

# The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset

Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman and Ingmar Posner

**Abstract**—In this paper we present *The Oxford Radar RobotCar Dataset*, a new dataset for researching scene understanding using Millimetre-Wave FMCW scanning radar data. The target application is autonomous vehicles where this modality is robust to environmental conditions such as fog, rain, snow, or lens flare, which typically challenge other sensor modalities such as vision and LIDAR.

The data were gathered in January 2019 over thirty-two traversals of a central Oxford route spanning a total of 280 km of urban driving. It encompasses a variety of weather, traffic, and lighting conditions. This 4.7 TB dataset consists of over 240,000 scans from a Navtech CTS350-X radar and 2.4 million scans from two Velodyne HDL-32E 3D LIDARs; along with six cameras, two 2D LIDARs, and a GPS/INS receiver. In addition we release ground truth optimised radar odometry to provide an additional impetus to research in this domain. The full dataset is available for download at:

[ori.ox.ac.uk/datasets/radar-robotcar-dataset](http://ori.ox.ac.uk/datasets/radar-robotcar-dataset)

## I. INTRODUCTION

While many of the challenges in urban autonomy have been met successfully with lasers and cameras, radar offers the field of robotics an alternative modality for robust sensing. The Frequency-Modulated Continuous-Wave (FMCW) class of radar provides a 360°-view of the scene and is capable of detecting targets at ranges far exceeding those of automotive 3D LIDAR. These advantages are particularly valuable to autonomous vehicles which need to see further if they are to travel safely at higher speeds or to operate in wide open spaces where there is a dearth of distinct features. Moreover, these vehicles must function reliably in unstructured environments and require a sensor such as radar that thrives in all conditions – rain, snow, dust, fog, or direct sunlight.

This dataset builds upon the *Oxford RobotCar Dataset* [1], one of the the largest available datasets for autonomous driving research. The original dataset release consisted of over 20 TB of vehicle-mounted monocular and stereo imagery, 2D and 3D LIDAR, as well as inertial and GPS data collected over a year of driving in Oxford, UK. More than 100 traversals of a 10 km route were performed over this period to capture scene variation over a range of timescales, from the 24 h day/night illumination cycle to long-term seasonal variations. As a valuable resource for self-driving research, the vehicle software and mechatronics have been maintained since the original dataset was gathered and released: now configured with a millimetre-wave radar and two additional 3D LIDARs. The current appearance of the vehicle with

Authors are from the Oxford Robotics Institute, University of Oxford, UK. {dbarnes,mattgadd,pmurcutt,pnewman,ingmar}@robots.ox.ac.uk

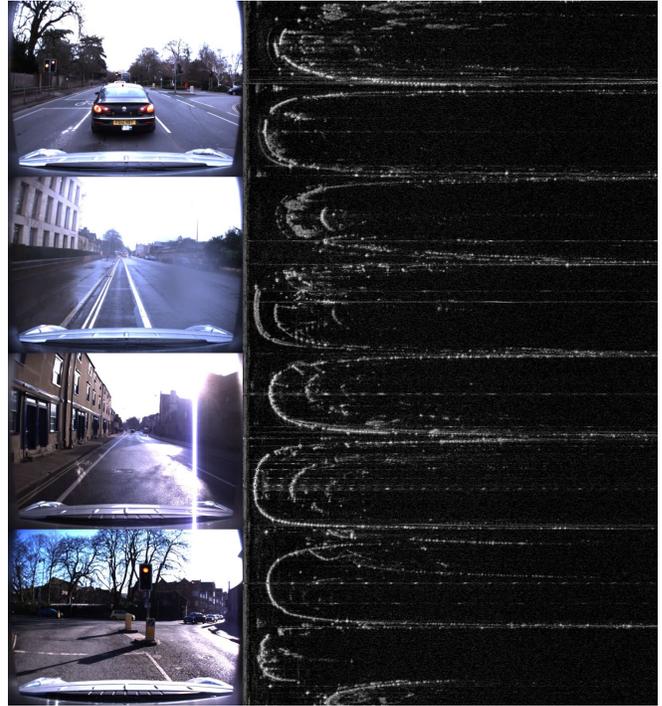


Fig. 1. The Oxford Radar RobotCar Dataset for complex and robust scene understanding with Millimetre-Wave FMCW scanning radar data. We collected 32 traversals of a central Oxford route with the Oxford RobotCar platform during the month of January, 2019. Despite weather conditions such as rain, direct sunlight, and fog which are challenging for traditional modalities such as vision (left), radar (right) holds the promise of consistent sensor observations for mapping, localisation, and scene understanding. Sample pairs are taken from different locations of the driven route.

these additional sensors can be seen in Figure 2. Along with the raw sensor recordings from all sensors, we provide an updated set of calibrations, ground truth trajectory for the radar sensor as well as MATLAB and Python development tools for utilising the data.

By sharing this large-scale radar dataset with researchers we aim to accelerate research into this promising modality for mobile robotics and autonomous vehicles of the future.

## II. RELATED WORK

A number of LIDAR- and vision-based autonomous driving datasets, such as [2]–[8], are available to the community and were primarily collected in order to develop competencies in these modalities. This dataset release is meant to advocate the increased exploitation of FMCW radar for vehicle autonomy. We therefore present radar data *alongside* the camera and LIDAR data typically appearing

these datasets with the goal of replicating and advancing these competencies with this promising sensor modality.

Similar radar sensors have been used in a variety of domains for mapping, navigation, and perception [9]–[11]. Some publications using similar, if not identical, FMCW radar for state estimation prior to the release of this dataset include [12]–[16]. To this end, Section VI discusses the optimised ground truth radar odometry data released as part of this dataset to help further research in this area.

The Navtech radar dataset presented in [17] is concurrent to this release. Although significantly smaller in size than our release, the comparable setups should provide a great opportunity for cross-validating approaches between datasets in different geographical locations. The *Marulan datasets* presented in [18] also use FMCW radar, but only configured to a maximum range of 40 m. Additionally, while these datasets are collected under variable conditions, they represent fairly static outdoor scenes that are not representative of urban driving.

### III. THE RADAR ROBOTCAR PLATFORM

The dataset was collected using the Oxford RobotCar platform as in [1], an autonomous-capable Nissan LEAF, illustrated with sensor layout in Figure 2. For this release, the RobotCar was equipped with the following sensors which were not in the original release:

- 1 x Navtech CTS350-X Millimetre-Wave FMCW radar, 4 Hz, 400 measurements per rotation, 163 m range, 4.38 cm range resolution, 1.8° beamwidth
- 2 x Velodyne HDL-32E 3D LIDAR, 360° HFoV, 41.3° VFoV, 32 planes, 20 Hz, 100 m range, 2 cm range resolution

In addition to the original sensors as in [1]:

- 1 x Point Grey Bumblebee XB3 (BBX3-13S2C-38) trinocular stereo camera, 1280×960×3, 16 Hz, 1/3" Sony ICX445 CCD, global shutter, 3.8 mm lens, 66° HFoV, 12/24 cm baseline
- 3 x Point Grey Grasshopper2 (GS2-FW-14S5C-C) monocular camera, 1024×1024, 11.1 Hz, 2/3" Sony ICX285 CCD, global shutter, 2.67 mm fisheye lens (Sunex DSL315B-650-F2.3), 180° HFoV
- 2 x SICK LMS-151 2D LIDAR, 270° FoV, 50 Hz, 50 m range, 0.5° resolution
- 1 x NovAtel SPAN-CPT ALIGN inertial and GPS navigation system, 6 axis, 50 Hz, GPS/GLONASS, dual antenna

As the main focus of this release, the Navtech CTS350-X radar was mounted at the centre of the vehicle aligned to the vehicle axes. We used a pair of Velodyne HDL-32E 3D LIDARs instead of the LD-MRS 3D LIDAR used in [1] for drastically improved 3D scene understanding. In addition to providing twice the range and intensity returns, the Velodynes provide a full 360° HFoV with 41.3° VFoV for full coverage around the vehicle.

Sensor drivers for both the Navtech CTS350-X and Velodyne HDL-32E devices were developed internally to provide

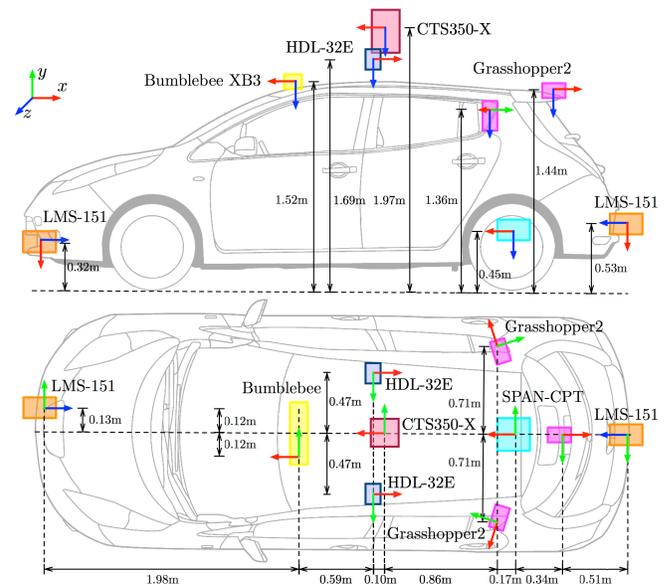


Fig. 2. The Radar RobotCar platform (top) and sensor location diagram (bottom) with the Navtech CTS350-X radar mounted in the centre. Coordinate frames show the origin and direction of each sensor mounted on the vehicle with the convention:  $x$ -forward (red),  $y$ -right (green),  $z$ -down (blue). Measurements shown are approximate; the development tools include exact  $SE(3)$  extrinsic calibrations for all sensors.

accurate synchronisation and timestamping with the other sensors.. For further details on sensors from the original release, compute specifications, and data logging procedures please consult the original dataset paper [1].

### IV. RADAR DATA

The Navtech CTS350-X is a FMCW scanning radar without Doppler information, configured to return 3768 power readings at a range resolution of 4.38 cm across 400 azimuths at a frequency of 4 Hz (corresponding to a maximum range of 163 m and 0.9° azimuth resolution). Other configurations of the Navtech CTS350-X are able to provide range in excess of 650 m or higher rotation frequencies. However, for this dataset shorter range, high resolution data was deemed most

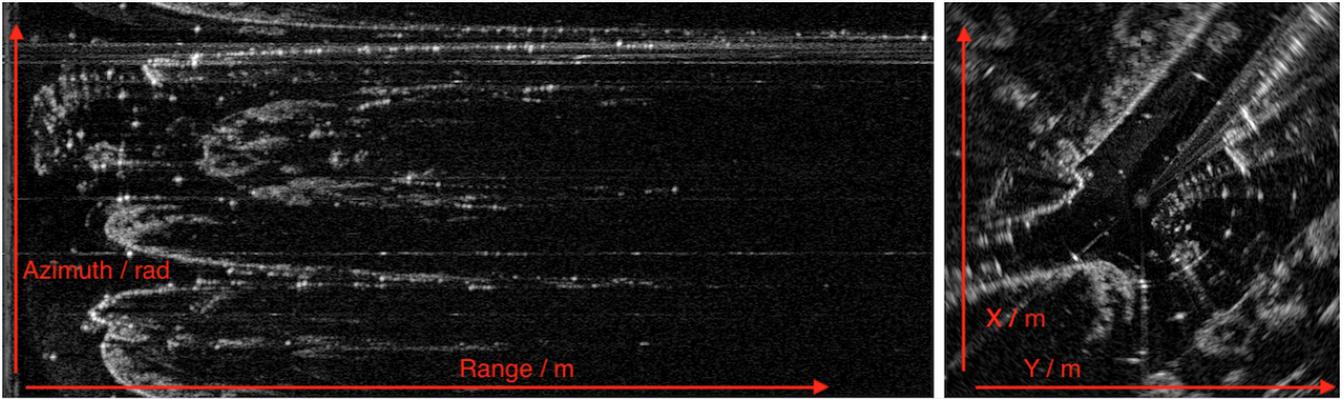


Fig. 3. Example sensor data from the Navtech CTS350-X radar. Raw radar power returns in polar form (left) for a full sweep of  $0 \rightarrow 2\pi$  over a range of  $0 \rightarrow 163\text{m}$  and the corresponding scan in Cartesian form (right), with the vehicle in the center and axes from  $-50\text{m} \rightarrow 50\text{m}$ . Tools required to parse the data and perform the polar-to-Cartesian conversion are provided in the SDK discussed in Section VII.

useful in urban scenarios where straight line distances over  $163\text{m}$  are rare.

This type of radar rotates about its vertical axis while continuously transmitting and receiving frequency-modulated radio waves similar to a spinning LIDAR. The frequency shift between the transmitted and received waves is used to compute the range of an object, and the received power is a function of the object’s reflectivity, size, shape, and orientation relative to the receiver. One full rotation and its 2D power data can be represented by a matrix in which each row corresponds to an azimuth and each column to a range, as shown in Figure 3, where the intensity represents the highest power reflection within a range bin.

The radar operates at frequencies of  $76\text{GHz}$  to  $77\text{GHz}$ , ensuring consistent measurements through harsh local conditions such as dust, rain, and snow. The main beam spread is  $1.8^\circ$  between  $-3\text{dB}$  points horizontally and vertically; with an additional cosec squared fill-in beam pattern up to  $40^\circ$  below the horizontal which permits detection of objects beneath the main beam.

## V. DATA COLLECTION

This dataset release follows the original Oxford RobotCar Dataset route in Oxford, UK and consists of 32 traversals in different traffic, weather, and lighting conditions in January 2019 totalling  $280\text{km}$  of urban driving. The vehicle was driven manually throughout the period of data collection; no autonomous capabilities were used. The total download size of the dataset is  $4.7\text{TB}$ . Figure 4 shows a random selection of images taken from the dataset, illustrating the variety of situations encountered. Table I lists summary statistics for the raw data collected through the entire month-long collection while Table II lists summary statistics for processed data which are also made available for download.

Every effort was made to follow the exact route for every traversal. However, this was not always possible and slight diversions were made infrequently. Additionally, two partial traversals are included which do not cover the entire route. The GPS/INS data can be used to identify diversions. However, similarly to [1], the accuracy of the fused INS

Sensor	Type	Count	Size
Bumblebee XB3	Image	2,887,776	2.2 TB
Grasshopper 2	Image	2,963,601	1.6 TB
LMS-151	2D Scan	5,988,123	67.3 GB
SPAN-CPT GPS	3D Position	300,814	35.4 MB
SPAN-CPT INS	6DoF Position	3,008,085	491.7 MB
Navtech CTS350-X	Radar Scan	240,088	106.1 GB
Velodyne Raw	3D Scan	2,405,785	91.0 GB

TABLE I

SUMMARY STATISTICS FOR COLLECTED DATA.

Sensor	Type	Count	Size
Stereo Visual Odometry (VO)	6DoF Position	961,487	89.0 MB
GT Radar Odometry	3DoF Position	240,024	28.6 MB
Velodyne Binary	3D Scan	2,405,785	774.3 GB

TABLE II

SUMMARY STATISTICS FOR PROCESSED DATA.

solution varied significantly during the course of data collection. Instead, we suggest using the optimised radar odometry shown in Figure 6 and discussed in Section VI as the best available solution of the underlying motion of the radar.

### A. Sensor Calibration

We include in this release a full set of extrinsic calibration data needed to utilise the additional Navtech and Velodyne sensors while the intrinsics and extrinsics of the sensors from [1] remain unchanged. Figure 2 illustrates the extrinsic configuration of sensors on the Radar RobotCar platform. The new LIDAR and radar sensors’ extrinsics were calibrated by manually taking measurements of the as-built positions of the sensors as a seed and then performing pose optimisation to minimise the error between laser and radar co-observations. Precise extrinsic calibrations for each sensor are included in the development tools to be discussed in Section VII. As per [1] the sensor extrinsics are not guaranteed to have remained constant throughout the lifetime of the vehicle. However, given the relatively short duration of this trial, little degradation is expected. Given the large overlap in observable environment and diversity of sensor modalities,



Fig. 4. Random pairs of Bumblebee XB3 images (left) with the temporally closest Navtech CTS350-X radar scan (right) from the Oxford Radar RobotCar Dataset, showing the challenging diversity of weather, lighting, and traffic conditions encountered during the period of data collection in Oxford, UK in January 2019.

```

oxford-radar-robotcar-dataset
├── yyyy-mm-dd-HH-MM-SS-radar-oxford-10k # (-partial)
│   ├── gt
│   │   └── radar_odometry.csv # ground truth radar odometry
│   ├── radar
│   │   ├── <timestamp>.png # Navtech radar data
│   │   ├── ...
│   │   ├── velodyne_left
│   │   │   ├── <timestamp>.bin # Velodyne binary sensor data
│   │   │   ├── <timestamp>.png # Velodyne raw sensor data
│   │   │   ├── ...
│   │   │   ├── velodyne_right
│   │   │   │   ├── <timestamp>.bin # Velodyne binary sensor data
│   │   │   │   ├── <timestamp>.png # Velodyne raw sensor data
│   │   │   │   ├── ...
│   │   ├── radar.timestamps
│   │   ├── velodyne_left.timestamps
│   │   ├── velodyne_right.timestamps
│   │   └── # Plus the original Oxford RobotCar Dataset layout
│   ├── ...
│   └── ...
└── ...

```

Fig. 5. Directory layout for the Oxford Radar RobotCar Dataset. When downloading multiple zip archives from multiple traversals, extracting them all in the same directory will preserve the folder structure shown here.

this dataset provides an excellent test-bed for work on cross-modality calibration and we encourage using our estimates as initial seeds for further research.

## B. Data Formats

Figure 5 shows the typical directory structure for a single dataset. In contrast to [1] we do not chunk sensor data into smaller files. Therefore each zip file download corresponds to the complete sensor data for one dataset traversal (or processed sensor output such as stereo VO) with the folder structure inside the archive illustrated in Figure 5. The formats for each data type are as follows:

1) *Radar scans*: are stored as lossless-compressed PNG files in polar form with each row representing the sensor reading at each azimuth and each column representing the raw power return at a particular range. The files are structured as `<dataset>/radar/<timestamp>.png` where `<timestamp>` is the starting UNIX timestamp of the capture, measured in microseconds. In the configuration used there are 400 azimuths per sweep (rows) and 3768 range bins (columns).

To give users all the raw data they could need we also embed the following *per azimuth* metadata into the PNG image within the first 11 columns as follows:

- UNIX timestamp as `int64` in cols 1-8.
- Sweep counter as `uint16` in cols 9-10; converted to angle in radians with:

$$\text{angle} = \text{sweep\_counter} / 2800 * \pi$$

- Finally, a *valid* flag as `uint8` in col 11.

The *valid* flag is included as there are a very small number of data packets carrying azimuth returns that are infrequently dropped. To this end, in order to simplify usage for users, we have interpolated adjacent returns so that each provided radar scan has 400 azimuths (rows). If this is not desirable it

is advised to simply drop any row which has the *valid* flag set to zero.

2) *3D Velodyne LIDAR scans*: are provided in two formats, a raw form which encapsulates all the raw data recorded from the sensor for users to do with as they please, or in binary form representing the non-motion compensated pointcloud for a particular scan.

*Raw scans*: are released as lossless PNG files with each column representing the sensor reading at each azimuth. The files are structured `<dataset>/<laser>/<timestamp>.png`, where `<laser>` is `velodyne_left` or `velodyne_right` and `<timestamp>` is the starting UNIX timestamp of the capture, measured in microseconds. To give users all the raw data they could need we embed *per azimuth* metadata into the PNG within the following rows:

- Raw intensities for each laser as `uint8` in rows 1-32.
- Raw ranges for each laser as `uint16` in rows 33-96, converted to metres with:

$$\text{ranges (metres)} = \text{ranges\_raw} * 0.02$$

- Sweep counter as `uint16` in rows 97-98; converted to angle in radians with:

$$\text{angle} = \text{sweep\_counter} / 18000 * \pi$$

- Finally, *approximate* UNIX timestamps as `int64` in rows 99-106

Timestamps are received for each data packet from the Velodyne LIDAR which includes 12 sets of readings for all 32 lasers. We have linearly interpolated timestamps at each azimuth reading. However, the original received timestamps can be extracted by simply taking every twelfth timestamp.

*Binary scans*: are released as single-precision floating point values packed into a binary file representing the non-motion compensated pointcloud generated from the corresponding raw scan, similar to the Velodyne scan format in [3]. The files are structured as `<dataset>/<laser>/<timestamp>.bin`, where `<laser>` is `velodyne_left` or `velodyne_right` and `<timestamp>` is the starting UNIX timestamp of the capture, measured in microseconds. Each scan consists of  $(x, y, z, I) \times N$  values, where  $x, y, z$  are the 3D Cartesian coordinates of the LIDAR return relative to the sensor (in metres), and  $I$  is the measured intensity value.

3) *Ground Truth Radar Odometry*: The files `<dataset>/gt/radar_odometry.csv` contain the  $SE(2)$  relative pose solution as detailed in Section VI, consisting of the source and destination frame UNIX timestamps (chosen to be in the middle of the corresponding radar scans), the six-vector Euler parameterisation  $(x, y, z, \alpha, \beta, \gamma)$  of the  $SE(3)$  relative pose relating the two frames (where  $z, \alpha, \beta$  are all zero but included for compatibility with other pose sources, most notably in the original SDK) and the *starting* source and destination frame UNIX timestamps of the corresponding radar scans which can be used as the `<timestamp>` to load the corresponding radar scan files.

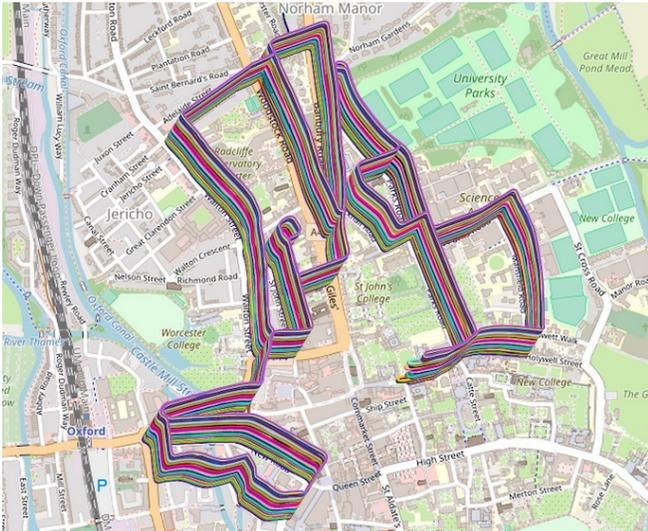


Fig. 6. Optimised radar odometry plotted on OpenStreetMap [19] for each of the 32 dataset traversals, where each run is offset for visualisation purposes. The trajectories were generated by optimising robust VO [20], visual loop closures [21], and GPS/INS as constraints. Map data copyrighted OpenStreetMap contributors and available from [openstreetmap.org](http://openstreetmap.org).

## VI. GROUND TRUTH RADAR ODOMETRY

Alongside this dataset we provide ground truth  $SE(2)$  radar odometry temporally aligned to the radar data to help further research using this modality for motion estimation, map building, and localisation. The poses were generated by performing a large-scale optimisation with Ceres Solver [22] incorporating VO, visual loop closures, and GPS/INS constraints with the resulting trajectories shown in Figure 6.

Specifically, we include all 32 dataset traversals and calculate robust VO using the approach proposed in [20], in which each image is masked with a neural network before generating odometry estimates using [23]. Visual loop closures are then found within and across each traversal using FAB-MAP [21]. For each traversal we optimise the VO, GPS/INS, and individual loop closures in the radar frame to obtain an approximately accurate global  $SE(2)$  pose estimate. Finally, all 32 pose chains are jointly optimised with all constraints before interpolating to create the ground truth, time-synchronised radar odometry.

## VII. DEVELOPMENT TOOLS

We provide a set of MATLAB and Python development tools for easy access to and manipulation of the newly provided data formats; where tools for sensors from the original dataset, such as for imagery, remain unchanged. The new tools include simple functions to load and display radar and Velodyne scans as well as more complex functionality such as converting the polar radar data into Cartesian form and converting raw Velodyne data into a pointcloud. To simplify usage these tools have been merged back into the original Oxford RobotCar Dataset SDK<sup>1</sup>. We also provide, and plan to extend, additional functionality useful to the community such as a batch downloader script for this dataset

and deep learning data loaders; for up to date information on these please refer to the dataset website.

### A. Radar Loading and Conversion to Cartesian

The MATLAB and Python functions `LoadRadar.m` and `load_radar` read a raw radar scan from a specified directory and at a specified timestamp, and return the per-azimuth UNIX timestamps ( $\mu\text{s}$ ), azimuth angles (rad), and power returns (dB) as well as the range resolution (cm) as described previously. For this data release radar resolution will always equal 4.38 cm.

The functions `RadarPolarToCartesian.m` and `radar_polar_to_cartesian` take the azimuth angles (rad), power returns (dB) and radar range resolution (cm) from a decoded radar scan and converts the polar scan into Cartesian form according to a desired Cartesian resolution (m) and Cartesian size (px).

The scripts `PlayRadar.m` and `play_radar.py` produce an animation of the available radar scans from a dataset directory as well as performing polar-to-Cartesian conversion as shown in Figure 3; please consult this script and the individual functions for demo usage.

### B. Velodyne Loading and Conversion to Pointcloud

Similarly, the MATLAB and Python functions `LoadVelodyneRaw.m` and `load_velodyne_raw` read a raw Velodyne scan from a specified directory and at a specified timestamp, of the form `<timestamp>.png`, and return ranges (m), intensities (uint8), azimuth angles (rad), and approximate timestamps ( $\mu\text{s}$ ) as described previously.

The functions `VelodyneRawToPointcloud.m` and `velodyne_raw_to_pointcloud` take the ranges (m), intensities (uint8), and azimuth angles (rad) from a decoded raw Velodyne scan and produce a pointcloud in Cartesian form including per-point intensity values.

The functions `LoadVelodyneBinary.m` and `load_velodyne_binary` read a binary Velodyne scan from a specified directory and at a specified timestamp, of the form `<timestamp>.bin`, and returns a pointcloud in Cartesian form including per-point intensity values.

Finally, the scripts `PlayVelodyne.m` and `play_velodyne.py` produce an animation of the available Velodyne scans from a dataset directory, as shown in Figure 7; please consult this script and the individual functions for demo usage.

## VIII. SUMMARY AND FUTURE WORK

We have presented the *The Oxford Radar RobotCar Dataset*, a new large-scale dataset focused on further exploitation of millimetre-wave FMCW scanning radar sensors for large-scale and long-term vehicle autonomy and mobile robotics. Although this modality has received relatively little attention in this context, we anticipate that this release will help foster discussion of its uses and encourage new and interesting areas of research not previously possible.

In the future, we would like to continue to collect and share large-scale radar datasets in new and challenging conditions and more precisely fine-tune the current extrinsic

<sup>1</sup>[github.com/ori-mrg/robotcar-dataset-sdk](https://github.com/ori-mrg/robotcar-dataset-sdk)

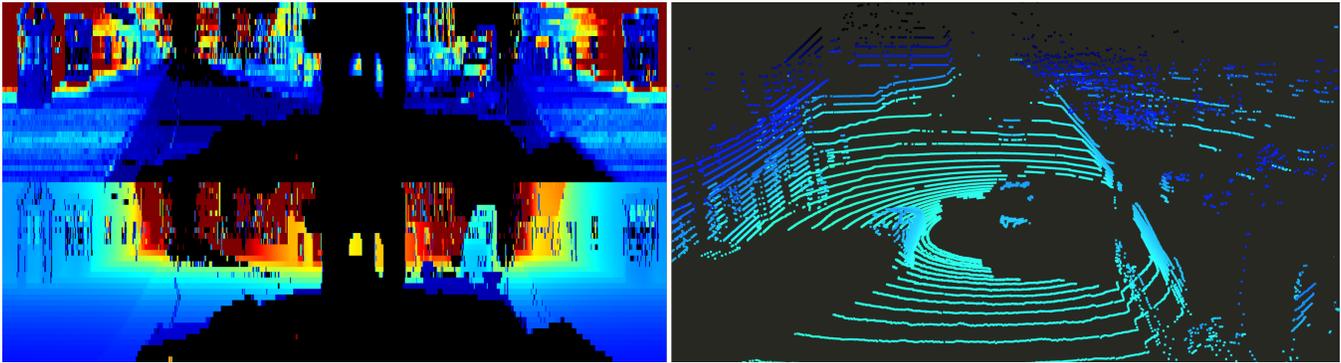


Fig. 7. Example sensor data from the Velodyne HDL-32E 3D LIDAR. A raw Velodyne scan (left) stores intensities (top) and ranges (bottom) for each azimuth (columns) whereas a binary scan stores the Cartesian pointcloud (right). Tools required to parse the data and perform the raw-to-pointcloud conversion are provided in the SDK mentioned in Section VII. Here the raw scan (left) is shown with invalid pixels set to black and stretched colourmap to improve visibility for the reader.

calibration parameters, perhaps by using publicly available toolboxes designed for radar-LIDAR-camera systems such as [24]. Finally, we would like to investigate semantic scene understanding in radar, perhaps with additionally collecting doppler data, to show that it is a viable alternative for otherwise commonly used sensors like vision and LIDAR.

#### IX. ACKNOWLEDGEMENTS

The authors thank all the members of the Oxford Robotics Institute (ORI) who performed scheduled driving over the data collection period. We would also like to thank our partners at Navtech Radar, without whom this dataset release would not have been possible.

Dan Barnes is supported by the UK EPSRC Doctoral Training Partnership. Matthew Gadd is supported by Innovate UK under CAV2 – Stream 1 CRD (DRIVEN). Paul Newman and Ingmar Posner are supported by EPSRC Programme Grant EP/M019918/1.

#### REFERENCES

- [1] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 year, 1000 km: The Oxford RobotCar dataset,” *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.
- [2] G. Pandey, J. R. McBride, and R. M. Eustice, “Ford campus vision and lidar data set,” *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.
- [3] 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.
- [4] J.-L. Blanco-Claraco, F.-Á. Moreno-Dueñas, and J. González-Jiménez, “The Málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario,” *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014.
- [5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.
- [6] 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.
- [7] 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.
- [8] 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 av dataset 2019,” <https://level5.lyft.com/dataset/>, 2019.
- [9] J. Callmer, D. Törnqvist, F. Gustafsson, H. Svensson, and P. Carlbom, “Radar slam using visual features,” *EURASIP Journal on Advances in Signal Processing*, vol. 2011, no. 1, p. 71, 2011.
- [10] G. Reina, J. Underwood, G. Brooker, and H. Durrant-Whyte, “Radar-based perception for autonomous outdoor vehicles,” *Journal of Field Robotics*, vol. 28, no. 6, pp. 894–913, 2011.
- [11] M. Adams, M. D. Adams, and E. Jose, *Robotic navigation and mapping with radar*. Artech House, 2012.
- [12] D. Vivet, P. Checchin, and R. Chapuis, “Localization and mapping using only a rotating FMCW radar sensor,” *Sensors*, vol. 13, no. 4, pp. 4527–4552, 2013.
- [13] F. Schuster, C. G. Keller, M. Rapp, M. Haueis, and C. Curio, “Landmark based radar slam using graph optimization,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 2559–2564.
- [14] S. H. Cen and P. Newman, “Precise Ego-Motion Estimation with Millimeter-Wave Radar under Diverse and Challenging Conditions,” *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, 2018.
- [15] S. Cen and P. Newman, “Radar-only ego-motion estimation in difficult settings via graph matching,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada*, 2019.
- [16] R. Aldera, D. De Martini, M. Gadd, and P. Newman, “Fast Radar Motion Estimation with a Learnt Focus of Attention using Weak Supervision,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada*, 2019.
- [17] Y. S. Park, J. Jeong, Y. Shin, and A. Kim, “Radar Dataset for Robust Localization and Mapping in Urban Environment,” in *ICRA 2019 Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR, Montreal, Canada*, 2019.
- [18] T. Peynot, S. Scheduling, and S. Terho, “The marulan data sets: Multi-sensor perception in a natural environment with challenging conditions,” *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1602–1607, 2010.
- [19] OpenStreetMap contributors, “Planet dump retrieved from <https://planet.osm.org>,” <https://www.openstreetmap.org>, 2017.
- [20] D. Barnes, W. Maddern, G. Pascoe, and I. Posner, “Driven to distraction: Self-supervised distractor learning for robust monocular visual odometry in urban environments,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1894–1900.
- [21] M. Cummins and P. Newman, “FAB-MAP: Probabilistic localization and mapping in the space of appearance,” *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.
- [22] S. Agarwal, K. Mierle, and Others, “Ceres solver,” <http://ceres-solver.org>.
- [23] W. Churchill, “Experience based navigation: Theory, practice and implementation,” Ph.D. dissertation, University of Oxford, 2012.
- [24] J. Domhof, J. F. P. Kooij, and D. M. Gavrilu, “An Extrinsic Calibration Tool for Lidar, Camera and Radar,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada*, 2019.