaluation for all the different sizes. Fig. 4 shows the result of this experiment. Even though the overall detection performance seems to be reducing with larger group sizes, the detectors do not completely fail for larger group sizes and the effect seems to be very similar for both detectors. What Fig. 4 also shows, is that a significant amount of people in the JRDB dataset are observed in the

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 18:18:24 UTC from IEEE Xplore. Restrictions apply.

	Desktop (TITAN RTX)	Laptop (RTX 2080)	Jetson AGX Xavier
CenterPoint	32.4	19.8	6.0
DR-SPAAM	59.1	37.1	8.9

TABLE II: Detector inference speed (frames per second) on three different platforms.

Subsampling factor	$AP_{centroid}$ (Default)	$AP_{centroid}$ (2D-visible)	FPS
1	47.6	77.3	8.9
2	46.5	76.5	18.2 ($\times$2.0)
3	45.2	75.9	26.6 ($\times$3.0)
4	43.7	74.9	32.5 ($\times$3.7)
5	42.0	73.3	36.4 ($\times$4.1)

TABLE III: DR-SPAAM performance and inference speed (frames per second) on Jetson AGX Xavier with different spatial subsampling.

vicinity of at least one other person making it significantly more realistic than some older datasets where one would often only observe a single person in an empty scene.

4) Runtime: The runtime is a typical practical constraint when deploying detectors. Especially in robotic applications, computational resources are often limited and we cannot rely on powerful desktop GPUs. Table II shows the runtime of the evaluated detectors on three different platforms: a powerful desktop machine, a laptop equipped with a decent GPU, and the lower-powered Jetson AGX Xavier. Preprocessing (*e.g.* computing *cutouts*, or voxelization) and postprocessing (*e.g.* non-maximum-suppression) steps are included in the measurement, and no batching is used for inference. In other words, these numbers reflect the end-to-end runtime when the detector is deployed. Both DR-SPAAM and CenterPoint achieve real-time performance on strong desktop or laptop GPUs, but not on embedded GPUs. In general, DR-SPAAM is roughly twice as fast as CenterPoint, with the margin being smaller on the Jetson AGX.[2]

DR-SPAAM can, thanks to its *cutout*-based design, obtain higher runtime by subsampling the scan, without significantly lowering the detection performance [20]. Table III shows the performance and runtime of DR-SPAAM with different subsampling factors on the Jetson AGX. With 3 times subsampling, DR-SPAAM can run at 26.6 FPS, while losing only 1.4% AP. In applications where the onboard computation is limited, DR-SPAAM presents a more favorable trade-off between performance and runtime, compared to the more accurate, yet slower CenterPoint.

C. Qualitative Results

Fig. 5 shows several qualitative results of both detectors. These results are obtained by using a detection threshold resulting in an equal error rate (EER), meaning the precision is equal to the recall for that threshold. We show cases where both detectors, one of the two, or neither detected a person. We specifically focus on the true positives and false negatives as these cases should contain a person. The predicted centroids, either coming from CenterPoint or DR-SPAAM, are well-localized to the actual person. The error in orientation estimation sometimes leads to misaligned boxes for CenterPoint, but the overlap with ground truth boxes is often sufficiently high.

Even though the numbers so far suggested that CenterPoint typically outperforms DR-SPAAM, we can here see that there are also cases where only one of the two detectors is able to detect a person. While Fig. 5 shows some cases where it clearly makes sense that a detector fails, *e.g.* due to missing points or partial occlusions. Most of the cases where only one of the detectors fails clearly show a person though and it is unclear why it was not detected. Interestingly, many ground truth bounding boxes missed by both detectors indeed do not look like they contain a person, potentially suggesting annotation errors.

To show these are not only a small set of cherry picked cases, Table IV shows exactly how often these four different cases happen. While CenterPoint is more often the only detector to detect a person, there is a small but significant number of people only detected by DR-SPAAM, too. This suggests that an ensemble of both methods could be an interesting approach when both sensors are available.

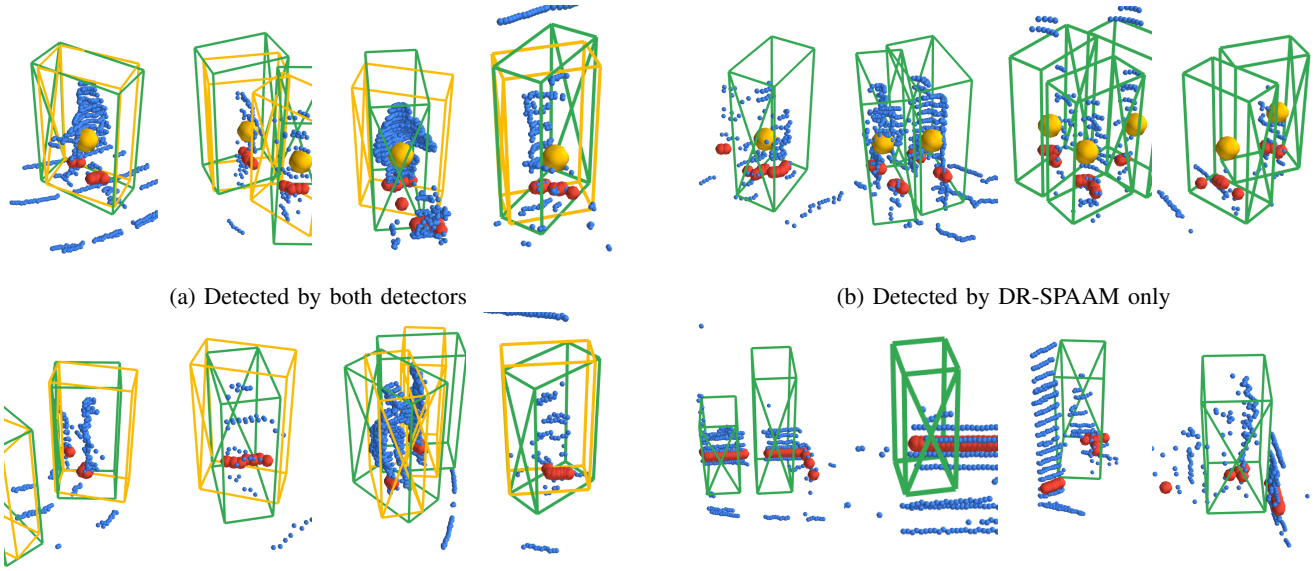
VI. Discussion

We performed all experiments with two state-of-the-art LiDAR-based detectors, while we could have tried to adapt CenterPoint to run on 2D LiDAR data, allowing for a more direct comparison. However, this would require a significant amount of tuning to make sure CenterPoint yields a representative performance on 2D LiDAR and our goal is not to develop a new 2D LiDAR-based detector. Instead, we here rely on an existing and well-tuned state-of-the-art method. With our detector evaluations, we have set a lower-bound for the person detection performance on JRDB using 2D and 3D LiDAR sensors, which will be further improved by future developments.

Apart from LiDAR-based person detection, one could consider other sensor modalities. In particular, image-based person detectors have a long history. While the person class is now typically seen as one of many classes by most deep-learning-based detectors [35], [7], a whole range of person-specific detectors existed before [13], [3]. Furthermore, RGB-D based person detectors are frequently used in robotics, which have the additional advantage that a person's position can be estimated [19], [25], [27]. While both types of image-based detectors can perform robust person detection, they have the drawback that the field-of-view of most cameras is fairly limited, as also shown in [28]. Image-based detectors will likely remain a viable source for person detections, however, a single LiDAR sensor can be sufficient to cover the complete surrounding, which would

[2]We additionally measured the memory usage on the Jetson AGX, 5311MB for DR-SPAAM, 6318MB for CenterPoint. These numbers are likely to vary depending on specific system setups (we used L4T 32.4.4 and PyTorch 1.6).

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 18:18:24 UTC from IEEE Xplore. Restrictions apply.



(a) Detected by both detectors

(b) Detected by DR-SPAAM only

(c) Detected by CenterPoint only

(d) Missed by both detectors

Fig. 5: Qualitative results from both detectors evaluated at equal error rates. Green boxes represent the ground truth, yellow boxes/spheres are detections by CenterPoint and DR-SPAAM respectively. The blue and red spheres are the 3D and 2D LiDAR points. These examples are picked from the less crowded areas to avoid visual clutter. Note that in many cases where one of the detectors fails, the data seems to be sufficient for a detection to be possible.

	Total GT		Detected by			
			Both	CenterPoint	DR-SPAAM	None
Default	189 579	(50.7%)	95 711 (50.5%)	53 136 (28.0%)	7 212 (3.8%)	33 520 (17.7%)
2D-visible	96 039		71 619 (74.6%)	12 432 (12.9%)	5 263 (5.5%)	6 725 (7.0%)

TABLE IV: Detection statistics obtained using an equal error rate threshold for both CenterPoint and DR-SPAAM. The majority of persons is detected by both detectors, whereas a significant amount is only detected by one of the two.

typically require multiple cameras, significantly increasing compute requirements.

VII. Conclusion

In this paper we investigated differences between 2D and 3D LiDAR-based person detection for mobile robots. For this we perform direct comparisons betweeen the state-of-the-art CenterPoint [49] and DR-SPAAM [20] detectors on the JackRabbot dataset [31]. We found that, when only visible persons are considered, detectors from both 2D and 3D LiDAR sensors perform on a similar level, but in 2D LiDAR sensors persons are more prone to being invisible to the sensor (and thus impossible to detect). While 3D LiDAR sensors overall provide a more robust solution to person detection, 2D LiDAR sensors provide better a trade-off between performance and runtime, a favorable trait for mobile robots with limited onboard computations.

Further analysis of the detectors showed that overall both their detections are well-localized and they are not significantly affected by crowds. Since a significant number of people are only detected by one of the two detectors, if computational resources allow it, an ensemble of the detectors could be an interesting possibility to further boost the detection performance. While one might have assumed the superiority of 3D LiDAR-based person detection, the similar performance and robustness on visible persons for these two sensor types comes as somewhat of a surprise. We believe these insights are valuable during sensor and detector selection for robots that will navigate around persons.

Acknowledgements: We would like to thank Timm Linder and Jen Jen Chung for valuable feedback. This project was funded by the EU H2020 project "CROWDBOT" (779942). Most experiments were performed on the RWTH Aachen University CLAIX 2018 GPU Cluster (rwth0485).

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *arXiv:1702.01105*, 2017.

[2] Kai O Arras, Óscar Martínez Mozos, and Wolfram Burgard. Using Boosted Features for the detection of People in 2D Range Data. In *ICRA*, 2007.

[3] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc Van Gool. Pedestrian detection at 100 frames per second. In *CVPR*, 2012.

[4] Lucas Beyer*, Alexander Hermans*, and Bastian Leibe. DROW: Real-Time Deep Learning based Wheelchair Detection in 2D Range Data. *RA-L*, 2(2):585–592, 2016.

[5] Lucas Beyer, Alexander Hermans, Timm Linder, Kai Oliver Arras, and Bastian Leibe. Deep Person Detection in 2D Range Data. *RA-L*, 3(3):2726–2733, 2018.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 18:18:24 UTC from IEEE Xplore. Restrictions apply.

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *ECCV*, 2020.

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *3DV*, 2017.

[9] Q. Chen, L. Sun, Z. Wang, Kui Jia, and A. Yuille. Object as Hotspots: An Anchor-Free 3D Object Detection Approach via Firing of Hotspots. In *ECCV*, 2020.

[10] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast Point R-CNN. In *ICCV*, 2019.

[11] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CVPR*, 2019.

[12] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017.

[13] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.

[14] Martin Engelcke, Dushyant Rao, D. Wang, C. Tong, and I. Posner. Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks. In *ICRA*, 2017.

[15] Ajo Fod, Andrew Howard, and Maja J. Mataric. A laser-based people tracker. In *ICRA*, 2002.

[16] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The KITTI dataset. *IJRR*, 32(11):1231–1237, 2013.

[17] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *CVPR*, 2018.

[18] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. AFDetV2: Rethinking the Necessity of the Second Stage for Object Detection from Point Clouds. In *AAAI*, 2022.

[19] Omid Hosseini Jafari, Dennis Mitzel, and Bastian Leibe. Real-Time RGB-D based People Detection and Tracking for Mobile Robots and Head-Worn Cameras. In *ICRA*, 2014.

[20] Dan Jia, Alexander Hermans, and Bastian Leibe. DR-SPAAM: A Spatial-Attention and Auto-regressive Model for Person Detection in 2D Range Data. In *IROS*, 2020.

[21] Dan Jia, Mats Steinweg, Alexander Hermans, and Bastian Leibe. Self-Supervised Person Detection in 2D Range Data using a Calibrated Camera. In *ICRA*, 2021.

[22] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft Level 5 Perception Dataset 2020. `https://level5.lyft.com/dataset/`, 2019.

[23] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection From Point Clouds. In *CVPR*, 2019.

[24] Angus Leigh, Joelle Pineau, Nicolas Olmedo, and Hong Zhang. Person tracking and Following with 2D Laser Scanners. In *ICRA*, 2015.

[25] Benjamin Lewandowski, Jonathan Liebner, Tim Wengefeld, Steffen Mueller, and Horst Michael Gross. A Fast and Robust 3D Person Detector and Posture Estimator for Mobile Robotic Application. In *ICRA*, 2019.

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Kaiming He. Focal Loss for Dense Object Detection. In *ICCV*, 2017.

[27] Timm Linder, Kilian Y Pfeiffer, Narunas Vaskevicius, Robert Schirmer, and Kai O Arras. Accurate detection and 3D localization of humans using a novel YOLO-based RGB-D fusion approach and synthetic training data. In *ICRA*, 2020.

[28] Timm Linder, Narunas Vaskevicius, Robert Schirmer, and Kai O Arras. Cross-Modal Analysis of Human Detection for Robotics: An Industrial Case Study. In *IROS*, 2021.

[29] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. TANet: Robust 3D Object Detection from Point Clouds with Triple Attention. In *AAAI*, 2020.

[30] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019.

[31] Roberto Martin-Martin*, Mihir Patel*, Hamid Rezatofighi*, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. JRDB: A Dataset and Benchmark for Visual Perception for Navigation in Human Environments. *PAMI*, 2021.

[32] Caroline Pantofaru. ROS leg_detector package. `https://wiki.ros.org/leg_detector`, 2010. Accessed 2018-02-22.

[33] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds. In *ICCV*, 2019.

[34] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 2017.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015.

[36] Matthias Scheutz, J. McRaven, and Gyorgy Cserey. Fast, reliable, adaptive, bimodal people tracking for indoor environments. In *IROS*, 2004.

[37] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. People Tracking with Mobile Robots Using Sample-Based Joint Probabilistic Data Association Filters. *IJRR*, 22(2):99–116, 2003.

[38] Shaoshuai Shi, Chaoxu Guo, L. Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *CVPR*, 2020.

[39] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. In *CVPR*, 2019.

[40] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *PAMI*, 2020.

[41] Shuran Song, Samuel P. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. In *CVPR*, 2015.

[42] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

[43] H. Thomas, C. Qi, Jean-Emmanuel Deschaud, B. Marcotegui, F. Goulette, and L. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *ICCV*, 2019.

[44] Yue Wang, A. Fathi, Abhijit Kundu, David A. Ross, C. Pantofaru, T. Funkhouser, and J. Solomon. Pillar-based Object Detection for Autonomous Driving. In *ECCV*, 2020.

[45] Zhi Yan, Tom Duckett, and Nicola Bellotto. Online Learning for 3D LiDAR-based Human Detection: Experimental Analysis of Point Cloud Clustering and Classification Methods. *Autonomous Robots*, 44(2):147–164, 2020.

[46] Bo Li Yan Yan, Yuxing Mao. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 2018.

[47] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3DSSD: Point-based 3D Single Stage Object Detector. In *CVPR*, 2020.

[48] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. STD: Sparse-to-Dense 3D Object Detector for Point Cloud. In *ICCV*, 2019.

[49] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D Object Detection and Tracking. In *CVPR*, 2021.

[50] Wu Zheng, Weiliang Tang, Sijin Chen, Li Jiang, and Chi-Wing Fu. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud. In *AAAI*, 2021.

[51] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud. In *CVPR*, 2021.

[52] Y. Zhou, P. Sun, Y. Zhang, Dragomir Anguelov, J. Gao, Tom Ouyang, J. Guo, J. Ngiam, and Vijay Vasudevan. End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds. In *CoRL*, 2019.

[53] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In *CVPR*, 2017.

[54] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection. *arXiv:1908.09492*, 2019.

Authorized licensed use limited to: ShanghaiTech University. Downloaded on August 15,2026 at 18:18:24 UTC from IEEE Xplore. Restrictions apply.