1 Year, 1000km: The Oxford RobotCar Dataset

Will Maddern, Geoffrey Pascoe, Chris Linegar and Paul Newman

***Abstract*— We present a challenging new dataset for autonomous driving: the Oxford RobotCar Dataset. Over the period of May 2014 to December 2015 we traversed a route through central Oxford twice a week on average using the Oxford RobotCar platform, an autonomous Nissan LEAF. This resulted in over 1000km of recorded driving with almost 20 million images collected from 6 cameras mounted to the vehicle, along with LIDAR, GPS and INS ground truth. Data was collected in all weather conditions, including heavy rain, night, direct sunlight and snow. Road and building works over the period of a year significantly changed sections of the route from the beginning to the end of data collection. By frequently traversing the same route over the period of a year we enable research investigating long-term localisation and mapping for autonomous vehicles in real-world, dynamic urban environments. The full dataset is available for download at: http://robotcar-dataset.robots.ox.ac.uk**

I. Introduction

Autonomous vehicle research is critically dependent on vast quantities of real-world data for development, testing and validation of algorithms before deployment on public roads. Following the benchmark-driven approach of the computer vision community, a number of vision-based autonomous driving datasets have been released including [1], [2], [3], [4], [5], notably the KITTI dataset in [6] and the recent Cityscapes dataset in [7]. These datasets focus primarily on the development of algorithmic competencies for autonomous driving: motion estimation as in [8], [9], stereo reconstruction as in [10], [11], pedestrian and vehicle detection as in [12], [13] and semantic classification as in [14], [15]. However, these datasets do not address many of the challenges of long-term autonomy: chiefly, localisation in the same environment under significantly different conditions as in [16], [17], and mapping in the presence of structural change over time as in [18], [19].

In this paper we present a large-scale dataset focused on long-term autonomous driving. We have collected more than 20TB of image, LIDAR and GPS data by repeatedly traversing a route in central Oxford, UK over the period of a year, resulting in over 1000km of recorded driving. Sample 3D visualisations of the data collected are shown in Fig. 1. By driving the same route under different conditions over a long time period, we capture a large range of variation in scene appearance and structure due to illumination, weather, dynamic objects, seasonal effects and construction. Along with the raw recordings from all the sensors, we provide a full set of intrinsic and extrinsic sensor calibrations, as well as MATLAB development tools for accessing and manipulating the raw sensor data.

Authors are from the Mobile Robotics Group, Dept. Engineering Science, University of Oxford, UK. {wm,gmp,chrisl,pnewman}@robots.ox.ac.uk

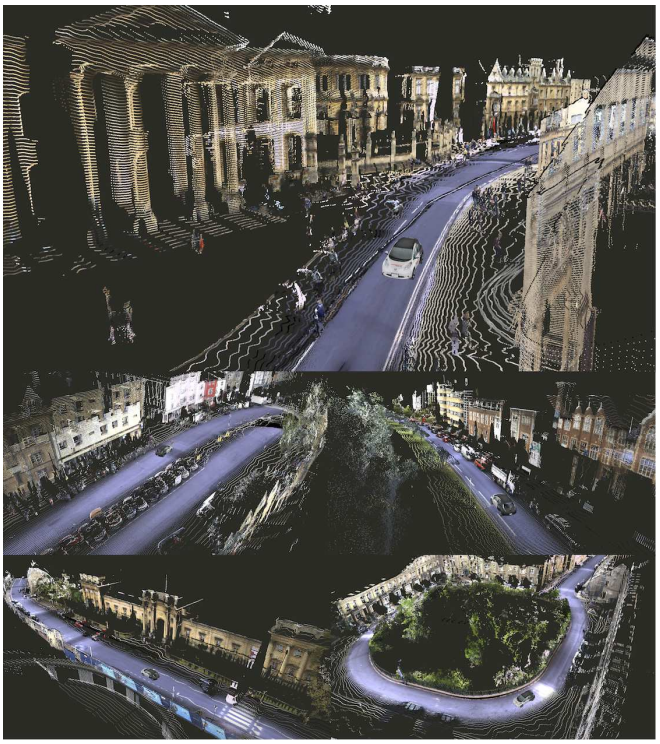


Fig. 1. The Oxford RobotCar Dataset for long-term road vehicle autonomy. Data was collected using the RobotCar platform repeatedly traversing an approximately 10km route in central Oxford, UK for over a year. This resulted in over 100 traversals of the same route, capturing the large variation in appearance and structure of a dynamic city environment over long periods of time. The image, LIDAR and GPS data collected enable research into long-term mapping and localisation for autonomous vehicles, with sample 3D maps built from different areas in the dataset shown here.

By providing this large-scale dataset to researchers, we hope to accelerate research towards long-term autonomy for the mobile robots and autonomous vehicles of the future.

II. The RobotCar Platform

The data was collected using the Oxford RobotCar platform, an autonomous-capable Nissan LEAF, illustrated in Fig. 2. The RobotCar is equipped with the following sensors:

- 1 x Point Grey Bumblebee XB3 (BBX3-13S2C-38) trinocular stereo camera, $1280 \times 960 \times 3$, 16Hz, 1/3" Sony ICX445 CCD, global shutter, 3.8mm lens, 66° HFoV, 12/24cm baseline
- 3 x Point Grey Grasshopper2 (GS2-FW-14S5C-C) monocular camera, 1024×1024, 11.1Hz, 2/3" Sony

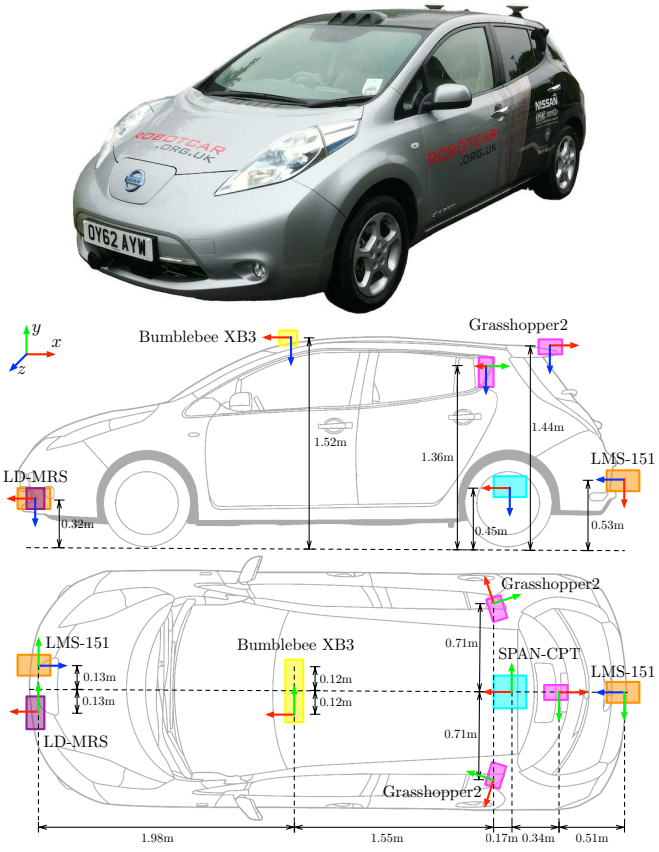


Fig. 2. The RobotCar platform (top) and sensor location diagram. Coordinate frames show the origin and direction of each sensor mounted to the vehicle with the convention: x-forward (red), y-right (green), z-down (blue). Measurements listed are approximated to the nearest centimetre; the development tools include exact $\mathbb{SE}(3)$ extrinsic calibrations for all sensors.

ICX285 CCD, global shutter, 2.67mm fisheye lens (Sunex DSL315B-650-F2.3), 180° HFoV

- 2 x SICK LMS-151 2D LIDAR, 270° FoV, 50Hz, 50m range, 0.5° resolution
- 1 x SICK LD-MRS 3D LIDAR, 85° HFoV, 3.2° VFoV, 4 planes, 12.5Hz, 50m range, 0.125° resolution
- 1 x NovAtel SPAN-CPT ALIGN inertial and GPS navigation system, 6 axis, 50Hz, GPS/GLONASS, dual antenna

The locations of the sensors on the vehicle is shown in Fig. 2. All sensors on the vehicle were logged using a PC running Ubuntu Linux with two eight-core Intel Xeon E5-2670 processors, 96GB quad-channel DDR3 memory and a RAID 0 (striped) array of eight 512GB SSDs, for a total capacity of 4 terabytes. All sensor drivers and logging processes were developed internally to provide accurate synchronisation and timestamping of the recorded data.

Both LMS-151 2D LIDAR sensors are mounted in a "push-broom" configuration, to provide a scan of the environment around the vehicle during forward motion. By combining these scans with a global or local pose source, an accurate 3D reconstruction of the environment is formed as illustrated in Fig. 1; software to generate 3D pointclouds

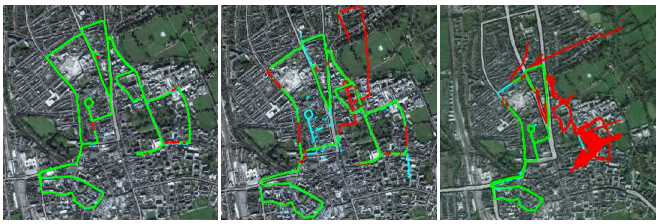


Fig. 3. Primary 10km data collection route illustrating variations in GPS reception and INS quality. Most datasets exhibited good GPS reception (left) shown in green. In some locations, poor GPS reception caused drift in the INS solution (middle, red), although raw GPS measurements (cyan) were still available. Occasionally a loss of reception caused significant positioning error over a large portion of the route (right).

is provided with the MATLAB development tools. The LD-MRS sensor is mounted in a forward-facing configuration to detect obstacles in front of the vehicle.

The combination of the Bumblebee XB3 and three Grasshopper2 cameras provides a full 360 degree visual coverage of the scene around the vehicle. The three Grasshopper2 cameras were synchronised using the automatic method provided by Point Grey for cameras sharing a Firewire 800 bus[1], yielding an average frame rate of 11.1Hz. The Bumblebee XB3 camera was logged at the maximum frame rate of 16Hz.

A custom auto-exposure controller was developed for the Bumblebee XB3 to provide well-exposed images of the ground and surrounding buildings, without attempting to correctly expose the sky or the front of the vehicle. However, this occasionally resulted in overexposed frames due to the slower response of the software auto-exposure control. The hardware region-of-interest auto-exposure controller was used for the Grasshopper2 to provide well-exposed images.

III. Data Collection

The data collection spans the period of 6 May 2014 to 13 December 2015, and consists of 1010.46km of recorded driving in central Oxford, UK. The vehicle was driven manually throughout the period of data collection; no autonomous capabilities were used. The total uncompressed size of the data collected is 23.15TB. The primary data collection route, shown in Fig. 3, was traversed over 100 times over the period of a year. Fig. 4 presents a montage of images taken from the same location on different traversals, illustrating the range of appearance changes over time. Table I lists summary statistics for the entire year-long dataset.

Traversal times were chosen to capture a wide range of conditions, including pedestrian, cyclist and vehicle traffic, light and heavy rain, direct sun, snow, and dawn, dusk and night. To reduce the buildup of dust and moisture on the camera lenses and LIDAR windows, they were cleaned with a microfiber cloth before each traversal. Labels for each condition have been added to each traversal, and traversals can be grouped by labels for easy collection of a particular condition. Fig. 5 presents the different condition labels and

[1] https://www.ptgrey.com/kb/10252

Fig. 4. Montage of over 100 traversals of the same location between May 2014 and December 2015, illustrating the large changes in appearance over the range of conditions. Along with short-term lighting and weather changes, long-term changes due to seasonal variations are evident. Note the alignment of the blue sign on the left shoulder of the road.

TABLE I

COLLECTED DATA SUMMARY STATISTICS

Sensor	Type	Count	Size
Bumblebee XB3	Image	11,070,651	13.78TB
Grasshopper2	Image	8,485,839	9.08TB
LMS-151	2D Scan	25,618,605	255.95GB
LD-MRS	3D Scan	3,226,183	31.76GB
SPAN-CPT GPS	3D Position	1,188,943	496MB
SPAN-CPT INS	6DoF Position	11,535,144	4.74GB
Stereo VO	6DoF Position	3,690,067	422MB

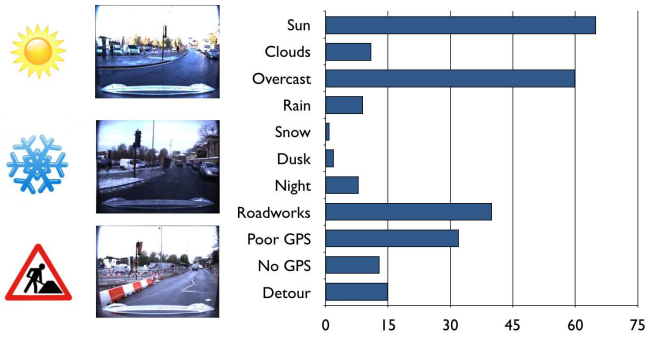


Fig. 5. Number of traversals for different condition labels. A number of environmental factors influenced the route, including changes in illumination and appearance, roadworks and construction causing detours, and atmospheric conditions affecting GPS reception. Traversals can be sorted by label for easy download and investigation of a particular environmental condition.

Fig. 6. Illustration of the variation in driving conditions captured during data collection. The appearance of the scene changed significantly due to lighting (direct sun, overcast, night), weather (rain, snow), and occlusions by other road users (vehicles, pedestrians, cyclists).

the number of traversals with each label. Fig. 6 illustrates the changes in appearance from a single location due to different conditions.

Due to roadworks and changes in traffic conditions, it was not possible to retrace the exact route for every traversal. In some sections of the route, there were significant structural changes over the period of data collection. Fig. 7 illustrates the change in structure of an intersection due to redevelopment in late 2014.

The quality of GPS reception and the accuracy of the fused INS solution varied significantly during the course of data collection, as illustrated in Fig. 3. We provide additional quality information as reported by the SPAN-CPT (satellite count, positioning error, fusion status) to aid in use of the GPS+Inertial data. However, we do not suggest directly using it as "ground truth" for benchmarking localisation and mapping algorithms, especially if using multiple traversals spaced over many months.

A. Sensor Calibration

We provide a full set of calibration data for each sensor to use for multi-sensor fusion and 3D reconstruction. Fig. 2 illustrates the extrinsic configuration of sensors on the RobotCar platform. All sensor drivers make use of the TICSync library of [20] to ensure sub-millisecond clock calibration between different sensors on different communication buses. Intrinsic and extrinsic calibrations are provided for each sensor as follows:

1) Intrinsics: For the Bumblebee XB3 images, two sets of rectification look-up-tables are provided in the development kit: one for the 12cm narrow baseline configuration, consisting of the `center` and `right` images, and one for the 24cm wide baseline configuration, consisting of the `left` and `right` images. Care must be taken to select the correct configuration of images for narrow- or wide-baseline stereo if downloading only a portion of the dataset. The values in the Bumblebee XB3 look-up-tables are provided by Point Grey as part of a factory calibration.

For the Grasshopper2 images, one undistortion look-up-table is provided for each camera, generated using a calibrated model obtained using the OCamCalib toolbox of [21]. However, perspective undistortion of fisheye images often results in highly non-uniform images, and different models of fisheye distortion are often preferable for different applications. As such we also provide the original images of a checkerboard sequence used as input for the OCamCalib toolbox calibration.

2) Extrinsics: For each LIDAR scanner, an initial calibration estimate was provided by the camera-LIDAR calibration process described in [22]. For the 2D LIDAR scanners, the calibration estimate was refined using the online calibration approach described in [23], using visual odometry derived from the Bumblebee XB3 as the relative pose source. The calibration from the GPS+Inertial sensor to the Bumblebee XB3 was computed using a modified version of [23], using the trajectory from the inertial sensor as the relative pose

Fig. 7. Structural change in the environment over time. The original intersection (top left) was redeveloped over a period of months, resulting in redirection of traffic during construction (top right, bottom left). The final intersection (bottom right) has a completely different road layout, replacing traffic lights with a roundabout.

source to construct the 3D LIDAR mesh. The LD-MRS was calibrated using a pointcloud built from 2D LIDAR scans using the approach described in [24]. All extrinsics are provided with the MATLAB development tools.

Lifelong sensor calibration is a challenging problem related to long-term localisation and mapping. While we provide our best estimates of the extrinsics using the above methods, we cannot guarantee they will be exact for a specific traversal. We encourage researchers investigating online calibration to use our extrinsics as an initial guess for long-term estimation of sensor calibration parameters.

B. Data Formats

For distribution we have divided up the datasets into individual routes, each corresponding to a single traversal. To reduce download file sizes, we have further divided up each traversal into *chunks*, where each chunk corresponds to an approximately 6 minute segment of the route. Within one traversal, chunks from different sensors will overlap in time (e.g. left stereo chunk 2 covers the same time period as LD-MRS chunk 2); however, chunks do not correspond between different traversals.

Each chunk has been packaged as a `tar` archive for downloading; the folder structure inside the archive is illustrated in Fig. 8. No `tar` file should exceed 5GB in size. It is intended that all `tar` archives are extracted in the same directory: this will preserve the folder structure in Fig. 8 when multiple chunks and/or traversals are downloaded. Each chunk archive also contains a full list of all sensor timestamps for the traversal in the `<sensor>.timestamps` file, as well as the list of condition tags as illustrated in Fig. 5 in the `tags.csv` file. The timestamps file contains ASCII formatted data, with each line corresponding to the UNIX timestamp and chunk ID of a single image or LIDAR scan.

We have converted our internal logging formatted sensor

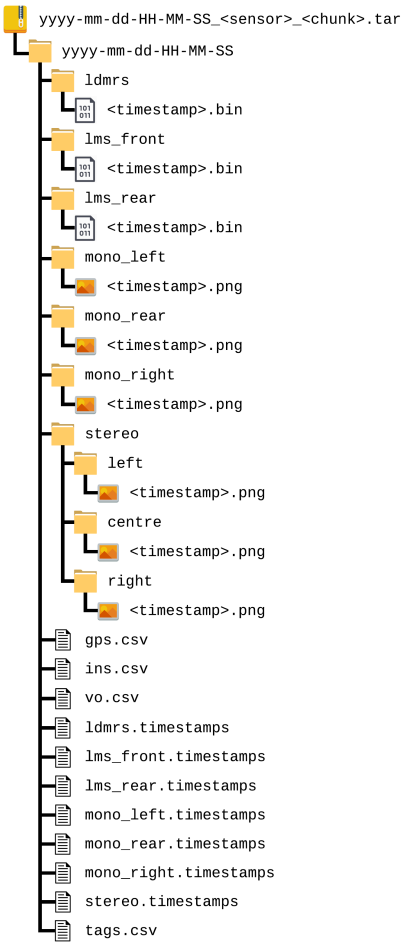


Fig. 8. Directory layout for a single data set. When downloading multiple `tar` archives from multiple traversals, extracting them all in the same directory will preserve the folder structure.

data to standard data formats for portability. The formats for each data type are as follows:

1) Images: All images are stored as lossless-compressed PNG files[2] in unrectified 8-bit raw Bayer format. The files are structured as `<camera>/<sensor>/<timestamp>.png`, where `<camera>` is `stereo` for Bumblebee XB3 images or `mono` for Grasshopper2 images, `<sensor>` is any of `left, centre, right` for Bumblebee XB3 images and `left, rear, right` for Grasshopper2 cameras, and `<timestamp>` is the UNIX timestamp of the capture, measured in microseconds. The top left 4 pixel Bayer pattern for the Bumblebee XB3 images is GBRG, and for Grasshopper2 images is RGGB. The raw Bayer images can be converted to RGB using the MATLAB `demosaic` function, the OpenCV `cvtColor` function or similar methods. Image undistortion and rectification is explained in the following section on Development Tools.

2) 2D LIDAR scans: The 2D LIDAR returns for each scan are stored as double-precision floating point values packed

[2] `https://www.w3.org/TR/PNG/`

Fig. 9. Left-to-right: raw Bayer image (provided in PNG format), demosaiced colour image, and rectified perspective image from (top) Bumblebee XB3 and (bottom) Grasshopper2 cameras. MATLAB functions for demosaicing and undistortion are provided in the development tools.

into a binary file, similar to the Velodyne scan format in [6]. The files are structured as `<laser>/<timestamp>.bin`, where `<laser>` is `lms_front` or `lms_rear`. Each 2D scan consists of 541 triplets of (x, y, R), where x, y are the 2D Cartesian coordinates of the LIDAR return relative to the sensor (in metres), and R is the measured infrared reflectance value. No correction for the motion of the vehicle has been performed when projecting the points into Cartesian coordinates; this can be optionally performed by interpolating a timestamp for each point based on the 15ms rotation period of the laser. For a scan with filename `<timestamp>.bin`, the (x, y, R) triplet at index 0 was collected at `<timestamp>` and the triplet at index 540 was collected at `<timestamp>+15e3`.

3) 3D LIDAR scans: The 3D LIDAR returns from the LD-MRS are stored in the same packed double-precision floating point binary format as the 2D LIDAR scans. The files are structured as `ldmrs/<timestamp>.bin`. Each 3D scan consists of triplets of (x, y, z), which is the 3D Cartesian coordinates of the LIDAR return relative to the sensor (in metres). The LD-MRS does not provide infrared reflectance values for LIDAR returns.

4) GPS+Inertial: GPS and inertial sensor data from the SPAN-CPT are provided in an ASCII-formatted csv file. Two separate files are provided: `gps.csv` and `ins.csv`. `gps.csv` contains the GPS-only solution of latitude (deg), longitude (deg), altitude (m) and uncertainty (m) at 5Hz, and `ins.csv` contains the fused GPS+Inertial solution, consisting of 3D UTM position (m), velocity (m/s), attitude (deg) and solution status at 50Hz.

5) Visual Odometry (VO): Local errors in the GPS+Inertial solution (due to loss or reacquisition of satellite signals) can lead to discontinuites in local maps built using this sensor as a pose source. For some applications a smooth local pose source that is not necessarily globally accurate is preferable, for example local 3D pointcloud construction as shown in Fig. 10. We have processed the full set of Bumblebee XB3 wide-baseline stereo imagery using our visual odometry system described in [25] and provide the

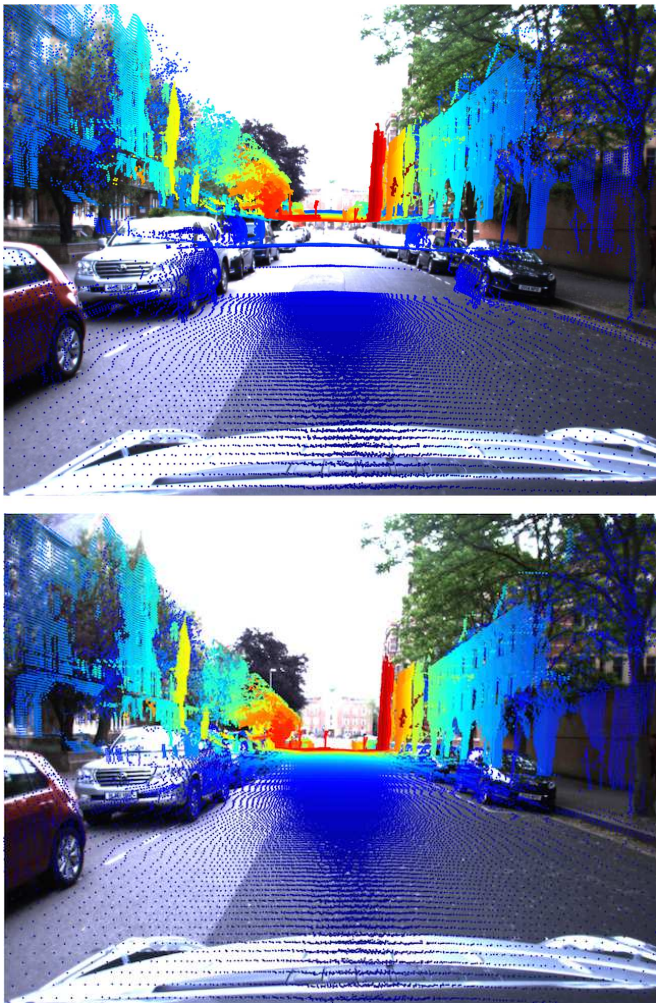


Fig. 10. Local 3D pointclouds generated using (top) INS poses and (bottom) local VO poses, produced with the included development tools. In locations with poor GPS reception, there are discontinuities in the INS pose solution leading to corrupted local 3D pointclouds. Visual odometry provides a smooth local pose estimate but drifts over larger distances.

relative pose estimates as a reference local pose source. The file `vo.csv` contains the relative pose solution, consisting of the source and destination frame timestamps, along with the six-vector Euler parameterisation $(x, y, z, \alpha, \beta, \gamma)$ of the $\mathbb{SE}(3)$ relative pose relating the two frames. Our visual odometry solution is accurate over hundreds of metres (suitable for building local 3D pointclouds) but drifts over larger scales, and is also influenced by scene content and exposure levels; we provide it only as a reference and not as a ground truth relative pose system.

IV. Development Tools

We provide a set of simple MATLAB development tools for easy access to and manipulation of the dataset. The MATLAB functions provided include simple functions to load and display imagery and LIDAR scans, as well as more advanced functions involving generating 3D pointclouds from push-broom 2D scans, and projecting 3D pointclouds into camera images. Fig. 11 illustrates sample uses of the development

tools.

A. Image Demosaicing and Undistortion

The function `LoadImage.m` reads a raw Bayer image from a specified directory and at a specified timestamp, and returns a MATLAB format RGB colour image. The function can also take an optional look-up table argument (provided by the function `ReadCameraModel.m`) which will then rectify the image using the undistortion look-up tables. Examples of raw Bayer, RGB and rectified images are shown in Fig. 9. The function `PlayImages.m` will produce an animation of the available images from a specified directory, shown in Fig. 11(a).

B. 3D Pointcloud Generation

The function `BuildPointcloud.m` combines a 6DoF trajectory from the INS with 2D LIDAR scans to produce a local 3D pointcloud, as pictured in Fig. 11(b). The resulting pointcloud is coloured using LIDAR reflectance information, highlighting lane markings and other reflective objects. The size of the pointcloud can be controlled using the `start_timestamp` and `end_timestamp` arguments.

C. Pointcloud Projection Into Images

The function `ProjectLaserIntoCamera.m` combines the above two tools as follows: `LoadImage.m` is used to retrieve and undistort a raw image from a specified directory at a specified timestamp, then `BuildPointcloud.m` is used to generate a local 3D pointcloud around the vehicle at the time of image capture. The 3D pointcloud is then projected into the 2D camera image using the camera intrinsics, producing the result shown in Fig. 11(c).

V. Lessons Learned

In the process of collecting, storing and processing the data collected from the RobotCar we learned a number of valuable lessons, which we summarise here for others attempting similar large-scale dataset collection:

1) Log raw data only: For each sensor we ensure that only the raw packets data received over the wire and an accurate host timestamp are logged to disk; we perform no parsing, compression or filtering (e.g. Bayer demosaicing). We also enable all optional log messages from all sensors (e.g. status messages). This maximises the utility of the data, which may only be apparent months or years after collection, and minimises 'baked-in' decisions about pre-processing. For example, methods that depend on photometric error as in [26] or precise colourimetry as in [27] are strongly affected by image compression and Bayer demosaicing; by logging raw packets directly from the camera we minimise the restrictions on future research.

2) Use forward-compatible formats: When collecting data over periods of multiple years, it is inevitable that software errors arise and are fixed, and the functionality of tools to manipulate data improve over time. Therefore it is important that data is logged in forward-compatible binary formats that are not tied to a particular software version. Internally we use Google Protocol Buffers[3] to manage our data formats; this allows us to change or extend message definitions, access message data in multiple languages using code generators and maintain binary compatibility with older software versions.

3) Separate logged and processed data: When processing and manipulating the data, it is often useful to generate associated metadata (e.g. index files, LIDAR pointclouds, visual odometry results). However, the tools that generate this metadata will change and improve over long periods, and it is important that the processed logs remain distinct from the original raw recordings. In practice we maintain two log directories on our data servers; one read-only directory that only the RobotCar-mounted computer can upload to, and one read-write directory that mirrors the logged data and allows users to add metadata. Therefore, if one of the metadata tools is upgraded, there is no risk of losing logged raw data when deleting or replacing all previous metadata generated by the tool.

VI. Summary and Future Work

We have presented the Oxford RobotCar Dataset, a new large-scale dataset focused on long-term autonomy for autonomous road vehicles. With the release of this dataset we intend to challenge current approaches to long-term localisation and mapping, and enable research investigating lifelong learning for autonomous vehicles and mobile robots.

In the near future we hope to offer a benchmarking service similar to the KITTI benchmark suite[4], providing the opportunity for researchers to publicly compare long-term localisation and mapping methods using a common ground truth and evaluation criteria. We also encourage researchers to develop their own application-specific benchmarks derived from the data presented in this paper, e.g. using the open source structure-from-motion of [28] or the optimisation package of [29], which we will endeavour to support.

VII. Acknowledgements

The authors wish to thank Chris Prahacs and Peter Get for maintaining the RobotCar hardware, and Alex Stewart, Winston Churchill and Tom Wilcox for maintaining the data collection software. The authors also thank all the members of the Oxford Robotics Institute who performed scheduled driving over the data collection period.

Will Maddern is supported by EPSRC Programme Grant EP/M019918/1. Geoffrey Pascoe and Chris Linegar are supported by Rhodes Scholarships. Paul Newman is supported by EPSRC Leadership Fellowship EP/I005021/1.

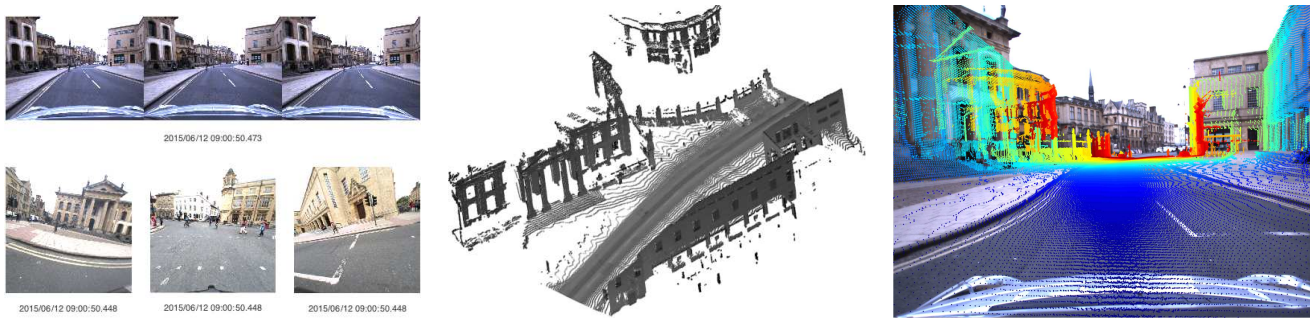
References

[1] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 304–311.

[2] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[3] `https://developers.google.com/protocol-buffers/`

[4] `http://www.cvlibs.net/datasets/kitti/`



(a) (b) (c)

Fig. 11. Samples from the MATLAB development tools. (a) Image undistortion and playback from all vehicle-mounted cameras, (b) 3D pointcloud generation from push-broom LIDAR and INS pose, (c) projecting a 3D pointcloud into a 2D camera image using known extrinsics and intrinsics.

[3] 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.

[4] D. Pfeiffer, S. Gehrig, and N. Schneider, "Exploiting the power of stereo confidences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 297–304.

[5] 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.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[7] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[9] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 963–968.

[10] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 807–814.

[11] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian conference on computer vision*. Springer, 2010, pp. 25–38.

[12] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[13] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *European Conference on Computer Vision*. Springer, 2014, pp. 613–627.

[14] I. Posner, M. Cummins, and P. Newman, "Fast probabilistic labeling of city maps," in *Proceedings of Robotics: Science and Systems IV*, Zurich, Switzerland, June 2008.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[16] C. McManus, B. Upcroft, and P. Newman, "Learning place-dependant features for long-term vision-based localisation," *Autonomous Robots, Special issue on Robotics Science and Systems 2014*, pp. 1–25, 2015.

[17] C. Linegar, W. Churchill, and P. Newman, "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[18] G. D. Tipaldi, D. Meyer-Delius, and W. Burgard, "Lifelong localization in changing environments," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1662–1678, 2013.

[19] W. Maddern, G. Pascoe, and P. Newman, "Leveraging experience for large-scale LIDAR localisation in changing cities," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015.

[20] A. Harrison and P. Newman, "TICSync: Knowing when things happened," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 356–363.

[21] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 5695–5701.

[22] A. Kassir and T. Peynot, "Reliable automatic camera-laser calibration," in *Proceedings of the 2010 Australasian Conference on Robotics & Automation*. ARAA, 2010.

[23] G. Pascoe, W. Maddern, and P. Newman, "Direct visual localisation and calibration for road vehicles in changing city environments," in *IEEE International Conference on Computer Vision: Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*, Santiago, Chile, December 2015.

[24] W. Maddern, A. Harrison, and P. Newman, "Lost in translation (and rotation): Fast extrinsic calibration for 2D and 3D LIDARs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.

[25] W. Churchill, "Experience based navigation: Theory, practice and implementation," Ph.D. dissertation, University of Oxford, Oxford, United Kingdom, 2012.

[26] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*. Springer, 2014, pp. 834–849.

[27] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman, "Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles," in *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, vol. 2, 2014, p. 3.

[28] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org, 2012.