MilliNoise: a Millimeter-wave Radar Sparse Point Cloud Dataset in Indoor Scenarios

Walter Brescia
walter.brescia@poliba.it
Politecnico di Bari
Bari, Italy

Pedro Gomes
pedro.gomes.19@ucl.ac.uk
University College London
London, United Kingdom

Laura Toni
l.toni@ucl.ac.uk
University College London
London, United Kingdom

Saverio Mascolo
saverio.mascolo@poliba.it
Politecnico di Bari
Bari, Italy

Luca De Cicco
luca.decicco@poliba.it
Politecnico di Bari
Bari, Italy

ABSTRACT

Millimeter-wave (mmWave) radar sensors produce Point Clouds (PCs) that are much sparser and noisier than other PC data (e.g., LiDAR), yet they are more robust in challenging conditions such as in the presence of fog, dust, smoke, or rain. This paper presents MilliNoise, a point cloud dataset captured in indoor scenarios through a mmWave radar sensor installed on a wheeled mobile robot. Each of the 12M points in the MilliNoise dataset is accurately labeled as *true*/*noise* point by leveraging known information of the scenes and a motion capture system to obtain the ground truth position of the moving robot. Each frame is carefully pre-processed to produce a fixed number of points for each cloud, enabling classification tools which require data with a fixed shape. Moreover, MilliNoise has been post-processed by labeling each point with the distance to its closest obstacle in the scene, which allows casting the denoising task into the regression framework. Along with the dataset, we provide researchers with the tools to visualize the data and prepare it for statistical and machine learning analysis.

MilliNoise is available at: https://github.com/c3lab/MilliNoise

CCS CONCEPTS

• **General and reference → Measurement**; • **Hardware → Sensor devices and platforms**; • **Information systems →** *Multimedia content creation.*

ACM Reference Format:
Walter Brescia, Pedro Gomes, Laura Toni, Saverio Mascolo, and Luca De Cicco. 2024. MilliNoise: a Millimeter-wave Radar Sparse Point Cloud Dataset in Indoor Scenarios. In *ACM Multimedia Systems Conference 2024 (MMSys '24), April 15–18, 2024, Bari, Italy.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3625468.3652189

1 INTRODUCTION

In the evolving landscape of multimedia systems, the role of 3D data sources has grown significantly, requiring new ways to capture, process, and utilize information. *Point Cloud (PC)* data, produced by 3D Light Detection and Ranging (LiDAR) sensors and stereocameras, is a powerful instrument for capturing the features of three-dimensional scenes. These technologies have induced significant advancements in applications encompassing autonomous navigation and robotics [15, 31], 3D modeling [23], augmented reality/virtual reality [1, 22, 32], and 6DoF video streaming [12, 29].

Despite their advantages, LiDAR and stereocamera systems are often challenged by environmental conditions, such as poor visibility in adverse weather or difficulties in detecting transparent or reflective surfaces [13, 19]. This limits the utilization of such media content for reliable machine applications. Furthermore, these technologies may be cost prohibitive and pose concerns regarding power consumption, thus hindering their adoption in mobile applications.

Millimeter-wave (mmWave) radars are sensors that operate at wavelengths of the order of millimeters. They are equipped with transmitter antennas that emit signals and receiver antennas that capture reflections of these signals. The processing of received signals enables the generation of a 3D PC that describes the environment, including information on the velocity, intensity and orientation of the reflecting objects. Operating within the high-frequency spectrum, typically ranging from 30 to 300 GHz, mmWave radars are capable of detecting multiple objects, even when they are obstructed by other elements. Moreover, these sensors are more robust in adverse environmental conditions, such as fog, dust, smoke, and rain [1, 7, 19, 31]. In this paper, we consider *Frequency-Modulated Continuous Waves (FMCW)* mmWave sensors with integrated antennas, referred to as *Antenna-On-Package* (AoP). Unlike conventional pulse radar sensors, FMCW sensors periodically and continuously emit chirp signals, instead of short pulses [28]. AoP sensors allow a compact package, resulting in significantly reduced production and retail costs, which helped increasing the adoption of mmWave sensors in several research and industrial fields.

Even though mmWave sensors can be considered as an interesting alternative and complement to LiDAR and stereocamera systems, especially in mobile scenarios, the PC data they generate introduces several peculiar challenges. Noise in PC data, usually caused by sensor limitations, environmental factors, or inherent system noise, can significantly degrade application quality. Moreover, unlike 3D LiDAR and stereocamera, these sensors provide sparser PCs, with a reduced amount of details for each detected

MMSys '24, April 15–18, 2024, Bari, Italy
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0412-3/24/04
https://doi.org/10.1145/3625468.3652189

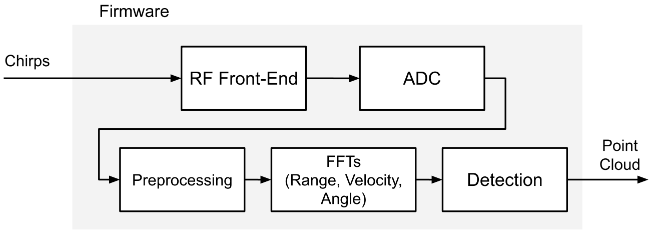


Figure 1: Sensor Pipeline employed to produce PCs from sensor's readings

object. In short, mmWave data suffers from key limitations, specifically (i) the presence of noise and (ii) the sparseness of the acquired data, which substantially limit the adoption of these data sources for multimedia systems. It is therefore an open question if we can use such PC data for machine processing.

Contributions: This work provides accurately labeled sparse PCs generated by FMCW mmWave radar sensors, to allow the design of learning-based denoising approaches. To this end, we present MilliNoise, a dataset that collects different indoor scenarios, such as wide and narrow hallways, tight and loose turns, and shelves, built by properly arranging a set of obstacles. Each point in the MilliNoise dataset is accurately labeled through a motion tracking system, allowing the discrimination of points describing actual objects in the scene from those representing noise, with a sub-millimeter accuracy. The dataset also includes the intensity and velocity values for each point captured by the mmWave sensor. Furthermore, we provide point-wise distances to the closest obstacles, allowing the application of regression methodologies for denoising the MilliNoise dataset.

2 BACKGROUND ON MMWAVE RADARS

In this section, we briefly present the working principles of mmWave FMCW radars, such as the one used to collect the MilliNoise dataset, highlighting the way PCs are obtained by processing the radar's raw data and the main sources of noise. The interested reader is referred to [17] for a comprehensive treatment of mmWave FMCW radar processing.

Working Principles: In a nutshell, the sensor emits a set of signals to obtain information about the environment. Each signal, referred to as *chirp*, is a sinusoidal wave with a frequency that increases linearly over time. A set of such signals then composes the *chirp frame*. Each receiver builds a two-dimensional matrix of N_c chirps per frame by N_s samples per signal. Then, by joining all these matrices, a three-dimensional matrix, denoted as a *radar cube*, is built.

The pipeline employed to produce the PCs from the emitted chirp frames is shown in Figure 1. The radar cube allows to extract information such as: the distance and angle of points belonging to objects in the scene from which the (x, y, z) coordinates can be computed, the points' intensities and velocities.

In particular, the radar cube is analyzed using the Fast Fourier Transform (FFT). The distance of a given obstacle is associated with each peak in the cube: the further away the sensor is, the longer the delay between the emitting and receiving peak [17]. Instead, the value of each peak identifies the intensity of the related point.

The angle estimation is obtained by considering the angle of arrival of the same emitted signal on multiple antennas. By leveraging the known physical displacement between antennas, angle can be extracted by measuring the phase shift of the identical signal received by different antennas. The radar cube allows estimating velocity information of the objects by utilizing multiple chirps. In particular, the velocity of the object reflecting the emitted waves induces a phase shift of the signals from the same chirp on the same antenna. Unlike angle estimation, which involves multiple receivers, this approach leverages the measurements of the phase shift from a single antenna. Further processing of this phase shift through the Doppler-FFT, which accounts for the Doppler effect in the frequency domain, allows the estimation of the velocity of the observed objects. The final PC will contain a number of points depending on the number of reflections captured by the receiving antenna. As a consequence, this procedure generates PCs with a time-varying number of points.

Main challenges: Similarly to any other PC data, mmWave PCs are a set of unordered points that lack correlation with their position in the data structure, thereby inhibiting the application of processing techniques reliant on this type of positional information.

However, unlike other PCs, mmWave ones present peculiar limitations that pose new challenges for data processing. When dealing with low-cost mmWave FMCW AoP radars, constraints on antenna size and their maximum displacements impose limitations on angle and range accuracy. This limitation is evident in the resulting PCs, where the detected points exhibit improved accuracy when the angle is sufficiently small (typically in the $[-10°, 10°]$ range in practical scenarios) [18]. Moreover, mmWave sensors are susceptible to noise, presenting randomly scattered points or artificial reflections, such as clutter and multipath reflections. This challenge becomes more prominent in confined indoor spaces where *ghost* objects and multipath rays may appear. Effectively addressing noise is crucial to improve the signal-to-noise ratio and mitigate the risk of misinterpreting these artifacts as false positives in the processing pipeline. Therefore, the importance of the MilliNoise dataset, which focuses on indoor scenes, becomes evident. Furthermore, mmWave PCs are highly sparse, reaching a few hundred points per frame, instead of the typical several thousand points produced by depth-cameras and LiDARs [19].

3 RELATED WORK

Existing mmWave radar datasets fall into two main classes: datasets for human sensing and datasets for autonomous navigation. In the former, the datasets [8, 22, 25, 26, 30] consist of sequences of human activities acquired using a static mmWave radar. They primarily serve as a base for action recognition applications. By contrast, autonomous navigation datasets, including ours, involve a mmWave radar mounted on a moving system, such as a robot or vehicle, to capture the surrounding 3D space as it moves. The acquired data is then processed for autonomous navigation tasks. These navigation datasets are often multimodal, with mmWave sensors combined with LiDAR or RGB/thermal cameras for richer information. In this section, we provide an overview of PCs datasets

Table 1: Comparison of mmWave Datasets for autonomous navigation tasks

Dataset	Runs	Frames (Points)	Environment	Annotation
nuScenes [5]	1000	1.3M (N/A)	Outdoor	Bounding Box
Astyx [20]	N/A	500 (N/A)	Outdoor	Object oriented
CARRADA [21]	N/A	12.6k (35M)	Outdoor	Bounding Box
RadarScenes [24]	158	N/A (N/A)	Outdoor	Point-Wise
Ghost Dataset [6]	111	N/A (35M)	Outdoor	Point-Wise (manual)
OdomBeyondVision [16]	119	N/A (N/A)	Indoor	Not annotated
MilliNoise (ours)	49	58k (11.6M)	Indoor	Point-Wise sub-mm accuracy

acquired through mmWave sensors for autonomous navigation systems. A summary of the main features of datasets containing mmWave PCs for autonomous navigation scenarios is presented in Table 1 which also includes the MilliNoise dataset presented in this paper.

Over the last few years, research on autonomous navigation has benefited from several large-scale LiDAR datasets, such as *KITTI* [10] and *Waymo* [27]. These datasets capture outdoor automobile scenes as dense PCs and come with high-fidelity annotation tools [3]. By comparison, radar datasets are not as high-quality due to the limitations of FMCW radars. More specifically, they suffer from significantly more noise and are difficult to provide ground-truth annotations. The *nuScenes* [5] is a multimodal dataset also containing radar data. It provides 1000 outdoor scenes, each lasting 20s, with objects annotated with bounding boxes. *Astyx* [20] is a smaller dataset of 500 frames containing around 3000 labeled 3D objects. Data are semi-automatically labeled and manually refined to provide ground truth annotations. In *CARRADA* [21], the dataset is collected using a mmWave sensor synchronized with an RGB camera. The dataset provides a total of 12666 frames, 7193 of which are annotated. Here, annotations are made by a deep-learning method and tracked by the SORT algorithm [4]. Similarly, in [14], an automatic label generation based on position, velocities, and previous semantic annotation for *RadarScenes* [24] dataset is proposed.

Given the difficulty in providing annotations for mmWave data in the above datasets, the annotation process is often simplified. The common approach is either to use a second modality as a reference (e.g. LiDAR) or based on acquisition characteristics (e.g. based on the Doppler effect [11]). However, these types of labeling strategies may lack precision and are therefore not guaranteed to be accurate. Within this context, the *Ghost-Data* [15] dataset offers mmWave PCs manually annotated. The *Ghost-Data* dataset is an outdoor dataset of 21 automobile-rehearsed scenarios of the main object (a pedestrian or cyclist) moving near one or two reflective surfaces. In-depth knowledge of the scenes allows manually labeling of the points. The points are not only labeled as real or noise but the noise points are also labeled according to first or second-order reflection. However, not all points were labeled. A significant number of points were simply labeled as "background".

It is important to note that all the navigation datasets presented so far have been acquired from outdoor scenes. In the current literature, there is a gap in datasets of indoor environments for autonomous navigation. One of the few datasets available for indoor scenes is the OdomBeyondVision [16]. However, this dataset was designed specifically for the ego-motion estimation task. The OdomBeyondVision points are not labeled, which limits research for navigation tasks beyond ego-motion estimation, such as denoising or obstacle detection.

Our proposed MilliNoise dataset seeks to fill this gap in the literature by providing point-wise labeled scenes recorded in an indoor environment. To our best knowledge, MilliNoise is the first mmWave PC dataset of *labeled indoor* scenes on long trajectories and with multiple obstacle positions. Furthermore, given our knowledge of the environment, the labels of the MilliNoise are accurate with sub-millimeter precision and guaranteed to be correct.

4 THE MILLINOISE DATASET

This section describes the acquisition and pre-processing steps used to create the MilliNoise dataset, followed by a description of the dataset features, organization and tools to access and manipulate it.

4.1 Capturing MilliNoise

The MilliNoise dataset has been captured using the acquisition system shown in Figure 2. The data has been collected via a mmWave radar sensor mounted on a mobile robot which followed several paths generated by a path planning algorithm in indoor scenarios populated by obstacles placed in an arena equipped with a motion tracking system.

mmWave Sensor Setup: To capture the surrounding 3D space, a differential drive robot, the *Turtlebot3 Waffle Pi*[1], has been equipped with the Texas Instruments' mmWave sensor AWR6843AOP[2]. As shown in Figure 2, the sensor has been mounted in the front of the robot. The collected data includes not only the points' coordinates but also intensity and velocity values for each point (see Section 2).

Motion Tracking Setup: Besides the mmWave sensor data, the positions of both obstacles and robot are recorded using *Vicon Tracker*[3] motion capture system. This system, based on an array of 8 infrared cameras, utilizes passive reflective markers to detect and recognize objects in the scene, providing objects' pose measurements with an error below a tenth of a millimeter.

Scenario Setup: We aimed at emulating realistic indoor environments, such as wide and narrow hallways, tight and loose turns, shelves and so on. To reach this goal, six courses populated by several obstacles were implemented using boxes of different sizes in a 6 × 6 meters arena, thus generating various possible *scenarios*.

Formally, a scenario is defined by a set of obstacles O_i ($i = 1, \ldots, N$) properly arranged in the environment. Figure 3 depicts

[1] https://www.robotis.us/turtlebot-3-waffle-pi/
[2] https://www.ti.com/product/AWR6843AOP
[3] https://www.vicon.com/software/tracker/

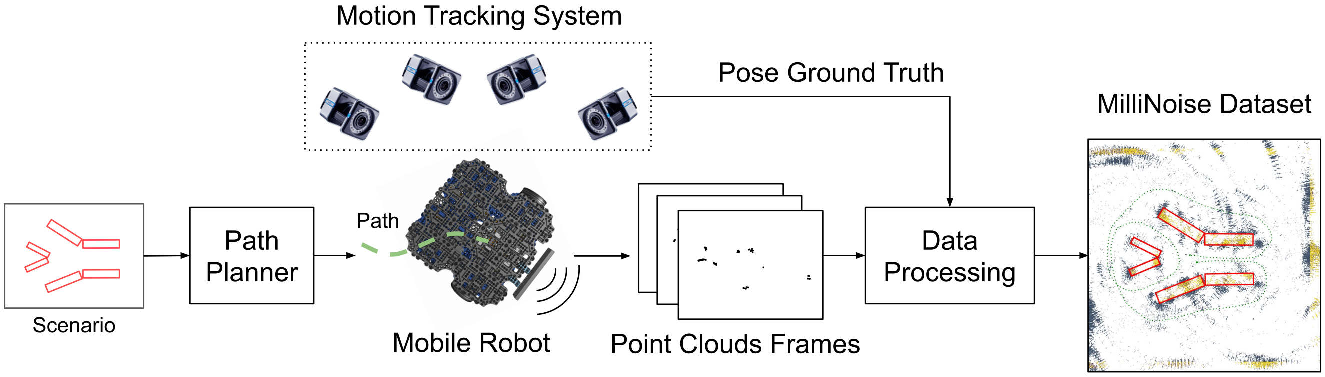


Figure 2: The MilliNoise dataset acquisition system

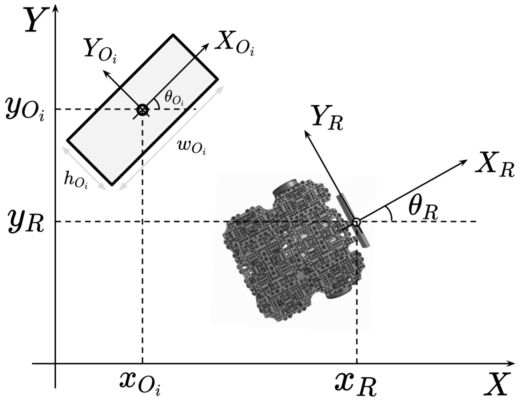


Figure 3: Scenario definition

the robot next to the *i*-th obstacle and reports the notation we use to define a scenario. XY is the coordinate system of the motion tracker, which we define as the *global coordinate system*. Each obstacle O_i in the scene is defined as follows:

$$O_i = (x_{O_i}, y_{O_i}, \theta_{O_i}, w_{O_i}, h_{O_i}) \quad (1)$$

where x_{O_i} and y_{O_i} are the coordinates of the origin of the obstacle in the global coordinate system, θ_{O_i} denotes the orientation of the obstacle with respect to the global coordinate system, w_{O_i} and h_{O_i} are the width and the height of the object, respectively. The robot position and orientation in the global coordinate system is given by the tuple (x_R, y_R, θ_R). Notice that the origin of the robot's coordinate system $X_R Y_R$ is placed on the sensor. The point clouds acquired by the mmWave sensor as the robot moves in the environment are expressed in the robot's coordinate system.

Figure 4 shows the six scenarios collected in the MilliNoise dataset. For each of those scenarios, several collision-free paths were generated using Dijkstra's path planner, a state-of-the-art algorithm for global path planning. These paths were traversed by the robot using the Adaptive Monte Carlo Localization (AMCL) [9] while capturing PCs produced by the mmWave sensor.

We denote each instance of a robot traversing a given scenario as a *run*. Given the combined use of tracking and mmWave sensors, each run contains the scene's global information as well as the robot's individual "perspective". This is, each run contains: (i) the time-varying pose of the robot (x_R, y_R, θ_R) and the fixed position of the obstacles $(x_{O_i}, y_{O_i}, \theta_{O_i})$ in the global coordinates reference system; (ii) the point clouds of the robot surroundings in the robot's coordinate system. Finally, notice that data have been captured at 2 Hz associating them to the current timestamp. Every point cloud captured from the mmWave sensor in this time window is collected under the same frame.

4.2 Point Cloud Pre-Processing

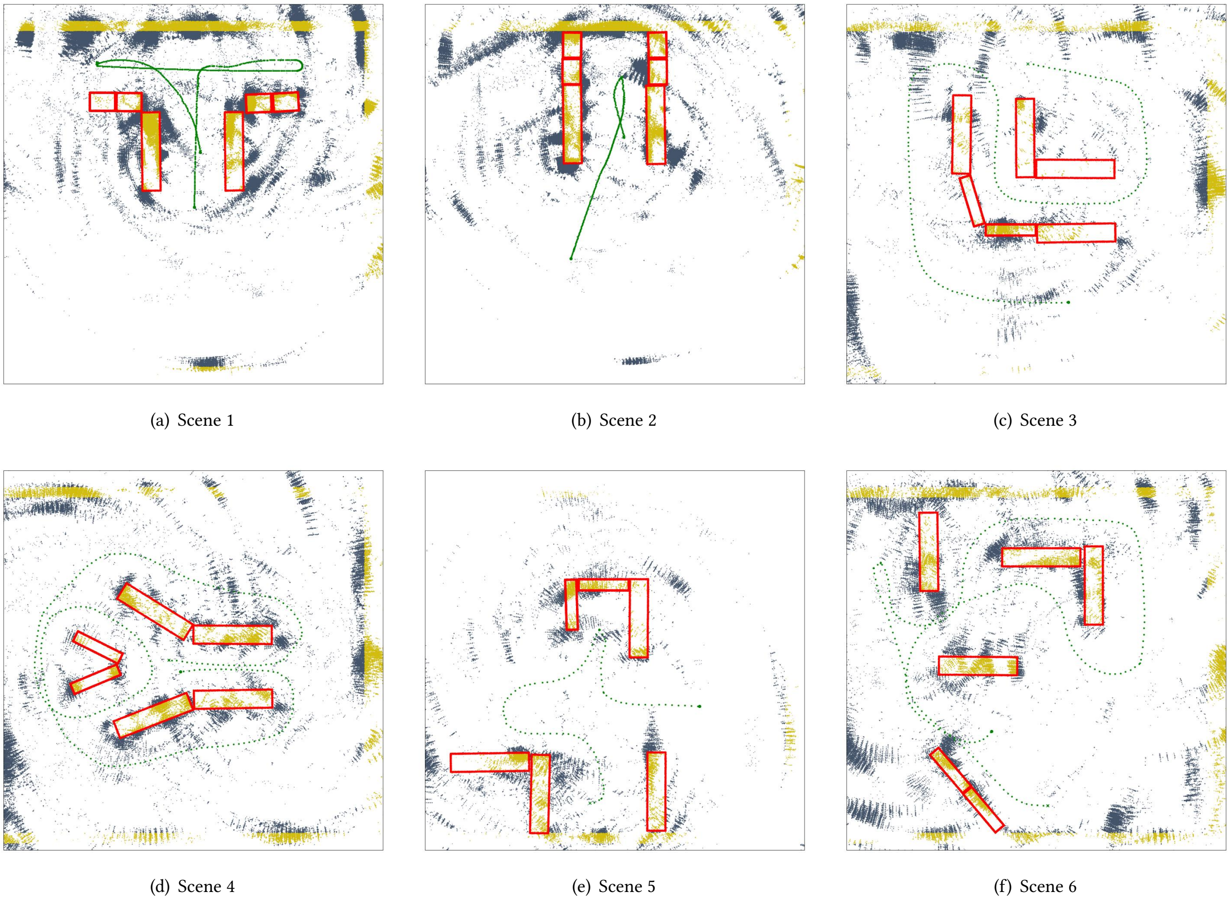
In order to simplify the elaboration of the PCs, the sensor's firmware has been modified to consistently produce a fixed number of points per cloud frame. In particular, the number of points captured by the sensor for each frame was set at 200. Given the specific scenarios under consideration (i.e. short range), the number of points produced by the sensor for each cloud may fall below the desired target number.

Augmenting the desired number of points is a non-trivial task; the additional points introduced should align with the distribution of points actually captured by the sensor. It is worth noting that, due to the sparsity inherent in the PC generated by these sensors, each PC describes multiple regions, often associated to different objects in the scene. As a consequence, rather than considering the distribution of the whole PC, we consider the distribution of points within each smaller region.

To this end, we employ the *Agglomerative Hierarchical Clustering* (AHC) [2] algorithm to reliably generate new points that maintain the integrity of the original PC in terms of both shape and distribution. AHC is a hierarchical clustering method commonly used for data grouping. It follows a bottom-up search, iteratively "agglomerating" the closest data points. For each detected subgroup of points, the algorithm computes the centroid which is added to the PC with fixed fake velocity and intensity values. This allows one to differentiate between the original points and those artificially added to reach the desired total points in the PC.

4.3 Data Labeling

One of the main advantages of MilliNoise is the accurate point-wise annotation. We use the scene information provided by the



(a) Scene 1

(b) Scene 2

(c) Scene 3

(d) Scene 4

(e) Scene 5

(f) Scene 6

Figure 4: Example runs of each of the six scenes available in the MilliNoise dataset. Obstacles are shown in red. Clean (noise) points are shown in yellow (blue). The trajectory followed by the robot is shown with a dashed line.

tracking system, i.e. the poses of the obstacles and mobile robot, to annotate the point cloud acquired by the mmWave radar. To this end, we *roto-translate* each point cloud acquired by the mmWave sensor from the robot coordinate system to the global coordinate system. After appropriate roto-translation and based on the known placement of obstacles, it is possible to automatically label each point as *real* or *noise*, as well as compute the distance of the point to the closest obstacle. Below, we describe how the two labels are assigned.

Real or Noise Label: Given the obstacles and points' roto-translated positions, if a point falls within an obstacle, it is labeled as *real*/*true*, whereas if it does not fall within the spatial region of any obstacle it is labeled as a *noise*/*false* point. For practical reasons, the walls of the room where the dataset was collected were considered as obstacles (real objects).

Distance to Object Label: Similarly, we compute the distance between each point and its closest obstacle. In particular, points falling within obstacles are assigned with a distance equal to zero. Hence, real points have a distance value equal to zero, while noise points will have increasingly higher distance values according to how far they are from the closest obstacle.

The distance annotation is particularly relevant in mmWave datasets since the boundary regions of objects pose several challenges. As shown in Figure 4, large clusters of noise tend to appear on the border of the objects. Their proximity to real objects means they are difficult to detect, possibly affecting navigation tasks. To tackle this issue, the distance label can be used to shift the denoising task from a binary classification problem into a regression one. This allows the system to learn that noise points at the borders of objects are not the same as noise points in the navigation routes. Our hope is that the distance label can help the development of more robust navigation systems capable of distinguishing negligible false points at an object's border from false points in an otherwise empty region where the robot can move freely.

4.4 Dataset features

The MilliNoise dataset collects 49 individual runs captured in a diverse set of 6 real-world scenarios totalling around 58k frames for a duration of 8 hours of data collection. This enables to carry out in-depth exploration of temporal dynamics in the point clouds and its effect on noise points. Figure 4 shows an example run for each of the six considered scenarios, depicting true points and noise points, as well as the path followed by the mobile robot. In total, around 12 million individually labeled 3D points have been collected, with 60.15% of noise points (7 million points) and the remaining part of real points (5 million). Besides its (x_i, y_i, z_i) coordinates, each point is equipped with velocity and intensity information, which paves the way to analyses investigating the impact of these additional variables on the noise points.

4.5 Dataset Organization

In the following, we provide more technical information on how MilliNoise is organized and how to use it.

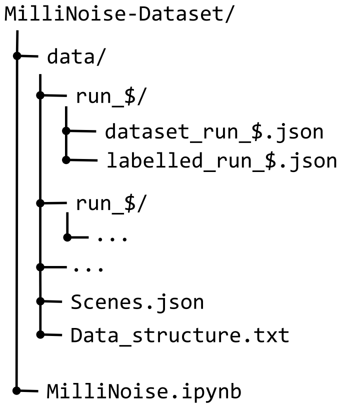
The dataset is organized in folders. Each folder contains data related to a single run. In particular, each folder contains the raw data and its post-elaborated version, which includes the labels and the distance to the closest obstacle.

For each run, we provide the robot's position and captured point clouds in the form of a JSON file with the following pair <*key, value*>:

- `scene_ID`: the ID of the scene of that particular run;
- `timestamp`: the timestamp at which the given frame has been captured;
- `data_pose`: the list of robots' coordinates (x_R, y_R) in meters and orientation θ_R in radians;
- `data_mmwave`: this field contains data coming from the sensor. In particular, it contains the list of frames, in which each point is described by its coordinates in meters, intensity and radial velocity in m/s: $x, y, z, intensity, velocity$. The labeled dataset extends this field with the labels and the distance in meters from the closest obstacle $(x, y, z, intensity, velocity, l, d)$.

Regarding the point cloud augmentation process described in Section 4.2, it is worth noting that one can easily filter out added points. In fact, points added through the augmentation process are assigned with an intensity value equal to 255. Notice that such intensity value is never obtained by real data points acquired by the sensor.

The data folder also contains a text file describing the structure of the data and a file with information of each scene. The scenes' file contains the following information: the scene's ID, the number of obstacles, the ID of each obstacle and the related position and sizes. A Jupyter notebook placed in the main folder provides examples on how to load, process, and visualize the mmWave data. Figure 5 depicts the resulting directory structure of MilliNoise.



```
MilliNoise-Dataset/
├─ data/
│  ├─ run_$/
│  │  ├─ dataset_run_$.json
│  │  └─ labelled_run_$.json
│  ├─ run_$/
│  │  └─ ...
│  ├─ ...
│  ├─ Scenes.json
│  └─ Data_structure.txt
└─ MilliNoise.ipynb
```

Figure 5: The directory structure of the dataset.

4.6 Dataset Tools

In order to further improve usability and help bootstrapping applications, we provide a few tools as examples on how to access and use the MilliNoise dataset.

In particular, the `MilliNoise.ipynb` script provides several useful functions, the main ones being:

- `load_run`: it takes as input the directory containing the run one wants to load and returns the robot's poses and the mmWave frames in the form of list of lists;
- `apply_rototranslation`: it requires the mmWave frames and the list of robot's poses and returns a *numpy* array containing the mmWave frames rototranslated to build a map;
- `plot_scene`: this function 3D-plots the (numpy array) mmWave frames in input; optionally, it accepts the list of robot's poses to plot the trajectory, the list of points representing the obstacles, the points' sizes, the x and y limits to focus on a particular region of interest and the destination directory (and file name) to save the produced plot;
- `plot_2d_scene`: this function is analogous to *plot_scene* but projects the plot on the 2D XY plane;
- `load_train_test_data`: this function reads the provided path searching for mmWave data and returns the list of all frames from all runs in the directory; it optionally accepts a list to filter runs as train, test and validation data;
- `get_data_and_label`: it expects in input a list of mmWave labeled frames and returns the numpy arrays of frames and related points' labels. This function allows one to directly use data for training purposes.

5 CONCLUSIONS

In this work, we presented MilliNoise, a labeled dataset with point clouds collected in indoor scenarios using a moving mmWave FMCW radar mounted on a mobile robot. To the best of our knowledge, MilliNoise is the first dataset collected with such an acquisition system in indoor scenarios. A distinctive feature of MilliNoise is that it provides an accurate point-wise labeling of each of its 12M points. In particular, label accuracy is only related to the accuracy of the motion tracking system, which is below a tenth of a millimeter. Furthermore, the sensor's firmware has been modified to provide a fixed number of points for each frame, enabling the employment of methodologies which require a fixed number of points for each frame. We believe that the MilliNoise dataset will prove valuable for designing learning-based methods to denoise mmWave PCs which will, in turn, make these devices more efficient and useful in autonomous navigation systems and multimedia scenarios.

ACKNOWLEDGMENTS

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART", CUP: D93C22000910001), CISCO and the EPSRF-SFI grant EP/T03324X/1.

REFERENCES

[1] Yasin Almalioglu, Mehmet Turan, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. 2021. Milli-RIO: Ego-Motion Estimation With Low-Cost Millimetre-Wave Radar. *IEEE Sensors Journal* 21, 3 (2021), 3314–3323. https://doi.org/10.1109/JSEN.2020.3023243

[2] Michael R. Anderberg. 1973. *Cluster Analysis for Applications*. Academic Press.

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Jürgen Gall, and Cyrill Stachniss. 2021. Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. *The International Journal of Robotics Research* 40, 8-9 (2021), 959–967. https://doi.org/10.1177/02783649211006735

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. 2016. Simple online and realtime tracking. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*. 3464–3468. https://doi.org/10.1109/ICIP.2016.7533003

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*. 11621–11631.

[6] Mahdi Chamseddine, Jason Rambach, Didier Stricker, and Oliver Wasenmuller. 2021. Ghost target detection in 3d radar data using point cloud based deep neural network. In *Proc. of the 25th International Conference on Pattern Recognition (ICPR)*. 10398–10403.

[7] Yuwei Cheng, Hu Xu, and Yimin Liu. 2021. Robust Small Object Detection on the Water Surface through Fusion of Camera and Millimeter Wave Radar. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 15243–15252. https://doi.org/10.1109/ICCV48922.2021.01498

[8] Han Cui, Shu Zhong, Jiacheng Wu, Zichao Shen, Naim Dahnoun, and Yiren Zhao. 2023. MiliPoint: A Point Cloud Dataset for mmWave Radar. arXiv:2309.13425 [cs.LG]

[9] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. 1999. Monte Carlo localization for mobile robots. In *Proc. of the IEEE International Conference on Robotics and Automation*, Vol. 2. 1322–1328. https://doi.org/10.1109/ROBOT.1999.772544

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*. 3354–3361. https://doi.org/10.1109/CVPR.2012.6248074

[11] Thomas Griebel, Dominik Authaler, Markus Horn, Matti Henning, Michael Buchholz, and Klaus Dietmayer. 2021. Anomaly Detection in Radar Data Using PointNets. In *Proc. of the IEEE International Intelligent Transportation Systems Conference (ITSC)* (Indianapolis, IN, USA). 2667–2673. https://doi.org/10.1109/ITSC48978.2021.9564730

[12] Bo Han, Yu Liu, and Feng Qian. 2020. ViVo: visibility-aware mobile volumetric video streaming. In *Proc. of the 26th ACM Annual International Conference on Mobile Computing and Networking* (London, United Kingdom) *(MobiCom '20)*. Article 11, 13 pages. https://doi.org/10.1145/3372224.3380888

[13] Jui-Te Huang, Chen-Lung Lu, Po-Kai Chang, Ching-I Huang, Chao-Chun Hsu, Zu Lin Ewe, Po-Jui Huang, and Hsueh-Cheng Wang. 2021. Cross-Modal Contrastive Learning of Representations for Navigation Using Lightweight, Low-Cost Millimeter Wave Radar for Adverse Environmental Conditions. *IEEE Robotics and Automation Letters* 6, 2 (April 2021), 3333–3340. https://doi.org/10.1109/lra.2021.3062011

[14] Johannes Kopp, Dominik Kellner, Aldi Piroli, and Klaus Dietmayer. 2023. Tackling Clutter in Radar Data - Label Generation and Detection Using PointNet++. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. 1493–1499. https://doi.org/10.1109/ICRA48891.2023.10160222

[15] Florian Kraus, Nicolas Scheiner, Werner Ritter, and Klaus Dietmayer. 2021. The Radar Ghost Dataset – An Evaluation of Ghost Objects in Automotive Radar Data. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 8570–8577. https://doi.org/10.1109/IROS51168.2021.9636338

[16] Peize Li, Kaiwen Cai, Muhamad Risqi U. Saputra, Zhuangzhuang Dai, and Chris Xiaoxuan Lu. 2022. OdomBeyondVision: An Indoor Multi-modal Multi-platform Odometry Dataset Beyond the Visible Spectrum. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 3845–3850. https://doi.org/10.1109/IROS47612.2022.9981865

[17] Xinrong Li, Xiaodong Wang, Qing Yang, and Song Fu. 2021. Signal processing for TDM MIMO FMCW millimeter-wave radar sensors. *IEEE Access* 9 (2021), 167959–167971.

[18] Yang Li, Yutong Liu, Yanping Wang, Yun Lin, and Wenjie Shen. 2020. The Millimeter-Wave Radar SLAM Assisted by the RCS Feature of the Target and IMU. *Sensors* 20, 18 (2020). https://doi.org/10.3390/s20185421

[19] Chris Xiaoxuan Lu, Stefano Rosa, Peijun Zhao, Bing Wang, Changhao Chen, John A. Stankovic, Niki Trigoni, and Andrew Markham. 2020. See through smoke: robust indoor mapping with low-cost mmWave radar. In *Proc. of the 18th International Conference on Mobile Systems, Applications, and Services* (Toronto, Ontario, Canada) *(MobiSys '20)*. 14–27. https://doi.org/10.1145/3386901.3388945

[20] Michael Meyer and Georg Kuschk. 2019. Automotive Radar Dataset for Deep Learning Based 3D Object Detection. In *Proc. of 16th European Radar Conference (EuRAD)*. 129–132.

[21] Arthur Ouaknine, Alasdair Newson, Julien Rebut, Florence Tupin, and Patrick Pérez. 2021. CARRADA Dataset: Camera and Automotive Radar with Range-Angle- Doppler Annotations. In *Proc. of the 25th International Conference on Pattern Recognition (ICPR)*. 5068–5075. https://doi.org/10.1109/ICPR48806.2021.9413181

[22] Sameera Palipana, Dariush Salami, Luis A. Leiva, and Stephan Sigg. 2021. Pantomime: Mid-Air Gesture Recognition with Sparse Millimeter-Wave Radar Point Clouds. *Proc. of the ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 1, Article 27, 27 pages. https://doi.org/10.1145/3448110

[23] Victor Sanchez and Avideh Zakhor. 2012. Planar 3D modeling of building interiors from point cloud data. In *Proc. of the 19th IEEE International Conference on Image Processing*. 1777–1780. https://doi.org/10.1109/ICIP.2012.6467225

[24] Ole Schumann, Markus Hahn, Nicolas Scheiner, Fabio Weishaupt, Julius F. Tilly, Jürgen Dickmann, and Christian Wöhler. 2021. RadarScenes: A Real-World Radar Point Cloud Data Set for Automotive Applications. In *Proc. of the IEEE 24th International Conference on Information Fusion (FUSION)*. 1–8. https://doi.org/10.23919/FUSION49465.2021.9627037

[25] Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. 2020. mm-Pose: Real-Time Human Skeletal Posture Estimation Using mmWave Radars and CNNs. *IEEE Sensors Journal* 20, 17 (2020), 10032–10044. https://doi.org/10.1109/JSEN.2020.2991741

[26] Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani Srivastava. 2019. RadHAR: Human Activity Recognition from Point Clouds Generated through a Millimeter-wave Radar. In *Proc. of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems* (Los Cabos, Mexico) *(mmNets'19)*. 51–56. https://doi.org/10.1145/3349624.3356768

[27] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*. 2446–2454.

[28] Arthur Venon, Yohan Dupuis, Pascal Vasseur, and Pierre Merriaux. 2022. Millimeter Wave FMCW RADARs for Perception, Recognition and Localization in Automotive Applications: A Survey. *IEEE Transactions on Intelligent Vehicles* 7, 3 (2022), 533–555. https://doi.org/10.1109/TIV.2022.3167733

[29] Irene Viola and Pablo Cesar. 2023. Chapter 15 - Volumetric video streaming: Current approaches and implementations. In *Immersive Video Technologies*, Giuseppe Valenzise, Martin Alain, Emin Zerman, and Cagri Ozcinar (Eds.). Academic Press, 425–443. https://doi.org/10.1016/B978-0-32-391755-1.00021-3

[30] Yi-Hung Wu, Hsin-Che Chiang, Shervin Shirmohammadi, and Cheng-Hsin Hsu. 2023. A Dataset of Food Intake Activities Using Sensors with Heterogeneous Privacy Sensitivity Levels. In *Proc. of the 14th Conference on ACM Multimedia Systems* (Vancouver, BC, Canada) *(MMSys '23)*. 416–422. https://doi.org/10.1145/3587819.3592553

[31] Huan Yin, Xuecheng Xu, Yue Wang, and Rong Xiong. 2021. Radar-to-lidar: Heterogeneous place recognition via joint learning. *Frontiers in Robotics and AI* 8 (2021). https://doi.org/10.3389/frobt.2021.661199

[32] Guangcheng Zhang, Xiaoyi Geng, and Yueh-Jaw Lin. 2021. Comprehensive mPoint: A Method for 3D Point Cloud Generation of Human Bodies Utilizing FMCW MIMO mm-Wave Radar. *Sensors* 21, 19 (2021). https://doi.org/10.3390/s21196455